Quantum machine learning with adaptive linear optics

Ulysse Chabaud[1,2], Damian Markham[2,3], and Adel Sohbi[4]

[1]Université de Paris, IRIF, CNRS, France

[2]Laboratoire d'Informatique de Paris 6, CNRS, Sorbonne Université, 4 place Jussieu, 75005 Paris, France

[3]JFLI, CNRS, National Institute of Informatics, University of Tokyo, Tokyo, Japan

[4]School of Computational Sciences, Korea Institute for Advanced Study, Seoul 02455, Korea

We study supervised learning algorithms in which a quantum device is used to perform a computational subroutine—either for prediction via probability estimation, or to compute a kernel via overlap estimation of quantum states. We design implementations of these quantum subroutines using Boson Sampling architectures in linear optics, supplemented by adaptive measurements. We then challenge these quantum algorithms by deriving classical simulation algorithms for the tasks of output probability estimation and overlap estimation and we obtain different classical simulability regimes for these two tasks in terms of the number of adaptive measurements and input photons. In both cases, our results set explicit limits to the range of parameters for which a quantum advantage with adaptive linear optics over classical machine learning algorithms can be envisaged: we show that the number of input photons and the number of adaptive measurements cannot be simultaneously small compared to the number of modes. Interestingly, our analysis leaves open the possibility of a quantum advantage with a single adaptive measurement.

1 Introduction

Quantum computers promise dramatic advantages over their classical counterparts [1, 2], but a fault-tolerant universal quantum computer is still far from being available [3]. The quest for near-term quantum speedup has thus led to the introduction of various subuniversal models—models that are believed to have an intermediate com-putational power between classical and universal quantum computing—such as Boson Sampling [4] or IQP circuits [5], recently culminating with the experimental demonstration of random circuit sampling [6].

Finding practical applications for these subuniversal models, other than the demonstration of quantum speedup, is a timely issue, as it may enable interesting quantum advantages in the era of Noisy Intermediate-Scale Quantum devices [3].

Recently, there has been an increased interest on the possibility of enhancing classical machine learning algorithms using quantum computers, which includes the development of quantum neural networks [7–15] and the development of quantum kernel methods [16–21].

In particular, recent proposals have been driven by subuniversal models such as Gaussian Boson Sampling [22, 23] or IQP circuits [5, 16]. In the latter, the authors considered supervised learning algorithms in which some computational subroutines are executed in a quantum way, namely the estimation of the output probabilities of quantum circuits, or the estimation of the overlap of the output states of quantum circuits. They showed that IQP circuits alone could not provide a quantum advantage for these subroutines and therefore considered minimal extensions of these circuits, in terms of circuit depth.

Hereafter, we study the use of Boson Sampling interferometers [4], with input photons, for similar quantum machine learning tasks. Instead of extending the depth, we allow for adaptive measurements—intermediate measurements that drive the rest of the computation—which provide a natural analogy with the circuit depth in the linear optics picture [24]: by encoding qubits into single-photons and using a sufficient number of adaptive measurements, one can perform universal quantum computing [25].

We give a detailed prescription for performing

quantum machine learning with classical data, using adaptive linear optics for computational subroutines such as probability estimation and overlap estimation.

We also examine the classical simulability of these quantum subroutines. More precisely, we give classical simulation algorithms whose runtimes are explicitly dependent on: (i) the number of modes m, (ii) the number of adaptive measurements k, (iii) the number of input photons n and (iv) the number of photons r detected during the adaptive measurements. This effectively sets a limit on the range of parameters for which adaptive linear optics may provide an advantage for machine learning over classical computers using our methods, thus identifying the regimes where a quantum advantage can be envisaged. Boson Sampling instances correspond to the case with no adaptive measurement, while the Knill–Laflamme–Milburn scheme for universal quantum computing [25] corresponds to the case where the number of adaptive measurements scales linearly with the size of the computation.

For probability estimation, we show that the classical simulation is efficient whenever the number of adaptive measurements or the number of input photons is constant. Moreover, the number of input photons and the number of adaptive measurements cannot be simultaneously small compared to the number of modes (see Table 2).

For overlap estimation, we show a similar behaviour, although in this case our results do not rule out the possibility of a quantum advantage with a single adaptive measurement (see Tables 3 and 4). Our main technical contribution is an expression for the inner product of the output states of two adaptive unitary interferometers which is essentially independent of the number of adaptive measurements.

The rest of the paper is organised as follows. In section 2, we provide a background on quantum machine learning with classical data and classical simulation of quantum computations. In section 3, we introduce the model of adaptive linear optics which we consider. We give a prescription for performing probability estimation and overlap estimation with instances of this model, and detail how to use these as subroutines for machine learning problems. In section 4, we derive two classical simulation algorithms, one for each of these two tasks, and analyze the running time

of these algorithms. We conclude in section 5.

2 Background

2.1 Kernel methods for quantum machine learning

2.1.1 Encoding classical data with quantum states

We consider a typical machine learning problem, such as classification problems, where a classical dataset $\mathcal{D} = \{\vec{x}_1, \ldots, \vec{x}_{|\mathcal{D}|}\}$ from an input set $\mathcal{X}$ is given. One method to use quantum computers to solve such problems is to encode classical data onto quantum states such that there exist a so-called *feature map* $\vec{x}_l \rightarrow |\phi(\vec{x}_l)\rangle$ which can be processed by a quantum computer.

Definition 1 (Feature map [17]). *Let $\mathcal{F}$ be a Hilbert space, called feature space, $\mathcal{X}$ an input set and $\vec{x} \in \mathcal{X}$. A feature map is a map $\phi : \mathcal{X} \rightarrow \mathcal{F}$ from inputs to vectors in the Hilbert space.*

Many machine learning algorithms perform well in linear cases such as the support vector machines (SVM) which will be use in this article. However, many real world problems require non-linearity to make successful predictions. By using *kernel methods* one can use estimation methods that are linear in terms of the kernel evaluations.

Definition 2 (Kernel [17]). *Let $\mathcal{X}$ be a non empty set called input set. A function $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{C}$ is called a kernel if for any finite subset $\mathcal{D} = \{\vec{x}_1, \ldots, x_{|\mathcal{D}|}\}$ with $M \geq 2$ the Gram matrix K with entries $K_{l,l'} = \kappa(\vec{x}_l, \vec{x}_{l'})$ is positive semidefinite.*

The kernel corresponds to a dot product in a feature space (here in a high-dimensional Hilbert space). In [17], it is shown that the notion of *feature map* in Hilbert space and *kernel* can be connected. A straightforward way is to define a kernel K from a feature map ϕ as follows:

$$\kappa(\vec{x}_l, \vec{x}_{l'}) = \langle \phi(\vec{x}_l)|\phi(\vec{x}_{l'})\rangle_{\mathcal{F}}, \qquad (1)$$

where $x_m \in \mathcal{X}$, $x_{m'} \in \mathcal{X}$ and $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ is the inner product over the Hilbert space $\mathcal{F}$.

2.1.2 Using Feature Hilbert Spaces for Machine Learning

There are two main ways to use feature Hilbert spaces:

- The *quantum variational classification* [16] or *explicit approach* [17]: the entire model computation is on a quantum device trained by a hybrid variational quantum-classical or classical algorithms [12, 13, 26]. In this case the probability distribution over the possible outcomes is used for the classification. Hence, while the feature map is used to encode the data, the kernel is not known directly from the quantum device. In [16], the probability to obtain a certain binary output b_i for the classical data input x is given by:

$$\Pr(b_i) = |\langle b_i | W(\theta) U_{\phi(x)} | 0^{\otimes n} \rangle|^2, \quad (2)$$

where $U_{\phi(x)}$ encodes the feature map and $W(\theta)$ corresponds to the quantum SVM.

- The *quantum kernel estimation* [16] or *implicit approach* [17]: is a quantum-assisted method where the quantum device is only used to evaluate the kernel and the rest of the machine learning algorithm is carried on by a classical algorithm. In [16], the kernel between two classical data x_m and $x_{m'}$ is given by:

$$K_{m,m'} = \langle 0^{\otimes n} | U^{\dagger}_{\phi(x_m)} U_{\phi(x_{m'})} | 0^{\otimes n} \rangle. \quad (3)$$

In order to challenge and justify the use of a quantum assisted method it is important to consider how hard would it be for a classical machine to compute the same quantities. Indeed, if the output probability or kernel could be computed directly classically, there would be no need for a quantum computer for this task. Hence, when it comes to the classical hardness, the first method requires that estimating the output probability in Eq. (2) is hard, while the second method requires that estimating the overlap in Eq. (3) is hard (for classical computers). We give formal definitions for these classical simulation tasks in the following section.

2.2 Classical simulation of quantum computations

Depending on the approach used for simulating classically the functioning of quantum devices, several notions of simulability are commonly used. One example is to ask the classical simulation algorithm to mimic the output of the quantum computation [27, 28]. Informally,

a quantum computation is *weakly simulable* if there exists a classical algorithm which outputs samples from its output probability distribution in time polynomial in the size of the quantum computation. Various relaxations of this definition are possible, allowing the classical sampling to be approximate rather than exact, or to abort with a small probability. The existence of such an efficient classical simulation has been ruled out for various subuniversal models of quantum computing, such as IQP circuits [5] or Boson Sampling [4], under complexity-theoretic conjectures—both in the exact case and the approximate case, up to additional conjectures.

While weak simulation of quantum computations is arguably the most commonly studied, other notions of classical simulation may be useful: if the output samples of a quantum computation are used to compute a quantity which may be computed efficiently classically by other means, it is no longer necessary to simulate the whole quantum device. We consider two concrete examples which are prominent for variational quantum algorithms in quantum machine learning: probability estimation and overlap estimation [16, 17].

Definition 3 (Probability estimation)**.** *Let P be a probability distribution over M outcomes and let $\epsilon, \delta > 0$. Given any outcome $\boldsymbol{x}$ in the sample space of P, probability estimation refers to the computational task of outputting an estimate $\tilde{P}[\boldsymbol{x}]$ such that*

$$\left| \tilde{P}[\boldsymbol{x}] - P[\boldsymbol{x}] \right| \leq \epsilon, \quad (4)$$

with probability greater than $1 - \delta$, in time $O(\text{poly}(\frac{M}{\epsilon}) \log \frac{1}{\delta})$.

Efficient probability estimation amounts to outputting an estimate of the probability of a fixed outcome with an additive error polynomially small and with exponentially small probability of failure, in polynomial time in the size of the computation. One may use the samples from a quantum computation in order to perform efficiently probability estimation for any given outcome: given a quantum device of size M which outputs samples from some probability distribution and a fixed outcome $\boldsymbol{x}$ in the sample space, one may run the device poly M times, recording the value 1 whenever the outcome $\boldsymbol{x}$ is obtained and the value 0 otherwise. Then, the frequency of the outcome $\boldsymbol{x}$ over the poly M uses of the

quantum device is a polynomially precise additive estimate of the probability of the outcome $\boldsymbol{x}$ with exponentially small probability of failure, by virtue of Hoeffding inequality [29].

Weak simulation is at least as hard as probability estimation, since by the previous reasoning one may obtain polynomially precise additive estimates of probabilities from samples of the probability distribution. Moreover, they are some quantum computations for which weak simulation is hard for classical computers (assuming widely believed conjectures from complexity theory), but probability estimation can be done efficiently classically. This is the case for IQP circuits [5, 16], Boson Sampling [4] and even the period-finding subroutine of Shor's factoring algorithm [2]. We detail the latter case in Appendix A, for the sake of clarifying the relations between these different notions of classical simulation.

A more general computational task than probability estimation in the context of quantum computing is the following:

Definition 4 (Overlap estimation). *Let $|\phi\rangle$ and $|\psi\rangle$ be quantum output states of two quantum computations efficiently describable of size M and let $\epsilon, \delta > 0$. Overlap estimation refers to the computational task of outputting an estimate $\tilde{O}$ such that*

$$\left| \tilde{O} - |\langle \phi|\psi\rangle|^2 \right| \leq \epsilon, \qquad (5)$$

with probability greater than $1 - \delta$, in time $O(\text{poly}\left(\frac{M}{\epsilon}\right) \log \frac{1}{\delta})$.

The overlap between two quantum states is a measure of their distinguishability [30] and overlap estimation thus is related to quantum state discrimination. Several techniques exist to perform quantumly the overlap estimation of two states $|\phi\rangle$ and $|\psi\rangle$ [31]. One of them is to perform the swap test [32] with various copies of both states.

Overlap estimation can be also done efficiently classically for IQP circuits. Nonetheless, this family of circuits has been identified as a promising venue for implementing quantum machine learning algorithms [16], when enlarged to contain similar circuits with bigger depth. Motivated by this approach, we consider hereafter the case of another subuniversal model: Boson Sampling [4] with input photons, supplemented with adaptive measurements.

3 Quantum machine learning with adaptive linear optics

In what follows, we study Boson Sampling architectures [4], supplemented with a given number of adaptive measurements—that is, some of the modes are measured throughout the computation and the rest of the computation can depend on their outcome—which we refer to as adaptive linear optics. We derive in this section quantum algorithms for performing probability and overlap estimation with adaptive linear optics, together with a prescription for utilising these algorithms as subroutines in supervised learning algorithms.

3.1 Adaptive linear optics

Hereafter, we detail the computational model of adaptive linear optics which we consider. We first introduce a few notations.

In the linear optics picture, the input states we consider are multimode photon number Fock states over m modes (we use bold math for multi-index notations, see Table 1):

$$|\boldsymbol{s}\rangle = \frac{1}{\sqrt{\boldsymbol{s}!}} \hat{a}_1^{\dagger s_1} \ldots \hat{a}_m^{\dagger s_m} |0\rangle^{\otimes m}, \qquad (6)$$

where s_i and $\hat{a}_i^\dagger$ are respectively the number of photons and the creation operator for the ith mode. We identify these states with m-tuples of integers $\boldsymbol{s} = (s_1, \ldots, s_m) \in \mathbb{N}^m$. Following [4], let us define, for all $n \in \mathbb{N}$,

$$\Phi_{m,n} := \{\boldsymbol{s} = (s_1, \ldots, s_m) \in \mathbb{N}^m, \ |\boldsymbol{s}| = n\}. \quad (7)$$

This set corresponds to the m-mode Fock states with total number of photons equal to n, and we have $|\Phi_{m,n}| = \binom{m+n-1}{n}$.

We consider a unitary interferometer of size m, described by an $m \times m$ unitary matrix $U = (u_{ij})_{1 \leq i,j \leq m}$. Unlike in the circuit picture, the matrix U does not act on the computational basis, which is the infinite multimode Fock basis, but rather describes the linear evolution of the creation operator of each mode. More precisely,

$$\begin{pmatrix} \hat{a}_1^\dagger \\ \vdots \\ \hat{a}_m^\dagger \end{pmatrix} \mapsto U \begin{pmatrix} \hat{a}_1^\dagger \\ \vdots \\ \hat{a}_m^\dagger \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^m u_{1k}\hat{a}_k^\dagger \\ \vdots \\ \sum_{k=1}^m u_{mk}\hat{a}_m^\dagger \end{pmatrix}. \quad (8)$$

We write $\hat{U}$ instead the unitary action of the interferometer on the multimode Fock basis. Be-

$\|\boldsymbol{s}\|$	$s_1 + \cdots + s_m$
$\boldsymbol{s}!$	$s_1! \cdots s_m!$
$\|\boldsymbol{s}\rangle$	$\|s_1 \ldots s_m\rangle$
$\boldsymbol{s} + \boldsymbol{t}$	$(s_1 + t_1, \ldots, s_m + t_m)$
$\boldsymbol{0}^m$	$(0, \ldots, 0)$
$\boldsymbol{1}^m$	$(1, \ldots, 1)$

Table 1: Multi-index notations. For all $m \in \mathbb{N}^*$, we write $\boldsymbol{s} = (s_1, \ldots, s_m) \in \mathbb{N}^m$ and $\boldsymbol{t} = (t_1, \ldots, t_m) \in \mathbb{N}^m$.

cause the interferometer conserves the total number of photons, for all $p, q \in \mathbb{N}$, all $\boldsymbol{s} \in \Phi_{m,p}$ and all $\boldsymbol{t} \in \Phi_{m,q}$

$$\langle \boldsymbol{s}|\hat{U}|\boldsymbol{t}\rangle = 0 \qquad (9)$$

whenever $p \neq q$. Let $n \in \mathbb{N}$ and $\boldsymbol{s} = (s_1, \ldots, s_m) \in \Phi_{m,n}$ and $\boldsymbol{t} = (t_1, \ldots, t_m) \in \Phi_{m,n}$. Combining Eq. (6) and Eq. (8) we obtain [4]

$$\langle \boldsymbol{s}|\hat{U}|\boldsymbol{t}\rangle = \frac{\mathrm{Per}(U_{\boldsymbol{s},\boldsymbol{t}})}{\sqrt{\boldsymbol{s}!}\sqrt{\boldsymbol{t}!}}, \qquad (10)$$

where $U_{\boldsymbol{s},\boldsymbol{t}}$ is the $n \times n$ matrix obtained from U by repeating s_i times its ith row and t_j times its jth column for $i, j = 1, \ldots, m$, and where the permanent of a $r \times r$ matrix $A = (a_{ij})_{1 \le i,j \le r}$ is defined as

$$\mathrm{Per}\, A = \sum_{\sigma \in \mathcal{S}_r} \prod_{i=1}^{r} a_{i\sigma(i)}, \qquad (11)$$

where $\mathcal{S}_r$ is the symmetric group over $\{1, \ldots, r\}$.

We write $\mathrm{Pr}_{m,n}[.|\boldsymbol{t}]$ the probability distribution of the outputs over $\Phi_{m,n}$ of the unitary interferometer U acting on an input $|\boldsymbol{t}\rangle$. With the previous notations we obtain, for all $p, q \in \mathbb{N}$, all $\boldsymbol{s} \in \Phi_{m,p}$, and all $\boldsymbol{t} \in \Phi_{m,q}$,

$$\mathrm{Pr}_{m,n}[\boldsymbol{s}|\boldsymbol{t}] = \frac{|\mathrm{Per}(U_{\boldsymbol{s},\boldsymbol{t}})|^2}{\boldsymbol{s}!\boldsymbol{t}!}\delta_{pq}. \qquad (12)$$

In what follows, we fix the input state $|\boldsymbol{t}\rangle = |\boldsymbol{1}^n \boldsymbol{0}^{m-n}\rangle$, with single-photon states in the first n modes, and vacuum states in all other modes, where the superscript indicates the size of the string $(0, \ldots, 0)$ or $(1, \ldots, 1)$ when there is a possible ambiguity. We consider the case of linear optical quantum computing with adaptive photon-number measurements, which we refer to

as adaptive linear optics (Fig. 1). We denote by $k \in \{0, \ldots, m\}$ the number of single-mode adaptive measurements. Without loss of generality, we assume that the first k modes are adaptively measured throughout the computation, and we write $\boldsymbol{p} = (p_1, \ldots, p_k)$ the adaptive measurement outcomes. For $r \in \mathbb{N}$ and $\boldsymbol{p} \in \Phi_{k,r}$, let us define

$$U^{\boldsymbol{p}} := [\mathbb{1}_k \oplus U_k(p_1, \ldots, p_k)] \ldots [\mathbb{1}_1 \oplus U_1(p_1)]\, U_0, \qquad (13)$$

where $\mathbb{1}_j$ is the identity matrix of size j and where the unitary matrices U_j depend on the measurement outcomes $p_1, \ldots, p_j$ for all $j \in \{1, \ldots, k\}$. An adaptive interferometer $\mathcal{U}$ over m modes with n input photons and k adaptive measurements is then represented as a family of nonadaptive unitary interferometers

$$\mathcal{U} := \{U^{\boldsymbol{p}}|\boldsymbol{p} \in \Phi_{k,r}, 0 \le r \le n\}, \qquad (14)$$

for each possible adaptive measurement outcome $\boldsymbol{p}$. The matrix $U^{\boldsymbol{p}}$ describes the interferometer in Fig. 1, when the adaptive measurement outcome $\boldsymbol{p} = (p_1, \ldots, p_k) \in \Phi_{k,r}$ has been obtained. In this case, the output state is a pure state which reads:

$$\mathrm{Tr}_k\left[(|\boldsymbol{p}\rangle\langle\boldsymbol{p}| \otimes \mathbb{1}_{m-k})\hat{U}^{\boldsymbol{p}}\,|\boldsymbol{t}\rangle\langle\boldsymbol{t}|\,\hat{U}^{\boldsymbol{p}\dagger}\right], \qquad (15)$$

where the partial trace is over the first k modes and where $|\boldsymbol{p}\rangle$ denotes the k-mode Fock state $|p_1 \ldots p_k\rangle$. At the end of the computation, all the remaining $m - k$ modes are measured with photon-number detection, yielding the final outcome $\boldsymbol{s} = (s_1, \ldots, s_{m-k}) \in \Phi_{m-k,n-r}$.

3.2 Quantum probability and overlap estimation

We now detail how to perform probability estimation or overlap estimation with an adaptive linear optical interferometer as described in the previous section. These tasks may be performed in particular as subroutines for machine learning algorithms, as detailed in the next section.

For estimating an output probability with a quantum circuit, one may run the circuit $O(\mathrm{poly}\, m)$ times, obtaining classical outcomes, for which the frequency gives a polynomially precise additive estimate of the probability which can be computed efficiently. In the case of a circuit with adaptive measurements, one only looks at the final measurement outcomes and the same holds for adaptive linear optical computations.
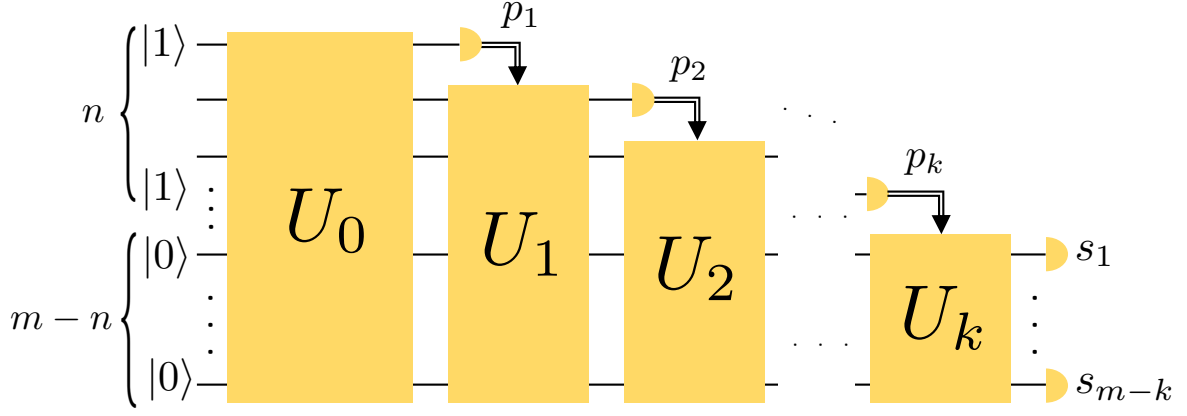
Figure 1: Linear optical computing model with k adaptive measurements and input state $|1\dots10\dots0\rangle$, with n photons over m modes. The output modes are measured using photon number detection. For all $j \in \{1, \dots, k\}$, the unitary interferometer U_j, acting on $m - j$ modes, depends on the measurement outcomes $p_1, \dots, p_j$. The adaptive measurement outcomes $p_1, \dots, p_k$ are used to drive the computation, whose final outcome is $s_1, \dots, s_{m-k}$.

For estimating the overlap of output states of two unitary quantum circuits, one may run both circuits in parallel and compare their quantum output states, for example with the swap test [32]. Doing so a polynomial number of times provides a polynomially precise estimate of the overlap. Alternatively, writing C_1 and C_2 the unitary circuits, one may build the circuit $C_1 C_2^\dagger$ and project the output quantum state onto the input state.

In the case of circuits with adaptive measurements, there is an output state for each adaptive measurement outcome. In particular, if the number of possible adaptive measurement outcomes is exponential in the size of the computation, then the probability distribution for these outcomes has to be concentrated on a polynomial number of events for the quantum overlap estimation to be efficient. This is because in order to compute a polynomially precise estimate of the overlap, say, $|\langle \phi|\psi\rangle|^2$, the states $|\phi\rangle$ and $|\psi\rangle$, both corresponding to specific adaptive measurement results, have to be obtained a polynomial number of times.

For adaptive linear optics over m modes with n input photons and k adaptive measurements, the number of possible adaptive measurement outcomes is given by

$$\sum_{r=0}^{n} |\Phi_{k,r}| = \sum_{r=0}^{n} \binom{k + r - 1}{r} = \binom{n+k}{n}, \tag{16}$$

where the sum is over the total number of pho-

tons detected at the stage of the adaptive measurements. Hence, for overlap estimation to be efficient, either the probability distribution for the adaptive measurements outcomes is concentrated on a polynomial number of outcomes, or $\binom{n+k}{n} = O(\text{poly } m)$, which is the case for example when $n = O(1)$ and $k = O(m)$, $n = O(\log m)$ and $k = O(\log m)$, or $n = O(m)$ and $k = O(1)$. In what follows, we do not assume concentration of the adaptive measurement outcome probability distribution and consider general interferometers with adaptive measurements. In this context, the quantum efficient regime for overlap estimation thus corresponds to $\binom{n+k}{n} = O(\text{poly } m)$.

Let $\mathcal{U} = \{U^{\boldsymbol r}|\boldsymbol r \in \Phi_{k,r}, \, 0 \le r \le n\}$ be an adaptive linear interferometer with n input photons and k adaptive measurements and let $|\phi\rangle$ and $|\psi\rangle$ be two output states. Let $\boldsymbol p$ and $\boldsymbol q$ denote the outcomes of the adaptive measurements for $|\phi\rangle$ and $|\psi\rangle$, respectively, so that $U^{\boldsymbol p}$ is the interferometer for $|\phi\rangle$ and $U^{\boldsymbol q}$ is the interferometer for $|\psi\rangle$, with input Fock state $|\boldsymbol t\rangle$. With Eq. (15) we obtain

$$\begin{aligned}
&|\langle\phi|\psi\rangle|^2 \\
&= \text{Tr}\left[\text{Tr}_k[(|\boldsymbol p\rangle\langle\boldsymbol p| \otimes \mathbb{1}_{m-k})\hat U^{\boldsymbol p}|\boldsymbol t\rangle\langle\boldsymbol t|\hat U^{\boldsymbol p\dagger}]|\psi\rangle\langle\psi|\right] \\
&= \text{Tr}\left[(|\boldsymbol p\rangle\langle\boldsymbol p| \otimes \mathbb{1}_{m-k})\hat U^{\boldsymbol p}|\boldsymbol t\rangle\langle\boldsymbol t|\hat U^{\boldsymbol p\dagger}(\mathbb{1}_k \otimes |\psi\rangle\langle\psi|)\right] \\
&= \text{Tr}\left[\hat U^{\boldsymbol p}|\boldsymbol t\rangle\langle\boldsymbol t|\hat U^{\boldsymbol p\dagger}(|\boldsymbol p\rangle\langle\boldsymbol p| \otimes |\psi\rangle\langle\psi|)\right] \\
&= \text{Tr}\left[|\boldsymbol t\rangle\langle\boldsymbol t|\hat U^{\boldsymbol p\dagger}(|\boldsymbol p\rangle\langle\boldsymbol p| \otimes |\psi\rangle\langle\psi|)\hat U^{\boldsymbol p}\right],
\end{aligned} \tag{17}$$

Because of the conservation of the total number of photons, the overlap between the states $|\phi\rangle$ and

$|\psi\rangle$ is zero if $|\boldsymbol{p}| \neq |\boldsymbol{q}|$.

Otherwise, it can be estimated using a polynomial number of copies of the state $|\psi\rangle$ as follows: send the input $|\boldsymbol{p}\rangle \otimes |\psi\rangle$ into the interferometer with unitary matrix $U^{\boldsymbol{p}\dagger}$ and measure the photon number in each output mode; record the value 1 if the measurement pattern matches the Fock state $|\boldsymbol{t}\rangle$ and the value 0 otherwise. Then, the mean of the obtained values yields a polynomially precise estimate of the overlap $|\langle\phi|\psi\rangle|^2$ by Eq. (17) and Hoeffding inequality.

Note that this overlap estimation requires the preparation of the Fock state $|\boldsymbol{p}\rangle$ but no adaptive measurements. By symmetry, one could estimate the overlap alternatively using a polynomial number of copies of the state $|\phi\rangle$ and preparing the Fock state $|\boldsymbol{q}\rangle$, or by mixing copies of $|\phi\rangle$ and $|\psi\rangle$. In practice, one may run the adaptive interferometer $\mathcal{U}$ and apply the above procedure for $|\phi\rangle$ or $|\psi\rangle$ "on the fly", depending on whether the adaptive measurement outcome obtained is equal to $\boldsymbol{q}$ or $\boldsymbol{p}$ (see Algorithm 1 for overlap estimation including this state preparation step).

3.3 Support Vector Machine with adaptive linear optics

3.3.1 Support vector machine with quantum kernel methods

We consider the training dataset $\mathcal{T}$, with $|\mathcal{T}|$ points of the following form $\{(\vec{x}_1, y_1), \ldots, (\vec{x}_{|\mathcal{T}|}, y_{|\mathcal{T}|})\}$, where $\vec{x}_l \in \mathbb{R}^d$, $y_l \in C$, $\forall l \in \{1 \ldots, |\mathcal{T}|\}$ and where $C = \{-1, 1\}$ in the case of binary classification.

We also consider a feature map in the form of unitary operators that is a subset of the family introduced in Eq. (14): $\mathcal{U}_\phi = \{U_l^{\boldsymbol{p}_l}\}_l$ in which for each classical data x_l there is a unitary operator $U_l^{\boldsymbol{p}_l}$.

The idea of Support Vector Machine (SVM) [33–35] is to find the maximum-margin hyperplane that divides the group of points by their y values. Such hyperplane is defined with a vector $\vec{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$. With the hard-margin condition we have,

$$y_l(\vec{w}.\vec{x}_l + b) \geq 1, \quad \forall l \in \{1 \ldots, |\mathcal{T}|\}. \quad (18)$$

In this case, a decision function over a new point $\vec{x}$ can be constructed directly from the hyperplane as follows:

$$f(\vec{x}) = \text{sign}(\vec{w}.\vec{x} + b). \quad (19)$$

Algorithm 1: Overlap estimation for adaptive linear optics

Input: Adaptive interferometer $\mathcal{U} = \{U^{\boldsymbol{r}} | \boldsymbol{r} \in \Phi_{k,r}, 0 \leq r \leq n\}$, adaptive measurement outcomes $\boldsymbol{p}, \boldsymbol{q}$.
Parameters: Input $|\boldsymbol{t}\rangle$ and number of shots T.
Set the state preparation counter $c_{sp} = 0$.
Set the overlap counter $c_{over} = 0$.
while $c_{sp} < T$ **do**
  Run $\mathcal{U}$ on input $|\boldsymbol{t}\rangle$, obtaining adaptive measurement outcome $\boldsymbol{r}$ and output state $|\chi\rangle$.
  if $\boldsymbol{r} = \boldsymbol{p}$ **then**
    $c_{sp} \to c_{sp} + 1$.
    Run $U^{\boldsymbol{q}\dagger}$ on input $|\boldsymbol{q}\rangle \otimes |\chi\rangle$.
    Measure in photon number basis.
    Record measurement outcome $\boldsymbol{s}$.
    if $\boldsymbol{s} = \boldsymbol{t}$ **then**
      $c_{over} \to c_{over} + 1$,
    else
      $c_{over} \to c_{over}$.
  else
  if $\boldsymbol{r} = \boldsymbol{q}$ **then**
    $c_{sp} \to c_{sp} + 1$.
    Run $U^{\boldsymbol{p}\dagger}$ on input $|\boldsymbol{p}\rangle \otimes |\chi\rangle$.
    Measure in photon number basis.
    Record measurement outcome $\boldsymbol{s}$.
    if $\boldsymbol{s} = \boldsymbol{t}$ **then**
      $c_{over} \to c_{over} + 1$,
    else
      $c_{over} \to c_{over}$.
  else
return c_{over}/T.

When we have access to a feature map ϕ, the vector $\vec{w}$ can be also written as

$$\vec{w} = \sum_{l \in \{1 \ldots, |\mathcal{T}|\}} \alpha_l y_l \phi(x_l), \quad (20)$$

and the decision function becomes

$$f(\vec{x}) = \text{sign}\left(\sum_{l=1}^{|\mathcal{T}|} \alpha_l y_l \kappa(\vec{x}_l, \vec{x}) + b\right), \quad (21)$$

where the kernel $\kappa(\vec{x}_l, \vec{x})$ has an interpretation in terms of the feature map ϕ as given in Eq. (1).

In order to find the best hyperplane, one needs to use a quadratic programming solver. In Sec. 3.3.2, we consider the case where

the programming solver is composed of a hybrid quantum-classical algorithm whereas in Sec. 3.3.3, such solver will be external to the quantum device. The optimisation program that needs to be solved is:

$$\max L_D = \sum_{l=1}^{|\mathcal{T}|} \alpha_l - \frac{1}{2} \sum_{l,l'} \alpha_l y_l \alpha_{l'} y_{l'} \kappa(\vec{x}_l, \vec{x}_{l'})$$

$$\text{s.t.} \sum_{l=1}^{|\mathcal{T}|} \alpha_l y_l = 0$$

$$0 \leq \alpha_l \leq 1/2n\lambda, \quad \forall l \in \{1 \ldots, |\mathcal{T}|\}, \tag{22}$$

where λ is a parameter that is introduced for the soft-margin condition. By solving such problem, one obtains the coefficients $\{\alpha_l^*\}$ and $b*$ of the hyperplane (see Eq. (20)) that are optimum.

3.3.2 Explicit method: probability estimation

The explicit method corresponds to the case where the prediction is obtained from the probability distribution when using a quantum device. Inspired by [16], we propose to use adaptive linear optics for the feature map and use its output state as an input state of a second quantum device for the SVM. We show that Boson Sampling can be used to realize a SVM algorithm by using a hybrid variational quantum-classical for the training phase [12, 13, 26].

We write the Boson Sampling interferometer operation $BS(\vec{\theta})$, where $\vec{\theta}$ are the parameters of the beam splitters and phase shifters of the interferometer, since any interferometer can be implemented efficiently with only phase shifters and beam splitters [36]. In order to make a prediction we need to bin the outcomes. For a given function $g : \Phi_{m,n} \to \{-1, +1\}$, we can write the following observable for the binning:

$$\mathcal{N} = \sum_{s \in \Phi_{m,n}} g(s) |\boldsymbol{s}\rangle \langle \boldsymbol{s}|. \tag{23}$$

The observable $\mathcal{N}$ can also be decomposed in term of projectors as $\mathcal{N} = \Pi_{(+1)} - \Pi_{(-1)}$ and $\Pi_y = (\mathbb{1} + y\mathcal{N})/2$.

For a given data point $\vec{x}$ with the associated operation $\hat{U}_x^{\boldsymbol{p}_x}$, the probability to obtain the outcome y is:

$$\Pr(y|\boldsymbol{p}_x) = \text{Tr}\left[\Pi_y BS(\vec{\theta})\hat{U}_x^{\boldsymbol{p}_x} |\boldsymbol{t}\rangle \langle \boldsymbol{t}| \hat{U}_x^{\boldsymbol{p}_x\dagger} BS(\vec{\theta})^\dagger\right]$$

$$= \frac{1}{2}\left(1 + y \langle \boldsymbol{t}| \hat{U}_x^{\boldsymbol{p}_x\dagger} BS(\vec{\theta})^\dagger \mathcal{N} BS(\vec{\theta})\hat{U}_x^{\boldsymbol{p}_x} |\boldsymbol{t}\rangle\right). \tag{24}$$

Algorithm 2: Explicit method: prediction phase

Input: Training dataset $\mathcal{T}$, feature maps $\mathcal{U}_\phi$, optimisation routine.
Parameters: Input $|\boldsymbol{t}\rangle$, number of shots T, initial be parameters $\vec{\theta}_0$.
Set $\vec{\theta} = \vec{\theta}_0$.
while $J(\vec{\theta})$ *not converged* **do**
  for $l = 1$ *to* $|\mathcal{T}|$ **do**
    Set $c_{p_l} = 0$.
    Set $c_y = 0 \; \forall y \in C$.
    while $c_{p_l} < T$ **do**
      Run $BS(\vec{\theta})U_l^{\boldsymbol{p}_l}$ on $|\boldsymbol{t}\rangle$, obtaining adaptive measurement outcome $\boldsymbol{r}$ and outcome label $\boldsymbol{y}$.
      if $\boldsymbol{r} = \boldsymbol{p}_l$ **then**
        $c_{p_l} \to c_{p_l} + 1$.
        $c_y \to c_y + 1$.
      else
    end while
    Compute $\Pr(y|\boldsymbol{p}_l) = c_y/T$.
  end for
  Compute $J(\vec{\theta})$ and update $\vec{\theta}$.
end while
return Value $J(\vec{\theta}^*)$ and final $\vec{\theta}^*$.

In the quantum circuit model, it has been explicitly shown in [16] that the equivalent of Eq. (24) in the circuit picture is related to the decision function of an SVM. This proof relies on decomposing the variational quantum circuit in the Pauli basis. Moreover, in [37], the authors provide a quantum circuit that simulates Boson Sampling with arbitrarily distinguishable particles. Hence, the unitary $BS(\vec{\theta})$ can be decomposed in the Pauli basis (for qudits), hence providing the same relation for Eq. (24) to the decision function of an SVM.

Experimentally the probability, $\Pr(y|\boldsymbol{p}_x)$, can be obtained via the following approximation

$$\Pr(y|\boldsymbol{p}_x) \approx \frac{T_{y|x}}{T_x}, \tag{25}$$

where T_x is the number of times where the value $\boldsymbol{p}_x$ has been recorded and $T_{y|x}$ is the number of times where the value y has been recorded after that the value has been also recorded $\boldsymbol{p}_x$.

In order to train the Boson Sampler it is necessary to use a cost function $J(\vec{\theta})$ and an opti-

Algorithm 3: Explicit method: prediction phase

Input: Unlabeled data x, feature map $\mathcal{U}_{\phi(x)}$, optimal parameters $\vec{\theta}^*$.
Parameters: Input $|\boldsymbol{t}\rangle$, number of shots T_x.
Set $c_{p_x} = 0$.
Set $c_y = 0 \ \forall y \in C$.
while $c_{p_x} < T_x$ **do**
  Run $BS(\vec{\theta}^*)\mathcal{U}_{\phi(x)}$ on $|\boldsymbol{t}\rangle$, obtaining adaptive measurement outcome $\boldsymbol{r}$ and outcome label $\boldsymbol{y}$.
  if $\boldsymbol{r} = \boldsymbol{p}_x$ **then**
    $c_{p_x} \to c_{p_x} + 1$.
    $c_y \to c_y + 1$.
  else
  $\overline{}$
  end while
Compute $\Pr(y|\boldsymbol{p}_x) = c_y / T_x$.
return Return $\mathrm{argmax}_y\{\Pr(y|\boldsymbol{p}_x)\}$.

Algorithm 4: Implicit method: training phase

Input: Dataset $\mathcal{T}$, Feature maps $\mathcal{U}$, quadratic programming solver.
Parameters: Input $|\boldsymbol{t}\rangle$ and number of shots T.
for $l = 1$ *to* $|\mathcal{T}|$ **do**
  for $l' = 1$ *to* $|\mathcal{T}|$ **do**
    Run $U_l^{\boldsymbol{p}_l \otimes T}$ on input $|\boldsymbol{t}\rangle^{\otimes T}$.
    Run Algo. 1 with input $U_l^{\boldsymbol{p}_l}, U_{l'}^{\boldsymbol{p}_{l'}}$.
    Store output at kernel matrix entry $K_{l,l'} = \kappa(\vec{x}_l, \vec{x}_{l'})$.
  endfor
endfor
Solve the optimisation problem in Eq. (22) with $\mathcal{T}$ and K.
return Return $\{\alpha_l^*\}$ and b.

Algorithm 5: Implicit method: prediction phase

Input: Dataset $\mathcal{T}$, unlabeled data $\vec{x}$, optimal SVM parameters $\{\alpha_l^*\}$ and b
Parameters: Input $|\boldsymbol{t}\rangle$ and number of shots T.
Set the variable $f = b$.
for $l = 1$ *to* $|\mathcal{T}|$ **do**
  Run $U_x^{\boldsymbol{p}_x \otimes T}$ on input $|\boldsymbol{t}\rangle^{\otimes T}$.
  Run Algo. 1 with parameters $U_x^{\boldsymbol{p}_x \otimes T}|\boldsymbol{t}\rangle^{\otimes T}, U_l^{\boldsymbol{p}_l}$.
  $f \to f + \alpha_l y_l \kappa(\vec{x}_l, \vec{x})$.
endfor
return Return $\mathrm{sign}(f)$.

---

misation routine. For the optimisation routine, standard variational optimisation methods can be used such as hybrid variational quantum-classical or classical algorithms [12, 13, 26]. Different cost functions can be used such as the empirical risk function [16] or the square-loss function [17]. However, other type of cost function are not suitable for this kind of problems such as the ones usually used in gate synthesis or quantum state preparation [38, 39].

The algorithm for the training phase is described in Algorithm 2 and the prediction phase is described in Algorithm 3.

3.3.3   Implicit method: overlap estimation

The implicit method corresponds to the case where the kernel is computed by a quantum device, while the rest of the machine learning algorithm can be proceed on a classical machine. In adaptive linear optics, the kernel can be estimated using the Algorithm 1. It is possible to proceed to both the training and prediction phase using this hybrid quantum-classical method. The algorithm for the training phase is described in Algorithm 4 and the prediction phase is described in Algorithm 5.

4   Classical simulation of adaptive linear optics

The complexity of probability estimation and overlap estimation of quantum computations has been well studied in the circuit model [40, 41]. In this section, we challenge the previous quantum algorithms by giving classical algorithms for probability estimation and overlap estimation for adaptive linear optics over m modes. We identify various complexity regimes for different numbers of input photons $n \leq m$, different number of adaptive measurements $k \leq m$, and different number of photons $r \leq n$ detected during the adaptive measurements. To do so, we derive generic expressions for the output probabilities and for the overlap of output states of linear opti-

cal interferometers with adaptive measurements.

4.1 Classical probability estimation

Firstly, we obtain a classical algorithm for probability estimation of adaptive linear optics over m modes with n input photons and k adaptive measurements.

We first consider the case $k = 0$, i.e., Boson Sampling. The probability of the outcome $\boldsymbol{s} \in \Phi_{m,n}$ for the interferometer U given the input $\boldsymbol{t} = (\boldsymbol{1}^n, \boldsymbol{0}^{m-n}) \in \Phi_{m,n}$ is given by Eq. (12):

$$\Pr_{m,n}[\boldsymbol{s}] = \frac{1}{\boldsymbol{s}!} |\operatorname{Per}(U_{\boldsymbol{s},\boldsymbol{t}})|^2, \qquad (26)$$

where $U_{\boldsymbol{s},\boldsymbol{t}}$ is the $n \times n$ matrix obtained from U by repeating s_i times its ith row for $i \in \{1, \ldots, m\}$ and removing its jth column for $j = \{n+1, \ldots, m\}$. When $|\boldsymbol{s}| \neq n$ however, the probability is 0, since the input $\boldsymbol{t}$ has n photons and the linear interferometer does not change the total number of photons.

The permanent of a square matrix of size n can be computed exactly in time $O(n2^n)$, thanks to Ryser's formula [42]. However, polynomially precise estimates of the permanent of a square unitary matrix can be obtained in polynomial time in the size of the matrix using an algorithm due to Gurvits [43], later generalised to matrices with repeated lines or columns [44], so probability estimation can be done classically efficiently, which was already noted in [4].

We now turn to the case $k > 0$—which to our knowledge has not been treated elsewhere—using the notations of section 3.1. This case is a direct extension of the case $k = 0$. With Eq. (12), for $r \in \mathbb{N}$, $\boldsymbol{p} \in \Phi_{k,r}$ and $\boldsymbol{s} \in \Phi_{m-k,n-r}$, the probability of an total outcome $(\boldsymbol{p}, \boldsymbol{s}) \in \Phi_{m,n}$ (adaptive measurement and final outcome) is given by

$$\Pr_{m,n}^{\text{total}}[\boldsymbol{p}, \boldsymbol{s}] = \frac{1}{\boldsymbol{p}!\boldsymbol{s}!} \left|\operatorname{Per}\left(U_{(\boldsymbol{p},\boldsymbol{s}),\boldsymbol{t}}^{\boldsymbol{p}}\right)\right|^2. \qquad (27)$$

Let $r \in \{0, \ldots, n\}$ and let $\boldsymbol{s} \in \Phi_{m-k,n-r}$. Then, the probability of obtaining the final outcome $\boldsymbol{s}$ after the adaptive measurements reads

$$\begin{aligned}
\Pr_{m,n}^{\text{final}}[\boldsymbol{s}] &= \sum_{\boldsymbol{p} \in \Phi_{k,r}} \Pr_{m,n}^{\text{total}}[\boldsymbol{p}, \boldsymbol{s}] \\
&= \frac{1}{\boldsymbol{s}!} \sum_{\boldsymbol{p} \in \Phi_{k,r}} \frac{1}{\boldsymbol{p}!} \left|\operatorname{Per}\left(U_{(\boldsymbol{p},\boldsymbol{s}),\boldsymbol{t}}^{\boldsymbol{p}}\right)\right|^2.
\end{aligned} \qquad (28)$$

The sum is taken over the elements of $\Phi_{k,r}$, which has $\binom{k+r-1}{r}$ elements, where $r \leq n$ is the total

r \ k	$O(1)$	$O(\log m)$	$O(m)$
$O(1)$	🟩	🟩	🟩
$O(\log m)$	🟩	🟩	🟥
$O(m)$	🟩	🟥	🟥

Table 2: Simulability regimes for probability estimation as a function of the parameters r (the total number of photons detected during the adaptive measurements) and k (the number of single-mode adaptive measurements). In green is the parameter regime for which the classical probability estimation is efficient, i.e., takes polynomial time in m, while in red is the regime where it is no longer efficient.

number of photons detected during the adaptive measurements. Each permanent in the sum can be estimated additively with polynomial precision in time $O(\operatorname{poly} m)$, using the generalised algorithm from [44] (see Appendix B for details). Hence, probability estimation can be done classically efficiently whenever the sum has a polynomial number of terms. In particular, as long as both k and r are $O(\log m)$, the output probability can be estimated efficiently. The simulability regimes are summarised in Table 2.

The universal quantum computing regime corresponds to $n = O(m)$ and $k = O(m)$ [25]. The time complexity of the classical simulation is $O\left(\binom{k+r-1}{r} \operatorname{poly} m\right)$ and there is a possibility of subuniversal quantum advantage for probability estimation for $n = O(\log m)$ and $k = O(m)$, or $n = O(m)$ and $k = O(\log m)$. Moreover, the fraction $\frac{r}{n}$ of input photons detected during the adaptive measurements has to be sufficiently large to prevent efficient classical simulation.

4.2 Classical overlap estimation

In this section, we obtain a classical algorithm for overlap estimation of adaptive linear optics over m modes with n input photons and k adaptive measurements.

Once again, we start with $k = 0$. With Eq. (10), the output state of an m-mode inter-

ferometer U with input state $\boldsymbol{t} \in \Phi_{m,n}$ reads

$$|\phi\rangle = \sum_{\boldsymbol{s} \in \Phi_{m,n}} \langle \boldsymbol{s}|\hat{U}|\boldsymbol{t}\rangle\, |\boldsymbol{s}\rangle$$
$$= \sum_{\boldsymbol{s} \in \Phi_{m,n}} \frac{\mathrm{Per}\,(U_{\boldsymbol{s},\boldsymbol{t}})}{\sqrt{\boldsymbol{s}!\boldsymbol{t}!}}\, |\boldsymbol{s}\rangle, \qquad (29)$$

where $U_{\boldsymbol{s},\boldsymbol{t}}$ is the $n \times n$ matrix obtained from U by repeating s_i times its ith row for $i \in \{1,\dots,m\}$ and repeating t_j times its jth row for $j \in \{1,\dots,m\}$. The composition of two interferometers is another interferometer which unitary representation is the product of the unitary representations of the composed interferometers. Hence, the inner product of the output states $|\phi\rangle$ and $|\psi\rangle$ of two m-mode interferometers U and V with the same input state $\boldsymbol{t} \in \Phi_{m,n}$, is equal to the matrix element $\boldsymbol{t}, \boldsymbol{t}$ of $\hat{U}^\dagger \hat{V}$:

$$\langle \phi|\psi\rangle = \sum_{\boldsymbol{u},\boldsymbol{v} \in \Phi_{m,n}} \langle \boldsymbol{t}|\hat{U}^\dagger|\boldsymbol{u}\rangle \langle \boldsymbol{v}|\hat{V}|\boldsymbol{t}\rangle \langle \boldsymbol{u}|\boldsymbol{v}\rangle$$
$$= \sum_{\boldsymbol{s} \in \Phi_{m,n}} \langle \boldsymbol{t}|\hat{U}^\dagger|\boldsymbol{s}\rangle \langle \boldsymbol{s}|\hat{V}|\boldsymbol{t}\rangle$$
$$= \langle \boldsymbol{t}|\hat{U}^\dagger \hat{V}|\boldsymbol{t}\rangle \qquad (30)$$
$$= \frac{\mathrm{Per}\left[(U^\dagger V)_{\boldsymbol{t},\boldsymbol{t}}\right]}{\boldsymbol{t}!},$$

where we used in the third line $\boldsymbol{t} \in \Phi_{m,n}$ and the fact that $\hat{U}^\dagger \hat{V}$ conserves the space $\Phi_{m,n}$. With the input $\boldsymbol{t} = (\mathbf{1}^n, \mathbf{0}^{m-n})$ with n photons in m modes, this reduces to

$$\langle \phi|\psi\rangle = \mathrm{Per}\left[(U^\dagger V)_n\right], \qquad (31)$$

where $(U^\dagger V)_n$ is the $n \times n$ top left submatrix of $U^\dagger V$. Hence, the inner product and the overlap may be approximated to a polynomial precision efficiently, since this is the case for the permanent [43, 44].

We now consider the case $k > 0$. Let $r \in \mathbb{N}$ and let $\boldsymbol{p} \in \Phi_{k,r}$. With Eqs. (10) and (15), writing $\mathrm{Pr}_{m,n}^{\mathrm{adap}}[\boldsymbol{p}]$ the probability of the adaptive measurement outcome $\boldsymbol{p}$, the output state of the interferometer $U^{\boldsymbol{p}}$ with k adaptive measurements with input $\boldsymbol{t} = (\mathbf{1}^n, \mathbf{0}^{m-n})$ in Fig. 1, when the adaptive measurement outcome $\boldsymbol{p}$ is obtained, reads

$$\frac{1}{\sqrt{\mathrm{Pr}_{m,n}^{\mathrm{adap}}[\boldsymbol{p}]}}\, |\psi_{\boldsymbol{p}}\rangle, \qquad (32)$$

where

$$|\psi_{\boldsymbol{p}}\rangle := \sum_{\boldsymbol{s} \in \Phi_{m-k,n-r}} \frac{\mathrm{Per}\left(U_{(\boldsymbol{p},\boldsymbol{s}),\boldsymbol{t}}^{\boldsymbol{p}}\right)}{\sqrt{\boldsymbol{p}!\boldsymbol{s}!}}\, |\boldsymbol{s}\rangle \qquad (33)$$

and where $\mathrm{Pr}_{m,n}^{\mathrm{adap}}[\boldsymbol{p}] = \langle \psi_{\boldsymbol{p}}|\psi_{\boldsymbol{p}}\rangle$. More generally, the inner product of two (not normalised) output states $|\psi_{\boldsymbol{p}}\rangle$ and $|\psi_{\boldsymbol{q}}\rangle$ of m-mode interferometers $U^{\boldsymbol{p}}$ and $V^{\boldsymbol{q}}$ with k adaptive measurements thus is zero if $|\boldsymbol{p}| \neq |\boldsymbol{q}|$. When $|\boldsymbol{p}| = |\boldsymbol{q}| = r$, it is given by

$$\langle \psi_{\boldsymbol{p}}|\psi_{\boldsymbol{q}}\rangle = \frac{1}{\sqrt{\boldsymbol{p}!\boldsymbol{q}!}}$$
$$\times \sum_{\boldsymbol{s} \in \Phi_{m-k,n-r}} \frac{1}{\boldsymbol{s}!}\, \mathrm{Per}\left(U_{\boldsymbol{t},(\boldsymbol{p},\boldsymbol{s})}^{\boldsymbol{p}\dagger}\right) \mathrm{Per}\left(V_{(\boldsymbol{q},\boldsymbol{s}),\boldsymbol{t}}^{\boldsymbol{q}}\right). \qquad (34)$$

This expression is a sum of $|\Phi_{m-k,n-r}|$ terms, which is exponential in m whenever n is not constant. In what follows, we show that the expression in Eq. (34) may be rewritten as a sum over fewer terms using properties of the permanent [45], which constitutes our main technical result:

Lemma 1. *Let $r \in \mathbb{N}$. The inner product of two (not normalised) output states $|\psi_{\boldsymbol{p}}\rangle$ and $|\psi_{\boldsymbol{q}}\rangle$ of m-mode interferometers $U^{\boldsymbol{p}}$ and $V^{\boldsymbol{q}}$ with adaptive measurements outcome $\boldsymbol{p}, \boldsymbol{q} \in \Phi_{k,r}$ is given by*

$$\langle \psi_{\boldsymbol{p}}|\psi_{\boldsymbol{q}}\rangle = \frac{1}{\sqrt{\boldsymbol{p}!\boldsymbol{q}!}}$$
$$\times \sum_{\substack{\boldsymbol{i},\boldsymbol{j} \in \{0,1\}^n \\ |\boldsymbol{i}|=|\boldsymbol{j}|=r}} \mathrm{Per}\left(A^{\boldsymbol{i}}\right) \mathrm{Per}\left(B^{\boldsymbol{j}}\right) \mathrm{Per}\left(C^{\boldsymbol{i},\boldsymbol{j}}\right), \qquad (35)$$

where for all $\boldsymbol{i}, \boldsymbol{j} \in \{0,1\}^n$ such that $|\boldsymbol{i}| = |\boldsymbol{j}| = r$,

$$A^{\boldsymbol{i}} = U_{(\boldsymbol{i},\mathbf{0}^{m-n}),(\boldsymbol{p},\mathbf{0}^{m-k})}^{\boldsymbol{p}\dagger} \qquad (36)$$

is an $r \times r$ matrix which can be obtained efficiently from $U^{\boldsymbol{p}}$,

$$B^{\boldsymbol{j}} = V_{(\boldsymbol{q},\mathbf{0}^{m-k}),(\boldsymbol{j},\mathbf{0}^{m-n})}^{\boldsymbol{q}} \qquad (37)$$

is an $r \times r$ matrix which can be obtained efficiently from $V^{\boldsymbol{q}}$, and

$$C^{\boldsymbol{i},\boldsymbol{j}} = U_{(\mathbf{1}^n-\boldsymbol{i},\mathbf{0}^{m-n}),(\mathbf{0}^k,\mathbf{1}^{m-k})}^{\boldsymbol{p}\dagger} V_{(\mathbf{0}^k,\mathbf{1}^{m-k}),(\mathbf{1}^n-\boldsymbol{j},\mathbf{0}^{m-n})}^{\boldsymbol{q}} \qquad (38)$$

is an $(n-r) \times (n-r)$ matrix which can be obtained efficiently from $U^{\boldsymbol{p}}$ and $V^{\boldsymbol{q}}$.

We give a detailed proof in Appendix C. By Lemma 1, the inner product is expressed as the modulus squared of a sum over $\binom{n}{r}^2$ products of three permanents, of square matrices of sizes $|\boldsymbol{p}| = r$, $|\boldsymbol{q}| = r$ and $(n-r)$, respectively. In the

n \ k	$O(1)$	$O(\log m)$	$O(m)$
$O(1)$			
$O(\log m)$			$\cdots$
$O(m)$		$\cdots$	$\cdots$

Table 3: Simulability regimes for classical overlap estimation as a function of the parameters n and k. In green is the parameter regime for which the classical overlap estimation may be efficient (i.e., may take polynomial time in m) depending on the value of r, and in red is the regime where the classical algorithm is no longer efficient. The cells containing a symbol "$\cdots$" correspond to parameter regimes for which the quantum algorithm for estimating the overlap is no longer efficient.

n \ r	$O(1)$	$O(\log n)$	$O(n)$
$O(1)$			
$O(\log m)$			
$O(m)$			

Table 4: Simulability regimes for classical overlap estimation as a function of the parameters n (total number of input photons) and r (total number of photons detected in the adaptive measurements). The regimes are obtained from the running time of the classical algorithm using Stirling's equivalent $n! \sim \sqrt{2\pi n}(\frac{n}{e})^n$.

worst case, when $r = n/2$, the sum has at most $O(4^n)$ terms, up to a polynomial factor in n. In particular, when $n = O(\log m)$ or $r = O(1)$, the inner product reduces to a sum of a polynomial number of terms, which can all be computed in time $O(\text{poly } m)$ with Ryser's formula [42]. An interesting fact is that the cost of computing the inner product does not depend explicitely on the number $k > 0$ of adaptive measurements. However, it does depend explicitly on r the total number of photons detected during the adaptive measurements, and a larger number of adaptive measurements k favors the detection of a larger number of photons r.

The overlap of normalised ouput states is given by

$$\frac{|\langle\psi_{\boldsymbol{p}}|\psi_{\boldsymbol{q}}\rangle|^2}{\langle\psi_{\boldsymbol{p}}|\psi_{\boldsymbol{p}}\rangle \langle\psi_{\boldsymbol{q}}|\psi_{\boldsymbol{q}}\rangle}, \tag{39}$$

which may also be computed efficiently when $n = O(\log m)$. In this case, the classical algorithm for overlap estimation simply computes the above expression, using Lemma 1 for each of the inner products. The running time of this classical algorithm thus is $O(\binom{n}{r}^2 \text{poly } m)$ and its efficiency is summarised as a function of n and k in Table 3 and as a function of n and the number of photons r detected during the adaptive measurements in Table 4.

Since the quantum efficient regime corresponds to $\binom{k+n}{n} = O(\text{poly } m)$ there is a possibility of quantum advantage for overlap estimation when $k = O(1)$ and $n = O(m)$. However, like for probability estimation, the fraction $\frac{r}{n}$ of input photons

detected during the adaptive measurements has to be sufficiently large to prevent efficient classical simulation. In this case, when the number of adaptive measurements satisfies $k = O(1)$, the interferometers employed need to concentrate many photons during the adaptive measurements.

5 Conclusion

In this work, we have given a roadmap for performing quantum variational classification and quantum kernel estimation using adaptive linear optical interferometers. We have investigated the classical simulation transition between Boson Sampling [4] and the Knill–Laflamme–Milburn scheme for universal quantum computing [25], in terms of the number of adaptive measurements performed. In particukar, we have derived classical algorithms for simulating the quantum computational subroutines involved: output probability estimation and output state overlap estimation.

In the case of probability estimation, the possible regimes for quantum advantage are incompatible with near-term implementations: both the number of adaptive measurements k and the number of input photons n must be greater than $\log m$, where m is the number of modes. On the other hand, for overlap estimation, there is a possibility of near-term beyond-classical computing with adaptive linear optics using a single adaptive measurement, which requires the preparation of photon number states. Note that the interferometer should be concentrating many photons r at the stage of the adaptive measurements in

order to obtain possibly hard to estimate overlaps. Using more adaptive measurements does not increase the complexity (apart from polynomial factors in m), but may ease the detection of a larger number of photons during the adaptive measurements.

Our results suggest regimes where quantum advantage for machine learning with adaptive linear optics is possible, in the parameter regimes where our classical simulation algorithms fail to be efficient: it is an interesting open question whether better classical simulation algorithms for the computational subroutines involved can be found, or even classical algorithms solving directly the machine learning problems efficiently. In any case, our results restrict the parameter regimes for which such a quantum advantage may be possible: to hope for a quantum advantage, SVM with adaptive linear optics has to take place either in the bunching regime—concentrating many photons in the adaptive measurements—or using a number of adaptive measurements scaling with the size of the problem. In both cases, this imposes strong experimental requirements.

Still, our results have identified a sweet spot for quantum kernel estimation with adaptive linear optics using a single adaptive measurement, which would be interesting to demonstrate experimentally. In the longer term, variational classification with adaptive linear optics could also be interesting since it may enable quantum advantage in a regime where the quantum algorithm for overlap estimation is no longer efficient.

As previously mentioned, it is a pressing question whether more efficient classical algorithms may be derived. In practical settings, taking into account photon losses could help providing more efficient classical simulation algorithms. We leave these considerations for future work.

Acknowledgements

References

[1] R. P. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys* **21**, (1982).

[2] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*, pp. 124–134, Ieee. 1994.

[3] J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum* **2**, 79 (2018).

[4] S. Aaronson and A. Arkhipov, "The computational Complexity of Linear Optics," *Theory of Computing* **9**, 143 (2013).

[5] M. J. Bremner, R. Josza, and D. Shepherd, "Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy," *Proc. R. Soc. A* **459**, 459 (2010).

[6] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature* **574**, 505–510 (2019).

[7] M. Schuld, I. Sinayskiy, and F. Petruccione, "The quest for a Quantum Neural Network," *Quantum Information Processing* **13**, 2567–2586 (2014). https://doi.org/10.1007/s11128-014-0809-8.

[8] J. Romero, J. P. Olson, and A. Aspuru-Guzik, "Quantum autoencoders for efficient compression of quantum data," *Quantum Science and Technology* **2**, 045001 (2017). https://doi.org/10.1088%2F2058-9565%2Faa8072.

[9] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, "Quantum machine learning: a classical perspective," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **474**, 20170551 (2018). https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2017.0551.

[10] V. Dunjko and H. J. Briegel, "Machine learning & artificial intelligence in the quantum domain: a review of recent progress," *Reports on Progress in Physics* **81**, 074001 (2018). https://doi.org/10.1088%2F1361-6633%2Faab406.

[11] C. Zoufal, A. Lucchi, and S. Woerner, "Quantum Generative Adversarial Networks

for learning and loading random distributions," *npj Quantum Information* **5**, 103 (2019). https://doi.org/10.1038/s41534-019-0223-2.

[12] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, "Continuous-variable quantum neural networks," *Phys. Rev. Research* **1**, 033063 (2019). https://link.aps.org/doi/10.1103/PhysRevResearch.1.033063.

[13] E. Farhi and H. Neven, "Classification with Quantum Neural Networks on Near Term Processors," 2018.

[14] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann, and R. Wolf, "Training deep quantum neural networks," *Nature Communications* **11**, 808 (2020). https://doi.org/10.1038/s41467-020-14454-2.

[15] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, "The power of quantum neural networks," 2020.

[16] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," *Nature* **567**, 209 (2019).

[17] M. Schuld and N. Killoran, "Quantum machine learning in feature Hilbert spaces," *Physical review letters* **122**, 040504 (2019).

[18] C. Blank, D. K. Park, J.-K. K. Rhee, and F. Petruccione, "Quantum classifier with tailored quantum kernel," *npj Quantum Information* **6**, 41 (2020). https://doi.org/10.1038/s41534-020-0272-6.

[19] K. Bartkiewicz, C. Gneiting, A. Černoch, K. Jiráková, K. Lemr, and F. Nori, "Experimental kernel-based quantum machine learning in finite feature space," *Scientific Reports* **10**, 12356 (2020). https://doi.org/10.1038/s41598-020-68911-5.

[20] Y. Liu, S. Arunachalam, and K. Temme, "A rigorous and robust quantum speed-up in supervised machine learning," 2020.

[21] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, "Power of data in quantum machine learning," 2020.

[22] C. S. Hamilton, R. Kruse, L. Sansoni, S. Barkhofen, C. Silberhorn, and I. Jex, "Gaussian boson sampling," *Physical review letters* **119**, 170501 (2017).

[23] M. Schuld, K. Brádler, R. Israel, D. Su, and B. Gupt, "A quantum hardware-induced graph kernel based on Gaussian boson sampling," *arXiv preprint arXiv:1905.12646* (2019).

[24] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest, "Measurement-based quantum computation," *Nature Physics* **5**, 19–26 (2009).

[25] E. Knill, R. Laflamme, and G. J. Milburn, "A scheme for efficient quantum computation with linear optics," *Nature* **409**, 46–52 (2001).

[26] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," *Phys. Rev. A* **98**, 032309 (2018). https://link.aps.org/doi/10.1103/PhysRevA.98.032309.

[27] B. M. Terhal and D. P. DiVincenzo, "Classical simulation of noninteracting-fermion quantum circuits," *Physical Review A* **65**, 032325 (2002).

[28] H. Pashayan, S. D. Bartlett, and D. Gross, "From estimation of quantum probabilities to simulation of quantum circuits," *Quantum* **4**, 223 (2020).

[29] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American statistical association* **58**, 13–30 (1963).

[30] D. Dieks, "Overlap and distinguishability of quantum states," *Physics Letters A* **126**, 303–306 (1988).

[31] M. Fanizza, M. Rosati, M. Skotiniotis, J. Calsamiglia, and V. Giovannetti, "Beyond the swap test: optimal estimation of quantum state overlap," *Physical Review Letters* **124**, 060503 (2020).

[32] H. Buhrman, R. Cleve, J. Watrous, and R. De Wolf, "Quantum fingerprinting," *Physical Review Letters* **87**, 167902 (2001).

[33] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg, 1995.

[34] C. J. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery* **2**,

121–167 (1998). https://doi.org/10.1023/A:1009715923555.

[35] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[36] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, "Experimental realization of any discrete unitary operator," *Physical review letters* **73**, 58 (1994).

[37] A. E. Moylett and P. S. Turner, "Quantum simulation of partially distinguishable boson sampling," *Phys. Rev. A* **97**, 062329 (2018). https://link.aps.org/doi/10.1103/PhysRevA.97.062329.

[38] 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," 2018.

[39] K. Heya, Y. Suzuki, Y. Nakamura, and K. Fujii, "Variational Quantum Gate Optimization," 2018.

[40] H. Pashayan, J. J. Wallman, and S. D. Bartlett, "Estimating outcome probabilities of quantum circuits using quasiprobabilities," *Physical review letters* **115**, 070501 (2015).

[41] S. Bravyi, D. Gosset, and R. Movassagh, "Classical algorithms for quantum mean values," *arXiv preprint arXiv:1909.11485* (2019).

[42] N. Albert and S. Wilf Herbert, *Combinatorial algorithms: for computers and calculators*. Academic Press, 1978.

[43] L. Gurvits, "On the complexity of mixed discriminants and related problems," in *International Symposium on Mathematical Foundations of Computer Science*, pp. 447–458, Springer. 2005.

[44] S. Aaronson and T. Hance, "Generalizing and derandomizing Gurvits's approximation algorithm for the permanent," *arXiv preprint arXiv:1212.0025* (2012).

[45] J. K. Percus, *Combinatorial methods*, vol. 4. Springer Science & Business Media, 2012.

A Classical probability estimation for Shor's period-finding algorithm

In this section, we detail the classical probability estimation for Shor's period-finding algorithm [2].

If N is an n bits integer to factor, the period-finding subroutine measures the output state

$$\frac{1}{N} \sum_x \sum_y e^{\frac{2i\pi xy}{N}} |y\rangle |f(x)\rangle \tag{40}$$

in the computational basis, where f is a periodic function over $\{0, \dots, N-1\}$ which can be evaluated efficiently. The probability of obtaining an outcome $y_0, f(x_0)$ is given by

$$\Pr[y_0, f(x_0)] = \left| \frac{1}{N} \sum_{f(x)=f(x_0)} e^{\frac{2i\pi xy_0}{N}} \right|^2. \tag{41}$$

Now let

$$g_{x_0,y_0} : x \mapsto \begin{cases} e^{\frac{2i\pi xy_0}{N}} & \text{if } f(x) = f(x_0), \\ 0 & \text{otherwise.} \end{cases} \tag{42}$$

The function g_{x_0,y_0} can be evaluated efficiently and we have

$$\Pr[y_0, f(x_0)] = \left| \mathop{\mathbb{E}}_{x \leftarrow N} [g_{x_0,y_0}(x)] \right|^2, \tag{43}$$

where $\mathop{\mathbb{E}}_{x \leftarrow N}$ denotes the expected value for x drawn uniformly randomly from $\{0, \dots, N-1\}$. By virtue of Hoeffding inequality, this quantity may be estimated efficiently (in n, the number of bits of N) classically by sampling uniformly a polynomial number of values in $\{0, \dots, N-1\}$ and computing the modulus squared of the mean of g_{x_0,y_0} for these values.

However, note that probability estimation of quantum circuits is a BQP-complete computational task almost by definition, since given a polynomially precise estimate of the probability of acceptance of an input x to a quantum circuit, one may determine whether it is accepted or rejected by the circuit. In particular, unless factoring is in P, probability estimation for the quantum circuit corresponding to Shor's algorithm *as a whole* is hard and weak simulation of the period-finding subroutine is also hard, since in Shor's algorithm the output samples from the period-finding subroutine are used for a different classical computation than probability estimation, namely obtaining promising candidates for the period.

B Efficiency of classical output probability estimation

For an interferometer U over m modes with n input photons and k adaptive measurements with outcomes $\boldsymbol{p} \in \Phi_{k,r}$, the expression of the probability of an outcome $\boldsymbol{s}$ obtained in the main text reads:

$$\Pr_{m,n}^{\text{final}}[\boldsymbol{s}] = \frac{1}{\boldsymbol{s}!} \sum_{\boldsymbol{p} \in \Phi_{k,r}} \frac{1}{\boldsymbol{p}!} \left| \text{Per}\left(U_{(\boldsymbol{p},\boldsymbol{s}),\boldsymbol{t}}^{\boldsymbol{p}} \right) \right|^2, \tag{44}$$

where $\boldsymbol{t} = (\mathbf{1}^n, \mathbf{0}^{m-n})$. This is a sum over $|\Phi_{k,r}| = \binom{k+r-1}{r}$ moduli squared of permanents of square matrices of size $n \leq m$. Permanents can be approximated efficiently using a simple randomized algorithm due to Gurvits:

Lemma 2 ([43]). *Let A be an $m \times m$ matrix. Then, $\text{Per}\, A$ may be estimated classically with additive precision $\pm\epsilon\|A\|^m$ with high probability, in time $O(\frac{m^2}{\epsilon^2})$, where $\|A\|$ is the largest singular value of A.*

This algorithm has been refined in the case of matrices with repeated lines or repeated columns:

Lemma 3 ([44]). *Let B be an $m \times n$ matrix. Let $\boldsymbol{q} = (q_1, \ldots, q_m) \in \Phi_{m,n}$ and let $A = B_{\boldsymbol{q},\mathbf{1}^n}$ be the $n \times n$ matrix obtained from B by repeating q_i times its ith line. Then, $\operatorname{Per} A$ may be estimated classically with additive precision $\pm\epsilon \cdot \frac{q_1! \ldots q_m!}{\sqrt{q_1^{q_1} \cdots q_m^{q_m}}} \|B\|^m$ with high probability, in time $O(\frac{mn}{\epsilon^2})$, where $\|B\|$ is the largest singular value of B. Moreover,*

$$|\operatorname{Per} A| \leq \frac{q_1! \ldots q_m!}{\sqrt{q_1^{q_1} \cdots q_m^{q_m}}} \|B\|^m. \tag{45}$$

This lemma also allows to approximate efficiently $|\operatorname{Per} A|^2$. Indeed, let $0 < \epsilon < 1$ and let z be an estimate of $\operatorname{Per} A$ with additive error $\pm\epsilon \cdot \frac{q_1! \ldots q_m!}{\sqrt{q_1^{q_1} \cdots q_m^{q_m}}} \|B\|^m$, then $|z|^2$ is a good estimate of $|\operatorname{Per} A|^2$:

$$
\begin{aligned}
\left| |z|^2 - |\operatorname{Per} A|^2 \right| &= \left| |z| - |\operatorname{Per} A| \right| \cdot \left(|z| + |\operatorname{Per} A| \right) \\
&\leq \epsilon \cdot \frac{q_1! \ldots q_m!}{\sqrt{q_1^{q_1} \cdots q_m^{q_m}}} \|B\|^m \cdot \left(|\operatorname{Per} A| + \epsilon \cdot \frac{q_1! \ldots q_m!}{\sqrt{q_1^{q_1} \cdots q_m^{q_m}}} + |\operatorname{Per} A| \right) \\
&\leq \epsilon \cdot \frac{q_1! \ldots q_m!}{\sqrt{q_1^{q_1} \cdots q_m^{q_m}}} \|B\|^m \cdot \left(\epsilon \cdot \frac{q_1! \ldots q_m!}{\sqrt{q_1^{q_1} \cdots q_m^{q_m}}} + 2 \frac{q_1! \ldots q_m!}{\sqrt{q_1^{q_1} \cdots q_m^{q_m}}} \|B\|^m \right) \\
&\leq 3\epsilon \cdot \frac{q_1!^2 \ldots q_m!^2}{q_1^{q_1} \cdots q_m^{q_m}} \|B\|^{2m},
\end{aligned}
\tag{46}
$$

where we used Eq. (45) in the third line.

In particular, when B is an $m \times n$ submatrix of a unitary matrix (implying $\|B\| \leq 1$), we may compute an estimate of $|\operatorname{Per} A|^2$, where $A = B_{\boldsymbol{q},\mathbf{1}^n}$ is the $n \times n$ matrix obtained from B by repeating q_i times its ith line, with additive precision $\pm\epsilon \cdot \frac{q_1!^2 \ldots q_m!^2}{q_1^{q_1} \cdots q_m^{q_m}}$ with high probability, in time $O(\frac{mn}{\epsilon^2})$. The matrices in Eq. (44) are submatrices of unitary matrices, with repeated lines. Hence, estimating independently all the terms in the sum in Eq. (44), we obtain, in time $O(\frac{mn}{\epsilon^2} \cdot |\Phi_{k,r}|)$ and with high probability, an estimate $\tilde{P}$ of the probability $\operatorname{Pr}_{m,n}^{\text{final}}[\boldsymbol{s}]$ such that

$$
\begin{aligned}
\left| \tilde{P} - \operatorname{Pr}_{m,n}^{\text{final}}[\boldsymbol{s}] \right| &\leq \epsilon \cdot \frac{1}{s_1! \ldots s_{m-k}!} \sum_{\boldsymbol{p} \in \Phi_{k,r}} \frac{1}{p_1! \ldots p_k!} \cdot \frac{p_1!^2 \ldots p_k!^2 s_1!^2 \ldots s_{m-k}!^2}{p_1^{p_1} \cdots p_k^{p_k} s_1^{s_1} \cdots s_{m-k}^{s_{m-k}}} \\
&\leq \epsilon \cdot \frac{s_1! \ldots s_{m-k}!}{s_1^{s_1} \cdots s_{m-k}^{s_{m-k}}} \sum_{\boldsymbol{p} \in \Phi_{k,r}} \frac{p_1! \ldots p_k!}{p_1^{p_1} \cdots p_k^{p_k}} \\
&\leq \epsilon \cdot |\Phi_{k,r}|,
\end{aligned}
\tag{47}
$$

where we used that for all $n \in \mathbb{N}$, $n! \leq n^n$. Note that a tighter bound may be obtained, but the above one is sufficient for our needs. In particular, when $|\Phi_{k,r}| = O(\text{poly } m)$, the above procedure provides a polynomially precise additive estimate of the probability $\operatorname{Pr}_{m,n}^{\text{final}}[\boldsymbol{s}]$ in time $O(\text{poly } m)$, with high probability.

We have

$$
\begin{aligned}
|\Phi_{k,r}| &= \binom{k+r-1}{r} \\
&= \frac{k}{k+r} \cdot \frac{(k+r)!}{k! r!}.
\end{aligned}
\tag{48}
$$

This quantity is polynomial in k (resp. in r) when $r = O(1)$ (resp. $k = O(1)$). Moreover,

$$
\begin{aligned}
\binom{k+r-1}{r} &\leq \sum_{j=0}^{k+r-1} \binom{k+r-1}{j} \\
&= 2^{k+r-1},
\end{aligned}
\tag{49}
$$

so for $k = O(\log m)$ and $r = O(\log m)$, we have $|\Phi_{k,r}| = O(\text{poly } m)$. For all other cases, that is $k = O(\log m)$ and $r = O(m)$, $k = O(m)$ and $r = O(\log m)$, or $k = O(m)$ and $r = O(m)$, $|\Phi_{k,r}|$ is superpolynomial in m, using Stirling's equivalent $n! \sim \sqrt{2\pi n}(\frac{n}{e})^n$.

C   Proof of Lemma 1

We recall Lemma 1 from the main text:

Lemma 1. *Let* $r \in \mathbb{N}$. *The inner product of two (not normalised) output states* $|\psi_{\boldsymbol{p}}\rangle$ *and* $|\psi_{\boldsymbol{q}}\rangle$ *of* m*-mode interferometers* $U^{\boldsymbol{p}}$ *and* $V^{\boldsymbol{q}}$ *with adaptive measurements outcome* $\boldsymbol{p}, \boldsymbol{q} \in \Phi_{k,r}$ *is given by*

$$\langle \psi_{\boldsymbol{p}} | \psi_{\boldsymbol{q}} \rangle = \frac{1}{\sqrt{\boldsymbol{p}!\boldsymbol{q}!}} \sum_{\substack{\boldsymbol{i},\boldsymbol{j} \in \{0,1\}^n \\ |\boldsymbol{i}| = |\boldsymbol{j}| = r}} \mathrm{Per}\left(A^{\boldsymbol{i}}\right) \mathrm{Per}\left(B^{\boldsymbol{j}}\right) \mathrm{Per}\left(C^{\boldsymbol{i},\boldsymbol{j}}\right), \tag{50}$$

where for all $\boldsymbol{i}, \boldsymbol{j} \in \{0,1\}^n$ *such that* $|\boldsymbol{i}| = |\boldsymbol{j}| = r$,

$$A^{\boldsymbol{i}} = U^{\boldsymbol{p}\dagger}_{(\boldsymbol{i},\boldsymbol{0}^{m-n}),(\boldsymbol{p},\boldsymbol{0}^{m-k})} \tag{51}$$

is an $r \times r$ *matrix which can be obtained efficiently from* $U^{\boldsymbol{p}}$,

$$B^{\boldsymbol{j}} = V^{\boldsymbol{q}}_{(\boldsymbol{q},\boldsymbol{0}^{m-k}),(\boldsymbol{j},\boldsymbol{0}^{m-n})} \tag{52}$$

is an $r \times r$ *matrix which can be obtained efficiently from* $V^{\boldsymbol{q}}$, *and*

$$C^{\boldsymbol{i},\boldsymbol{j}} = U^{\boldsymbol{p}\dagger}_{(\boldsymbol{1}^n-\boldsymbol{i},\boldsymbol{0}^{m-n}),(\boldsymbol{0}^k,\boldsymbol{1}^{m-k})} V^{\boldsymbol{q}}_{(\boldsymbol{0}^k,\boldsymbol{1}^{m-k}),(\boldsymbol{1}^n-\boldsymbol{j},\boldsymbol{0}^{m-n})} \tag{53}$$

is an $(n-r) \times (n-r)$ *matrix which can be obtained efficiently from* $U^{\boldsymbol{p}}$ *and* $V^{\boldsymbol{q}}$.

Proof. We consider the expression for the inner product obtained in Eq. (34):

$$\langle \psi_{\boldsymbol{p}} | \psi_{\boldsymbol{q}} \rangle = \frac{1}{\sqrt{\boldsymbol{p}!\boldsymbol{q}!}} \sum_{\boldsymbol{s} \in \Phi_{m-k,n-r}} \frac{1}{\boldsymbol{s}!} \mathrm{Per}\left(U^{\boldsymbol{p}\dagger}_{\boldsymbol{t},(\boldsymbol{p},\boldsymbol{s})}\right) \mathrm{Per}\left(V^{\boldsymbol{q}}_{(\boldsymbol{q},\boldsymbol{s}),\boldsymbol{t}}\right). \tag{54}$$

It is reminiscent of the permanent composition formula [45]: for all $m, n, c \in \mathbb{N}^*$, all $s \in \mathbb{N}$, all $\boldsymbol{u} \in \Phi_{m,s}$ and all $\boldsymbol{v} \in \Phi_{n,s}$,

$$\mathrm{Per}\left[(MN)_{\boldsymbol{u},\boldsymbol{v}}\right] = \sum_{\boldsymbol{s} \in \Phi_{c,s}} \frac{1}{\boldsymbol{s}!} \mathrm{Per}\left(M_{\boldsymbol{u},\boldsymbol{s}}\right) \mathrm{Per}\left(N_{\boldsymbol{s},\boldsymbol{v}}\right) \tag{55}$$

where M is a $m \times c$ matrix and N is a $n \times c$ matrix. However, this formula is not directly applicable to the expression in Eq. (54). In order to obtain a suitable expression, we first make use of the Laplace formula for the permanent: we expand the permanent of $U^{\boldsymbol{p}\dagger}_{\boldsymbol{t},(\boldsymbol{p},\boldsymbol{s})}$ along the columns that are repeated according to $\boldsymbol{p}$ and we expand the permanent of $V^{\boldsymbol{q}}_{(\boldsymbol{q},\boldsymbol{s}),\boldsymbol{t}}$ along the rows that are repeated according to $\boldsymbol{q}$. The generalised Laplace column expansion formula for the permanent reads: let $n \in \mathbb{N}^*$, let W be an $n \times n$ matrix, and let $\boldsymbol{j} \in \{0,1\}^n$. Then,

$$\mathrm{Per}\left(W\right) = \sum_{\substack{\boldsymbol{i} \in \{0,1\}^n \\ |\boldsymbol{i}| = |\boldsymbol{j}|}} \mathrm{Per}\left(W_{\boldsymbol{i},\boldsymbol{j}}\right) \mathrm{Per}\left(W_{\boldsymbol{1}^n-\boldsymbol{i},\boldsymbol{1}^n-\boldsymbol{j}}\right), \tag{56}$$

where $W_{\boldsymbol{i},\boldsymbol{j}}$ is the matrix obtained from W by keeping only the kth rows and l^{th} columns such that $i_k = 1$ and $j_l = 1$, respectively, and $W_{\boldsymbol{1}^n-\boldsymbol{i},\boldsymbol{1}^n-\boldsymbol{j}}$ is the matrix obtained from W by keeping only the kth rows and l^{th} columns such that $i_k = 0$ and $j_l = 0$, respectively. This formula is obtained by applying the Laplace expansion formula for one column various times, for each column with index l such that $j_l = 1$, and the same formula holds for rows.

We first apply the general column expansion formula in Eq. (56) to the matrix $U^{\boldsymbol{p}\dagger}_{\boldsymbol{t},(\boldsymbol{p},\boldsymbol{s})}$ with $\boldsymbol{j} = (\mathbf{1}^r, \mathbf{0}^{n-r}) \in \{0,1\}^n$, obtaining

$$\mathrm{Per}\left(U^{\boldsymbol{p}\dagger}_{\boldsymbol{t},(\boldsymbol{p},\boldsymbol{s})}\right) = \sum_{\substack{\boldsymbol{i}\in\{0,1\}^n \\ |\boldsymbol{i}|=r}} \mathrm{Per}\left[\left(U^{\boldsymbol{p}\dagger}_{\boldsymbol{t},(\boldsymbol{p},\boldsymbol{s})}\right)_{\boldsymbol{i},\boldsymbol{j}}\right] \mathrm{Per}\left[\left(U^{\boldsymbol{p}\dagger}_{\boldsymbol{t},(\boldsymbol{p},\boldsymbol{s})}\right)_{\mathbf{1}^n-\boldsymbol{i},\mathbf{1}^n-\boldsymbol{j}}\right]. \tag{57}$$

Let us consider the matrix $\left(U^{\boldsymbol{p}\dagger}_{\boldsymbol{t},(\boldsymbol{p},\boldsymbol{s})}\right)_{\boldsymbol{i},\boldsymbol{j}}$ appearing in this last expression, for $\boldsymbol{i} \in \{0,1\}^n$. Its rows are obtained by keeping the first n lines of $U^{\boldsymbol{p}\dagger}$ since $\boldsymbol{t} = (\mathbf{1}^n, \mathbf{0}^{m-n})$, then by keeping only the lth rows such that $i_l = 1$. Its columns are obtained by repeating p_l times the lth column for $l \in \{1,\dots,k\}$ and s_l times for $l \in \{k+1,\dots,m\}$, then by only keeping the first r columns since $\boldsymbol{j} = (\mathbf{1}^r, \mathbf{0}^{n-r})$. However, since $|\boldsymbol{p}| = |\boldsymbol{j}| = r$, these are the columes repeated according to $\boldsymbol{p}$. Hence,

$$\left(U^{\boldsymbol{p}\dagger}_{\boldsymbol{t},(\boldsymbol{p},\boldsymbol{s})}\right)_{\boldsymbol{i},\boldsymbol{j}} = U^{\boldsymbol{p}\dagger}_{(\boldsymbol{i},\mathbf{0}^{m-n}),(\boldsymbol{p},\mathbf{0}^{m-k})}, \tag{58}$$

where $U^{\boldsymbol{p}\dagger}_{(\boldsymbol{i},\mathbf{0}^{m-n}),(\boldsymbol{p},\mathbf{0}^{m-k})}$ is the matrix obtained from $U^{\boldsymbol{p}\dagger}$ by keeping only the lth rows such that $i_l = 1$ and removing the others, and by repeating p_l times the lth column for $l \in \{1,\dots,k\}$ and removing the others. Similarly, with $|\boldsymbol{s}| = |\mathbf{1}^n - \boldsymbol{j}| = n - r$,

$$\left(U^{\boldsymbol{p}\dagger}_{\boldsymbol{t},(\boldsymbol{p},\boldsymbol{s})}\right)_{\mathbf{1}^n-\boldsymbol{i},\mathbf{1}^n-\boldsymbol{j}} = U^{\boldsymbol{p}\dagger}_{(\mathbf{1}^n-\boldsymbol{i},\mathbf{0}^{m-n}),(\mathbf{0}^k,\boldsymbol{s})}, \tag{59}$$

where $U^{\boldsymbol{p}\dagger}_{(\mathbf{1}^n-\boldsymbol{i},\mathbf{0}^{m-n}),(\mathbf{0}^k,\boldsymbol{s})}$ is the matrix obtained from $U^{\boldsymbol{p}\dagger}$ by keeping only the lth rows such that $i_l = 0$ and removing the others, and by repeating s_l times the lth column for $l \in \{k+1,\dots,m\}$ and removing the others. With Eqs. (57), (58) and (59) we obtain

$$\begin{aligned} \mathrm{Per}\left(U^{\boldsymbol{p}\dagger}_{\boldsymbol{t},(\boldsymbol{p},\boldsymbol{s})}\right) &= \sum_{\substack{\boldsymbol{i}\in\{0,1\}^n \\ |\boldsymbol{i}|=r}} \mathrm{Per}\left(U^{\boldsymbol{p}\dagger}_{(\boldsymbol{i},\mathbf{0}^{m-n}),(\boldsymbol{p},\mathbf{0}^{m-k})}\right)\mathrm{Per}\left(U^{\boldsymbol{p}\dagger}_{(\mathbf{1}^n-\boldsymbol{i},\mathbf{0}^{m-n}),(\mathbf{0}^k,\boldsymbol{s})}\right) \\ &= \sum_{\substack{\boldsymbol{i}\in\{0,1\}^n \\ |\boldsymbol{i}|=r}} \mathrm{Per}\left(A^{\boldsymbol{i}}\right)\mathrm{Per}\left(U^{\boldsymbol{p}\dagger}_{(\mathbf{1}^n-\boldsymbol{i},\mathbf{0}^{m-n}),(\mathbf{0}^k,\boldsymbol{s})}\right), \end{aligned} \tag{60}$$

where we have defined, for all $\boldsymbol{i} \in \{0,1\}^n$ such that $|\boldsymbol{i}| = r$,

$$A^{\boldsymbol{i}} := U^{\boldsymbol{p}\dagger}_{(\boldsymbol{i},\mathbf{0}^{m-n}),(\boldsymbol{p},\mathbf{0}^{m-k})}, \tag{61}$$

which is an $r \times r$ matrix independent of $\boldsymbol{s}$ that can be obtained efficiently from $U^{\boldsymbol{p}}$.

The same reasoning with the general row expansion formula for the matrix $V^{\boldsymbol{q}}_{(\boldsymbol{q},\boldsymbol{s}),\boldsymbol{t}}$ and the rows $\boldsymbol{i} = (\mathbf{1}^r, \mathbf{0}^{n-r})$ gives

$$\begin{aligned} \mathrm{Per}\left(V^{\boldsymbol{q}}_{(\boldsymbol{q},\boldsymbol{s}),\boldsymbol{t}}\right) &= \sum_{\substack{\boldsymbol{j}\in\{0,1\}^n \\ |\boldsymbol{j}|=r}} \mathrm{Per}\left[\left(V^{\boldsymbol{q}}_{(\boldsymbol{q},\boldsymbol{s}),\boldsymbol{t}}\right)_{\boldsymbol{i},\boldsymbol{j}}\right]\mathrm{Per}\left[\left(V^{\boldsymbol{q}}_{(\boldsymbol{q},\boldsymbol{s}),\boldsymbol{t}}\right)_{\mathbf{1}^n-\boldsymbol{i},\mathbf{1}^n-\boldsymbol{j}}\right] \\ &= \sum_{\substack{\boldsymbol{j}\in\{0,1\}^n \\ |\boldsymbol{j}|=r}} \mathrm{Per}\left(V^{\boldsymbol{q}}_{(\boldsymbol{q},\mathbf{0}^{m-k}),(\boldsymbol{j},\mathbf{0}^{m-n})}\right)\mathrm{Per}\left(V^{\boldsymbol{q}}_{(\mathbf{0}^k,\boldsymbol{s}),(\mathbf{1}^n-\boldsymbol{j},\mathbf{0}^{m-n})}\right), \end{aligned} \tag{62}$$

where $V^{\boldsymbol{q}}_{(\boldsymbol{q},\mathbf{0}^{m-k}),(\boldsymbol{j},\mathbf{0}^{m-n})}$ is the matrix obtained from $V^{\boldsymbol{q}}$ by repeating q_l times the lth row for $l \in \{1,\dots,k\}$ and removing the others and by keeping only the lth columns such that $j_l = 1$, and where $V^{\boldsymbol{q}}_{(\mathbf{0}^k,\boldsymbol{s}),(\mathbf{1}^n-\boldsymbol{j},\mathbf{0}^{m-n})}$ is the matrix obtained from $V^{\boldsymbol{q}}$ by repeating s_l times the lth row for $l \in \{k+1,\dots,m\}$ and removing the others and by keeping only the lth columns such that $j_l = 0$. Defining, for all $\boldsymbol{j} \in \{0,1\}^n$ such that $|\boldsymbol{j}| = r$,

$$B^{\boldsymbol{j}} := V^{\boldsymbol{q}}_{(\boldsymbol{q},\mathbf{0}^{m-k}),(\boldsymbol{j},\mathbf{0}^{m-n})}, \tag{63}$$

the expression in Eq. (62) rewrites

$$\text{Per}\left(V^{\boldsymbol{q}}_{(\boldsymbol{q},\boldsymbol{s}),\boldsymbol{t}}\right) = \sum_{\substack{\boldsymbol{j}\in\{0,1\}^n \\ |\boldsymbol{j}|=r}} \text{Per}\left(B^{\boldsymbol{j}}\right)\text{Per}\left(V^{\boldsymbol{q}}_{(\boldsymbol{0}^k,\boldsymbol{s}),(\boldsymbol{1}^n-\boldsymbol{j},\boldsymbol{0}^{m-n})}\right),\tag{64}$$

where $B^{\boldsymbol{j}}$ are $r\times r$ matrices independent of $\boldsymbol{s}$ and can be obtained efficiently from $V^{\boldsymbol{q}}$.

Plugging Eqs. (60) and (64) in Eq. (54) we obtain

$$\begin{aligned}\langle\psi_{\boldsymbol{p}}|\psi_{\boldsymbol{q}}\rangle = \frac{1}{\sqrt{\boldsymbol{p}!\boldsymbol{q}!}}\sum_{\substack{\boldsymbol{i},\boldsymbol{j}\in\{0,1\}^n \\ |\boldsymbol{i}|=|\boldsymbol{j}|=r}}&\left[\text{Per}\left(A^{\boldsymbol{i}}\right)\text{Per}\left(B^{\boldsymbol{j}}\right)\right.\\ &\left.\times\sum_{\boldsymbol{s}\in\Phi_{m-k,n-r}}\frac{1}{\boldsymbol{s}!}\text{Per}\left(U^{\boldsymbol{p}\dagger}_{(\boldsymbol{1}^n-\boldsymbol{i},\boldsymbol{0}^{m-n}),(\boldsymbol{0}^k,\boldsymbol{s})}\right)\text{Per}\left(V^{\boldsymbol{q}}_{(\boldsymbol{0}^k,\boldsymbol{s}),(\boldsymbol{1}^n-\boldsymbol{j},\boldsymbol{0}^{m-n})}\right)\right].\end{aligned}\tag{65}$$

The sum appearing in the second line may now be expressed as a single permanent using the permanent composition formula: for all $\boldsymbol{i},\boldsymbol{j}\in\{0,1\}^n$ such that $|\boldsymbol{i}|=|\boldsymbol{j}|=r$, let us define the $(n-r)\times(m-k)$ matrix

$$\tilde{U}^{\boldsymbol{p},\boldsymbol{i}} := U^{\boldsymbol{p}\dagger}_{(\boldsymbol{1}^n-\boldsymbol{i},\boldsymbol{0}^{m-n}),(\boldsymbol{0}^k,\boldsymbol{1}^{m-k})},\tag{66}$$

and the $(m-k)\times(n-r)$ matrix

$$\tilde{V}^{\boldsymbol{q},\boldsymbol{j}} := V^{\boldsymbol{q}}_{(\boldsymbol{0}^k,\boldsymbol{1}^{m-k}),(\boldsymbol{1}^n-\boldsymbol{j},\boldsymbol{0}^{m-n})},\tag{67}$$

so that

$$U^{\boldsymbol{p}\dagger}_{(\boldsymbol{1}^n-\boldsymbol{i},\boldsymbol{0}^{m-n}),(\boldsymbol{0}^k,\boldsymbol{s})} = \tilde{U}^{\boldsymbol{p},\boldsymbol{i}}_{\boldsymbol{1}^{n-r},\boldsymbol{s}} \quad\text{and}\quad V^{\boldsymbol{q}}_{(\boldsymbol{0}^k,\boldsymbol{s}),(\boldsymbol{1}^n-\boldsymbol{j},\boldsymbol{0}^{m-n})} = \tilde{V}^{\boldsymbol{q},\boldsymbol{j}}_{\boldsymbol{s},\boldsymbol{1}^{n-r}}.\tag{68}$$

With the permanent composition formula in Eq. (55) we obtain

$$\sum_{\boldsymbol{s}\in\Phi_{m-k,n-r}}\frac{1}{\boldsymbol{s}!}\text{Per}\left(U^{\boldsymbol{p}\dagger}_{(\boldsymbol{1}^n-\boldsymbol{i},\boldsymbol{0}^{m-n}),(\boldsymbol{0}^k,\boldsymbol{s})}\right)\text{Per}\left(V^{\boldsymbol{q}}_{(\boldsymbol{0}^k,\boldsymbol{s}),(\boldsymbol{1}^n-\boldsymbol{j},\boldsymbol{0}^{m-n})}\right) = \text{Per}\left[\left(\tilde{U}^{\boldsymbol{p},\boldsymbol{i}}\tilde{V}^{\boldsymbol{q},\boldsymbol{j}}\right)_{\boldsymbol{1}^{n-r},\boldsymbol{1}^{n-r}}\right].\tag{69}$$

Since $\tilde{U}^{\boldsymbol{p},\boldsymbol{i}}\tilde{V}^{\boldsymbol{q},\boldsymbol{j}}$ is an $(n-r)\times(n-r)$ matrix we thus have

$$\sum_{\boldsymbol{s}\in\Phi_{m-k,n-r}}\frac{1}{\boldsymbol{s}!}\text{Per}\left(U^{\boldsymbol{p}\dagger}_{(\boldsymbol{1}^n-\boldsymbol{i},\boldsymbol{0}^{m-n}),(\boldsymbol{0}^k,\boldsymbol{s})}\right)\text{Per}\left(V^{\boldsymbol{q}}_{(\boldsymbol{0}^k,\boldsymbol{s}),(\boldsymbol{1}^n-\boldsymbol{j},\boldsymbol{0}^{m-n})}\right) = \text{Per}\left(\tilde{U}^{\boldsymbol{p},\boldsymbol{i}}\tilde{V}^{\boldsymbol{q},\boldsymbol{j}}\right).\tag{70}$$

Then, Eq. (65) rewrites

$$\langle\psi_{\boldsymbol{p}}|\psi_{\boldsymbol{q}}\rangle = \frac{1}{\sqrt{\boldsymbol{p}!\boldsymbol{q}!}}\sum_{\substack{\boldsymbol{i},\boldsymbol{j}\in\{0,1\}^n \\ |\boldsymbol{i}|=|\boldsymbol{j}|=r}}\text{Per}\left(A^{\boldsymbol{i}}\right)\text{Per}\left(B^{\boldsymbol{j}}\right)\text{Per}\left(C^{\boldsymbol{i},\boldsymbol{j}}\right),\tag{71}$$

where we have defined

$$\begin{aligned}C^{\boldsymbol{i},\boldsymbol{j}} &:= \tilde{U}^{\boldsymbol{p},\boldsymbol{i}}\tilde{V}^{\boldsymbol{q},\boldsymbol{j}}\\ &= U^{\boldsymbol{p}\dagger}_{(\boldsymbol{1}^n-\boldsymbol{i},\boldsymbol{0}^{m-n}),(\boldsymbol{0}^k,\boldsymbol{1}^{m-k})}V^{\boldsymbol{q}}_{(\boldsymbol{0}^k,\boldsymbol{1}^{m-k}),(\boldsymbol{1}^n-\boldsymbol{j},\boldsymbol{0}^{m-n})},\end{aligned}\tag{72}$$

is an $(n-r)\times(n-r)$ matrix which can be obtained efficiently from $U^{\boldsymbol{p}}$ and $V^{\boldsymbol{q}}$.

$\square$