Near-deterministic production of universal quantum photonic gates enhanced by machine learning

Krishna Kumar Sabapathy, Haoyu Qi, Josh Izaac, and Christian Weedbrook
Xanadu, 372 Richmond St W, Toronto ON, M5V 2L7, Canada

We introduce architectures for near-deterministic implementation of fully tunable weak cubic phase gates requisite for universal quantum computation. The first step is to produce a resource state which is a superposition of the first four Fock states with a probability $\geq 10^{-2}$, an increase by a factor of 10^4 over standard sequential photon-subtraction techniques. The resource state is produced from a quantum gadget that uses displaced squeezed states, interferometers and photon-number resolving detectors. The parameters of this gadget are trained using machine learning algorithms for variational circuits. Stacking these gadgets in parallel we build quantum resource farms in a scalable manner depending on the error tolerance. Using conventional teleportation techniques we can implement weak cubic phase gates, in principle, at a rate $\sim$ 100kHz dictated by the photon number resolving detectors. Our proposal is realizable with current photonic technologies without the need for quantum memories. The methods for non-Gaussian state preparation is of independent interest to the resource theory of non-Gaussianity.

Introduction — Universal quantum computation in continuous-variable systems requires non-Gaussian gates generated by Hamiltonians that are beyond quadratic in the quadrature operators [1, 2]. In this context the quadrature phase gates which are of the form $\Theta(\gamma) = \exp[i\gamma\hat{x}^n/\hbar]$ have played a very important role, especially the cubic phase gate corresponding to $n = 3$. Current methods to implement this gate are via gate teleportation where a resource state is prepared and used in a teleportation (measurement-based) circuit [3–10]. However, the methods of preparation of the resource state either require very high photon-counting [3, 11] or have a very low probability of success due to repeated photon subtractions, though some methods to improve the latter have been recently proposed [12–14].

We address this aspect of the resource state preparation wherein we constrain the resources to the simplest possible ones, namely, preparation of Gaussian multimode pure states followed by conditional photon detection measurements. For our purpose it suffices only to consider two-mode and three-mode architectures. We delegate the tuning of this multi-parameter constrained variational circuit to a machine learning algorithm that trains the circuit to learn the required state preparation, inspired by recent works [15–19]. It turns out that we can prepare a learned state with near-perfect fidelity to the target state and with comparatively high probability, leading to a preparation efficiency greater than 1% (sequential photon-subtraction techniques have an efficiency of $\sim 10^{-4}$%, see Supp. Sec. I). We then construct scalable quantum resource farms, depending on an error threshold, that are used to implement the cubic phase gates.

Gottesman-Kitaev-Preskill (GKP) gate teleporation — We now focus on the optical implementation of the lowest-order quadrature phase gate, namely, the cubic phase gate that we denote by $\mathsf{V}(\gamma) = \exp[i\gamma\hat{x}^3/\hbar]$, where γ is the gate strength. All known methods to implement the cubic phase gate involve preparing a suitable resource state and using measurement-based techniques (see for example Table II of [4]). We use the teleportation technique of GKP [3], but this can be translated into an adaptive gate teleportation presented using different gates and additional auxiliary squeezed states [8]. We take $\hbar = 2$ for the rest of the article.

For the cubic phase gate the resource state is the cubic phase state defined as $\mathsf{V}(\gamma)|0\rangle_p$, which is nonphysical due to the zero momentum ket $|0\rangle_p$. Therefore, as an approximation, we consider $\mathsf{V}(\gamma)\mathsf{S}(r)^\dagger|0\rangle$, where $\mathsf{S}(r)$ is the standard single-mode squeezing gate given by $\mathsf{S}(r) = \exp[r(\hat{a}^2 - \hat{a}^{\dagger 2})/2]$. For large squeezing, this state has a large Fock support and hence would be difficult to synthesize directly. To get around this we commute the squeeze operator across the cubic phase gate to obtain $\mathsf{S}(r)^\dagger\mathsf{V}(\gamma')|0\rangle$. Let us further assume that $\gamma' << 1$, then we can expand the cubic phase gate to first-order in gate strength to obtain $\mathsf{S}(r)^\dagger[1 + i\gamma'\hat{x}^3/2]|0\rangle$. This resource state was previously considered in [9, 10].

We will assume that we can apply the on-line squeezing gate using methods such as measurement-based squeezing [20–22] and we begin with what we denote as the resource state given by

$$|\phi\rangle = \frac{1}{\sqrt{1+5|a|^2/2}}\left[|0\rangle + ia\sqrt{\frac{3}{2}}|1\rangle + ia|3\rangle\right]. \quad (1)$$

The wavefunction of the squeezed resource state is

$$\tilde{\phi}(x) = \langle x|\,\mathsf{S}(r)^\dagger|\phi\rangle = \int dx'\phi(x')\,\langle x|\,\mathsf{S}(r)^\dagger|x'\rangle\,. \quad (2)$$

We note that $\langle x|\,\mathsf{S}(r)^\dagger|x'\rangle = e^{r/2}\langle x|e^r x'\rangle$. So we have that $\tilde{\phi}(x) = e^{-r}\int dy\phi(e^{-r}y)e^{r/2}\delta(x-y) = e^{-r/2}\phi(e^{-r}x)$.

We now use this squeezed resource state in a GKP teleporation scheme as depicted in Fig. 1; the output

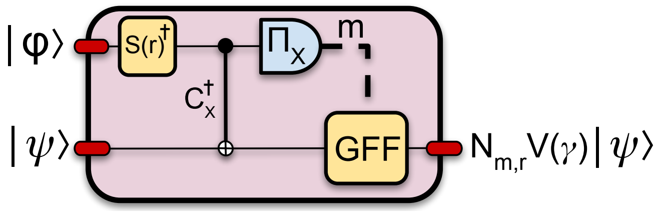

FIG. 1. The GKP teleportation for implementation of a cubic phase gate [3]. Here a state $|\psi\rangle$ is input to one mode along with a resource state $|\phi\rangle$ in the other. $\mathsf{S}(r)^\dagger$ denotes the squeezing operator, C_x the controlled-X gate given by $\exp[-i\hat{x}_1\hat{p}_2/2]$, Π_x the x-homodyne measurement with outcome labeled m, GFF the Gaussian feed-forward correction operator that needs to be applied, $\mathsf{N}_{m,r}$ the noise operator, and $\mathsf{V}(\gamma)$ the final cubic phase gate that is applied to the input state as shown in Eq. (4).

state is (Eq. 8 of Ref. [4])

$$\begin{aligned}|\psi_\mathrm{out}\rangle &= N\,\langle m^{(2)}|\,\mathsf{C}_\mathsf{x}{}^\dagger|\psi_\mathrm{in}\rangle|\tilde{\phi}\rangle = N\tilde{\phi}(\hat{x}+m)|\psi_\mathrm{in}\rangle\\ &= N^{'}\exp\left[-\frac{(\hat{x}+m)^2}{4e^{2r}}\right]\left[1+\frac{2ia}{\sqrt{6}e^{3r}}\frac{(\hat{x}+m)^3}{2}\right]|\psi_\mathrm{in}\rangle\\ &= N'\exp\left[-\frac{(\hat{x}+m)^2}{4e^{2r}}\right]\left[1+i\frac{\gamma}{2}(\hat{x}+m)^3\right]|\psi_\mathrm{in}\rangle, \qquad (3)\end{aligned}$$

where $\gamma = 2ae^{-3r}/\sqrt{6}$, N' is the normalization factor, m the homodyne measurement outcome, and $\mathsf{C}_\mathsf{x} = \exp[-i\hat{x}_1\hat{p}_2/2]$ is a gate that can be implemented [22–26]. We now assume that $\gamma \ll 1$, allowing us to approximate the terms in the second square bracket as a first-order expansion in the gate strength of a cubic phase gate, resulting in

$$|\psi_\mathrm{out}\rangle = N^{'}\exp\left[-\frac{(\hat{x}+m)^2}{4e^{2r}}\right]\exp\left[\frac{i\gamma(\hat{x}+m)^3}{2}\right]|\psi_\mathrm{in}\rangle.$$

Expanding the terms in the second operator and applying a Gaussian feed-forward $\mathsf{GFF}(\mathsf{m}) = \exp[-i\gamma(3m\hat{x}^2 + 3m^2\hat{x} + m^3)/2]$, we obtain the final action on the input state to be

$$|\psi_\mathrm{out}\rangle = N^{'}\mathsf{N}(m,r)\mathsf{V}(\gamma)|\psi_\mathrm{in}\rangle, \qquad (4)$$

where $\mathsf{N}(m,r) = \exp[-(\hat{x}+m)^2/(4e^{2r})]$ is the Gaussian damping noise operator that depends on the homodyne measurement outcome m.

So using the resource state $|\phi\rangle$, we can effect a transformation which is a weak cubic phase gate along with an unavoidable Gaussian noise factor. Note that the initial squeezing gate $\mathsf{S}(r)^\dagger$ not only reduces the strength of the final cubic phase gate but also negates the effect of the Gaussian noise operator as seen from Eq. (3).

Machine learning for state preparation—The resource state $|\phi\rangle$ can be prepared in the lab using standard sequential photon-subtraction/photon-addition techniques [27–31]. However, such a method is not scalable since the successful probability of three successive photon additions/subtractions is extremely low, due to the use of very high transmission beamsplitters [27, 32] as discussed in Supp. Sec. I.

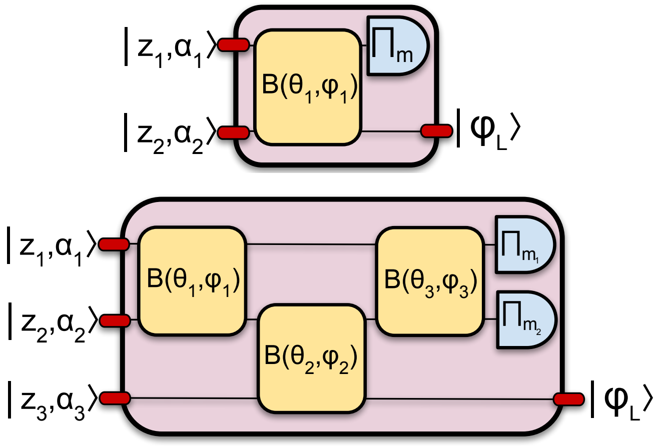

FIG. 2. Quantum gadgets of two (top) and three (bottom) modes. Here $\{z_i, \alpha_i\} \in \mathbb{C}$ denotes the input squeezing z and displacement α operators applied to the vacuum states. $\mathsf{B}(\theta,\phi)$ denotes a beamsplitter and Π_n a photon-number resolving detector post-selected to value n. $|\phi_L\rangle$ is the state obtained after training the circuit parameters.

To bypass this difficulty, we introduce very different architectures that use Gaussian states conditioned on non-Gaussian post-selected photon-number resolving (PNR) detectors, akin to Gaussian Boson Sampling circuits [33, 34]. Early works have considered conditional measurements on the outputs of beamsplitters for state preparation [35–41].

Our circuits are depicted in Fig. 2 for both the two-mode and three-mode architectures. We then use machine learning algorithms and the Strawberry Fields quantum simulator [42] to train these circuits against the target state given in Eq. (1), as presented in Algorithm 1. The algorithm executes a two-step optimization, where the circuit parameters are first trained to maximize the fidelity with the target state using basinhopping which is a global search heuristic. The second step is then to perform a local_search starting from the global optimum found by basinhopping, to further increase the probability of producing the trained state. We choose the cutoff dimension of each mode in Fig. 2 to be 15 such that there is a large enough Hilbert space to be explored but without too much overhead. The PNR detectors in the two-mode case is set to $m = 2$ and in the three-mode case to $(m_1, m_2) = (1, 2)$ as argued in Supp. Sec. II along with all the details of the numerical techniques.

We plot the fidelity and Wigner negative region overlap between the trained state $|\phi_L\rangle$ and the target state $|\phi\rangle$, and the probability of producing the trained state in Fig. 3. For parameter $a \in [0.3, 1]$ we have that the

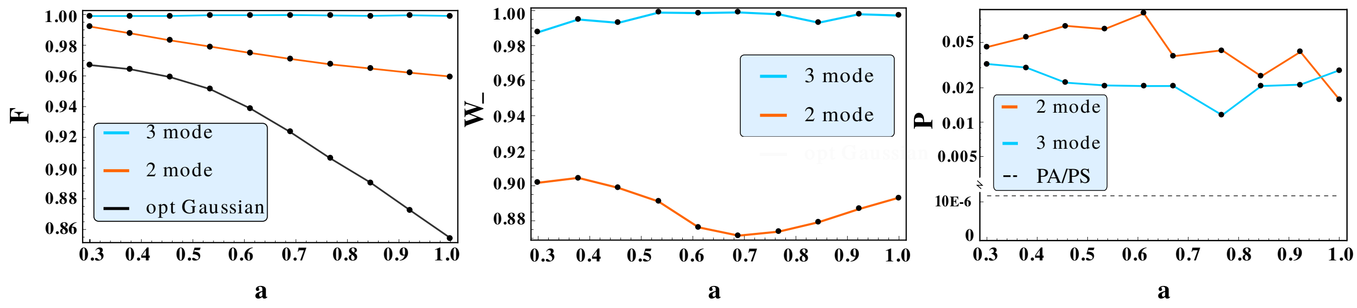

FIG. 3. Plot 1 depicts the optimal fidelity F of the machine learned state $|\phi_L\rangle$ in three-mode (top line) and two-mode (middle line) architectures, and the closest Gaussian state (bottom line) with respect to the target state $|\phi\rangle$ in Eq. (1). Plot 2 shows the overlap of the negative region of the Wigner function W_- between $|\phi_L\rangle, |\phi\rangle$. Plot 3 depicts the probability P of the optimal output state for the two-mode (top line) and three-mode (bottom line) architectures compared with three consecutive photon-additions/subtractions (PA/PS) with a probability $\sim 10^{-6}$. We see that for fidelity and Wigner negativity, the three-mode case performs better at the cost of a drop in probability.

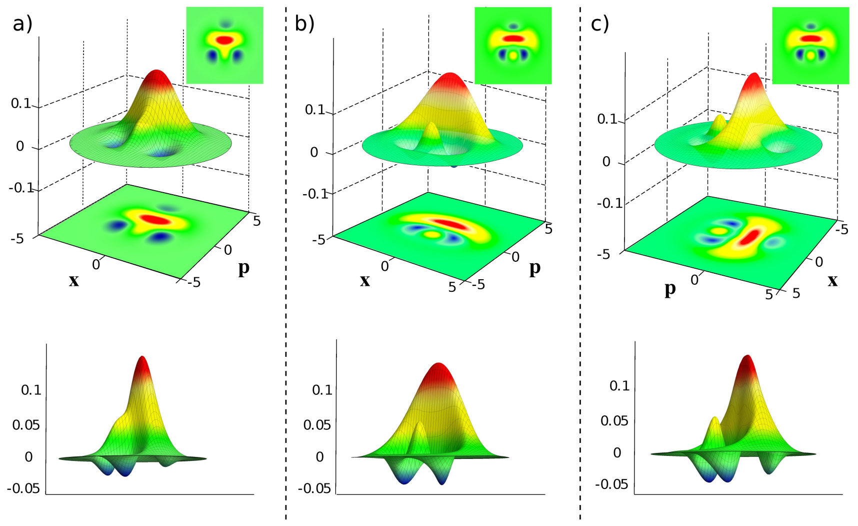

FIG. 4. The Wigner function of the trained state for three cases (a) $a = 0.3$ with three-mode architecture, (b) $a = 0.61$ with two-mode circuit, and (c) $a = 0.61$ with the three-mode architecture. The insets are the Wigner contour plot for the target states for the corresponding values of a. We see that the trained state using the three-mode circuit is indistinguishable with the target state. We find that for the two-mode case (b) with $a = 0.61$, one Wigner negative region is missing with respect to the three-mode case (c), thereby leading to lesser fidelity to the target state.

gate strength $\gamma \in [0.0122, 0.0407]$. The trained and target states are both pure, and we have that the fidelity is equivalent to the Wigner overlap [43], i.e.,

$$F(\phi_L, \phi) = |\langle\phi_L|\phi\rangle|^2 = \int dxdp\, W(x,p;\phi_L)W(x,p;\phi).$$

If the fidelity is high it implies that the Wigner overlap is also high as depicted in Fig. 4. We find that using the three-mode architecture the trained state is extremely close to the target state, and so is the Wigner negative overlap region. On the other hand, the two-mode circuit performs better than the three-mode circuit in terms of the probability of producing the trained state, but there is a substantial drop in the Wigner negative region overlap.

Scalability and quantum resource farms — The resource state preparation is probabilistic since it is conditioned on a particular post-selected measurement outcome as shown in Fig. 2. To convert this resource state preparation to a near-deterministic process, we propose an optical device which we call a quantum resource farm. We line up identical copies of the state preparation gadget in parallel, and connect all outputs to a single wire or sink as shown in Fig. 5. We now estimate the number of such gadgets that are required to render the device near-deterministic when allowing for a small error.

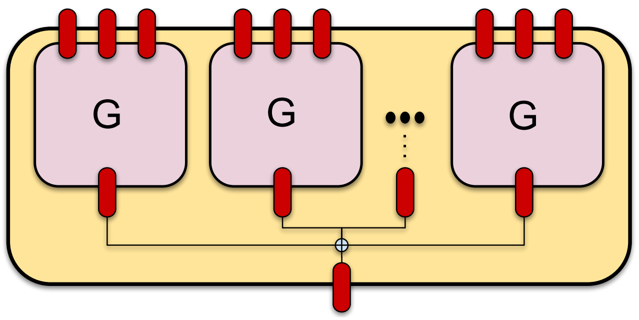

FIG. 5. A quantum resource farm that consists of a sufficient number n of quantum gadgets G of Fig. 2 (three-mode case shown here) placed in a parallel manner. The individual outputs are collected to one output to guarantee a near-deterministic resource state preparation depending on the error tolerance.

Algorithm 1 StatePrep_3mode

```
function loss($\boldsymbol{x}$, para, handle):
    # $\boldsymbol{x} = \{\boldsymbol{x}_s, \boldsymbol{x}_d, \boldsymbol{x}_\theta\}$ parameters for squeezing,
    # displacement and beamsplitter array
    para = $\{a, m_1, m_2, \text{cutoff\_dim}\}$
    $|\psi_{in}\rangle \leftarrow \mathsf{S}(\boldsymbol{x}_s)\mathsf{D}(\boldsymbol{x}_d)\,|\mathbf{0}\rangle$
    $|\psi_{out}\rangle \leftarrow \mathsf{U}(\boldsymbol{x}_\theta)\,|\psi_{in}\rangle$
    normIn, normOut $\leftarrow |\langle\psi_{in}\,|\psi_{in}\rangle|\,, |\langle\psi_{out}\,|\psi_{out}\rangle|$
    penalty = $100\times$|1-normIn| + $100\times$|1-normOut|
    $|\psi\rangle \leftarrow \langle m_1, m_2|\,\psi_{out}\rangle$
    prob $\leftarrow \langle\psi\,|\psi\rangle$
    $|\psi\rangle_L \leftarrow |\psi\rangle$ /prob
    fid = $|\langle\psi_L\,|\phi_T\rangle|^2$
    if handle == 'fid' then
        return -fid + penalty
    else if handle == 'fid_prob' then
        return -fid + prob + penalty
    end if
end function
procedure optimization(para, niter):
    # global exploration to optimize fidelity
    # basinhopping is a global search algorithm
    # further optimize the probability by local search
    initialize $\boldsymbol{x}_0$   # randomly chosen with proper range
    $\boldsymbol{x}_1$ ←basinhopping($\boldsymbol{x}_0$, loss, args=(para, 'fid'),niter)
    $\boldsymbol{x}_2$ ←local_search(loss, $\boldsymbol{x}_1$, args= (para, 'fid_prob'))
    save $\boldsymbol{x}_2$
end procedure
```

The model of our resource farm has n parallel quantum gadgets, and let p be the probability of success of producing the resource state from each one of them. Let ϵ be the error which captures the degree of determinism of the entire farm. The probability of producing the required resource state from the farm is denoted $P(F)$ and is given by the union of events of any of the constituent gadgets preparing this state. Even in the case where multiple resource states are simultaneously prepared, we continue to count it as a useful event. Thus, this success probability is identical to the complement event probability that none of the gadgets produce the required resource state;

$$P(F) = 1 - (1-p)^n. \tag{5}$$

Allowing an error ϵ for the failure of the state preparation from the farm, we set $P(F) \geq 1-\epsilon$. Therefore, the minimum required number of gadgets $n_{\min}$ is given by

$$n_{\min} = \left\lceil \frac{\log \epsilon}{\log(1-p)} \right\rceil, \tag{6}$$

where $\lceil y \rceil$ denotes the smallest integer $\geq y$. If we assume an ideal GKP-teleportation using this resource state, then the effective cubic phase gate can be implemented at the same rate as the resource state preparation. If we use a fewer number of gadgets than $n_{\min}$, the rate at which the cubic phase gate can be implemented will be proportionally reduced.

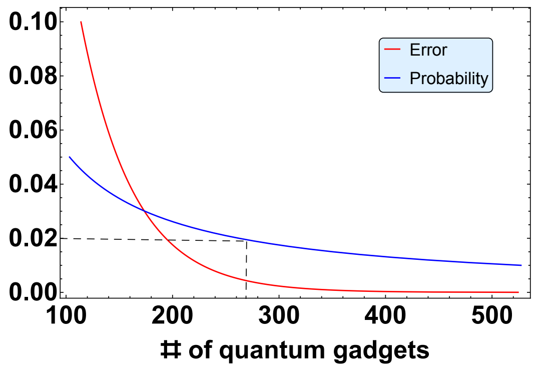

FIG. 6. The error (top line) for a probability fixed at 0.02 and probability (bottom line) with error fixed at 0.005 as a function of the number of quantum gadgets in the resource farm. As an example, if we consider $p = 0.02$ and $\epsilon = 0.005$, i.e., the gadget produces the resource state with efficiency of 2% and if we allow the farm to have a success rate of 99.5%, then we require ~ 260 gadgets (dotted lines) in the farm.

Examples — The top (red) line in Fig. 6 shows the variation of error ϵ as a function of the number of quantum gadgets $n_{\min}$ for a fixed probability of $p = 0.02$ of state preparation from a gadget. The bottom (blue) line shows the variation of p vs $n_{\min}$ for $\epsilon = 5 \cdot 10^{-3}$. For $p = 0.02$ (i.e., 2% efficiency) and $P(F) = 0.995$ ($\epsilon = 0.005$), $n_{\min} \sim 260$ (shown by dotted lines).

Conclusion — We proposed a scalable and efficient method for the production of weak cubic phase gates. While the method of gate teleportation is well-known, a conversion to a scalable architecture was made possible using the techniques of variational circuits in machine learning for the resource state preparation. We first constructed quantum gadgets that produce very general Gaussian pure states in two and three-modes which were conditioned on all but one of the modes to a post-selected photon-number resolving detector to obtain a non-Gaussian state. We then tuned the parameters of the gadget to produce a trained state that is of almost perfect fidelity with the target state.

We found that this architecture lead to a huge increase in the probability (of the order of 10^4) compared to conventional sequential photon-addition or subtraction methods. This gadget when cloned and combined in parallel, in what we term quantum resource farms, can lead to an almost deterministic production of the resource states, at a rate dictated by the PNR detectors which is currently $\sim$ 100khz. The crucial aspect is that our architecture can be realized with current state-of-the-art optical technologies, and we anticipate substantial improvements in future through technological advancements driven by demand. Further, with this architecture we are also in a position to estimate the cost required to build a resource farm with a few hundred gadgets of interest from a practical point of view. Also, having fewer gadgets would lead to a lower production rate of these gates while not compromising on the error.

Our proposal is only an initial step in this direction and there is scope for improvements and optimization using tools such as machine learning. Further, the final applied cubic phase gate using the resource state and gate-teleportation circuit tends to be noisy, and requires additional considerations. The detailed numerical analysis also seems to suggest that when the PNR detectors are replaced by threshold detectors, we do not obtain reasonable fidelities of the trained state with respect to the target state.

Acknowledgements — We thank Nathan Killoran, Koushik Ramachandran, Maria Schuld, Kamil Bradler, Casey Myers, Juan Miguel Arrazola, and Brajesh Gupt for useful discussions.

Software — All the numerical simulations were performed using open source software. The Strawberry Fields quantum simulator [42] is available at https://github.com/XanaduAI/strawberryfields. Our Python scripts for the optimization are built on Strawberry Fields and scipy package [44], and can be accessed at https://github.com/XanaduAI/constrained-quantum-learning.

[1] S. Lloyd and S. L. Braunstein, Phys. Rev. Lett. **82**, 1784 (1999), arXiv:9810082 [quant-ph].
[2] S. D. Bartlett, B. C. Sanders, S. L. Braunstein, and K. Nemoto, Phys. Rev. Lett. **88**, 4 (2002), arXiv:0109047 [quant-ph].
[3] D. Gottesman, A. Kitaev, and J. Preskill, Phys. Rev. A **64**, 123101 (2001), arXiv:0008040 [quant-ph].
[4] K. K. Sabapathy and C. Weedbrook, Phys. Rev. A **97**, 062315 (2018), arXiv:1802.05220.
[5] F. Arzani, N. Treps, and G. Ferrini, Phys. Rev. A **95**, 1 (2017), arXiv:1703.06693.
[6] K. Marshall, R. Pooser, G. Siopsis, and C. Weedbrook, Phys. Rev. A **91**, 032321 (2015), arXiv:1412.0336.
[7] P. Marek, R. Filip, H. Ogawa, A. Sakaguchi, S. Takeda, J.-i. Yoshikawa, and A. Furusawa, Phys. Rev. A **97**, 022329 (2018), arXiv:1708.02822.
[8] K. Miyata, H. Ogawa, P. Marek, R. Filip, H. Yonezawa, J.-i. Yoshikawa, and A. Furusawa, Phys. Rev. A **93**, 022301 (2016), arXiv:1507.08782.
[9] P. Marek, R. Filip, and A. Furusawa, Phys. Rev. A **84**, 1 (2011), arXiv:1105.4950.
[10] M. Yukawa, K. Miyata, H. Yonezawa, P. Marek, R. Filip, and A. Furusawa, Phys. Rev. A **88**, 053816 (2013).
[11] S. Ghose and B. C. Sanders, J. Mod. Opt. **54**, 855 (2007).
[12] J. Honer, R. Löw, H. Weimer, T. Pfau, and H. P. Büchler, Phys. Rev. Lett. **107**, 1 (2011), arXiv:1103.1319v1.
[13] P. Marek, J. Provazník, and R. Filip, arXiv:1808.08845 (2008).
[14] S. Rosenblum, O. Bechler, I. Shomroni, Y. Lovsky, G. Guendelman, and B. Dayan, Nat. Photonics **10**, 19 (2016), arXiv:1510.04042.
[15] M. Krenn, M. Malik, R. Fickler, R. Lapkiewicz, and A. Zeilinger, Phys. Rev. Lett **116**, 090405 (2016).
[16] A. A. Melnikov, H. P. Nautrup, M. Krenn, V. Dunjko, M. Tiersch, A. Zeilinger, and H. J. Briegel, Proceedings of the National Academy of Sciences , 201714936 (2018).
[17] J. Gao, L.-F. Qiao, Z.-Q. Jiao, Y.-C. Ma, C.-Q. Hu, R.-J. Ren, A.-L. Yang, H. Tang, M.-H. Yung, and X.-M. Jin, Phys. Rev. Lett **120**, 240501 (2018).
[18] L. Cincio, Y. Subaşı, A. T. Sornborger, and P. J. Coles, arXiv:1803.04114 (2018).
[19] J. M. Arrazola, T. R. Bromley, J. Izaac, C. R. Myers, K. Brádler, and N. Killoran, arXiv:1807.10781 (2018).
[20] Y. Miwa, J. I. Yoshikawa, P. Van Loock, and A. Furusawa, Phys. Rev. A **80**, 1 (2009), arXiv:0906.3141.
[21] K. Miyata, H. Ogawa, P. Marek, R. Filip, H. Yonezawa, J. I. Yoshikawa, and A. Furusawa, Phys. Rev. A **90**, 1 (2014), arXiv:1409.3754.
[22] J. I. Yoshikawa, Y. Miwa, A. Huck, U. L. Andersen, P. Van Loock, and A. Furusawa, Phys. Rev. Lett. **101**, 1 (2008), arXiv:0808.0551.
[23] R. Filip, P. Marek, and U. L. Andersen, Phys. Rev. A **71**, 1 (2005).
[24] R. Ukai, N. Iwata, Y. Shimokawa, S. C. Armstrong, A. Politi, J. I. Yoshikawa, P. Van Loock, and A. Furusawa, Phys. Rev. Lett. **106**, 1 (2011), arXiv:1001.4860.
[25] R. Ukai, S. Yokoyama, J. I. Yoshikawa, P. Van Loock, and A. Furusawa, Phys. Rev. Lett. **107**, 1 (2011), arXiv:1107.0514.
[26] S. Yokoyama, R. Ukai, S. C. Armstrong, J.-i. Yoshikawa, P. van Loock, and A. Furusawa, Phys. Rev. A **92**, 032304 (2015).
[27] M. Dakna, J. Clausen, L. Knöll, and D. G. Welsch, Phys. Rev. A **59**, 1658 (1999), arXiv:9807089 [quant-ph].
[28] M. Dakna, J. Clausen, L. Knöll, and D.-G. Welsch, Phys. Rev. A **60**, 726 (1999).
[29] M. Yukawa, K. Miyata, T. Mizuta, H. Yonezawa, P. Marek, R. Filip, and A. Furusawa, Opt. Express **21**, 5529 (2013), arXiv:1212.3396.

[30] J. Fiurášek, R. García-Patrón, and N. J. Cerf, Phys. Rev. A **72**, 1 (2005), arXiv:0503111 [quant-ph].
[31] F. Dell'Anno, S. De Siena, and F. Illuminati, Phys. Rep. **428**, 53 (2006), arXiv:0701050 [quant-ph].
[32] T. Gerrits, S. Glancy, T. S. Clement, B. Calkins, A. E. Lita, A. J. Miller, A. L. Migdall, S. W. Nam, R. P. Mirin, and E. Knill, Phys. Rev. A **82**, 1 (2010), 1004.2727.
[33] A. P. Lund, A. Laing, S. Rahimi-Keshari, T. Rudolph, J. L. OBrien, and T. Ralph, Phys. Rev. Lett. **113**, 100502 (2014).
[34] C. S. Hamilton, R. Kruse, L. Sansoni, S. Barkhofen, C. Silberhorn, and I. Jex, Phys. Rev. Lett. **119**, 170501 (2017).
[35] M. Ban, Journal of Modern Optics **43**, 1281 (1996).
[36] M. Dakna, L. Knöll, and D.-G. Welsch, Opt. Commun. **145**, 309 (1998), quant-ph/9703039.
[37] M. Dakna, L. Knöll, and D.-G. Welsch, Eur. Phys. J. D **3**, 295 (1998), quant-ph/9803077.
[38] M. Dakna, T. Anhut, T. Opatrný, L. Knöll, and D.-G. Welsch, Phys. Rev. A **55**, 3184 (1997), arXiv:quant-ph/9612011v1.
[39] J. Clausen, M. Dakna, L. Knöll, and D. Welsch, J. Opt. B **1**, 332 (1999).
[40] M. G. Paris, M. Cola, and R. Bonifacio, Phys. Rev. A **67**, 10 (2003).
[41] P. Knott, New J. Phys **18**, 073033 (2016).
[42] N. Killoran, J. Izaac, N. Quesada, V. Bergholm, M. Amy, and C. Weedbrook, preprint arXiv:1804.03159 (2018).
[43] K. E. Cahill and R. J. Glauber, Phys. Rev. **177**, 1857 (1969).
[44] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," (2001).
[45] D. J. Wales and J. P. K. Doye, J. Phys. Chem. A **101**, 5111 (1997).
[46] Z. Li and H. A. Scheraga, Proc. Natl. Acad. Sci. **84**, 6611 (1987).
[47] J.-F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal, *Numerical optimization: theoretical and practical aspects* (Springer Science & Business Media, 2006).
[48] K. K. Sabapathy and A. Winter, Phys. Rev. A **95**, 062309 (2017).
[49] Q. Zhuang, P. Shor, and J. Shapiro, Phys. Rev. A **97**, 052317 (2018).
[50] R. Takagi and Q. Zhuang, Phys. Rev. A **97**, 062337 (2018).
[51] F. Albarelli, M. G. Genoni, M. G. A. Paris, and A. Ferraro, Preprint arXiv:1804.05763 (2018).

Supplementary material

I. PHOTON-SUBTRACTION

The standard photon-subtraction probability is obtained from a high transmission beamsplitter where $U(\theta) = \exp[\theta(\hat{a}^\dagger\hat{b} - \hat{a}\hat{b}^\dagger)]$ with $\theta << 1$. In this limit we can expand the unitary operator to the first-order in gate strength to obtain $1+\theta(\hat{a}^\dagger\hat{b} - \hat{a}\hat{b}^\dagger)$. For an arbitrary state and vacuum state incident on the beamsplitter, we obtain $U(\theta)|\psi\rangle\,|0\rangle = N\,(|\psi\rangle\,|0\rangle - \theta\hat{a}\,|\psi\rangle\,|1\rangle)$, where $N = \sqrt{1+\theta^2\langle\hat{a}^\dagger a\rangle_\psi}$. If we measure a single photon in the second mode, the probability of a photon-subtraction on the input state is then given by $\theta^2 N^{-2}$. For beamsplitters with around 98% [27, 32] transmission strength, this results in a probability $\sim 10^{-2}$. The resource states such as the one we consider require three successive photon-subtractions which results in a net probability of 10^{-6}. If we use this latter probability in the Eq. (6) with $\epsilon \sim 10^{-3}$, we find that the required number of quantum gadgets is of the order of 10^5 which is unfeasible.

II. NUMERICAL TECHNIQUES

We discuss two optimization functions that were used for training the circuit parameters. The landscape of our loss functions usually have several local minima, which makes it hard for standard local optimization methods because there is a very strong dependency on the initial conditions. The first global search function that we used is called basinhopping which is described in Algorithm 2. Basinhopping is a stochastic algorithm which attempts to find the global minimum of a smooth scalar function [45]. The implementation we used in our simulation is from the `scipy` package.

Algorithm 2 basinhopping

```
procedure basinhopping($\boldsymbol{x}_0$, $f$, $args$, $niter$, $step\_size=1$):
    $\boldsymbol{x}_{\text{old}} \leftarrow$ local_search(loss, $\boldsymbol{x}_0$)
    for i in range($iter$) do
        $\Delta \leftarrow$ random$(0,1)$
        $\boldsymbol{x}_{\text{jump}} \leftarrow \boldsymbol{x}_{\text{old}} +$ $step\_size$ $\times\Delta$
        $\boldsymbol{x}_{\text{new}} \leftarrow$ local_minimize( $\boldsymbol{x}_{\text{jump}}$, loss, $args$)
        if accept($\boldsymbol{x}_{\text{new}}$, $\boldsymbol{x}_{\text{old}}$)== True then
            $\boldsymbol{x}_{\text{old}} \leftarrow \boldsymbol{x}_{\text{new}}$
        end if
    end for
end procedure
```

The algorithm is iterative with each cycle composed of the following features:

1. initialize the variables $\boldsymbol{x}_0$
2. perform local_minimize to minimize $f(\boldsymbol{x}, args)$ starting from $\boldsymbol{x}_0$, to reach a local minimum we call $\boldsymbol{x}_{\text{old}}$
3. randomly change the position of $\boldsymbol{x}_{\text{old}}$ with a tunable *step_size*
4. perform local_minimize starting from $\boldsymbol{x}_{\text{old}}$ to reach a local minimum we call $\boldsymbol{x}_{\text{new}}$
5. perform an acceptance test `accept`($\boldsymbol{x}_{\text{new}}$,$\boldsymbol{x}_{\text{old}}$): a simple rule could be that if $f(\boldsymbol{x}_{\text{new}}, args) < f(\boldsymbol{x}_{\text{old}}, args)$, $\boldsymbol{x}_{\text{old}} = \boldsymbol{x}_{\text{new}}$. However, the commonly used acceptance test is stochastic so as to maximize the likelihood of finding the global minimum. In the `scipy` implementation, the acceptance test used there is the Metropolis criterion of standard Monte Carlo algorithms [46], where the probability of acceptance is given by $\exp[-(f(\boldsymbol{x}_{\text{old}}, args) - f(\boldsymbol{x}_{\text{new}}, args))/T]$. Here T is a fictitious temperate to control the degree of randomness
6. go back to step 2 and repeat this process *niter* times

This global minimization method has been shown to be extremely efficient for a wide variety of problems in physics and chemistry. For a stochastic global heuristic there is no way to determine if the true global minimum has actually been found. In our simulation we set $niter = 40$, which is tuned to be able to produce reproducible results. *step_size* is set to be the default value from the `scipy` package. The algorithm for local_minimize can in principle be one of many options. In our simulation we choose sequential least squares programming (SLSP) [47] which seems to be the fastest one for our task among the local search algorithms available in `scipy`. When there are no constraints on the variables, SLSP reduces to the well-known Newton's method [47].

Remark. In principle the local_search function mentioned in Algorithm 1 can be any local minimization algorithm and in our simulation we use the well-known BFGS algorithm which is available from the `scipy` package. In principle, the local_search and the local_minimize algorithms can be the same. But we use different names due to the role they play in our main Algorithm. 1.

A. Second-optimization over probability

For our state preparation task, solely optimizing fidelity to the target state is insufficient. To have a scalable architecture the conditional probability of preparing the trained state needs to be sufficiently high as well. One possible strategy to achieve both high fidelity and probability is to train the circuit to maximize fidelity and then use that point as a seed to further optimize the probability. This is exactly what we did for the three-mode case (see Algorithm 1), where we found that the second optimization over probability did little harm to the fidelity.

However, this is not the case for two-mode circuit where we found that second optimization quickly deteriorates the pre-trained high fidelity. To tackle this problem, making considerations for the fact that the computation overhead for training two-mode circuit is moderate, we did a brute-force optimization over probability. That is, we repeat the basinhopping *nbh* times. We then pick out the global optimum, trained to optimize fidelity with the highest probability. In our simulation, we found that using $nbh = 20$ and $niter = 30$ is enough to obtain reproducible results. This procedure is summarized in Algorithm 3.

Algorithm 3 optimization of probability for the two-mode architecture

```
procedure prob_opt($nbh$):
    initialize $\boldsymbol{x}$_list, prob_list ← empty list
    for e in range($nbh$) do
        initialize $\boldsymbol{x}_0$
        $\boldsymbol{x}$ ← basinhopping($\boldsymbol{x}_0$, loss, $args$, $niter$)
        _ , prob, _ , _ ← objective($\boldsymbol{x}$)
        $\boldsymbol{x}$_list.append($\boldsymbol{x}$)
        prob_list.append(prob)
    end for
    MaxIndex ← max(prob_ls)
    save $\boldsymbol{x}$_list[MaxIndex]
end procedure
```

III. DETAILS OF TWO-MODE AND THREE-MODE ARCHITECTURES

Here we list a few additional details of the numerical simulations with regard to the two-mode and three-mode architectures.

A. Post-selection of the PNR detectors

We provide numerical reasons for our choice of post-selection of the PNR detectors. For the two-mode example we plot the fidelity and probability as a function of the post-selected PNR measurement in the first (left) plot of Fig. 7 for a fixed value of $a = 0.3$. We find that while the fidelity is increasing with higher photon measurement, the probability drops rapidly. As a reasonable trade-off we pick $m = 2$ which gives an increase with respect to the fidelity with a slight cost to the probability of producing the state.

In the second plot of Fig. 7 we compare the fidelity with respect to the target state parameter a for three settings, namely, (i) two-mode architecture with $m = 2$, (ii) three-mode architecture with $(m_1, m_2) = (1, 1)$, and (iii) three-mode architecture with $(m_1, m_2) = (1, 2)$.

There are two important findings that we wish to highlight. The first is that we achieve near-perfect fidelity for the three mode circuit for all values of $a \in [0.3, 1]$ when the PNR detectors are post-selected to values $(m_1, m_2) = (1, 2)$. The second interesting numerical observation is that the performance of the two-mode architecture with $m = 2$ is extremely close to the three-mode circuit with $(m_1, m_2) = (1, 1)$. In both cases a common feature is that the sum of the values of post-selected PNR detectors are the same. We anticipate that a deeper understanding is possible if we explore these aspects from the perspective of resource theory of non-Gaussian operations [48–51].

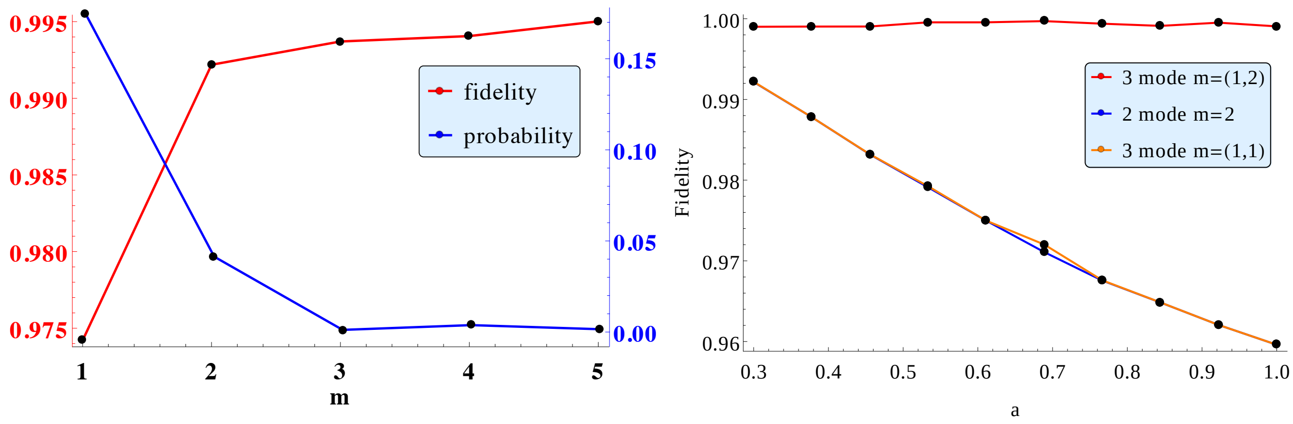

FIG. 7. Left panel: Plot of fidelity of the trained state to the target state, and probability of preparing the trained state as a function of post-selection of the PNR detectors in the two-mode circuit of Fig. 2. Here we use `basinhopping` with *niter* $= 20$ without further optimizing the probability. We make the choice of $m = 2$ which has a reasonable trade-off between fidelity and probability, both of which we would require to be high. Right panel: Comparison of the optimal fidelities for $m = (1, 1)$ and for $m = (1, 2)$ for the output state of the three-mode circuit, bench-marked with two-mode circuit with measurement $m = 2$, for various values of the target state parameter a. We find that measurement $m = (1, 1)$ has a very poor performance compared to using $m = (1, 2)$. For the three-mode case with $m = (1, 2)$ and $m = (1, 1)$, we use *niter* $= 40$ and *niter* $= 80$ respectively.

B. Trained parameters for two-mode circuits

The final optimization values for the two-mode circuit parameters is listed in Table.I. The fidelity of the trained state to the target state and its probability of production is presented in Fig. 3. We find that the required squeezing values r_1, r_2 are all ≤ 5.1 dB.

a	r_1	r_2	ϕ_1^r	ϕ_2^r	d_1	d_2	ϕ_1^d	ϕ_2^d	θ	ψ
0.3	0.37	-0.37	-3.08	0.59	-0.25	0.35	0.77	-0.98	-0.64	2.26
0.38	-0.43	-0.43	-1.72	-1.64	-0.36	-0.3	-2.05	-3.25	2.25	-1.93
0.46	-0.48	-0.48	-1.34	0.59	0.34	-0.39	-0.14	-0.93	2.49	2.85
0.53	-0.5	0.46	1.73	0.37	0.4	-0.29	-0.64	-1.2	0.97	-0.77
0.61	-0.54	-0.58	0.55	0.65	0.39	-0.41	0.75	2.28	-0.68	-1.26
0.67	-0.34	-0.48	3.1	0	-0.14	0.39	1.55	-1.57	0.61	-3.12
0.77	0.35	-0.51	0.85	0.01	0.14	-0.4	-1.14	1.57	0.63	2.72
0.84	0.28	-0.46	-0.57	0	0.07	0.38	4.42	-1.57	0.62	-2.86
0.92	-0.53	0.34	-1.84	0.02	-0.4	0.12	-2.49	7.9	-0.93	0.93
1	-0.23	0.43	-0.97	3.14	0.01	-0.38	-0.47	-1.57	3.76	-1.09

TABLE I. Optimal circuit parameters for the two-mode architecture with PNR $m = 2$. $\{r_i, \phi_i^r\}$ ($\{d_i, \phi_i^d\}$) are the magnitude and phase for squeezing (displacement) applied to the i-th vacuum mode. $\{\theta, \phi\}$ are the parameters for the beamsplitter.

C. Trained parameters for three-mode circuits

The final trained values of the three-mode circuit parameters is listed in Table.II. The fidelity of the trained state to the target state and its probability of production is presented in Fig. 3. We find that the required squeezing values r_1, r_2 are all ≤ 6.7 dB. Further, unlike the two-mode case earlier, it turns out that taking the displacements to be real provided better fidelity results.

a	r_1	r_2	r_3	ϕ_1^r	ϕ_2^r	ϕ_3^r	d_1	d_2	d_3	θ_1	θ_2	θ_3	ϕ_1	ϕ_2	ϕ_3
0.3	-0.27	0.65	0.66	-4.14	0.53	-1.94	0.38	-0.19	-0.47	2.29	-4.19	3.5	2.77	0.59	-0.2
0.38	-0.53	-0.75	-0.55	7.19	10.16	10.73	0.51	-0.03	0.53	2.3	-2.04	-4.11	-0.29	-1.99	-0.89
0.46	-0.75	-0.54	0.27	1.23	-5.31	1.53	-0.04	0.38	0.63	0.7	1.97	-0.88	-1.62	0.72	-1.79
0.53	0.71	0.67	-0.42	-2.07	0.06	-3.79	-0.02	0.34	0.02	-1.57	0.68	2.5	0.53	-4.51	0.72
0.61	0.72	0.65	-0.45	0.23	0.49	-3.81	0	-0.33	-0.01	-1.57	-2.46	0.63	-1.81	-0.22	6.42
0.67	0.52	0.56	0.74	0.09	3.35	-3.26	0.64	-0.33	0.02	-3.31	2.29	2.63	-3.33	1.41	-6.29
0.77	-0.55	0.73	0.03	1.14	0.98	-3.03	0.34	0.11	0.51	-2.3	-1.96	0.69	-3.22	4.32	-4.07
0.84	0.74	0.54	-0.48	0.5	-5.29	2.73	-0.01	-0.36	0.01	-1.52	0.7	-3.8	-0.95	-1.06	-1.24
0.92	-0.54	0.49	0.72	-0.44	2.36	3.14	0.15	-0.48	0.25	-1.3	2.27	0.67	-2.13	3.94	1.99
1	0.66	-0.38	-0.76	3.36	-0.73	0.4	-0.05	-0.82	0	1.56	-2.32	0.61	-3.67	-0.97	-0.71

TABLE II. Optimal circuit parameters for the three-mode architecture with PNR ($m_1 = 1, m_2 = 2$). $\{r_i, \phi_i^r\}$ are the magnitude and phase for squeezing applied to the i-th vacuum mode. d_i are the displacements applied to the i-th vacuum mode. It turns out that taking the displacements to be real gives rise to more stable solutions. $\{\theta_i, \phi_i\}$ correspond to the parameters for the beamsplitters.

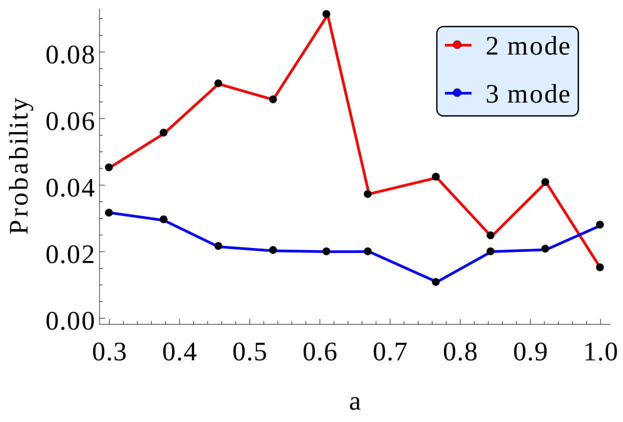

FIG. 8. Probability of preparing the trained states using the two-mode and three-mode architectures for various values of target state parameter a. Data here is obtained from Algorithm. 1 with *niter* = 40 and Algorithm. 3 with (*niter* = 30, *nbh* = 20). We find that for most cases the two-mode case performs better than the three-mode case, however, with a cost in the fidelity as mentioned in Fig. 2.

In Fig. 8 we plot the probability of generating the trained states in the two-mode and three-mode architectures. We find that in general the three-mode case outperforms the two-mode case. We wish to highlight that the number of steps for which we run the optimization algorithms are fixed for all values of state parameter a for the sake of reproducibility of the numerical results. For certain values of a an improvement in probability cannot be ruled out.

D. Generating random states

As a final numerical experiment we try to target a random state for each value of Fock state cutoff as outputs of the three-mode architecture with the same post-selected PNR detectors $(m_1, m_2) = (1, 2)$. in Fig. 9 we plot the average fidelity of the trained state to the target random state for various values of the cutoff dimension of the random state. As expected the average fidelity is monotonically decreasing with an increase in the output cutoff dimension of the target state.

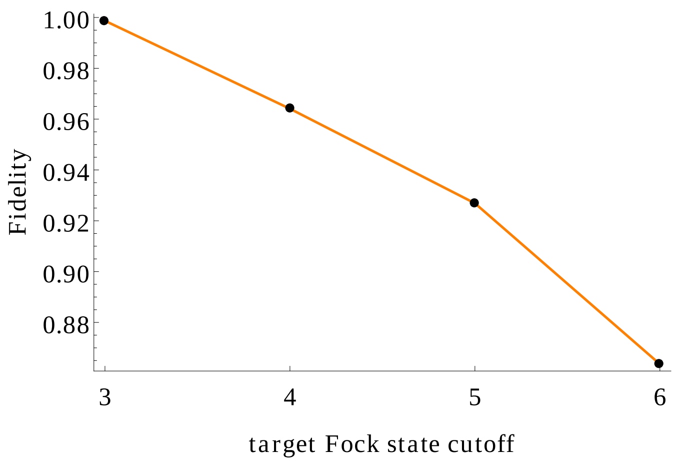

FIG. 9. Plot of the average fidelity of the trained state a target random state for various values of the Fock cutoff dimension. Here, the Fock cutoff dimension is one less than the physical cutoff dimension. We sampled 100 random states for each value of target cutoff dimension.