

Classical simulation of non-Gaussian fermionic circuits

Beatriz Dias^{1,2} and Robert König^{1,2}

¹Department of Mathematics, School of Computation, Information and Technology, Technical University of Munich, Garching, Germany

²Munich Center for Quantum Science and Technology, Munich, Germany

We propose efficient algorithms for classically simulating fermionic linear optics operations applied to non-Gaussian initial states. By gadget constructions, this provides algorithms for fermionic linear optics with non-Gaussian operations. We argue that this problem is analogous to that of simulating Clifford circuits with non-stabilizer initial states: Algorithms for the latter problem immediately translate to the fermionic setting. Our construction is based on an extension of the covariance matrix formalism which permits to efficiently track relative phases in superpositions of Gaussian states. It yields simulation algorithms with polynomial complexity in the number of fermions, the desired accuracy, and certain quantities capturing the degree of non-Gaussianity of the initial state. We study one such quantity, the fermionic Gaussian extent, and show that it is multiplicative on tensor products when the so-called fermionic Gaussian fidelity is. We establish this property for the tensor product of two arbitrary pure states of four fermions with positive parity.

Contents

1	Introduction	2
1.1	Efficiently simulable quantum computations	2
1.1.1	Clifford circuits / Stabilizer computations	4
1.1.2	Fermionic linear optics / Fermionic Gaussian computations	4
1.2	Classical simulation algorithms and measures of magic	5
1.3	The fermionic Gaussian extent	7
1.4	On the (sub)multiplicativity of the extent	8
1.5	Our contribution	9
1.6	Prior and related work	11
1.7	Outline	13
2	Background	13
2.1	Dirac and Majorana operators	13
2.2	Gaussian unitaries	15
2.3	Fermionic Gaussian (pure) states	16
2.4	Gaussianity condition	16
2.5	Covariance matrices, Gaussian states and Wick's theorem	17
2.6	Inner product formulas for Gaussian states	18

2.7	Gaussian evolution and occupation number measurement	18
2.8	The tensor product of two fermionic states	19
3	Tracking relative phases in fermionic linear optics	21
3.1	Subroutines	23
3.2	Computing overlaps and descriptions of evolved/measured states	28
3.3	Initial states for computation	37
4	Classical simulation of fermionic Gaussian circuits with non-Gaussian initial states	38
4.1	Extending simulation algorithms to superpositions	39
4.2	Sparsification: Relating \mathcal{D} -extent to approximate \mathcal{D} -rank	42
4.3	Fast norm estimation and approximate simulation	42
4.4	Fermionic linear optics with non-Gaussian initial states	44
4.5	Efficient additive-error strong simulation	46
5	Multiplicativity of the Gaussian fidelity for 4 fermions	47
5.1	Four-fermion Gaussian and non-Gaussian states	47
5.2	The Gaussian fidelity for 4-fermion states	49
5.3	Multiplicativity of the Gaussian fidelity for 4-fermion states	52
6	Multiplicativity of \mathcal{D}-fidelity implies that of \mathcal{D}-extent	55
6.1	Multiplicativity for finite dictionaries	55
6.2	Multiplicativity for infinite dictionaries	56
7	Multiplicativity of the Gaussian extent for four fermions	61
A	Alternative Gaussianity condition for 4-fermion states	61
B	Commutativity of the map θ and quadratic Majorana monomials	63

1 Introduction

While universal polynomial-time quantum computation is believed to exceed the capabilities of efficient classical algorithms, restricted classes of quantum computations are amenable to efficient classical simulation. Identifying such models and corresponding simulation algorithms is a central goal in the study of quantum computing. On the one hand, a good characterization of the boundary between the computational power of classical and quantum computational models provides insight into potential quantum advantages. On the other hand, efficient classical simulation methods can be used to assess the merits and scalability of quantum information-processing proposals. For example, the resilience of certain quantum codes against restricted noise models has successfully been studied by means of classical simulation methods giving threshold estimates for large-scale systems, see e.g., [1–6] for an incomplete list of relevant references.

1.1 Efficiently simulable quantum computations

Most known examples of efficiently simulable quantum computations can be summarized by the following ingredients:

- (i) A set \mathcal{D} of states with the property that each element $\Psi \in \mathcal{D}$ has a succinct classical description d_Ψ . In the following, we will refer to \mathcal{D} as a dictionary.

- (ii) A set \mathcal{E} of operations (unitary or non-unitary evolutions), again with the property that each element $E \in \mathcal{E}$ has a succinct classical description d_E . Following resource-theoretic conventions, we call \mathcal{E} the set of free operations.
- (iii) A set \mathcal{M} of measurements (quantum instruments) with an efficient classical descriptions d_M for each $M \in \mathcal{M}$, and the property that every post-measurement state (associated with different measurement outcomes) obtained by applying $M \in \mathcal{M}$ to a state $\Psi \in \mathcal{D}$ belongs to \mathcal{D} .

A triple $(\mathcal{D}, \mathcal{E}, \mathcal{M})$ gives rise to a (generally restricted) quantum computational model by composing these ingredients. A typical (non-adaptive) computation proceeds by preparing an initial state $\Psi \in \mathcal{D}$, applying a sequence $\{E_t\}_{t=1}^T \subset \mathcal{E}$ of operations, and performing measurements $\{M_k\}_{k=1}^L \subset \mathcal{M}$ in succession. Assuming for simplicity that \mathcal{E} consists of a set of unitaries, and that for each $k \in [L] = \{1, \dots, L\}$, the measurement M_k realizes a POVM $M_k = \{M_m^{(k)}\}_{m \in \mathcal{M}_k}$ with outcomes in a set \mathcal{M}_k , such a computation produces a sample $m = (m_1, \dots, m_L) \in \mathcal{M}_1 \times \dots \times \mathcal{M}_L$ from the distribution

$$p(m_1, \dots, m_L) = \langle \Psi_T, M_{m_L}^{(L)} \dots M_{m_1}^{(1)} \Psi_T \rangle \quad \text{where} \quad \Psi_T = E_T \circ \dots \circ E_1(\Psi). \quad (1)$$

More generally, one may consider circuits where operations are chosen adaptively depending on intermediate measurement results, assuming that the dependence is given by an efficiently computable function.

The task of classically simulating the computational model associated with $(\mathcal{D}, \mathcal{E}, \mathcal{M})$ comes in two flavors. The input in both cases is the collection $(d_\Psi, \{d_{E_t}\}_{t=1}^T, \{d_{M_k}\}_{k=1}^L)$ of descriptions of the initial state, the set of operations applied, and the measurements. The problem of weak simulation then consists in producing a sample $m \in \mathcal{M}_1 \times \dots \times \mathcal{M}_L$ drawn according to the (ideal) output distribution $p(m)$ of the circuit given by Eq. (1). In contrast, the problem of strong simulation consists of computing the output probability $p(m)$ for a given (potential) measurement outcome m .

Relaxing the requirements of weak and strong simulation, one may allow for an approximation error. For weak simulation, this is typically formalized by demanding that the (probabilistic) classical algorithm outputs a sample m drawn from a distribution \tilde{p} which is δ -close in L^1 -distance (for a chosen error parameter $\delta > 0$) to the ideal output distribution p . Similarly, for strong simulation, the output \tilde{p} is required to be close to the value $p(m)$ with a controlled (additive or multiplicative) error.

In cases where a computational model specified by $(\mathcal{D}, \mathcal{E}, \mathcal{M})$ is amenable to efficient classical simulation, associated classical simulation algorithms are typically constructed by considering evolution and measurement separately. The basic problem then consists in constructing efficient classical algorithms with the following functionalities:

- (a) An algorithm `evolve` which, given classical descriptions d_Ψ of a state $\Psi \in \mathcal{D}$ and d_E of an evolution operation $E \in \mathcal{E}$, computes a classical description $d_{E(\Psi)}$ of the evolved state $E(\Psi)$.
- (b) Given a classical description d_Ψ of a state $\Psi \in \mathcal{D}$, a classical description d_M of a measurement $M \in \mathcal{M}$ (with associated set of measurement outcomes \mathcal{M}_M) and a measurement outcome $m \in \mathcal{M}_M$,
 - (b1) an algorithm `measureprob` which outputs the probability $p(m)$ (determined by Born's rule) of obtaining measurement outcome m , and
 - (b2) an algorithm `postmeasure` which outputs a classical description of the post-measurement state

when applying the measurement M to Ψ .

It is clear that a triple (evolve, measureprob, postmeasure) of such algorithms immediately gives rise to an efficient algorithm for strong simulation of the model $(\mathcal{D}, \mathcal{E}, \mathcal{M})$, with a runtime

$$T \cdot \text{time}(\text{evolve}) + L \cdot (\text{time}(\text{measureprob}) + \text{time}(\text{postmeasure}))$$

which is linear in the number T of operations applied, and linear in the number L of measurements. Assuming that for any measurement $M \in \mathcal{M}$, the set of measurement outcomes \mathcal{M}_M associated with M is of constant cardinality, the triple (evolve, measureprob, postmeasure) also gives rise to a randomized algorithm for weak simulation: Such an algorithm is obtained by using measureprob to compute the entire distribution $\{p(m)\}_{m \in \mathcal{M}_M}$ of measurement outcomes (when applying a measurement M), and then drawing $m \in \mathcal{M}_M$ randomly according to this distribution. The runtime of this probabilistic algorithm is

$$T \cdot \text{time}(\text{evolve}) + L \cdot (\text{time}(\text{measureprob}) \cdot w + \text{time}(\text{postmeasure}))$$

where $w = \max_{M \in \mathcal{M}} |\mathcal{M}_M|$ bounds the maximal cardinality of the set of measurement outcomes.

1.1.1 Clifford circuits / Stabilizer computations

Perhaps the most well-known example of a computational model $(\mathcal{D}, \mathcal{E}, \mathcal{M})$ where efficient algorithms (evolve, measureprob, postmeasure) can be provided is the Gottesman-Knill-theorem for stabilizer computations on n qubits. Here \mathcal{D} is the set STAB_n of n -qubit stabilizer states (whose elements can be specified by their stabilizer generators, i.e., corresponding stabilizer tableaux), \mathcal{E} is the set of Clifford unitaries (described by symplectic matrices), and \mathcal{M} are measurements of single-qubit Pauli Z operators (described by an index $j \in [n]$). In this case, there are efficient algorithms with runtimes given in Table 1.

algorithm	time
evolve	$O(n)$
measureprob	$O(n)$
postmeasure	$O(n^2)$

Table 1: Runtimes of building blocks evolve, measureprob, postmeasure for classical simulation of n -qubit stabilizer circuits as given in [7]. Evolution corresponds to application of an n -qubit Clifford unitary, and each measurement is that of a Pauli observable Z_j with $j \in [n]$.

1.1.2 Fermionic linear optics / Fermionic Gaussian computations

A different class of efficiently simulable computations – the one we are interested in here – is that of fermionic linear optics on n fermions. We focus on pure-state computations: Here the dictionary \mathcal{D} consists of the set \mathcal{G}_n of pure fermionic Gaussian states. An element $\Psi \in \mathcal{G}_n$ in the dictionary can be described by its covariance matrix Γ_Ψ , an antisymmetric $2n \times 2n$ matrix with real entries. The set $\mathcal{E} = \mathcal{E}_{\text{Gauss}}$ can be taken as the set of Gaussian unitary operations. Each such unitary $U = U_R$ is fully determined by an element $R \in O(2n)$ of the orthogonal group on \mathbb{R}^{2n} , where $R \mapsto U_R$ defines a (projective) unitary

representation of $O(2n)$ on the space \mathcal{H}^n of n fermions. The set $\mathcal{M} = \mathcal{M}_{\text{number}}$ consists of all occupation number measurements. As in the case of stabilizer states, there are polynomial-time algorithms (`evolve`, `measureprob`, `postmeasure`) for classical simulation with runtimes summarized in Table 2. In particular, the covariance matrix $\Gamma_{U_R\Psi}$ of a Gaussian state Ψ evolved under a Gaussian unitary U_R can be computed in time $O(n^3)$ from Γ_Ψ and R . The outcome probability of observing 0 (respectively 1) when performing an occupation number measurement can be computed in time $O(1)$, and the covariance matrix of the post-measurement state can be computed in time $O(n^2)$ [8–10] (see also [3]).

algorithm	time
<code>evolve</code>	$O(n^3)$
<code>measureprob</code>	$O(1)$
<code>postmeasure</code>	$O(n^2)$

Table 2: Runtimes of building blocks `evolve`, `measureprob`, `postmeasure` for classical simulation of n -fermion linear optics circuits as proposed in [8–10], see also [3]. Evolution amounts to application of a fermionic Gaussian unitary. Measurement corresponds to measuring an observable $a_j^\dagger a_j$ (occupation number) for $j \in [n]$.

1.2 Classical simulation algorithms and measures of magic

A natural way of extending the power of a quantum computational model specified by $(\mathcal{D}, \mathcal{E}, \mathcal{M})$ consists in providing resources/capabilities that do not belong to the specified sets. “Magic states” are a prime example: Here a state $\Psi \notin \mathcal{D}$ not belonging to the dictionary is provided as an (initial) state in the quantum computation, thereby providing additional capabilities to the computational model. For example, non-Clifford unitaries can be realized by certain stabilizer-computations (sometimes referred to as “gadgets”) applied to so-called magic states [11]. Similarly, non-Gaussian initial states can be combined with fermionic linear optics operations to realize non-Gaussian operations [12, 13]. While such a magic state can even promote the computational model to universal quantum computation, this is generally not the case for all states Ψ . It is thus a natural question to quantify the degree of “magicness” provided by a state $\Psi \notin \mathcal{D}$. For the set STAB_n of n -qubit stabilizer states, corresponding magic monotones considered in the literature include the robustness of magic [14, 15], the exact and approximate stabilizer rank [16–18], the stabilizer extent [18, 19], the stabilizer nullity [20], the generalized robustness [21] and the magic entropy [22].

The maximum overlap of a given state Ψ with an element of the dictionary \mathcal{D} , i.e., the quantity

$$F_{\mathcal{D}}(\Psi) = \sup_{\varphi \in \mathcal{D}} |\langle \varphi, \Psi \rangle|^2, \quad (2)$$

is arguably one of the most direct ways of quantifying how far Ψ is from a “free” state, i.e., a state belonging to \mathcal{D} . Motivated by the analogously defined notion of stabilizer fidelity in Ref. [18], we call $F_{\mathcal{D}}(\Psi)$ the \mathcal{D} -fidelity of Ψ in the following. This quantity plays an important role in our arguments when considering multiplicativity properties. However, the \mathcal{D} -fidelity $F_{\mathcal{D}}(\Psi)$ is not a good quantifier of hardness of classical simulation because simply replacing Ψ by an element of \mathcal{D} typically leads to a significant approximation error.

From the point of view of classical simulation, a relevant magicness measure for a state $\Psi \notin \mathcal{D}$ relates to the (added) complexity when trying to simulate a quantum computation with initial state Ψ , built from a triple $(\mathcal{D}, \mathcal{E}, \mathcal{M})$ allowing for efficient classical simulation. One such measure, introduced in Ref. [16] for the case of stabilizer computations, is the \mathcal{D} -rank $\chi_{\mathcal{D}}(\Psi)$ of Ψ . (For $\mathcal{D} = \text{STAB}_n$, this is called the stabilizer rank of Ψ .) It is defined as the minimum number of terms when decomposing Ψ as a linear combination of elements of \mathcal{D} , i.e.,

$$\chi_{\mathcal{D}}(\Psi) = \min \left\{ \chi \in \mathbb{N} \mid \exists \{\varphi_j\}_{j=1}^{\chi} \subset \mathcal{D}, \{\gamma_j\}_{j=1}^{\chi} \subset \mathbb{C} \text{ such that } \Psi = \sum_{j=1}^{\chi} \gamma_j \varphi_j \right\}. \quad (3)$$

In the context of signal processing, the corresponding optimization problem is referred to as a sparse approximation problem. The \mathcal{D} -rank $\chi_{\mathcal{D}}(\Psi)$ appears naturally when constructing and analyzing simulation algorithms, but it suffers from a number of shortcomings: On the one hand, the set of states $\Psi \in \mathcal{H}$ whose \mathcal{D} -rank is less than the dimension of the Hilbert space \mathcal{H} is a set of zero Lebesgue measure [23, Proposition 4.1]. On the other hand, the quantity $\chi_{\mathcal{D}}(\Psi)$ relates to the classical simulation cost of exactly simulating dynamics involving the state Ψ . In practice, some approximation error is typically acceptable, and corresponding simulations can be achieved with lower cost. In other words, the quantity $\chi_{\mathcal{D}}(\Psi)$ does not accurately reflect the cost of approximate simulation.

A more operationally relevant quantity is the δ -approximate \mathcal{D} -rank $\chi_{\mathcal{D}}^{\delta}(\Psi)$ of Ψ introduced in Ref. [17], again for stabilizer computations. For a fixed approximation error $\delta > 0$, this is given by the minimum \mathcal{D} -rank of any state Ψ' that is δ -close to Ψ , i.e.,

$$\chi_{\mathcal{D}}^{\delta}(\Psi) = \min \{ \chi_{\mathcal{D}}(\Psi') \mid \Psi' \in \mathcal{H} \text{ such that } \|\Psi - \Psi'\| \leq \delta \}. \quad (4)$$

An exact classical simulation algorithm whose complexity scales with the exact \mathcal{D} -rank $\chi_{\mathcal{D}}(\Psi)$ provides an approximate simulation at a cost with an identical scaling in the approximate (instead of exact) \mathcal{D} -rank $\chi_{\mathcal{D}}^{\delta}(\Psi)$ of Ψ . Here approximate weak simulation means that instead of sampling from the ideal output distribution P of a circuit, the simulation samples from a distribution P' whose L^1 -distance from P is bounded by $O(\delta)$. Similarly, in approximate (strong) simulation, output probabilities are approximately computed with a controlled approximation error.

A different quantity of interest is obtained by replacing the ill-behaved rank function (i.e., size of the support) in the definition of the \mathcal{D} -rank $\chi_{\mathcal{D}}(\Psi)$ by the L^1 -norm of the coefficients when representing Ψ as a linear combination. In the context of stabilizer states the corresponding quantity was introduced by Bravyi et al. [18] under the term stabilizer extent: For a state $\Psi \in (\mathbb{C}^2)^{\otimes n}$ it is defined as

$$\xi_{\text{STAB}_n}(\Psi) = \inf \left\{ \|\gamma\|_1 \mid \gamma : \text{STAB}_n \rightarrow \mathbb{C} \text{ such that } \Psi = \sum_{\varphi \in \text{STAB}_n} \gamma(\varphi) \varphi \right\},$$

where $\|\gamma\|_1 = \sum_{\varphi \in \text{STAB}_n} |\gamma(\varphi)|$ denotes the 1-norm of γ . The corresponding convex optimization problem is known as the basis pursuit problem [24] (when STAB_n is replaced by e.g., a finite dictionary \mathcal{D}). Sufficient conditions for when the basis pursuit problem yields a solution of the sparse approximation problem were investigated in a series of works culminating in Fuchs' condition [25] (see also [26]). More importantly for (approximate) simulation, feasible solutions of the basis pursuit problem provide upper bounds on the sparse approximation problem. For the stabilizer rank, a sparsification result (see [18, Theorem 1]) gives an upper bound on the δ -approximate stabilizer rank $\chi_{\text{STAB}_n}^{\delta}(\Psi)$ in terms of the stabilizer extent $\xi_{\text{STAB}_n}(\Psi)$, for any $\delta > 0$ (see Section 4.2).

Building on earlier results [17], it was shown in Ref. [18] that a stabilizer circuit on n qubits with L Clifford gates initialized in a state Ψ can be weakly simulated with error δ in a time scaling as $O(\xi_{\text{STAB}_n}(\Psi)/\delta^2 \cdot \text{poly}(n, L))$. The error δ expresses the L^1 -norm distance of the distribution of simulated measurement outcomes from the output distribution of the actual quantum computation. Here we are not accounting for the time required to perform classical computations when adaptive quantum circuits are considered. In addition, Ref. [27] provided a classical algorithm for strong simulation of a circuit U with L Clifford gates and t T -gates initialized in a stabilizer state Ψ with an additive error δ . Their algorithm outputs an estimate of the probability $|\langle x, U\Psi \rangle|^2$ of obtaining measurement outcome $x \in \{0, 1\}^n$ up to an additive error δ , with success probability greater than $1 - p_f$. It has runtime $O(\xi_{\text{STAB}_n}(|T\rangle^{\otimes t}) \log(1/p_f) \cdot \text{poly}(n, L, \delta^{-1}))$, scaling linearly with the stabilizer extent $\xi_{\text{STAB}_n}(|T\rangle^{\otimes t})$ of t copies of the single-qubit magic state $|T\rangle$ associated with a T -gate [11].

1.3 The fermionic Gaussian extent

In the following, we generalize the notion of the extent beyond stabilizer computations to any dictionary \mathcal{D} . We refer to the corresponding quantity as the \mathcal{D} -extent $\xi_{\mathcal{D}}(\Psi)$ of Ψ . We assume throughout that we are interested in pure state quantum computations on a Hilbert space \mathcal{H} , and that the dictionary \mathcal{D} is a subset of pure states on \mathcal{H} . Then the \mathcal{D} -extent $\xi_{\mathcal{D}}(\Psi)$ of $\Psi \in \mathcal{H}$ is defined as

$$\xi_{\mathcal{D}}(\Psi) = \inf_{N \in \mathbb{N}} \inf_{\varphi_1, \dots, \varphi_N \in \mathcal{D}} \left\{ \|\gamma\|_1^2 \mid \gamma \in \mathbb{C}^N \text{ such that } \sum_{j=1}^N \gamma_j \varphi_j = \Psi \right\}. \quad (5)$$

Here $\|\gamma\|_1 = \sum_{j=1}^N |\gamma_j|$ is the L^1 -norm of the vector γ . That is, the \mathcal{D} -extent $\xi_{\mathcal{D}}(\Psi)$ is the L^1 -norm of the coefficients minimized over all decompositions of Ψ into a finite linear combination of elements of the dictionary \mathcal{D} . As mentioned above, quantities of the form (5) are well-studied in the context of signal-processing.

When the dictionary \mathcal{D} is a finite subset of a Hilbert space $\mathcal{H} \cong \mathbb{C}^d$, the \mathcal{D} -extent $\xi_{\mathcal{D}}(\Psi)$ of a state $\Psi \in \mathcal{H}$ can be expressed as a second-order cone program [28] (see also e.g., [29]), as in Appendix A of Ref. [19]. Second-order cone programs can be solved in time polynomial in $\max(d, |\mathcal{D}|)$. We are typically interested in cases where \mathcal{D} contains a basis of \mathcal{H} (such that every state can indeed be represented as a linear combination of dictionary elements): Here this runtime is at least polynomial in d . For example, in the case $\mathcal{D} = \text{STAB}_n$ of stabilizer states on n qubits, this leads to an exponential scaling in n^2 . Beyond algorithmic considerations related to the evaluation of the extent, the fact that $\xi_{\mathcal{D}}(\Psi)$ is given by a second-order cone program provides useful analytical insight by convex programming duality. Indeed, this fact has previously been exploited both for showing multiplicativity of the stabilizer extent for states of small dimension [18], as well as to show non-multiplicativity in high dimensions [19]. In Section 6, we also exploit this connection to relate the \mathcal{D} -fidelity $F_{\mathcal{D}}(\Psi)$ with the \mathcal{D} -extent $\xi_{\mathcal{D}}(\Psi)$.

In contrast, the \mathcal{D} -extent $\xi_{\mathcal{D}}(\Psi)$ for an infinite, i.e., continuously parameterized, dictionary \mathcal{D} constitutes additional mathematical challenges as an optimization problem. This is the case of interest here as we are considering the dictionary $\mathcal{D} = \mathcal{G}_n$ consisting of all n -fermion Gaussian states in the following. We call the associated quantity $\xi_{\mathcal{G}_n}(\Psi)$ the (fermionic) Gaussian extent of an n -fermion state Ψ . Our focus here is on discussing the role of the quantity $\xi_{\mathcal{G}_n}(\Psi)$ in the context of classically simulating fermionic linear optics, and its behavior on tensor products. A detailed discussion of the algorithmic problem of computing $\xi_{\mathcal{G}_n}(\Psi)$ for an arbitrary state Ψ , and finding a corresponding optimal

decomposition of Ψ into a linear combination of Gaussian states is beyond the scope of this work. We refer to e.g., [30] where semidefinite relaxations are given for the related atomic norm minimization problem in cases where the atomic set (corresponding to the dictionary) has algebraic structure. Similar techniques may be applicable to the fermionic Gaussian extent.

1.4 On the (sub)multiplicativity of the extent

Consider a situation where an operation $E \notin \mathcal{E}$ not belonging to the set \mathcal{E} of efficiently simulable operations is implemented by using a “magic” resource state $\Psi \notin \mathcal{D}$. For example, if $\mathcal{D} = \text{STAB}_n$ is the set of stabilizer states, \mathcal{E} the set of Clifford unitaries and \mathcal{M} the set of single-qubit Pauli- Z -measurements, then a non-Clifford gate (such as the T -gate) can be realized by an (adaptive) Clifford circuit at the cost of consuming a non-Clifford state (such as the state $|T\rangle$) [11]. Similar “gadget constructions” exist for fermionic linear optics, where non-Gaussian unitaries are realized by Gaussian unitaries and non-Gaussian states [12, 13]. A natural question arising in this situation is to characterize the cost of simulating the application of two independent magic gates $E_1, E_2 \notin \mathcal{E}$, each realized by efficiently simulable operations (belonging to \mathcal{E}) using magic states Ψ_1, Ψ_2 . For any reasonable simulation algorithm, we expect the required simulation effort to increase at most multiplicatively. Indeed, this feature is reflected in the submultiplicativity property

$$\xi_{\mathcal{D}_3}(\Psi_1 \otimes \Psi_2) \leq \xi_{\mathcal{D}_1}(\Psi_1)\xi_{\mathcal{D}_2}(\Psi_2) \quad \text{for all} \quad \Psi_1 \in \mathcal{H}_1 \text{ and } \Psi_2 \in \mathcal{H}_2 \quad (6)$$

of the \mathcal{D} -extent. In Eq. (6), we are considering Hilbert spaces $\mathcal{H}_1, \mathcal{H}_2$ and their tensor product $\mathcal{H}_3 = \mathcal{H}_1 \otimes \mathcal{H}_2$, as well as dictionaries $\mathcal{D}_j \subset \mathcal{H}_j$ for $j \in [3]$. The submultiplicativity property (6) follows immediately from the definition of the extent if the three dictionaries satisfy the inclusion property

$$\mathcal{D}_1 \otimes \mathcal{D}_2 \subset \mathcal{D}_3 . \quad (7)$$

In particular, this is satisfied e.g., when the dictionary $\mathcal{D}_j = \text{STAB}_{n_j} \subset (\mathbb{C}^2)^{\otimes n_j}$ is the set of n_j -qubit stabilizer states for $j \in [3]$, with $n_3 = n_1 + n_2$, or when considering the set of (even) Gaussian states (see below).

While the submultiplicativity property (6) is a trivial consequence of Eq. (7), the question of whether or not the stronger multiplicativity property

$$\xi_{\mathcal{D}_3}(\Psi_1 \otimes \Psi_2) = \xi_{\mathcal{D}_1}(\Psi_1)\xi_{\mathcal{D}_2}(\Psi_2) \quad \text{for all} \quad \Psi_1 \in \mathcal{H}_1 \text{ and } \Psi_2 \in \mathcal{H}_2 \quad (8)$$

holds for the \mathcal{D} -extent is a much less trivial problem. If the multiplicativity property (8) is satisfied, then computing the extent of a product state can be broken down into several smaller optimization problems: It suffices to compute the extent of each factor in the tensor product. Furthermore, the classical simulation cost (with typical algorithms) when applying several non-free (“magic”) gates constructed by gadgets increases at an exponential rate determined by the individual gates. In contrast, if the extent is not multiplicative (i.e., the equality in (8) is not satisfied for some states $\Psi_j \in \mathcal{H}_j$, $j \in [2]$), then such a simplification is not possible. More surprisingly, such a violation of multiplicativity implies that the classical simulation cost of applying certain non-free gates can be reduced by treating these jointly instead of individually. We note that in the slightly different context of so-called circuit knitting, similar savings in complexity have been shown to be significant [31].

Previous work established that the stabilizer extent is multiplicative even for multiple factors, that is,

$$\xi_{\text{STAB}_{n_1+\dots+n_r}}(\Psi_1 \otimes \dots \otimes \Psi_r) = \prod_{j=1}^r \xi_{\text{STAB}_{n_j}}(\Psi_j) \quad \text{for all} \quad \Psi_j \in (\mathbb{C}^2)^{\otimes n_j}, j \in [r]$$

if the factors are single-qubit, 2- or 3-qubit states, i.e., $n_j \in [3]$, see Ref. [18]. An example is the stabilizer extent of a tensor product of t copies of the magic (single-qubit) state $|T\rangle = (|0\rangle + e^{i\pi/4}|1\rangle)/\sqrt{2}$ associated with the T -gate. Multiplicativity for qubit states gives $\xi_{\text{STAB}_1}(|T\rangle^{\otimes t}) = \xi_{\text{STAB}_t}(|T\rangle)^t$, where $\xi_{\text{STAB}_1}(|T\rangle)$ is known to be approximately 1.17 [17]. This translates to an overhead exponential in t in the runtime of stabilizer computations supplemented with t T -gates. Surprisingly, the stabilizer extent has been shown not to be multiplicative (for all pairs of states) in high dimensions [19].

For (pure) Gaussian states, the Gaussian extent of a 1-, 2- and 3-mode pure fermionic state is trivially one because any 1-, 2- and 3-mode pure fermionic state is Gaussian [32] and is thus an element of the dictionary. Hence the Gaussian extent is (trivially) multiplicative if the factors are 1-, 2- or 3-mode fermionic states. The simplest non-trivial case is that of $n = 4$ fermionic modes in each factor.

1.5 Our contribution

Our results concern fermionic linear optics, the computational model introduced in Section 1.1.2 described by the triple $(\mathcal{G}_n, \mathcal{E}_{\text{Gauss}}, \mathcal{M}_{\text{number}})$ of fermionic Gaussian pure states on n fermions, Gaussian unitary operations and number state measurements. We propose classical simulation algorithms for the case where the initial state $\Psi \in \mathcal{H}_n$ is an arbitrary pure state in the n -fermion Hilbert space \mathcal{H}_n (instead of belonging to the set $\mathcal{G}_n \subset \mathcal{H}_n$ of Gaussian states). Our results are two-fold:

New simulation algorithms. We give algorithms realizing the functionalities described in Section 1.1 exactly for the triple $(\mathcal{H}_n, \mathcal{E}_{\text{Gauss}}, \mathcal{M}_{\text{number}})$. This immediately gives rise to efficient algorithms for weak and strong simulation of circuits with non-Gaussian initial states. The corresponding runtimes of these building blocks, which we refer to as $(\chi\text{evolve}, \chi\text{measureprob}, \chi\text{postmeasure})$, depend on the Gaussian rank $\chi = \chi_{\mathcal{G}_n}(\Psi)$ of the initial state Ψ and are summarized in Table 3.

algorithm	time
χevolve	$O(\chi n^3)$
$\chi\text{measureprob}$	$O(\chi^2 n^3)$
$\chi\text{postmeasure}$	$O(\chi n^3)$

Table 3: Runtimes of the building blocks χevolve , $\chi\text{measureprob}$, $\chi\text{postmeasure}$ for exact simulation of n -qubit fermionic linear optics circuits with a non-Gaussian initial state Ψ of Gaussian rank $\chi = \chi_{\mathcal{G}_n}(\Psi)$. Evolution corresponds to the application of a Gaussian unitary from a set $\mathcal{E}_{\text{Gauss}}$ of generators (specified below), and the set of measurements is given by occupation number measurements on each of the modes.

Key to the construction of these algorithms is a novel way of keeping track of relative phases in superpositions of Gaussian states, see Section 3. We argue that our techniques can be applied more generally to adapt simulation procedures developed, e.g., for Clifford circuits, to the setting of fermionic linear optics. In order to illustrate this procedure,

we apply it to the simulation algorithms of [17, 18] for efficient (approximate) classical simulation algorithms. In this way, we obtain new approximate simulation algorithms with runtimes depending linearly on the fermionic Gaussian extent $\xi = \xi_{\mathcal{G}_n}(\Psi)$ of the initial state Ψ , see Table 4 for a summary of the corresponding runtimes. They depend

algorithm	time
approx evolve	$O(\xi \delta^{-2} n^3)$
approx measure prob	$O(\xi \delta^{-2} \epsilon^{-2} p_f^{-1} n^{7/2})$
approx post measure	$O(\xi \delta^{-2} n^3)$

Table 4: Runtimes of building blocks `approx evolve`, `approx measure prob`, `approx post measure` for approximate simulation of n -qubit fermionic linear optics circuits with a non-Gaussian initial state Ψ of Gaussian extent $\xi = \xi_{\mathcal{G}_n}(\Psi)$. The parameters (ϵ, δ, p_f) determine the quality of the approximation.

inverse-polynomially on parameters (δ, ϵ, p_f) determining the accuracy of the simulation. The error δ describes a certain “offset”, i.e., a systematic error: Instead of simulating the dynamics of the circuit with the (ideal) initial state Ψ , the simulation algorithm emulates the dynamics when using a different starting state $\tilde{\Psi}$ which is δ -close to Ψ , i.e., which satisfies $\|\Psi - \tilde{\Psi}\| \leq \delta$. The algorithm `approx evolve` computes evolution exactly on the state used in the simulation (i.e., it preserves the approximation error δ relative to the ideal initial state). In contrast, the procedure `approx measure prob` can fail with probability p_f , and both `approx measure prob` and `approx post measure` introduce an additional error quantified by ϵ (if `approx measure prob` succeeds): Instead of returning the probability $p(0)$ of obtaining zero occupation number when measuring the state, the output of `approx measure prob` is a value \tilde{p} which satisfies $|\tilde{p} - p(0)| \leq O(\epsilon)$. Similarly, the output of `approx post measure` is a description of a state that is $O(\epsilon)$ -close to the actual post-measurement state. These parameters and runtimes are analogous to those obtained in [18] for simulating Clifford circuits with non-stabilizer initial states. In particular, they imply that a circuit with initial state Ψ involving T Gaussian unitaries and L occupation number measurements can be weakly simulated in time $\tilde{O}(\epsilon^{-2} \xi)$, such that the sampled measurement outcomes are ϵ -close in L^1 -distance to the ideal (joint) output distribution of all measurements. Here the notation $\tilde{O}(\cdot)$ suppresses a factor polynomial in n, T, L and $\log(\epsilon^{-1})$, see [17] for details.

On the multiplicativity of the Gaussian extent and the Gaussian fidelity. Motivated by the relevance of the Gaussian extent $\xi_{\mathcal{G}_n}(\Psi)$ for characterizing the complexity of classical simulation, we study multiplicativity properties of both the \mathcal{D} -fidelity $F_{\mathcal{D}}(\Psi)$ as well as the \mathcal{D} -extent $\xi_{\mathcal{D}}(\Psi)$ for a general infinite, i.e., continuously parameterized, dictionary \mathcal{D} . We show that multiplicativity of the \mathcal{D} -fidelity is closely related to that of the \mathcal{D} -extent: For a general family of (discrete or continuous) dictionaries $\mathcal{D}_j \subset \mathcal{H}_j$ for $j \in [3]$ with the property

$$\mathcal{D}_1 \otimes \mathcal{D}_2 \subset \mathcal{D}_3 ,$$

multiplicativity of the \mathcal{D} -fidelity, i.e.,

$$F_{\mathcal{D}_3}(\Psi_1 \otimes \Psi_2) = F_{\mathcal{D}_1}(\Psi_1) F_{\mathcal{D}_2}(\Psi_2) \quad \text{for all } \Psi_j \in \mathcal{H}_j \text{ for } j \in [2]$$

implies multiplicativity of the \mathcal{D} -extent, i.e.

$$\xi_{\mathcal{D}_3}(\Psi_1 \otimes \Psi_2) = \xi_{\mathcal{D}_1}(\Psi_1) \xi_{\mathcal{D}_2}(\Psi_2) \quad \text{for all } \Psi_j \in \mathcal{H}_j \text{ for } j \in [2] .$$

We note that for stabilizer states $\mathcal{D} = \text{STAB}_n$, a similar route was followed in Ref. [18] to show multiplicativity of the stabilizer extent ξ_{STAB_n} with respect to the tensor product of 1-, 2- and 3-qubit states. Our main contribution is an extension of this connection to the case of infinite dictionaries by the use of nets. We expect this connection to be helpful in proving or disproving multiplicativity of the extent more generally.

We subsequently make use of this connection to the Gaussian fidelity to show that the fermionic Gaussian extent is multiplicative for the tensor product of any two 4-mode fermionic states with positive parity, i.e.,

$$\xi_{\mathcal{G}_8}(\Psi_1 \otimes \Psi_2) = \xi_{\mathcal{G}_4}(\Psi_1)\xi_{\mathcal{G}_4}(\Psi_2) \quad \text{for all} \quad \Psi_1, \Psi_2 \in \mathcal{H}_+^4. \quad (9)$$

Here \mathcal{H}_+^4 denotes the set of 4-mode fermionic states with positive parity. The proof of (9) relies on the Schmidt decomposition of fermionic Gaussian states and specific properties of 4-mode (positive parity) fermionic states.

The result (9) gives the first non-trivial example of multiplicativity of the Gaussian extent. Multiplicativity for more general cases such as that of multiple 4-mode fermionic factors remains an open problem.

1.6 Prior and related work

The starting point of our work is the fact that fermionic Gaussian operations acting on Gaussian states can be efficiently simulated classically as shown in pioneering work by Terhal and DiVincenzo [9] and Knill [10]. The model and its simulability are closely related to that of matchgate computations introduced by Valiant [8], where so-called matchgates correspond to a certain subset of Gaussian operations (see also [33]). In analogy to the fermionic context, the efficient simulability of bosonic Gaussian circuits was recognized at around the same time [34, 35]. In an effort to identify commonalities between simulation algorithms for a variety of quantum computational models, Somma et al. [36] provided a unifying Lie algebraic treatment which gives a counterpart to the Gottesman-Knill theorem for the simulability of Clifford circuits [7, 37].

While matchgate circuits, fermionic and bosonic linear optics, and Clifford circuits provide rich classes of efficiently simulable models for the study of many-body dynamics associated with quantum circuits, it is desirable to extend the applicability of such simulation methods. There has been significant interest in this problem resulting in a range of approaches. We only briefly discuss these here to give an overview, without attempting to give an exhaustive treatment.

A first prominent approach is the use of quasi-probability distributions to describe states and corresponding dynamics. Such a description typically applies to a subset of density operators: For example, it has been shown in [38, 39] in the context of continuous-variable systems that circuits applying bosonic Gaussian operations to initial states with a positive Wigner function (a strict superset of the set of bosonic Gaussian states) can be simulated efficiently. Negativity of the Wigner function (both in the continuous-variable as well as the qubit context) thus serves as a resource for quantum computation, also see e.g., [40, 41]. It is also closely related to contextuality, see [42], and thus connects contextuality to the complexity of classical simulation [43, 44]. Not unlike the notorious sign problem in quantum Monte-Carlo methods applied in many-body physics, the run-times of corresponding (randomized) simulation algorithms scale with certain measures of “negativity” of the initial state.

The concept of a convex-Gaussian state was introduced and studied in [32] to extend the range of fermionic linear optics simulation methods. This is related to quasi-probability

representations in the sense that initial states of a particular form are shown to lead to efficient simulability. Here a density operator is called convex-Gaussian if it is a convex combination of fermionic Gaussian states. The utility of this concept was illustrated in [32] by showing a converse to the fault-tolerance threshold theorem: Sufficiently noisy quantum circuits can be simulated classically because the corresponding states turn out to be convex-Gaussian. A detailed characterization of convex-Gaussianity is necessary to translate this into explicit (numerical) threshold estimates. An infinite hierarchy of semidefinite programs was constructed in [32] to detect convex-Gaussianity, and this was subsequently shown to be complete [45]. This hierarchy also provides a way of determining whether a state is close to being convex-Gaussian [45].

A second important class of approaches are rank-based methods. Here the non-free resource (either a state or an operation) is decomposed into a linear combination of free (i.e., efficiently simulable) resources. Our work follows this approach, which is detailed in Section 4.1 for pure states. For Clifford computations, this involves writing general states as superpositions of stabilizer states. The development of such simulators was pioneered by Bravyi, Smith, and Smolin [16] with subsequent work dealing with approximate stabilizer decompositions [17].

The concept of low-rank (approximate) decompositions of quantum states or operations into more easily treatable basic objects appears in a variety of forms: For example, the work [18] also discusses – in addition to state vector decompositions – decompositions of non-Clifford unitaries into sums of Clifford operations. In Ref. [46], a similar approach was taken to approximately decompose non-Gaussian fermionic unitary operations into linear combinations of Gaussian channels. In all these cases, the main challenge is to identify optimal (or simply good) decompositions (e.g., in terms of rank or an extent-like quantity).

In more recent work, Mocherla, Lao and Browne [47] study the problem of simulating matchgate circuits using universality-enabling gates. They provide a simulation algorithm and associated runtime estimates for estimating expectation values of single-qubit observables in output states obtained by applying a matchgate circuit to a product state. This problem is closely related to the problem considered in this work as matchgate circuits efficiently describe evolution under a quadratic fermionic Hamiltonian. The approach taken in [47] is quite different from ours, however: The classical simulator keeps track of the density operator by tracking its coefficients in the Pauli (operator) basis, using the structure of corresponding linear maps associated with matchgates. The effect of a specific set of universality-enabling gates is then analyzed in detail. This extends the sparse simulation method for matchgate circuits to circuits augmented with such gates. The runtime estimates of [47] apply to certain universality-providing gates. In contrast, our constructions can in principle also be applied to (gadget-based) constructions of arbitrary gates and provide gate-specific information. For gates close to the identity, for example, this may provide additional resource savings (in terms of e.g., the rate of growth for several uses of such a gate).

Near the completion of our work, we became aware of concurrent independent work on fermionic circuits with non-Gaussian operations, see the papers [48, 49] which were posted simultaneously with our work to the arXiv. Reference [48] proposes an alternative classical simulation algorithm for such circuits whose basic building blocks have identical runtime as our algorithm. In particular, when applying a generator $\exp(\vartheta/2c_jc_k)$ of the Gaussian unitary group, the runtime of the corresponding state update routine is $O(n^3)$ as in our work. Specifying a circuit in terms of a sequence of generators is a common assumption when discussing restricted gate sets. If instead, a general Gaussian unitary

is specified by an element $R \in SO(2n)$, the runtimes are different: By decomposing R into $O(n^2)$ Givens rotations (generators), our techniques yield a runtime of order $O(n^5)$, whereas the algorithm of [48] runs in time $O(n^4)$ in this case. A more significant distinction is the underlying approach: While we rely on a minimal extension of the covariance matrix formalism, the approach of [48] involves an explicit decomposition of fermionic Gaussian unitaries inspired by a certain canonical form of Clifford circuits [18]. This decomposition of a Gaussian unitary U is subsequently used to specify a Gaussian state $U|0\rangle$ including its global phase.

The work [49] analyses the set of fermionic Gaussian states to investigate properties of the (exact and approximate) Gaussian rank, the Gaussian fidelity and the Gaussian extent. A major result obtained in [49] is the multiplicativity of the extent for a tensor product state with any number of 4-mode factors. This mirrors what is known for two-qubit and three-qubit factors in the stabilizer context [18]. The proof in Ref. [49] is inspired by Ref. [19] and makes use of tools from convex optimization. It involves the use of a net of Gaussian states to establish an upper bound on the Gaussian fidelity. While we only establish multiplicativity for two factors (i.e., a special case of their result), our approach is different and may be of independent interest: We use a net to show that multiplicativity of the Gaussian fidelity implies that of the Gaussian extent, and employ the Schmidt decomposition [50] of fermionic Gaussian states.

1.7 Outline

The paper is structured as follows. In Section 2, we give some background on fermionic linear optics, reviewing fermionic Gaussian operations and states, inner product formulas for Gaussian states and tensor products of fermionic systems. In Sections 3 and 4 we describe classical algorithms for simulation of Gaussian and non-Gaussian fermionic circuits, respectively. Specifically, in Section 3 we provide an algorithm `overlap` for computing the overlap of two Gaussian states, an algorithm `evolve` to simulate unitary evolution of a Gaussian state, and algorithms `measureprob` and `postmeasure` to simulate measurements of occupation numbers. All these algorithms keep track of the phase of the state. In Section 4 we extend the simulation described in Section 3 to allow for non-Gaussian input states. The remainder of this work is focused on the multiplicativity of the fermionic Gaussian extent. In Section 5, we prove the multiplicativity of the fermionic Gaussian fidelity for the tensor product of any two 4-mode fermionic states with positive parity. Section 6 is devoted to showing that the multiplicativity of the \mathcal{D} -fidelity implies multiplicativity of the \mathcal{D} -extent for general (finite and infinite, i.e., continuously parameterized) dictionaries. Finally, the results from Sections 5 and 6 are used to prove the main result in Section 7, namely the multiplicativity of the fermionic Gaussian extent for the tensor product of any two 4-mode fermionic states with positive parity.

2 Background

In this section, we give some background on fermionic linear optics to fix notation.

2.1 Dirac and Majorana operators

Throughout, we consider fermionic systems composed of n modes, with (Dirac) creation- and annihilation operators a_j^\dagger, a_j , $j \in [n]$, satisfying the canonical anticommutation rela-

tions

$$\{a_j, a_k^\dagger\} = \delta_{j,k} I \quad \text{and} \quad \{a_j, a_k\} = \{a_j^\dagger, a_k^\dagger\} = 0 \quad \text{for all} \quad j, k \in [n] .$$

The fermionic vacuum state $|0_F\rangle$ is the (up to a phase) unique unit vector satisfying $a_j |0_F\rangle = 0$ for all $j \in [n]$. For $x = (x_1, \dots, x_n) \in \{0, 1\}^n$, we define the number state $|x\rangle$ by

$$|x\rangle = (a_1^\dagger)^{x_1} \cdots (a_n^\dagger)^{x_n} |0_F\rangle . \quad (10)$$

The states $\{|x\rangle\}_{x \in \{0,1\}^n}$ are an orthonormal basis of the underlying Hilbert space $\mathcal{H}^n \cong (\mathbb{C}^2)^{\otimes n}$. A state $|x\rangle$ is a simultaneous eigenstate of the occupation number operators $a_j^\dagger a_j$, $j \in [n]$, where x_j is the eigenvalue of $a_j^\dagger a_j$. For later reference, we note that

$$a_j |x\rangle = (-1)^{\eta_j(x)} x_j |x \oplus e_j\rangle \quad \text{and} \quad a_j^\dagger |x\rangle = (-1)^{\eta_j(x)} \bar{x}_j |x \oplus e_j\rangle , \quad (11)$$

with the definition

$$\eta_j(x) = \sum_{k=1}^{j-1} x_k \quad \text{for} \quad j \in [n] , \quad (12)$$

where we write $\bar{0} = 1$ and $\bar{1} = 0$, where $e_j \in \{0, 1\}^n$ is given by $(e_j)_k = \delta_{j,k}$ for $k \in [n]$, and where \oplus denotes bitwise addition modulo 2.

It will be convenient to work with Majorana operators $\{c_j\}_{j=1}^{2n}$ defined by

$$c_{2j-1} = a_j + a_j^\dagger \quad \text{and} \quad c_{2j} = i(a_j - a_j^\dagger) . \quad (13)$$

Majorana operators are self-adjoint and satisfy the relations

$$\{c_j, c_k\} = 2\delta_{jk} I \quad \text{and} \quad c_j^2 = I \quad \text{for} \quad j, k \in [2n] .$$

For $\alpha \in \{0, 1\}^{2n}$, we call the self-adjoint operator

$$c(\alpha) = i^{|\alpha| \cdot (|\alpha|-1)/2} c_1^{\alpha_1} \cdots c_{2n}^{\alpha_{2n}}$$

a Majorana monomial. Here $|\alpha| = \sum_{j=1}^{2n} \alpha_j$ denotes the Hamming weight of $\alpha \in \{0, 1\}^{2n}$. The set $\{c(\alpha)\}_{\alpha \in \{0,1\}^{2n}}$ constitutes an orthonormal basis of the real vector space of self-adjoint operators on \mathcal{H}^n equipped with the (normalized) Hilbert-Schmidt inner product $\langle A, B \rangle = 2^{-n} \text{tr}(A^\dagger B)$. The Majorana monomials satisfy

$$c(y)c(x) = (-1)^{|x| \cdot |y| + x \cdot y} c(x)c(y) \quad \text{with} \quad x, y \in \{0, 1\}^{2n} ,$$

where $x \cdot y = \sum_{j=1}^n x_j y_j$. In particular, if either x or y have even Hamming weight then $c(x)c(y) = (-1)^{x \cdot y} c(y)c(x)$. In the following, we will denote the set of even- and odd-weight $2n$ -bit strings by $\{0, 1\}_+^{2n}$ and $\{0, 1\}_-^{2n}$, respectively.

The parity operator

$$P = i^n c_1 c_2 \cdots c_{2n}$$

is the Majorana monomial associated with $\alpha = 1^{2n} = (1, \dots, 1)$. The parity operator commutes with every even-weight Majorana monomial and anti-commutes with every odd-weight Majorana monomial, i.e., we have

$$Pc(\alpha) = (-1)^{|\alpha|} c(\alpha)P \quad \text{for every} \quad \alpha \in \{0, 1\}^{2n} . \quad (14)$$

The Hilbert space $\mathcal{H}^n = \mathcal{H}_+^n \oplus \mathcal{H}_-^n$ associated with n fermions decomposes into a direct sum of positive- and negative-parity vectors

$$\begin{aligned}\mathcal{H}_+^n &= \{\Psi \in \mathcal{H}^n \mid P\Psi = \Psi\} \quad \text{and} \\ \mathcal{H}_-^n &= \{\Psi \in \mathcal{H}^n \mid P\Psi = -\Psi\} .\end{aligned}$$

We call a state $\Psi \in \mathcal{H}^n$ of definite parity if either $\Psi \in \mathcal{H}_+^n$ or $\Psi \in \mathcal{H}_-^n$. An element $X \in \mathcal{B}(\mathcal{H}^n)$ belonging to the set $\mathcal{B}(\mathcal{H}^n)$ of linear operators on \mathcal{H}^n is called even (odd) if it is a linear combination of Majorana monomials $c(\alpha)$ with $\alpha \in \{0, 1\}^{2n}$ of even (odd) weight. An immediate consequence of these definitions is that a state $\Psi \in \mathcal{H}^n$ has definite parity if and only if $|\Psi\rangle\langle\Psi|$ is even (see e.g., [51, Proposition 1] for a proof).

2.2 Gaussian unitaries

A unitary operator U on \mathcal{H}^n is Gaussian if and only if it maps a Majorana operator c_j to a linear combination of Majorana operators, i.e.

$$Uc_jU^\dagger = \sum_{k=1}^{2n} R_{jk}c_k , \quad (15)$$

where $R \in O(2n)$ is a real orthogonal matrix. Ignoring overall phases, the group of Gaussian unitary operators is generated by operators of the form

$$U_{j,k}(\vartheta) = \exp(\vartheta/2c_jc_k) \quad \text{with} \quad \vartheta \in [0, 2\pi) \text{ and } j < k \in [2n]$$

and by operators

$$U_j = c_j \quad \text{with} \quad j \in [2n] .$$

The operator $U_{j,k}(\vartheta)$ implements the rotation

$$\begin{aligned}U_{j,k}(\vartheta)c_jU_{j,k}(\vartheta)^\dagger &= \cos(\vartheta)c_j - \sin(\vartheta)c_k \\ U_{j,k}(\vartheta)c_kU_{j,k}(\vartheta)^\dagger &= \sin(\vartheta)c_j + \cos(\vartheta)c_k \\ U_{j,k}(\vartheta)c_\ell U_{j,k}(\vartheta)^\dagger &= c_\ell \quad \text{for} \quad \ell \notin \{j, k\} .\end{aligned} \quad (16)$$

The operator $U_j = c_j$ leaves c_j invariant and flips the sign of each c_k with $k \neq j$, i.e., it implements the reflection

$$\begin{aligned}U_jc_jU_j^\dagger &= c_j \\ U_jc_kU_j^\dagger &= -c_k \quad \text{for} \quad k \neq j .\end{aligned} \quad (17)$$

We note that by relation (14), every generator $U_{j,k}(\vartheta)$ is parity-preserving, whereas every generator U_j reverses the parity, i.e.,

$$\begin{aligned}U_{j,k}(\vartheta)PU_{j,k}(\vartheta)^\dagger &= P \quad \text{for all} \quad k > j \in [n], \vartheta \in [0, 2\pi) , \\ U_jPU_j^\dagger &= -P \quad \text{for all} \quad j \in [n] .\end{aligned} \quad (18)$$

Every orthogonal matrix R gives rise to a Gaussian unitary U_R satisfying (15). The unitary U_R is unique up to a global phase, and $R \mapsto U_R$ is called the metaplectic representation. We can fix the global phase of U_R uniquely, e.g., by the following procedure. Every element $R \in O(2n)$ can be uniquely decomposed into a product

$$R = S_0S_1 \cdots S_L \quad (19)$$

with $L \leq \frac{2n(2n-1)}{2}$ and where

$$S_0 = \begin{cases} I & \text{if } R \in SO(2n) \\ R_1 & \text{otherwise} \end{cases},$$

where for each $r \in [L]$, the matrix S_r is of the form

$$S_r = R_{j_r, k_r}(\vartheta_r) \quad \text{for some} \quad j_r < k_r \in [2n], \vartheta_r \in [0, 2\pi) .$$

Here $R_1 \in O(2n)$ is associated with the unitary U_1 by Eq. (17), whereas $R_{j,k}(\vartheta) \in SO(2n)$ is associated with $U_{j,k}(\vartheta)$ according to Eq. (16). We note that $R_{j,k}(\vartheta) \in SO(2n)$ is a so-called Givens rotation, introduced in Ref. [52], and a decomposition of the form (19) can be found by a deterministic algorithm with runtime $O(n^3)$ (see e.g., Section 5.2.3 in Ref. [53]). In particular, application of this algorithm defines a function taking an arbitrary element $R \in O(2n)$ to a unique product of the form (19). Given the (unique) decomposition (19) of $R \in O(2n)$, we can then define U_R as the product

$$U_R = U_1 U_{j_1, k_1}(\vartheta_1) \cdots U_{j_L, k_L}(\vartheta_L) .$$

Overall, this defines a function $R \mapsto U_R$ from $O(2n)$ to the set of Gaussian unitaries, fixing the phase ambiguity. Throughout the remainder of this work, U_R will denote the Gaussian unitary uniquely fixed by R .

2.3 Fermionic Gaussian (pure) states

The set of pure fermionic Gaussian states is the orbit of the vacuum state $|0_F\rangle$ under the action of $O(2n)$ defined by the metaplectic representation, i.e., fermionic Gaussian states are of the form $U_R |0_F\rangle$ with U_R a fermionic Gaussian unitary. In more detail, every fermionic Gaussian state $e^{i\theta} U_R |0_F\rangle$ is uniquely specified by a pair (θ, R) with $\theta \in [0, 2\pi)$ and $R \in O(2n)$. We will denote the set of all fermionic Gaussian states by

$$\mathcal{G}_n = \left\{ e^{i\theta} U_R |0_F\rangle \mid \theta \in [0, 2\pi), R \in O(2n) \right\} .$$

By Eq. (18) and because $P|0_F\rangle = |0_F\rangle$, every pure fermionic Gaussian state Ψ has a fixed parity, i.e., it is an eigenvector of the parity operator P . This defines a disjoint partition $\mathcal{G}_n = \mathcal{G}_n^+ \cup \mathcal{G}_n^-$ of the set of fermionic Gaussian states into positive- and negative-parity states.

2.4 Gaussianity condition

In Ref. [54] Bravyi established a necessary and sufficient condition to determine if a (possibly mixed) state $\rho \in \mathcal{B}(\mathcal{H}^n)$ is Gaussian (see Theorem 1 therein). Here Gaussianity of a density operator ρ is defined by the condition that ρ has the form

$$\rho = K \exp \left(i \sum_{j,k=1}^{2n} A_{j,k} c_j c_k \right) \quad (20)$$

for an antisymmetric matrix $A = -A^T \in \text{Mat}_{2n \times 2n}(\mathbb{R})$ and a constant $K > 0$. We note that a pure state $\Psi \in \mathcal{H}^n$ is Gaussian if and only if the associated density operator $\rho = |\Psi\rangle\langle\Psi|$ is Gaussian. (This follows from the fact that $|0_F\rangle\langle 0_F| = \frac{1}{2^{n/2}} \exp \left(i \frac{\pi}{4} \sum_{j=1}^n c_{2j-1} c_{2j} \right)$. Indeed, if $\rho = |\Psi\rangle\langle\Psi|$ is a rank-one projection of the form (20), then it follows from

Williamson's normal form for antisymmetric matrices that there is $R \in O(2n)$ such that $RAR^T = \bigoplus_{j=1}^n \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$. This implies that $U_R|\Psi\rangle\langle\Psi|U_R^\dagger = |0_F\rangle\langle 0_F|$ and thus $|\Psi\rangle = U_R^\dagger|0_F\rangle$. Conversely, for a Gaussian state $|\Psi\rangle = e^{i\theta}U_R|0_F\rangle \in \mathcal{G}_n$, we can use the expression for $|0_F\rangle\langle 0_F|$ to argue that $|\Psi\rangle\langle\Psi|$ is of the form (20), i.e., Gaussian.) The characterization of Gaussian density operators established in [54] is the following.

Theorem 2.1 (Theorem 1 in [54]). *Define $\Lambda = \sum_{j=1}^{2n} c_j \otimes c_j$. An even state $\rho \in \mathcal{B}(\mathcal{H}^n)$ is Gaussian if and only if $[\Lambda, \rho \otimes \rho] = 0$.*

Based on this characterization [54], the following was shown in [32].

Lemma 2.2 (Corollary 1 in [32]). *Let $\rho \in \mathcal{B}(\mathcal{H}^n)$ be an even state. Then ρ is a Gaussian pure state if and only if $\Lambda(\rho \otimes \rho) = 0$.*

In the following, we only use the statement of Lemma 2.2 applied to pure states in order to distinguish between Gaussian and non-Gaussian pure states. We formulate this as follows:

Lemma 2.3. *Let $\Psi \in \mathcal{H}^n$ be a pure state with fixed parity. Then Ψ is Gaussian if and only if*

$$\Lambda(|\Psi\rangle \otimes |\Psi\rangle) = 0 .$$

Proof. This follows immediately from the equivalence of the concepts of Gaussianity of pure states (vectors) and density operators because the density operator $|\Psi\rangle\langle\Psi|$ is even for any fixed-parity state Ψ . \square

We note that there is an elegant representation-theoretic interpretation of this characterization of Gaussianity [55]. It is derived from the fact that Gaussian states are the orbit of the vacuum state $|0_F\rangle$ (a highest weight state) under the action of the metaplectic group, cf. [56, Section IV] and [57]. We use a version of this reformulation for 4 fermions, see Lemma 5.1 below, that has first been obtained in [58].

2.5 Covariance matrices, Gaussian states and Wick's theorem

The covariance matrix $\Gamma = \Gamma(\Psi) \in \text{Mat}_{2n \times 2n}(\mathbb{R})$ of a state $\Psi \in \mathcal{H}^n$ is the antisymmetric matrix with entries

$$\Gamma_{j,k}(\Psi) = \begin{cases} \langle\Psi, ic_j c_k \Psi\rangle & \text{for } j \neq k \\ 0 & \text{for } j = k \end{cases} \quad (21)$$

with $j, k \in [2n]$. It satisfies $\Gamma\Gamma^T = I$ for any state $\Psi \in \mathcal{H}^n$.

The expectation value of a Hermitian operator with respect to a Gaussian state Ψ is fully determined by its covariance matrix $\Gamma = \Gamma(\Psi)$. This is because the expectation value of a Majorana monomial $c(\alpha)$, $\alpha \in \{0, 1\}^{2n}$, is given by Wick's theorem

$$\langle\Psi, c(\alpha)\Psi\rangle = \begin{cases} \text{Pf}(\Gamma[\alpha]) & \text{if } |\alpha| \text{ is even} \\ 0 & \text{otherwise} \end{cases} . \quad (22)$$

Here $\Gamma[\alpha] \in \text{Mat}_{|\alpha| \times |\alpha|}(\mathbb{R})$ is the submatrix of Γ which includes all rows and columns with index $j \in [2n]$ such that $\alpha_j = 1$. Evaluating such expectation values, i.e., computing Pfaffians of $|\alpha| \times |\alpha|$ -matrices (with $|\alpha|$ even), takes time $O(|\alpha|^3)$. (Here and below we use the number of elementary arithmetic operations to quantify the time complexity of algorithms.)

2.6 Inner product formulas for Gaussian states

The modulus of the inner product of two Gaussian states Φ_1, Φ_2 with identical parity $\sigma \in \{\pm 1\}$ and covariance matrices Γ_1, Γ_2 is given by the expression [59]

$$|\langle \Phi_1, \Phi_2 \rangle|^2 = \sigma 2^{-n} \text{Pf}(\Gamma_1 + \Gamma_2) . \quad (23)$$

For three Gaussian states Φ_0, Φ_1, Φ_2 , the expression $\langle \Phi_0, \Phi_1 \rangle \cdot \langle \Phi_1, \Phi_2 \rangle \cdot \langle \Phi_2, \Phi_0 \rangle$ is invariant under a change of the global phase of any of the states, and can therefore be computed by the covariance matrix formalism. An explicit expression was derived by Löwdin in [59]. In Ref. [60] Bravyi and Gosset gave the formula

$$\langle \Phi_0, \Phi_1 \rangle \cdot \langle \Phi_1, \Phi_2 \rangle \cdot \langle \Phi_2, \Phi_0 \rangle = \sigma 4^{-n} i^n \text{Pf} \begin{pmatrix} i\Gamma_0 & -I & I \\ I & i\Gamma_1 & -I \\ -I & I & i\Gamma_2 \end{pmatrix} \quad (24)$$

for three Gaussian states $\{\Phi_j\}_{j=0}^2$ of identical parity $\sigma \in \{\pm 1\}$, where $\Gamma_j = \Gamma(\Phi_j)$ is the covariance matrix of Φ_j for $j = 0, 1, 2$. More generally, they obtained the formula

$$\langle \Phi_0, \Phi_1 \rangle \cdot \langle \Phi_1, c(\alpha)\Phi_2 \rangle \cdot \langle \Phi_2, \Phi_0 \rangle = \sigma 4^{-n} i^{n+|\alpha| \cdot (|\alpha|-1)/2} \text{Pf}(R_\alpha) , \quad (25)$$

for any even-weight Majorana monomial $c(\alpha)$, $\alpha \in \{0, 1\}_+^{2n}$, where

$$R_\alpha = \begin{pmatrix} i\Gamma_0 & -I & I & 0 \\ I & i\Gamma_1 & -I & 0 \\ -I & I & iD_\alpha \Gamma_2 D_\alpha & J_\alpha^T + iD_\alpha \Gamma_2 J_\alpha^T \\ 0 & 0 & -J_\alpha + iJ_\alpha \Gamma_2 D_\alpha & iJ_\alpha \Gamma_2 J_\alpha^T \end{pmatrix} \in \text{Mat}_{(6n+|\alpha|) \times (6n+|\alpha|)}(\mathbb{R}) \quad (26)$$

Here $D_\alpha = \text{diag}(\{1 - \alpha_j\}_{j=1}^{2n})$ is a diagonal matrix, whereas $J_\alpha \in \text{Mat}_{|\alpha| \times 2n}(\mathbb{R})$ has entries defined in terms of the indices $\{i \in [2n] \mid \alpha_i \neq 0\} = \{i_1 < \dots < i_r\}$ associated with non-zero entries of α , that is,

$$(J_\alpha)_{j,k} = \begin{cases} \delta_{i_j, k} & \text{if } j \leq r \\ 0 & \text{otherwise} \end{cases} .$$

In other words, $(J_\alpha)_{j,k} = 1$ if and only if k is the position of the j -th nonzero element of α .

As argued in [60], expressions (24) and (25) determine the inner product $\langle \Phi_1, \Phi_2 \rangle$ and an expression of the form $\langle \Phi_1, c(\alpha)\Phi_2 \rangle$ entirely in terms of covariance matrices, assuming that the remaining two overlaps $\langle \Phi_0, \Phi_1 \rangle, \langle \Phi_2, \Phi_0 \rangle$ with a Gaussian reference state Φ_0 are given and non-zero. In this situation, these quantities can be computed in time $O(n^3)$.

2.7 Gaussian evolution and occupation number measurement

Underlying the known classical simulation algorithms for fermionic linear optics is the fact that Gaussian unitaries and occupation number measurements preserve Gaussianity. Explicitly, this can be described as follows: Given a Gaussian state Ψ with covariance matrix $\Gamma(\Psi)$

- (i) the covariance matrix $\Gamma(U_R \Psi)$ of Ψ evolved under the Gaussian unitary U_R , $R \in O(2n)$, is given by $\Gamma(U_R \Psi) = R\Gamma(\Psi)R^T$.

- (ii) measurement of the observable $a_j^\dagger a_j$ for $j \in [n]$ gives the outcome $s \in \{0, 1\}$ with probability

$$\begin{aligned}
P_j(s) &= \|\Pi_j(s)\Psi\|^2 \\
&= \langle \Psi, \Pi_j(s)\Psi \rangle \\
&= \frac{1}{2} (1 + (-1)^s \langle \Psi, i c_{2j-1} c_{2j} \Psi \rangle) \\
&= \frac{1}{2} (1 + (-1)^s \Gamma_{2j-1, 2j}) .
\end{aligned} \tag{27}$$

where the third identity follows from the definition

$$\Pi_j(s) = \frac{1}{2} (1 + (-1)^s i c_{2j-1} c_{2j})$$

and the last identity follows from the definition of the covariance matrix in Eq. (21). Given that the measurement outcome is $s \in \{0, 1\}$, the post-measurement state

$$\Psi(s) = (\Pi_j(s) |\Psi\rangle) / \sqrt{P_j(s)}$$

is Gaussian with covariance matrix $\Gamma(\Psi(s))$ defined by the lower-diagonal entries (see e.g., [3, Proof of Lemma 4])

$$\Gamma(\Psi(s))_{k,\ell} = \begin{cases} (-1)^s & \text{if } (k, \ell) = (2j, 2j-1) \\ \Gamma_{k,\ell} + \frac{(-1)^s}{2P_j(s)} (\Gamma_{2j-1,\ell} \Gamma_{2j,k} - \Gamma_{2j-1,k} \Gamma_{2j,\ell}) & \text{otherwise} \end{cases} \tag{28}$$

for $k > \ell$.

In particular, the corresponding resulting covariance matrices can be computed in time $O(n^3)$ [8–10, 54] and $O(n^2)$ [3] for unitary evolution and measurement, respectively.

2.8 The tensor product of two fermionic states

Two density operators $\rho_j \in \mathcal{B}(\mathcal{H}^{n_j})$, $j \in [2]$, have a joint extension if and only if there is an element $\rho \in \mathcal{B}(\mathcal{H}^{n_1+n_2})$ such that

$$\text{tr}(c(\alpha_1 || \alpha_2) \rho) = \text{tr}(c(\alpha_1) \rho_1) \text{tr}(c(\alpha_2) \rho_2) \quad \text{for all} \quad \alpha_j \in \{0, 1\}^{2n_j}, j \in [2] . \tag{29}$$

Here $\alpha_1 || \alpha_2 \in \{0, 1\}^{2(n_1+n_2)}$ denotes the concatenation of α_1 and α_2 . Theorem 1 in [61] implies that if either ρ_1 or ρ_2 is even, then a unique joint extension ρ of (ρ_1, ρ_2) exists. Furthermore, this extension is even if and only if both ρ_1 and ρ_2 are even. Theorem 2 in [61] shows that if ρ is even and ρ_1 and ρ_2 are pure, then ρ is also pure.

In particular, this means that for states Ψ_1, Ψ_2 of definite parity, there is a unique joint pure extension $\rho = |\Psi\rangle\langle\Psi|$ of $(|\Psi_1\rangle\langle\Psi_1|, |\Psi_2\rangle\langle\Psi_2|)$. Since ρ is pure, this also means that Ψ is of definite parity. We will write $\Psi = \Psi_1 \tilde{\otimes} \Psi_2$ for this state in the following, and we call $\tilde{\otimes}$ the fermionic tensor product. Note that Ψ is only defined up to a global phase. It follows immediately from these definitions that

$$|\langle x, y | \Psi_1 \tilde{\otimes} \Psi_2 \rangle| = |\langle x, \Psi_1 \rangle \cdot \langle y, \Psi_2 \rangle| \quad \text{for all} \quad x \in \{0, 1\}^{n_1} \text{ and } y \in \{0, 1\}^{n_2} . \tag{30}$$

Proof. Let $x \in \{0, 1\}^{n_1}$ and $y \in \{0, 1\}^{n_2}$ be arbitrary. By definition, we have

$$\begin{aligned} |x, y\rangle\langle x, y| &= \left(\prod_{j=1}^{n_1} \frac{1}{2} (I + (-1)^{x_j} i c_{2j-1} c_{2j}) \right) \left(\prod_{k=1}^{n_2} \frac{1}{2} (I + (-1)^{y_k} i c_{2n_1+2k-1} c_{2n_1+2k}) \right) \\ &= \left(\sum_{\alpha \in \{0,1\}_+^{2n_1}} \gamma_x(\alpha) c(\alpha|0^{2n_2}) \right) \left(\sum_{\beta \in \{0,1\}_+^{2n_2}} \gamma_y(\beta) c(0^{2n_1}|\beta) \right). \end{aligned}$$

for certain coefficients $\{\gamma_x(\alpha)\}_{\alpha \in \{0,1\}_+^{2n_1}}$ and $\{\gamma_y(\beta)\}_{\beta \in \{0,1\}_+^{2n_2}}$. Since $\rho = |\Psi_1 \tilde{\otimes} \Psi_2\rangle\langle \Psi_1 \tilde{\otimes} \Psi_2|$ is an extension of $(|\Psi_1\rangle\langle \Psi_1|, |\Psi_2\rangle\langle \Psi_2|)$ and $c(\alpha|0^{2n_2})c(0^{2n_1}|\beta) = i^{-|\alpha|\cdot|\beta|} c(\alpha|\beta) = c(\alpha|\beta)$ for (even-weight) $\alpha \in \{0, 1\}_+^{2n_1}$ and $\beta \in \{0, 1\}_+^{2n_2}$, it follows that

$$\begin{aligned} |\langle x, y | \Psi_1 \tilde{\otimes} \Psi_2 \rangle|^2 &= \sum_{\alpha \in \{0,1\}_+^{2n_1}} \gamma_x(\alpha) \sum_{\beta \in \{0,1\}_+^{2n_2}} \gamma_y(\beta) \text{tr}(c(\alpha|\beta)\rho) \\ &= \langle \Psi_1, \left(\sum_{\alpha \in \{0,1\}_+^{2n_1}} \gamma_x(\alpha) c(\alpha) \right) \Psi_1 \rangle \cdot \langle \Psi_2, \left(\sum_{\beta \in \{0,1\}_+^{2n_2}} \gamma_y(\beta) c(\beta) \right) \Psi_2 \rangle \\ &= |\langle x, \Psi_1 \rangle|^2 \cdot |\langle y, \Psi_2 \rangle|^2. \end{aligned}$$

□

Refining Eq. (30), (relative) phase information between these matrix elements can be obtained from the explicit construction of $\Psi_1 \tilde{\otimes} \Psi_2$ given in [61, Section 3.1] (see also [51, Proof of Theorem 1]): Consider the isometry

$$\begin{aligned} U : \quad \mathcal{H}^{n_1+n_2} &\rightarrow \mathcal{H}^{n_1} \otimes \mathcal{H}^{n_2} \\ |x_1, \dots, x_{n_1+n_2}\rangle &\mapsto |x_1, \dots, x_{n_1}\rangle \otimes |x_{n_1+1}, \dots, x_{n_1+n_2}\rangle \end{aligned}$$

whose action is given by

$$U a_j U^\dagger = \begin{cases} a_j \otimes I & \text{if } j \in [n_1] \\ P_1 \otimes a_{j-n_1} & \text{if } j \in \{n_1+1, \dots, n_1+n_2\}. \end{cases}$$

where P_1 , the parity operator acting on \mathcal{H}_{n_1} , introduces phases. Then

$$\Psi_1 \tilde{\otimes} \Psi_2 = U^\dagger (\Psi_1 \otimes \Psi_2).$$

It is straightforward from this definition to check that $\Psi_1 \tilde{\otimes} \Psi_2$ is the extension of (Ψ_1, Ψ_2) and

$$\langle x, y | \Psi_1 \tilde{\otimes} \Psi_2 \rangle = (-1)^{|x|} \cdot \langle x, \Psi_1 \rangle \cdot \langle y, \Psi_2 \rangle \quad \text{for all } x \in \{0, 1\}^{n_1} \text{ and } y \in \{0, 1\}^{n_2} \quad (31)$$

We note that the fermionic tensor product preserves Gaussianity in the following sense.

Lemma 2.4. *Let $\Psi_j \in \mathcal{G}_{n_j}^+$ be positive-parity fermionic Gaussian states for $j \in [2]$. Then $\Psi_1 \tilde{\otimes} \Psi_2 \in \mathcal{G}_{n_1+n_2}^+$, i.e., it is an even fermionic Gaussian state.*

Proof. By definition of an extension (see Eq. (29)) and Wick's theorem (Eq. (22)), the tensor product $\Psi = \Psi_1 \tilde{\otimes} \Psi_2$ satisfies

$$\langle \Psi, c(\alpha_1|\alpha_2)\Psi \rangle = \begin{cases} \text{Pf}(\Gamma_1[\alpha_1])\text{Pf}(\Gamma_1[\alpha_1]) & \text{if both } |\alpha_1| \text{ and } |\alpha_2| \text{ are even} \\ 0 & \text{otherwise} \end{cases} \quad (32)$$

for all $\alpha_j \in \{0, 1\}^{2n_j}$, where Γ_j is the covariance matrix of Ψ_j for $j \in [2]$. Because the Pfaffian satisfies

$$\text{Pf}(A_1 \oplus A_2) = \text{Pf} \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} = \text{Pf}(A_1)\text{Pf}(A_2)$$

for block-matrices, it follows from (32) that the tensor product Ψ satisfies Wick's theorem (22) with covariance matrix $\Gamma_1 \oplus \Gamma_2$. In particular, it is Gaussian. \square

3 Tracking relative phases in fermionic linear optics

The covariance matrix $\Gamma(\Psi)$ of a fermionic Gaussian state $|\Psi\rangle = e^{i\theta}U_R|0_F\rangle \in \mathcal{G}_n$ fully determines expectation values by Wick's theorem, which is why Gaussian states and dynamics are amenable to efficient classical simulation (see Section 2.7). However, the description of Ψ in terms of the covariance matrix $\Gamma(\Psi)$ does not capture information on the global phase $e^{i\theta}$ of the state. For processes involving non-Gaussian states expressed as superpositions of Gaussian states, such phase information needs to be available for computing norms, expectation values and overlaps.

Here we provide an extended (classical) description of fermionic Gaussian states that incorporates phase information. A central feature of our construction is the fact that this description permits to compute overlaps (including relative phases, i.e., not only the absolute value) of Gaussian states in an efficient manner.

Our construction is motivated by and relies on expression (24), which relates the inner product $\langle\Psi_1, \Psi_2\rangle$ of two Gaussian states $\Psi_1, \Psi_2 \in \mathcal{G}_n$ to their inner products $\langle\Psi_0, \Psi_1\rangle, \langle\Psi_0, \Psi_2\rangle$ with a Gaussian reference state $\Psi_0 \in \mathcal{G}_n$ and their covariance matrices $\Gamma_0, \Gamma_1, \Gamma_2$. This suggests fixing a reference state $\Psi_0 \in \mathcal{G}_n$ and using the pair $(\Gamma(\Psi), \langle\Psi_0, \Psi\rangle)$ as a classical description of any state $|\Psi\rangle \in \mathcal{G}_n$ relevant in the computation. The problem with this idea is that $\langle\Psi_0, \Psi\rangle$ may vanish, preventing the application of (24). To avoid this problem, we use – instead of a single state Ψ_0 – a (potentially) different reference state for each state Ψ . Specifically, we will show that using number states, i.e., states of the form (10), suffices. This motivates the following definition.

Definition 3.1. *Let $|\Psi\rangle = e^{i\theta}U_R|0_F\rangle \in \mathcal{G}_n$ be a Gaussian state. We call a tuple*

$$d = (\Gamma, x, r) \in \text{Mat}_{2n \times 2n}(\mathbb{R}) \times \{0, 1\}^n \times \mathbb{C}$$

a (valid) description of $|\Psi\rangle$ if the following three conditions hold:

- (i) $\Gamma = \Gamma(\Psi)$ is the covariance matrix of $|\Psi\rangle$.
- (ii) $x \in \{0, 1\}^n$ is such that $\langle x, \Psi\rangle \neq 0$, where $|x\rangle$ is the number state defined by Eq. (10).

In our algorithms we will in fact ensure that $|\langle x, \Psi\rangle|^2 \geq 2^{-n}$, i.e., only a subset of valid descriptions is used. A description $d = (\Gamma, x, r)$ with this property, i.e., satisfying $|r|^2 \geq 2^{-n}$, will be called a good description. The restriction to good descriptions is necessary to make our algorithms work with finite-precision arithmetic.

- (iii) $r = \langle x, \Psi\rangle$.

More explicitly, necessary and sufficient conditions for $d = (\Gamma(\Psi), x, r)$ to constitute a description of Ψ are that

$$r \neq 0 \quad \text{and} \quad |r|^4 = 2^{-2n} \text{Det}(\Gamma(|x\rangle) + \Gamma(\Psi))$$

because of formula (23) for the overlap of two states and because $\text{Det}(\cdot) = \text{Pf}^2(\cdot)$. Here

$$\Gamma(|x\rangle) = \bigoplus_{j=1}^n \begin{pmatrix} 0 & (-1)^{x_j} \\ -(-1)^{x_j} & 0 \end{pmatrix} \quad (33)$$

is the covariance matrix of $|x\rangle$. Since a Gaussian state Ψ generally has non-zero overlap with more than a single occupation number state $|x\rangle$, there are several distinct valid descriptions of Ψ . We will denote the set of descriptions of $|\Psi\rangle \in \mathcal{G}_n$ by $\text{Desc}(\Psi)$.

We note that a description $d = (\Gamma, x, r)$ uniquely fixes a Gaussian state $\Psi \in \mathcal{G}_n$: The covariance matrix Γ determines all expectation values, and the global phase of Ψ is fixed by the overlap $\langle x, \Psi \rangle$, i.e., by r . Denoting by $\text{Desc}_n = \bigcup_{\Psi \in \mathcal{G}_n} \text{Desc}(\Psi)$ the set of all descriptions of fermionic Gaussian n -mode states, this means that we have a function

$$\begin{aligned} \Psi & : \text{Desc}_n & \rightarrow & \mathcal{G}_n \\ d & \mapsto & \Psi(d) . \end{aligned} \quad (34)$$

The main result of this section shows that expectation values, overlaps, and matrix elements of (Majorana) operators with respect to Gaussian states can be efficiently computed from their classical descriptions. Furthermore, when evolving a Gaussian state under a Gaussian unitary, the description of the resulting state can be computed efficiently. The same is true for the post-measurement state when applying an occupation number measurement.

For evolution, we note that it suffices to consider Gaussian unitaries of the form U_R where $R \in O(2n)$ belongs to the set of generators $\text{Gen}(O(2n))$ introduced in Section 2.2, that is,

$$\text{Gen}(O(2n)) = \{R_{j,k}(\vartheta) \mid j < k \in [2n], \vartheta \in [0, 2\pi)\} \cup \{R_j\}_{j=1}^{2n} .$$

Here $R_{j,k}(\vartheta)$ is a Givens rotation and $R_j = -\text{diag}(\{(-1)^{\delta_{j,k}}\}_{k=1}^{2n})$ a reflection. We note that each element of $\text{Gen}(O(2n))$ can be specified by a tuple $(j, k, \vartheta) \in [2n] \times [2n] \times [0, 2\pi)$ or an index $j \in [2n]$, respectively. We assume that this parameterization is used in the following algorithms (but leave this implicit).

To state the properties of our (deterministic) algorithms, it is convenient to express these as functions.

Theorem 3.2 (Overlap, evolution, and measurement). *Let $\Psi(d) \in \mathcal{G}_n$ be the Gaussian state associated with a description $d \in \text{Desc}_n$, see Eq. (34). Then the following holds:*

- (i) *The algorithm $\text{overlap} : \text{Desc}_n \times \text{Desc}_n \rightarrow \mathbb{C}$ given in Fig. 7 has runtime $O(n^3)$ and satisfies*

$$\text{overlap}(d_1, d_2) = \langle \Psi(d_1), \Psi(d_2) \rangle \quad \text{for all} \quad d_1, d_2 \in \text{Desc}_n .$$

- (ii) *The algorithm $\text{evolve} : \text{Desc}_n \times \text{Gen}(O(2n)) \rightarrow \text{Desc}_n$ given in Fig. 9 has runtime $O(n^3)$ and satisfies*

$$\Psi(\text{evolve}(d, R)) = U_R \Psi(d) \quad \text{for all} \quad d \in \text{Desc}_n \text{ and } R \in \text{Gen}(O(2n)) ,$$

where U_R denotes the Gaussian unitary associated with $R \in O(2n)$.

(iii) The algorithm `measureprob` : $\text{Desc}_n \times [n] \times \{0, 1\} \rightarrow \mathbb{R}$ given in Fig. 11 has runtime $O(1)$ and satisfies

$$\text{measureprob}(d, j, s) = \|\Pi_j(s)\Psi(d)\|^2 \quad \text{for all } d \in \text{Desc}_n, j \in [n], s \in \{0, 1\},$$

where $\Pi_j(s) = \frac{1}{2}(I + (-1)^s c_{2j-1} c_{2j})$ is the projection onto the eigenspace of $a_j^\dagger a_j$ with eigenvalue s .

(iv) The algorithm `postmeasure` : $\text{Desc}_n \times [n] \times \{0, 1\} \times [0, 1] \rightarrow \text{Desc}_n$ given in Fig. 12 has runtime $O(n^3)$. The algorithm satisfies

$$\Psi(\text{postmeasure}(d, j, s, p(d, j, s))) = \frac{\Pi_j(s)\Psi(d)}{\|\Pi_j(s)\Psi(d)\|}$$

for all $d \in \text{Desc}_n, j \in [n], s \in \{0, 1\}$, with $p(d, j, s) = \|\Pi_j(s)\Psi(d)\|^2$.

The output of both `evolve` and `postmeasure` is a good description for any input.

We argue that descriptions of relevant initial states can be obtained efficiently. Clearly, this is the case for any state of the form $|\Psi\rangle = U_{R_L} \cdots U_{R_1} |0_F\rangle$ obtained by applying a sequence $\{R_j\}_{j \in [L]} \subset \text{Gen}(O(2n))$ of generators to the vacuum state $|0_F\rangle$: Here we can use the algorithm `evolve` L times, producing a description of $|\Psi\rangle$ in time $O(Ln^3)$.

We will at times need a description of a state $|\Psi\rangle$ but do not require fixing its global phase. This is the case for example when subsequent computational states only involve phase-insensitive expressions, e.g., terms of the form $|\langle \Psi, \Phi \rangle|^2$. Such a description can be found efficiently from the covariance matrix Γ of $|\Psi\rangle$. Since the phase can be fixed arbitrarily, the problem here is to find $x \in \{0, 1\}^n$ such that $\langle x, \Psi \rangle$ is non-zero.

Theorem 3.3. *There is an algorithm `describe` : $\text{Mat}_{2n \times 2n}(\mathbb{R}) \rightarrow \text{Desc}_n$ with runtime $O(n^3)$ such that for any covariance matrix Γ , the state $\Psi(\text{describe}(\Gamma))$ is a Gaussian state with covariance matrix Γ , and `describe`(Γ) is a good description.*

For example, consider states of the form $|\Phi(\pi, y)\rangle = U_{R_\pi} |y\rangle$, where $R_\pi \in O(2n)$ is a permutation matrix specified by an element $\pi \in S_{2n}$ and $y \in \{0, 1\}^n$. (Such states are used in Ref. [60] to give a fast norm estimation algorithm, see Section 4.3.) The covariance matrix of this state is $\Gamma(\pi, y) = R_\pi \Gamma(|y\rangle) R_\pi^T$ (with $\Gamma(|y\rangle)$ defined by Eq. (33)). We thus conclude that $|\Psi(\text{describe}(\Gamma(\pi, y)))\rangle$ is proportional to $|\Phi(\pi, y)\rangle$ with a global phase $e^{i\theta}$ possibly depending on the pair (π, y) .

The remainder of this section is devoted to the proofs of Theorem 3.2 and Theorem 3.3: We describe the algorithms `evolve`, `overlap`, `measureprob`, `postmeasure` and `describe` in detail, providing pseudocode, and verify that these satisfy the desired properties.

3.1 Subroutines

Our algorithms make use of subroutines called `findsupport`, `relatebasiselements`, `overlaptriple` and `convert` which we describe here.

```

Require:  $\Gamma \in \text{Mat}_{2n \times 2n}(\mathbb{R})$  covariance matrix of a pure Gaussian state
1: function findsupport( $\Gamma$ )
2:    $\Gamma^{(0)} \leftarrow \Gamma$ 
3:    $x \leftarrow 0^n \in \{0, 1\}^n$ .
4:   for  $j \leftarrow 1$  to  $n$  do                                      $\triangleright$  simulate a measurement of  $a_j^\dagger a_j$ 
5:      $q_j \leftarrow \frac{1}{2}(I + \Gamma_{2j-1, 2j}^{(j-1)})$ 
6:     if  $q_j \geq 1/2$  then                                        $\triangleright$  choose the higher-probability outcome
7:        $x_j \leftarrow 0$ 
8:        $p_j \leftarrow q_j$ 
9:     else
10:       $x_j \leftarrow 1$ 
11:       $p_j \leftarrow 1 - q_j$ 
12:       $\Gamma^{(j)} \leftarrow 0 \in \text{Mat}_{2n \times 2n}(\mathbb{R})$            $\triangleright$  covariance matrix of the post-measurement
state
13:       $\Gamma_{2j, 2j-1}^{(j)} \leftarrow (-1)^{x_j}$ 
14:      for  $\ell \leftarrow 1$  to  $n - 1$  do
15:        for  $k \leftarrow \ell + 1$  to  $n$  do
16:          if  $(k, \ell) \neq (2j, 2j - 1)$  then
17:             $\Gamma_{k, \ell}^{(j)} \leftarrow \Gamma_{k, \ell}^{(j-1)} + \frac{(-1)^{x_j}}{2p_j} (\Gamma_{2j-1, \ell}^{(j-1)} \Gamma_{2j, k}^{(j-1)} - \Gamma_{2j-1, k}^{(j-1)} \Gamma_{2j, \ell}^{(j-1)})$ 
18:       $\Gamma^{(j)} \leftarrow \Gamma^{(j)} - (\Gamma^{(j)})^T$ 
19:   return  $x$ 

```

Figure 1: The algorithm findsupport: Given the covariance matrix Γ of a Gaussian state $|\Psi\rangle$, it computes $x \in \{0, 1\}^n$ such that $\langle x, \Psi \rangle \neq 0$.

The subroutine findsupport takes as input the covariance matrix Γ of a Gaussian state Ψ and produces a string $x \in \{0, 1\}^{2n}$ with the property that $\langle x, \Psi \rangle \neq 0$. It is given in Fig. 1. It has the following properties:

Lemma 3.4. *The algorithm findsupport : $\text{Mat}_{2n \times 2n}(\mathbb{R}) \rightarrow \{0, 1\}^n$ runs in time $O(n^3)$. It satisfies*

$$|\langle \text{findsupport}(\Gamma(\Psi)), \Psi \rangle|^2 \geq 2^{-n} \quad \text{for every} \quad \Psi \in \mathcal{G}_n, \quad (35)$$

where $\Gamma(\Psi)$ is the covariance matrix of Ψ .

Proof. The main idea of the algorithm is to mimic a measurement in the number state basis executed in a sequential manner. Consider the following process: Suppose we start with the state $\Psi^{(0)} = \Psi$, and then measure $a_j^\dagger a_j$ successively for $j = 1, \dots, n$. Let $P(x_j | x_1 \cdots x_{j-1})$ denote the conditional probability of observing the outcome $x_j \in \{0, 1\}$ (when measuring $a_j^\dagger a_j$), given that the previous measurements yielded (x_1, \dots, x_{j-1}) . According to Born's rule, this is given by

$$P(x_j | x_1 \cdots x_{j-1}) = \langle \Psi_{x_1 \cdots x_{j-1}}^{(j-1)}, \Pi_j(x_j) \Psi_{x_1 \cdots x_{j-1}}^{(j-1)} \rangle$$

where $\Psi_{x_1 \cdots x_{j-1}}^{(j-1)}$ is the post-measurement state after the first $(j - 1)$ measurements. The probability of observing the sequence $x \in \{0, 1\}^n$ of outcomes then is

$$|\langle x, \Psi \rangle|^2 = \prod_{j=1}^n P(x_j | x_1 \cdots x_{j-1}) \quad (36)$$

by Bayes' rule.

The algorithm `findsupport` simulates this process: For each $j \in [n]$, the quantity q_j computed in line 5 is equal to the conditional probability $P(0|x_1 \cdots x_{j-1})$ that the j -th measurement results in the outcome 0. Lines 6–11 ensure that the outcome $x_j \in \{0, 1\}$ with higher probability of occurrence is selected at each step, guaranteeing Property (35) (because of Eq. (36)). The matrix $\Gamma^{(j)}$ computed in steps 12–18 is the covariance matrix of the post-measurement state $\Psi_{x_1 \cdots x_j}^{(j)}$.

Each measurement is thus realized in time $O(n^2)$ yielding the overall complexity of $O(n^3)$. \square

The algorithm `relatebasiselements` is more straightforward: Given $x, y \in \{0, 1\}^n$, it outputs $(\alpha, \vartheta) \in \{0, 1\}^{2n} \times \mathbb{R}$ such that $c(\alpha) |x\rangle = e^{i\vartheta} |y\rangle$. That is, it finds a Majorana monomial $c(\alpha)$ which maps the basis state $|x\rangle$ to $|y\rangle$ up to a phase and computes the corresponding phase. In Fig. 2 we give pseudocode for this algorithm.

Require: $x \in \{0, 1\}^n, y \in \{0, 1\}^n$

- 1: **function** `relatebasiselements`(x, y)
- 2: $\alpha \leftarrow 0^{2n} \in \{0, 1\}^{2n}$.
- 3: **for** $j \leftarrow 1$ to n **do**
- 4: $\alpha_{2j-1} \leftarrow x_j \oplus y_j$
- 5: $\vartheta \leftarrow \frac{\pi}{4} |x \oplus y| + \pi \sum_{j=1}^n (x \oplus y)_j \eta_j(x)$
- 6: **return** (α, ϑ) $\triangleright (\alpha, \vartheta)$ is such that $c(\alpha) |x\rangle = e^{i\vartheta} |y\rangle$

Figure 2: Given $x, y \in \{0, 1\}^n$, the algorithm `relatebasiselements` computes $\alpha \in \{0, 1\}^{2n}$ and $\vartheta \in \mathbb{R}$ such that $c(\alpha) |x\rangle = e^{i\vartheta} |y\rangle$. The expression $\eta_j(x)$ in line 5 is defined by Eq. (12), and $x \oplus y \in \{0, 1\}^n$ denotes the bitwise addition modulo two for $x, y \in \{0, 1\}^n$.

Lemma 3.5. *The algorithm `relatebasiselements` : $\{0, 1\}^n \rightarrow \{0, 1\}^{2n} \times \mathbb{C}$ runs in time $O(n)$ and satisfies*

$$c(\alpha) |x\rangle = e^{i\vartheta} |y\rangle \quad \text{where } (\alpha, \vartheta) = \text{relatebasiselements}(x, y) \quad \text{for all } x, y \in \{0, 1\}^n .$$

Proof. Let $x, y \in \{0, 1\}^n$ be arbitrary. Define

$$\alpha_{2j-1} = x_j \oplus y_j \quad \text{and} \quad \alpha_{2j} = 0 \quad \text{for } j \in [n] ,$$

as in line 4 of algorithm `relatebasiselements`. Then

$$\begin{aligned} c(\alpha) |y\rangle &= i^{|\alpha| \cdot (|\alpha|-1)/2} c_1^{x_1 \oplus y_1} c_3^{x_2 \oplus y_2} \cdots c_{2n-1}^{x_n \oplus y_n} |y\rangle \\ &= i^{\left(\sum_{j=1}^n x_j \oplus y_j\right) \cdot \left(\left(\sum_{j=1}^n x_j \oplus y_j\right) - 1\right)/2} (-1)^{\sum_{j=1}^n (x_j \oplus y_j) \eta_j(x)} |y \oplus (x \oplus y)\rangle \\ &= i^{|x \oplus y| \cdot (|x \oplus y| - 1)/2} (-1)^{\sum_{j=1}^n (x \oplus y)_j \eta_j(x)} |x\rangle \end{aligned}$$

where in the second identity, we used that

$$c_{2j-1} |x\rangle = (-1)^{\eta_j(x)} |x \oplus e_j\rangle \quad \text{for all } x \in \{0, 1\}^n \text{ and } j \in [n]$$

because of Eq. (11). Because $i^{|x \oplus y| \cdot (|x \oplus y| - 1)/2} (-1)^{\sum_{j=1}^n (x \oplus y)_j \eta_j(x)} = e^{i\vartheta}$ for

$$\vartheta = \frac{\pi}{4} |x \oplus y| \cdot (|x \oplus y| - 1) + \pi \sum_{j=1}^n (x \oplus y)_j \eta_j(x) ,$$

comparison with line 5 of the algorithm `relatebasiselements` gives the claim. \square

The algorithm `overlaptriple` takes covariance matrices $\Gamma_0, \Gamma_1, \Gamma_2$ of three Gaussian states Φ_0, Φ_1, Φ_2 of the same parity $\sigma \in \{-1, 1\}$ and $\alpha \in \{0, 1\}_+^n$, as well as overlaps $u = \langle \Phi_0, \Phi_1 \rangle$, $v = \langle \Phi_1, c(\alpha)\Phi_2 \rangle$ (which both have to be non-zero), and computes the overlap $\langle \Phi_2, \Phi_0 \rangle$. It is obtained by direct application of the formula (25). For completeness, we include pseudocode in Fig. 3. Since this algorithm involves computing Pfaffians of matrices that have size linear in n , its runtime is $O(n^3)$.

Require: Γ_j covariance matrix of a Gaussian state Φ_j for $j = 0, 1, 2$
Require: $\sigma = \text{Pf}(\Gamma_0) = \text{Pf}(\Gamma_1) = \text{Pf}(\Gamma_2)$
Require: $\alpha \in \{0, 1\}_+^n$ such that $\langle \Phi_1, c(\alpha)\Phi_2 \rangle \neq 0$
Require: $\langle \Phi_0, \Phi_1 \rangle \neq 0$
Require: $u = \langle \Phi_0, \Phi_1 \rangle$ and $v = \langle \Phi_1, c(\alpha)\Phi_2 \rangle$
1: **function** `overlaptriple`($\Gamma_0, \Gamma_1, \Gamma_2, \alpha, u, v$)
2: $R_\alpha \in \text{Mat}_{(6n+|\alpha|) \times (6n+|\alpha|)}(\mathbb{R})$
3: $R_\alpha \leftarrow$ evaluate Eq. (26)
4: $o \leftarrow u^{-1} \cdot v^{-1} \cdot \sigma \cdot 4^{-n} i^{n+|\alpha| \cdot (|\alpha|-1)/2} \text{Pf}(R_\alpha)$ \triangleright compute the overlap $\langle \Phi_2, \Phi_0 \rangle$
5: **return** o

Figure 3: The algorithm `overlaptriple` takes as input the covariance matrices Γ_j of three Gaussian states Φ_j , $j = 0, 1, 2$ with identical parity, $\alpha \in \{0, 1\}_+^n$ and the overlaps $\langle \Phi_0, \Phi_1 \rangle$, $\langle \Phi_1, c(\alpha)\Phi_2 \rangle$. The latter have to be non-zero. The algorithm computes the overlap $\langle \Phi_2, \Phi_0 \rangle$ using Eq. (25).

Fig. 4 gives a graphical representation of what the algorithm `overlaptriple` achieves. These graphical representations will be helpful to construct and analyze other algorithmic building blocks.



(a) The input to the algorithm `overlaptriple` consists of (descriptions of) three Gaussian states Φ_0, Φ_1, Φ_2 and $\alpha \in \{0, 1\}_+^{2n}$, together with overlaps $u = \langle \Phi_0, \Phi_1 \rangle$ and $v = \langle \Phi_1, c(\alpha)\Phi_2 \rangle$ that are both non-zero.

(b) Applying `overlaptriple` provides the inner product $\langle \Phi_0, \Phi_2 \rangle$. In this diagrammatic representation, this completes the triangle with vertices Φ_0, Φ_1, Φ_2 .

Figure 4: A graphical representation of the functionality provided by the algorithm `overlaptriple`. Solid lines represent inner products that are given / have been computed, and are non-zero. Inner products of the form $\langle \Phi_1, c(\alpha)\Phi_2 \rangle$ are represented by arrows.

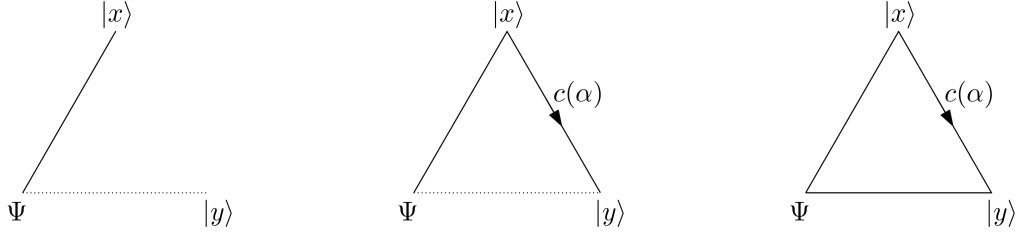
The algorithm `convert` takes a description $d = (\Gamma, x, r)$ of a Gaussian state $\Psi(d)$ and $y \in \{0, 1\}^n$ such that $\langle y, \Psi(d) \rangle \neq 0$, and outputs a description $d' = (\Gamma, y, s)$ of the same state. In other words, it converts a description d of the state involving the reference state $|x\rangle$ to a description d' of the same state but involving a different reference state $|y\rangle$. In Fig. 5 we give pseudocode for this algorithm.

Require: $d = (\Gamma, x, r) \in \text{Desc}_n$, $y \in \{0, 1\}^n$ such that $\langle y, \Psi(d) \rangle \neq 0$

- 1: **function** `convert`(d, y)
- 2: $(\alpha, \vartheta) \leftarrow \text{relatebasiements}(y, x)$ \triangleright find (α, ϑ) such that $c(\alpha) |y\rangle = e^{i\vartheta} |x\rangle$
- 3: $\Gamma_0 \leftarrow \Gamma$, $\Gamma_1 \leftarrow \Gamma(|x\rangle)$, $\Gamma_2 \leftarrow \Gamma(|y\rangle)$ \triangleright covariance matrices of $\Psi(d)$, $|x\rangle$ and $|y\rangle$
- 4: $u \leftarrow \bar{r}$ $\triangleright u = \langle \Psi(d), x \rangle$
- 5: $v \leftarrow e^{i\vartheta}$ $\triangleright v = \langle x, c(\alpha)y \rangle$
- 6: $w \leftarrow \text{overlaptriple}(\Gamma_0, \Gamma_1, \Gamma_2, \alpha, u, v)$ \triangleright compute the overlap $\langle y, \Psi(d) \rangle$
- 7: **return** (Γ, y, w)

Figure 5: The algorithm `convert` takes a description $d \in \text{Desc}_n$ and $y \in \{0, 1\}^n$ such that $\langle y, \Psi(d) \rangle \neq 0$. It outputs a description $d' \in \text{Desc}_n$ of $\Psi(d)$ such that the second entry of d' is equal to y , i.e., $d' = (\Gamma, y, s)$. It makes use of the subroutines `relatebasiements` and `overlaptriple`. For $x \in \{0, 1\}^n$, $\Gamma(|x\rangle)$ denotes the covariance matrix of the state $|x\rangle$, see Eq. (33).

The algorithm `convert` is illustrated in Fig. 6.



(a) The input to the algorithm `convert` specifies a Gaussian state Ψ , $x \in \{0, 1\}^n$ such that $\langle x, \Psi \rangle \neq 0$, the value $r = \langle x, \Psi \rangle$ and an element $y \in \{0, 1\}^n$ such that $\langle y, \Psi \rangle \neq 0$. The value of $\langle y, \Psi \rangle$ is not given.

(b) The algorithm applies the subroutine `relatebasiements` to find (α, ϑ) such that $c(\alpha) |y\rangle = e^{i\vartheta} |x\rangle$. In particular, after this step, the value $\langle x, c(\alpha)y \rangle = e^{i\vartheta}$ is known and it is non-zero.

(c) The algorithm then applies the subroutine `overlaptriple` to compute $w = \langle y, \Psi \rangle$. The triple (Γ, y, w) is a valid description of Ψ .

Figure 6: An illustration of the algorithm `convert`. Dotted lines represent inner products that are non-zero, but that are not provided / have not yet been computed by the algorithm.

Lemma 3.6. *The algorithm `convert` : $\text{Desc}_n \times \{0, 1\}^n \rightarrow \text{Desc}_n$ given in Fig. 5 runs in time $O(n^3)$. Assume that $d \in \text{Desc}_n$ and $y \in \{0, 1\}^n$ satisfy $\langle y, \Psi(d) \rangle \neq 0$. Then*

$$\Psi(\text{convert}(d, y)) = \Psi(d) . \quad (37)$$

Furthermore, denoting the output of `convert`(d, y) by $d' = (\Gamma', y', s')$ we have

$$y' = y$$

as well as

$$|s'|^2 = |\langle y', \Psi(d) \rangle|^2 = |\langle y, \Psi(d) \rangle|^2 . \quad (38)$$

Proof. Let us denote the input to `convert` by (d, y) , where $d = (\Gamma, x, r) \in \text{Desc}_n$ and $y \in \{0, 1\}^n$. Then

$$\langle x, \Psi(d) \rangle \neq 0 \quad (39)$$

since d is a description of $\Psi(d)$. Furthermore, for (α, ϑ) as defined in line 2 we have

$$\langle x, c(\alpha)y \rangle = e^{i\vartheta} \neq 0 \quad (40)$$

by definition of the algorithm `relatebasiselements`. In line 3 of `convert`, the matrices Γ_j , $j \in [3]$ are the covariance matrices of the states

$$(\Phi_0, \Phi_1, \Phi_2) = (\Psi(d), |x\rangle, |y\rangle) .$$

We note that Eq. (39) and the assumption $\langle y, \Psi(d) \rangle \neq 0$ imply that these three states have identical parity. The value w computed in line 6 using `overlaptriple` is equal to the overlap

$$w = \langle \Phi_2, \Phi_0 \rangle = \langle y, \Psi(d) \rangle , \quad (41)$$

because

$$\begin{aligned} u &= \bar{r} = \overline{\langle x, \Psi(d) \rangle} = \langle \Psi(d), x \rangle = \langle \Phi_0, \Phi_1 \rangle \neq 0 \\ v &= e^{i\vartheta} = \langle x, c(\alpha)y \rangle = \langle \Phi_1, c(\alpha)\Phi_2 \rangle \neq 0 \end{aligned}$$

by Eqs. (39) and (40). Eq. (41) together with the assumption $\langle y, \Psi(d) \rangle \neq 0$ show that the output (Γ, y, w) is a description of $\Psi(d)$. This completes the proof of Eq. (37).

Eq. (38) is trivially satisfied because

$$s' = w = \langle y, \Psi(d) \rangle .$$

The complexity of the algorithm is dominated by `overlaptriple`, which takes time $O(n^3)$. \square

3.2 Computing overlaps and descriptions of evolved/measured states

Based on the subroutines `findsupport`, `relatebasiselements`, `overlaptriple` and `convert`, we can now describe our main algorithms `overlap`, `evolve`, `measureprob` and `postmeasure` for overlaps, Gaussian unitary evolution, to compute the outcome probability and the post-measurement state when measuring the occupation number, respectively. We give pseudocode for each algorithm and establish the associated claims.

We give pseudocode for the algorithm `overlap` in Fig. 7 and we illustrate it in Fig. 8.

Require: $d_1 = (\Gamma_1, x_1, r_1), d_2 = (\Gamma_2, x_2, r_2) \in \text{Desc}_n$

- 1: **function** `overlap`(d_1, d_2)
- 2: $\sigma_1 \leftarrow \text{Pf}(\Gamma_1)$ \triangleright compute the parity σ_j of $\Psi(d_j)$
- 3: $\sigma_2 \leftarrow \text{Pf}(\Gamma_2)$
- 4: **if** $\sigma_1 \neq \sigma_2$ **then**
- 5: **return** 0 \triangleright states with different parities have zero overlap
- 6: $(\alpha, \vartheta) \leftarrow \text{relatebasiselements}(x_2, x_1)$ \triangleright (α, ϑ) satisfies $c(\alpha)|x_2\rangle = e^{i\vartheta}|x_1\rangle$
- 7: $\Gamma'_0 \leftarrow \Gamma_1, \Gamma'_1 \leftarrow \Gamma(|x_1\rangle), \Gamma'_2 \leftarrow \Gamma_2$ \triangleright covariance matrices of $\Psi(d_1), |x_1\rangle$
and $\Psi(d_2)$
- 8: $u \leftarrow \bar{r}_1$ $\triangleright u = \langle \Psi(d_1), x_1 \rangle$
- 9: $v \leftarrow e^{i\vartheta} r_2$ $\triangleright v = \langle x_1, c(\alpha)\Psi(d_2) \rangle$
- 10: $w \leftarrow \text{overlaptriple}(\Gamma'_0, \Gamma'_1, \Gamma'_2, \alpha, u, v)$ \triangleright compute the overlap $\langle \Psi(d_2), \Psi(d_1) \rangle$
- 11: **return** \bar{w}

Figure 7: The algorithm `overlap` takes descriptions $d_1, d_2 \in \text{Desc}_n$ and outputs the overlap $\langle \Psi(d_1), \Psi(d_2) \rangle$.

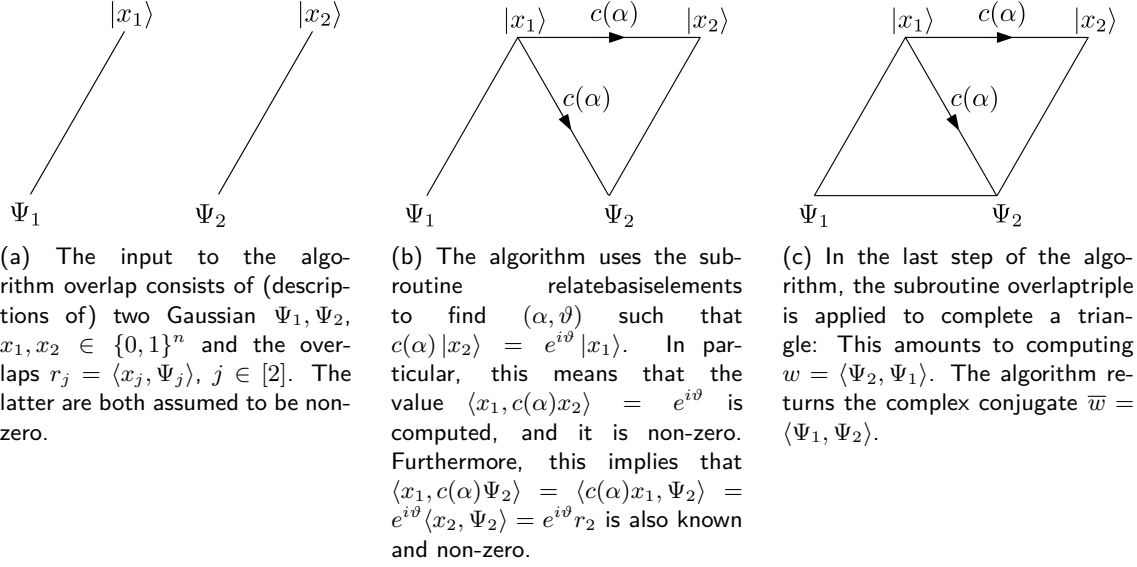


Figure 8: An illustration of the algorithm overlap.

Lemma 3.7. *The algorithm overlap : $\text{Desc}_n \times \text{Desc}_n \rightarrow \mathbb{C}$ given in Fig. 7 runs in time $O(n^3)$. It satisfies*

$$\text{overlap}(d_1, d_2) = \langle \Psi(d_1), \Psi(d_2) \rangle \quad \text{for all} \quad d_1, d_2 \in \text{Desc}_n . \quad (42)$$

Proof. Let $d_j = (\Gamma_j, x_j, r_j) \in \text{Desc}_n$ for $j \in [2]$. Then

$$r_j = \langle x_j, \Psi(d_j) \rangle \neq 0 \quad \text{for} \quad j \in [2] , \quad (43)$$

by assumption.

Line 4 treats the case where $\Psi(d_1)$ and $\Psi(d_2)$ have different parity, and are thus orthogonal. Starting from line 6, we can hence assume that the parities σ_1, σ_2 of $\Psi(d_1), \Psi(d_2)$ are identical, $\sigma = \sigma_1 = \sigma_2$. By Eq. (43), this implies that both $|x_1\rangle$ and $|x_2\rangle$ also have parity σ , that is,

$$\sigma(|x_1\rangle) = \sigma(|x_2\rangle) = \sigma(|\Psi(d_1)\rangle) = \sigma(|\Psi(d_2)\rangle) . \quad (44)$$

By definition of `relatebasiselements`, the pair (α, ϑ) computed in line 6 satisfies

$$c(\alpha)|x_2\rangle = e^{i\vartheta}|x_1\rangle . \quad (45)$$

Consider the triple of states

$$(\Phi_0, \Phi_1, \Phi_2) = (\Psi(d_1), |x_1\rangle, \Psi(d_2)) .$$

Then the matrices Γ'_j , $j \in [3]$ defined in line 7 of the algorithm are the covariance matrices of Φ_j , $j \in \{0, 1, 2\}$. We have

$$u = \bar{r}_1 = \overline{\langle x_1, \Psi(d_1) \rangle} = \langle \Psi(d_1), x_1 \rangle = \langle \Phi_0, \Phi_1 \rangle \neq 0 ,$$

by Eq. (43), and similarly

$$\begin{aligned}
v &= e^{i\vartheta} r_2 \\
&= e^{i\vartheta} \langle x_2, \Psi(d_2) \rangle \\
&= \langle e^{-i\vartheta} x_2, \Psi(d_2) \rangle \\
&= \langle c(\alpha) x_1, \Psi(d_2) \rangle && \text{with (45)} \\
&= \langle x_1, c(\alpha) \Psi(d_2) \rangle && \text{because } c(\alpha) \text{ is self-adjoint} \\
&= \langle \Phi_1, c(\alpha) \Phi_2 \rangle \\
&\neq 0 && \text{because } r_2 \neq 0 \text{ by Eq. (43) .}
\end{aligned}$$

Furthermore, by Eq. (44) the states (Φ_0, Φ_1, Φ_2) have identical parity. It thus follows from the properties of overlaptriple that the quantity w computed in step line 10 is equal to

$$w = \langle \Psi(d_2), \Psi(d_1) \rangle .$$

Since the output of the algorithm is the complex conjugate $\bar{w} = \langle \Psi(d_1), \Psi(d_2) \rangle$, this implies the claim (42).

The runtime of `overlap` is dominated by `overlaptriple`, hence it is of order $O(n^3)$. \square

We give pseudocode for the algorithm `evolve` in Fig. 9 and we illustrate it in Fig. 10.

Lemma 3.8. *The algorithm `evolve` : $\text{Gen}(O(2n)) \times \text{Desc}_n \rightarrow \text{Desc}_n$ given in Fig. 9 runs in time $O(n^3)$. Consider an arbitrary generator $R \in \text{Gen}(O(2n))$ and a description $d \in \text{Desc}_n$. Then*

$$\Psi(\text{evolve}(R, d)) = U_R \Psi(d) ,$$

that is, the output of `evolve` is a description of the evolved state $U_R \Psi(d)$. Furthermore, denoting the output by $d' = (\Gamma', x', r') = \text{evolve}(R, d)$ we have

$$|r'|^2 = |\langle x', \Psi(d') \rangle|^2 \geq 2^{-n} . \quad (46)$$

Proof. Let us denote the input of `evolve` by (R, d) where $R \in \text{Gen}(O(2n))$ and $d = (\Gamma, x, r) \in \text{Desc}_n$. The state $U_R \Psi(d)$ has covariance matrix $\Gamma_0 = R \Gamma R^T$ computed in line 2 (see Section 2.7). By the properties of `findsupport` (see Lemma 3.4), the state $|y\rangle$ with $y = \text{findsupport}(\Gamma_0) \in \{0, 1\}^n$ computed in line 3 is such that

$$|\langle y, U_R \Psi(d) \rangle|^2 \geq 2^{-n} . \quad (47)$$

In particular, it is non-zero. The remainder of the algorithm computes $\langle y, U_R \Psi(d) \rangle$.

We first show the following:

Claim 3.9. *Lines 4–15 compute $(z, s) \in \{0, 1\}^n \times \mathbb{C}$ such that*

$$|\langle z, U_R x \rangle|^2 \geq 1/2 \quad (48)$$

and

$$s = \langle z, U_R x \rangle . \quad (49)$$

Proof. Here we are using the fact that for any generator $R \in \text{Gen}(O(2n))$, the associated Gaussian unitary U_R has a local action on the mode operators. In particular, we can easily compute the image $U_R |x\rangle$ of a number state $|x\rangle$ under U_R . We distinguish two cases:

```

Require:  $R \in \text{Gen}(O(2n))$ 
Require:  $d = (\Gamma, x, r) \in \text{Desc}_n$ 
1: function evolve( $R, d$ )
2:    $\Gamma_0 \leftarrow R\Gamma R^T$  ▷ covariance matrix of  $U_R\Psi(d)$ 
3:    $y \leftarrow \text{findsupport}(\Gamma_0)$  ▷ gives  $y$  such that  $|\langle y, U_R\Psi(d) \rangle|^2 \geq 2^{-n}$ 
4:   if  $R = R_{j,k}(\vartheta)$  for  $j < k \in [2n], \vartheta \in \mathbb{R}$  then ▷  $R = R_{j,k}(\vartheta)$  is a Givens rotation
5:     if  $\cos^2(\vartheta/2) \geq 1/2$  then
6:        $z \leftarrow x$ 
7:        $s \leftarrow \cos(\vartheta/2)$ 
8:     else
9:        $z \leftarrow x \oplus e_j \oplus e_k$ 
10:       $\beta \leftarrow \beta_j(x) + \beta_k(x)$ 
11:       $s \leftarrow e^{i\pi\beta} \sin(\vartheta/2)$ 
12:   if  $R = R_j$  for  $j \in [2n]$  then ▷  $R = R_j$  is a reflection
13:      $z \leftarrow x \oplus e_j$ 
14:      $\beta \leftarrow \beta_j(x)$ 
15:      $s \leftarrow e^{i\pi\beta}$  ▷  $(s, z)$  satisfies  $s = \langle z, U_R x \rangle, |s|^2 \geq 1/2$ 
16:    $(\alpha, \gamma) \leftarrow \text{relatebasiselements}(y, z)$  ▷  $(\alpha, \gamma)$  satisfies  $c(\alpha) |y\rangle = e^{i\gamma} |z\rangle$ 
17:    $\Gamma_1 \leftarrow R\Gamma(|x\rangle)R^T, \Gamma_2 \leftarrow \Gamma(|y\rangle)$  ▷ covariance matrices of  $U_R|x\rangle$  and  $|y\rangle$ 
18:    $u \leftarrow \bar{r}$  ▷  $u = \langle U_R\Psi, U_R x \rangle$ 
19:    $v \leftarrow e^{i\gamma\bar{s}}$  ▷  $v = \langle U_R x, c(\alpha)y \rangle$ 
20:    $w \leftarrow \text{overlaptriple}(\Gamma_0, \Gamma_1, \Gamma_2, \alpha, u, v)$  ▷ compute the overlap  $\langle y, U_R\Psi(d) \rangle$ 
21:   return  $(\Gamma_0, y, w)$  ▷ return a description of  $U_R|\Psi(d)\rangle$ 

```

Figure 9: The algorithm evolve takes a description $d \in \text{Desc}_n$ and an orthogonal matrix $R \in \text{Gen}(O(2n))$ associated with the Gaussian unitary U_R and computes a description for the state $U_R\Psi(d)$. In this algorithm, the functions $\beta_s : \{0, 1\}^n \rightarrow \mathbb{R}$ for $s \in [n]$ are defined as $\beta_s(x) = \eta_s(x) + (x_s - \frac{1}{2}) \cdot (s+1)$, $x \in \{0, 1\}^n$ with $\eta_s(x)$ given in Eq. (12).

- (i) $R = R_{j,k}(\vartheta)$, $j < k \in [2n]$, $\vartheta \in [0, 2\pi)$ is a Givens-rotation (see Lines 4–11): In this case, R is associated with the unitary evolution operator

$$U_{j,k} = \exp(\vartheta/2 c_j c_k) = \cos(\vartheta/2)I + \sin(\vartheta/2)c_j c_k .$$

It maps a basis state $|x\rangle$, $x \in \{0, 1\}^n$, to

$$U_{j,k}(\vartheta) |x\rangle = \cos(\vartheta/2) |x\rangle + e^{i\pi(\beta_j(x) + \beta_k(x))} \sin(\vartheta/2) |x \oplus e_j \oplus e_k\rangle \quad (50)$$

where we introduced the quantities

$$\beta_s(x) = \eta_s(x) + \left(x_s - \frac{1}{2}\right) \cdot (s+1) \quad \text{for any } s \in [n] ,$$

with $\eta_s(x)$ defined in Eq. (12). To obtain Eq. (50), we used that

$$c_j |x\rangle = e^{i\pi\beta_j(x)} |x \oplus e_j\rangle \quad \text{for all } j \in [2n] \quad \text{and } x \in \{0, 1\}^n \quad (51)$$

because

$$c_{2j-1} |x\rangle = (-1)^{\eta_j(x)} |x \oplus e_j\rangle \quad \text{and} \quad c_{2j} |x\rangle = -i(-1)^{\eta_j(x) + x_j} |x \oplus e_j\rangle .$$

Eq. (50) motivates the following case distinction:

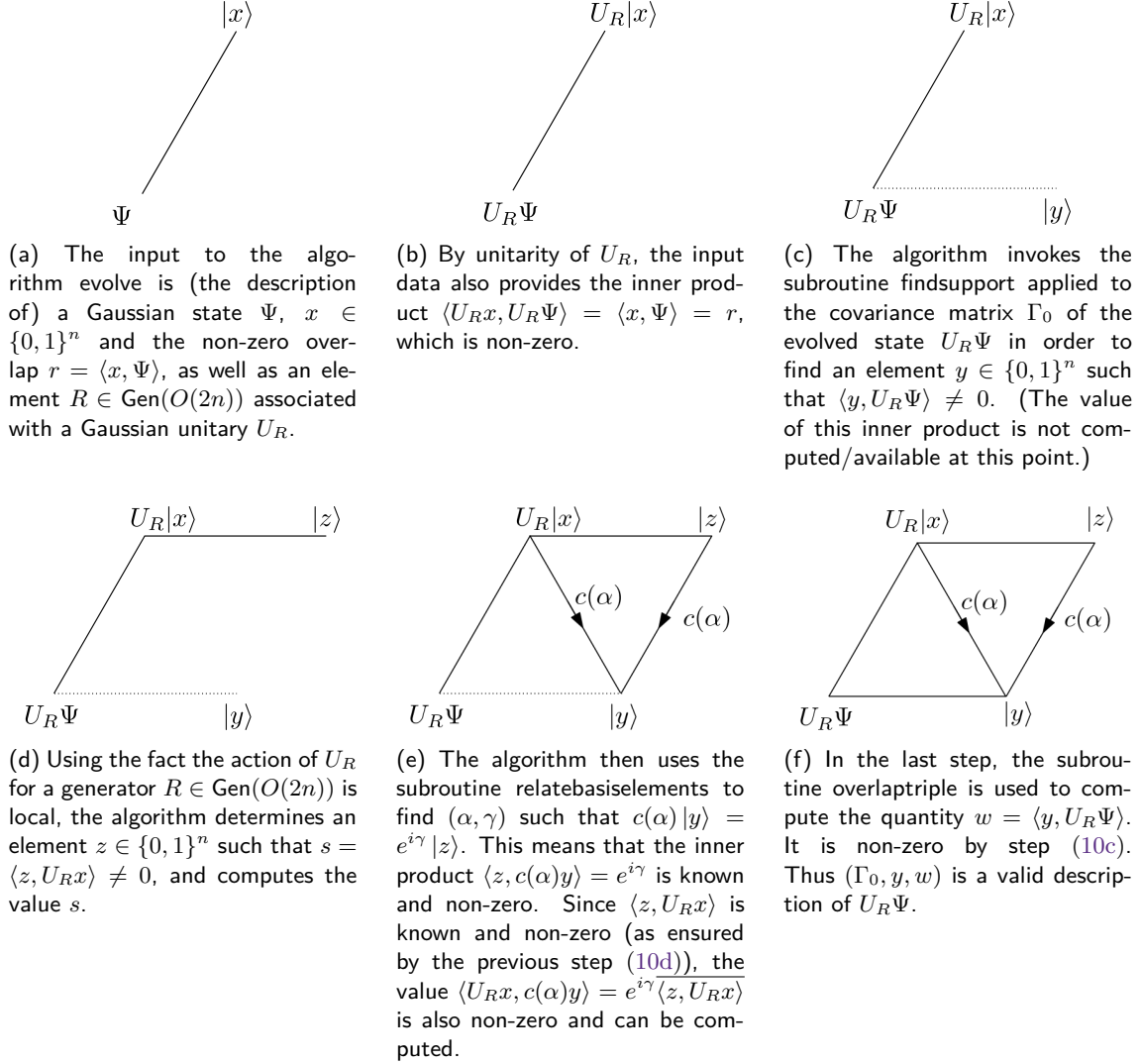


Figure 10: An illustration of the algorithm evolve. Dotted lines correspond to inner products whose value is non-zero, but has not been computed at that stage of the algorithm.

- (a) $\cos^2(\vartheta/2) \geq 1/2$ (see Lines 6–7): Here $|x\rangle$ has higher amplitude than $|x \oplus e_j \oplus e_k\rangle$ in the state $U_{j,k}(\vartheta)|x\rangle$. The algorithm picks $z = x$ (Line 6) and sets $s = \cos(\vartheta/2)$ (line 7). In particular, comparing with (50), it follows immediately that the claims (48) and (49) are satisfied.
- (b) $\cos^2(\vartheta/2) < 1/2$ (see Lines 9–11): In this case the algorithm ensures that

$$z = x \oplus e_j \oplus e_k \quad \text{by Line 9} \quad (52)$$

$$\beta = \beta_j(x) + \beta_k(x) \quad \text{by Line 10}$$

$$s = e^{i\pi(\beta_j(x) + \beta_k(x))} \sin(\vartheta/2) \quad \text{by Lines 10 and 11} \quad (53)$$

Because $\cos^2(\vartheta/2) + \sin^2(\vartheta/2) = 1$ we have

$$|s|^2 \geq \frac{1}{2}$$

by the assumption that $\cos^2(\vartheta/2) < 1/2$. To prove the claims (48) and (49), it thus suffices to show the second claim (49). But this again follows from (50) and the definitions (52) of z and (53) of s , i.e., we have $s = \langle z, U_{j,k}(\vartheta)x \rangle$.

- (ii) $R = R_j$, $j \in [2n]$ is a reflection (see Lines 12–15): Here R is associated with the unitary evolution operator

$$U_j = c_j .$$

Its action on $|x\rangle$, $x \in \{0, 1\}^n$, is described by Eq. (51), i.e., we have

$$U_j |x\rangle = e^{i\pi\beta_j(x)} |x \oplus e_j\rangle .$$

This state is proportional to $|x \oplus e_j\rangle$, showing that the choice

$$z = x \oplus e_j \quad (\text{Line 13})$$

$$\beta = \beta_j(x) \quad (\text{Line 14})$$

$$s = e^{i\pi\beta} \quad (\text{Line 15})$$

indeed ensures that the claims (48) and (49) are satisfied. □

Equipped with Claim 3.9, we can show that the algorithm `evolve` has the desired functionality. The matrix Γ_0 computed in Line 2 is the covariance matrix of the evolved state $U_R\Psi(d)$, whereas Γ_1, Γ_2 computed in Line 17 are the covariance matrices of $U_R|x\rangle$ and $|y\rangle$, respectively. Thus `overlaptriple` in Line 20 is invoked on the triple of states

$$(\Phi_0, \Phi_1, \Phi_2) = (U_R\Psi(d), U_R|x\rangle, |y\rangle) .$$

To check that the requirements of `overlaptriple` are satisfied, first observe that

$$\begin{aligned} u &= \bar{r} && \text{by Line 18} \\ &= \langle \Psi(d), x \rangle && \text{by definition of } r \\ &= \langle U_R\Psi(d), U_Rx \rangle && \text{by unitarity of } U_R \\ &= \langle \Phi_0, \Phi_1 \rangle . \end{aligned}$$

Furthermore, this is non-zero because r (part of the input) is non-zero by definition of the description $d = (\Gamma_0, x, r)$ of $\Psi(d)$.

By the defining property of the subroutine `relatebasiselements`, Line 16 of the algorithm computes $(\alpha, \gamma) \in \{0, 1\}^{2n} \times [0, 2\pi)$ such that

$$c(\alpha) |y\rangle = e^{i\gamma} |z\rangle . \quad (54)$$

We also have

$$\begin{aligned} v &= e^{i\gamma\bar{s}} && \text{by line 18} \\ &= e^{i\gamma\overline{\langle z, U_Rx \rangle}} && \text{by (49)} \\ &= \langle U_Rx, c(\alpha)y \rangle && \text{by (54)} \\ &= \langle \Phi_1, c(\alpha)\Phi_2 \rangle . \end{aligned}$$

Because $|s|^2 \geq 1/2$ (see Claim 3.9), this is non-zero.

We conclude from the the properties of `overlaptriple` that

$$\begin{aligned} w &= \langle \Phi_2, \Phi_0 \rangle && \text{see Line 20} \\ &= \langle y, U_R\Psi(d) \rangle . \end{aligned}$$

By construction of y using `findsupport`, we have

$$|w|^2 \geq 2^{-n} ,$$

see Eq. (47). In particular, we conclude that the triple (Γ_0, y, w) is a valid description of $U_R\Psi(d)$ with the desired property (46). This is what the algorithm returns.

The runtime of the algorithm `evolve` is dominated by the runtime $O(n^3)$ of the algorithm `overlaptriple`. □

We give pseudocode for the algorithm `measureprob` in Fig. 11.

Require: $d = (\Gamma, x, r) \in \text{Desc}_n$
Require: $j \in [n]$
Require: $s \in \{0, 1\}$
1: **function** `measureprob`(d, j, s)
2: **return** $\frac{1}{2}(1 + (-1)^s \Gamma_{2j-1, 2j})$

Figure 11: The subroutine `measureprob` takes as input a description $d = (\Gamma, x, r) \in \text{Desc}_n$ of a Gaussian state $\Psi(d)$, an integer $j \in [n]$ and a bit $s \in \{0, 1\}$. It outputs the probability of obtaining the measurement outcome s when measuring the occupation number operator $a_j^\dagger a_j$. The outcome probability does not depend on the global phase of $\Psi(d)$ (which is determined by its reference state x and the overlap r), but only on its covariance matrix Γ .

Lemma 3.10. *The algorithm `measureprob` : $\text{Desc}_n \times [n] \times \{0, 1\} \rightarrow \mathbb{R}$ given in Fig. 11 runs in time $O(1)$. It satisfies*

$$\text{measureprob}(d, j, s) = \langle \Psi(d), \Pi_j(s) \Psi(d) \rangle \quad \text{for all } d \in \text{Desc}_n, j \in [n], s \in \{0, 1\} ,$$

where $\Pi_j(s) = \frac{1}{2}(I + (-1)^s i c_{2j-1} c_{2j})$ is the projection onto the eigenvalue- s eigenspace of $a_j^\dagger a_j$.

Proof. We denote the input to `measureprob` by (d, j, s) where $d = (\Gamma, x, r) \in \text{Desc}_n$ is a description of a state $\Psi(d)$, $j \in [n]$ and $s \in \{0, 1\}$. Given the state $\Psi(d)$, the probability of obtaining measurement outcome s when measuring the occupation number operator $a_j^\dagger a_j$ is given by Eq. (27). This is the output of the algorithm in line 2 and gives the claim. Computing line 2 requires a constant number of arithmetic operations, giving the runtime $O(1)$. □

We give pseudocode for the algorithm `postmeasure` in Fig. 12 and we illustrate it in Fig. 13.

Lemma 3.11. *The algorithm `postmeasure` : $\text{Desc}_n \times [n] \times \{0, 1\} \times [0, 1] \rightarrow \text{Desc}_n$ given in Fig. 12 runs in time $O(n^3)$. Let $d \in \text{Desc}_n$, $j \in [n]$ and $s \in \{0, 1\}$ be arbitrary. Let $\Pi_j(s) = \frac{1}{2}(I + (-1)^s i c_{2j-1} c_{2j})$ be the projection onto the eigenvalue- s eigenspace of $a_j^\dagger a_j$ and let $p = \|\Pi_j(s) \Psi(d)\|^2$. Then*

$$\Psi(\text{postmeasure}(d, j, s, p)) = \frac{\Pi_j(s) \Psi(d)}{\|\Pi_j(s) \Psi(d)\|} ,$$

```

Require:  $d = (\Gamma, x, r) \in \text{Desc}_n$ 
Require:  $j \in [n]$ 
Require:  $s \in \{0, 1\}$ 
Require:  $p = \|\Pi_j(s)\Psi(d)\|^2 \in [0, 1]$   $\triangleright$  probability of outcome  $s$  when measuring  $a_j^\dagger a_j$ 
1: function postmeasure( $d, j, s, p$ )
2:    $\Gamma' \leftarrow 0 \in \text{Mat}_{2n \times 2n}(\mathbb{R})$   $\triangleright$  compute covariance matrix of post-measurement
   state  $\Psi'$ 
3:    $\Gamma'_{2j, 2j-1} \leftarrow (-1)^s$ 
4:   for  $\ell \leftarrow 1$  to  $n - 1$  do
5:     for  $k \leftarrow \ell + 1$  to  $n$  do
6:       if  $(k, \ell) \neq (2j, 2j - 1)$  then
7:          $\Gamma'_{k, \ell} \leftarrow \Gamma_{k, \ell} + \frac{(-1)^s}{2p} (\Gamma_{2j-1, \ell} \Gamma_{2j, k} - \Gamma_{2j-1, k} \Gamma_{2j, \ell})$ 
8:    $\Gamma' \leftarrow \Gamma' - (\Gamma')^T$ 
9:    $y \leftarrow \text{findsupport}(\Gamma')$   $\triangleright$  find  $y$  such that  $|\langle y, \Psi' \rangle|^2 \geq 2^{-n}$ 
10:   $(\alpha, \vartheta) \leftarrow \text{relatebaselements}(y, x)$   $\triangleright$   $(\alpha, \vartheta)$  are such that  $c(\alpha) |y\rangle = e^{i\vartheta} |x\rangle$ 
11:   $\Gamma_0 \leftarrow \Gamma$ ,  $\Gamma_1 \leftarrow \Gamma(|x\rangle)$ ,  $\Gamma_2 \leftarrow \Gamma(|y\rangle)$   $\triangleright$  covariance matrices of  $\Psi(d)$ ,  $|x\rangle$  and  $|y\rangle$ 
12:   $u \leftarrow \bar{r}$   $\triangleright u = \langle \Psi(d), x \rangle$ 
13:   $v \leftarrow e^{i\vartheta}$   $\triangleright v = \langle x, c(\alpha)y \rangle$ 
14:   $w \leftarrow \text{overlaptriple}(\Gamma_0, \Gamma_1, \Gamma_2, \alpha, u, v)$   $\triangleright w = \langle y, \Psi(d) \rangle$ 
15:  return  $(\Gamma', y, w/\sqrt{p})$   $\triangleright$  return a description of  $\Psi'$ 

```

Figure 12: The algorithm `postmeasure` takes as input a description $d \in \text{Desc}_n$, an integer $j \in [n]$, a bit $s \in \{0, 1\}$ and a real number $p \in [0, 1]$. Assuming $p = \|\Pi_j(s)\Psi(d)\|^2$, the algorithm outputs a description of the post-measurement state $\Psi' = (\Pi_j(s)\Psi(d))/\|\Pi_j(s)\Psi(d)\|$ when measuring the number operator $a_j^\dagger a_j$ and obtaining the outcome s where $s \in \{0, 1\}$. Here, $\Pi_j(s) = (I + (-1)^s i c_{2j-1} c_{2j})/2$ is the projection onto the eigenvalue s eigenspace of $a_j^\dagger a_j$.

that is, `postmeasure` computes a description of the post-measurement state when measuring $a_j^\dagger a_j$ and obtaining outcome s . Denoting the output of the algorithm by

$$d' = (\Gamma', x', r') = \text{postmeasure}(d, j, s, p) ,$$

we further have

$$|r'|^2 = |\langle x', \Psi(d') \rangle|^2 \geq 2^{-n} . \quad (55)$$

Proof. We denote the input to `postmeasure` by (d, j, s, p) , where $d \in \text{Desc}_n$, $j \in [n]$, $s \in \{0, 1\}$ and $p \in [0, 1]$. For brevity, let us denote the post-measurement state when measuring the observable $a_j^\dagger a_j$ and obtaining outcome s by

$$\Psi' = \frac{\Pi_j(s)\Psi(d)}{\|\Pi_j(s)\Psi(d)\|} .$$

In lines 2–7, the algorithm `postmeasure` computes the covariance matrix Γ' of Ψ' according to Eq. (28). In line 9 the algorithm uses `findsupport` to find $y \in \{0, 1\}^n$ such that

$$|\langle y, \Psi' \rangle|^2 \geq 2^{-n} . \quad (56)$$

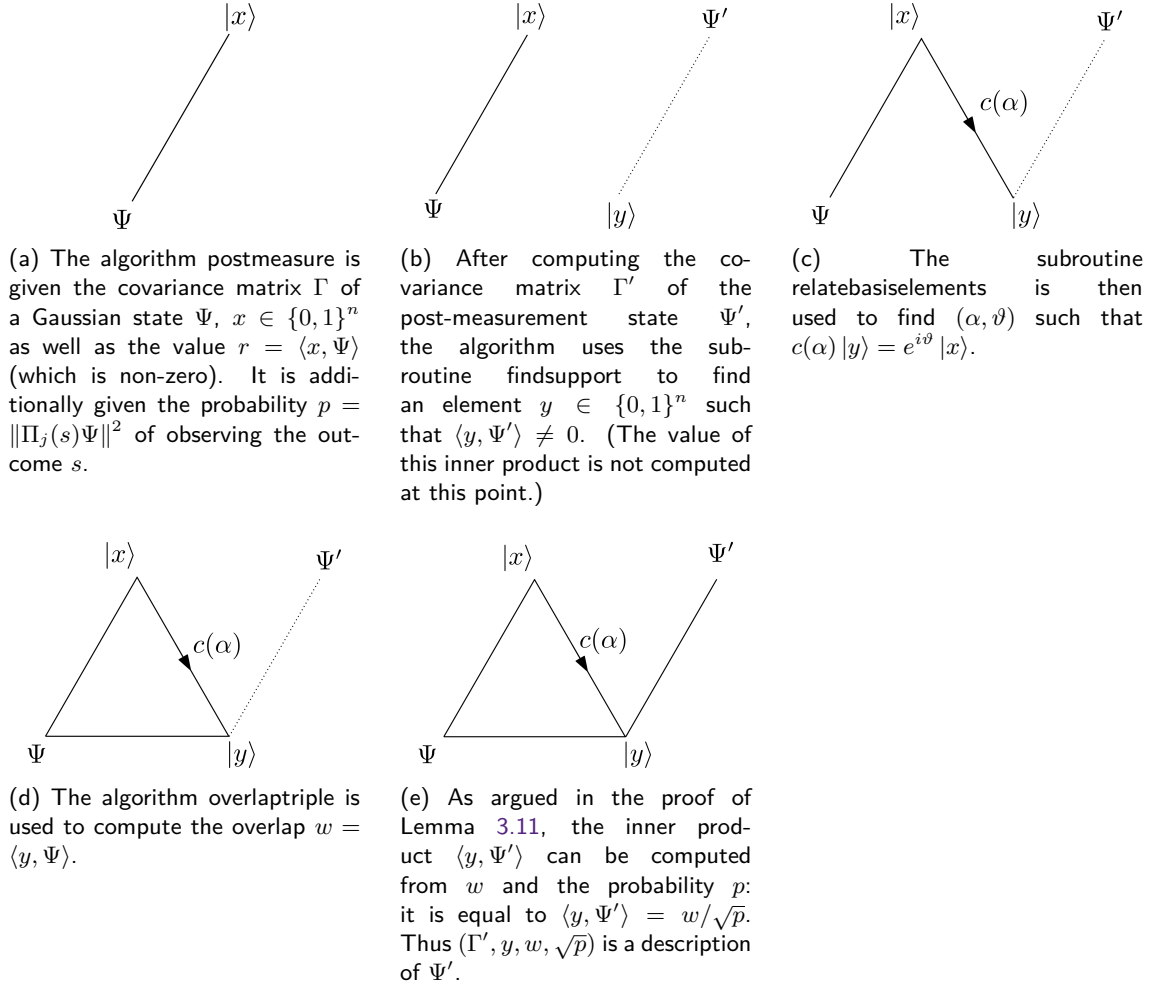


Figure 13: An illustration of the algorithm postmeasure, which computes a description of the post-measurement state $\Psi' = \frac{\Pi_j(s)\Psi}{\|\Pi_j(s)\Psi\|}$ given a description of Ψ .

Line 10 provides $(\alpha, \vartheta) \in \{0, 1\}^{2n} \times \mathbb{R}$ such that

$$c(\alpha) |y\rangle = e^{i\vartheta} |x\rangle . \quad (57)$$

Line 11 of the algorithm sets the matrices $(\Gamma_0, \Gamma_1, \Gamma_2)$ equal to the covariance matrices of the three states

$$(\Phi_0, \Phi_1, \Phi_2) = (\Psi(d), |x\rangle, |y\rangle) .$$

We check that the conditions for applying `overlaptriple` in Line (14) are satisfied. We have

$$\begin{aligned} u &= \bar{r} && \text{by Line 12} \\ &= \langle \Psi(d), x \rangle && \text{by definition of } r \\ &= \langle \Phi_0, \Phi_1 \rangle , \end{aligned}$$

and this is non-zero because $d = (\Gamma, x, r)$ is a valid description (hence $r \neq 0$). Similarly as before, we also have

$$\begin{aligned} v &= e^{i\vartheta} && \text{by Line 13} \\ &= \langle x, c(\alpha)y \rangle && \text{by Eq. (57)} \\ &= \langle \Phi_1, c(\alpha)\Phi_2 \rangle . \end{aligned}$$

In particular, this is also non-zero. The requirements to run `overlaptriple` in Line 14 are therefore met, and Line 14 returns

$$\begin{aligned} w &= \langle \Phi_2, \Phi_0 \rangle \\ &= \langle y, \Psi(d) \rangle . \end{aligned} \quad (58)$$

It remains to show that $(\Gamma', y, w/\sqrt{p})$ (the expression returned by the algorithm) is a valid description of the post-measurement state Ψ' , and to establish the bound

$$|w/\sqrt{p}|^2 \geq 2^{-n} \quad (59)$$

in order to prove Eq. (55).

Inserting $\Psi' = \Pi_j(s)\Psi/\sqrt{p}$, Eq. (56) implies that

$$2^{-n} \leq |\langle y, \Psi' \rangle|^2 = \frac{1}{p} |\langle y, \Pi_j(s)\Psi \rangle|^2 = \frac{1}{p} |\langle \Pi_j(s)y, \Psi \rangle|^2 \leq \frac{1}{p} \|\Pi_j(s)y\|^2 \cdot \|\Psi\|^2 \quad (60)$$

because $\Pi_j(s)$ is self-adjoint and with the Cauchy-Schwarz inequality. In particular, we have $\Pi_j(s)y \neq 0$ and thus

$$\Pi_j(s)y = y \quad (61)$$

since any number state $|y\rangle$ is an eigenvector of the projection $\Pi_j(s)$. Inserting (61) into (60) and using (58) we obtain the bound

$$2^{-n} \leq \frac{1}{p} |\langle y, \Psi \rangle|^2 = \frac{|w|^2}{p} ,$$

establishing (59). Eq. (61) and the self-adjointness of $\Pi_j(s)$ also imply that

$$\langle y, \Psi' \rangle = \frac{1}{\sqrt{p}} \langle y, \Pi_j(s)\Psi \rangle = \frac{1}{\sqrt{p}} \langle y, \Psi \rangle = \frac{w}{\sqrt{p}} .$$

Since Γ' is the covariance matrix of Ψ' and $p = \|\Pi_j(s)\Psi(d)\|^2$ is the probability of obtaining outcome s when measuring $a_j^\dagger a_j$, this shows that $(\Gamma', y, w/\sqrt{p})$ is a valid description of Ψ' as claimed.

The complexity of the algorithm is dominated by `overlaptriple`, which takes time $O(n^3)$. \square

3.3 Initial states for computation

Using the algorithm `evolve`, it is straightforward to generate a description of a state that is obtained by applying a sequence of Gaussian unitaries (generators) to the vacuum state. This is all that is typically needed to describe initial states.

In cases where we do not need to fix the overall phase, we can generate a description from the covariance matrix. The algorithm `describe` takes as input the covariance matrix Γ of a Gaussian state Ψ and outputs a description $d \in \text{Desc}_n$ of a Gaussian state which is equal to Ψ up to a global phase. It is given in Fig. 14 and it simply uses the subroutine `findsupport` and Eq. (23).

Require: $\Gamma \in \text{Mat}_{n \times n}(\mathbb{R})$ covariance matrix of a pure Gaussian state Ψ

- 1: **function** describe(Γ)
- 2: $y \in \{0, 1\}^n$
- 3: $y \leftarrow \text{findsupport}(\Gamma)$ \triangleright find y such that $|\langle y, \Psi \rangle|^2 \geq 2^{-n}$
- 4: $\sigma \leftarrow \text{Pf}(\Gamma)$
- 5: $r \leftarrow \sqrt{\sigma 2^{-n} \text{Pf}(\Gamma(|y\rangle) + \Gamma)}$ \triangleright compute $|\langle y, \Psi \rangle|$
- 6: **return** (Γ, y, r)

Figure 14: The algorithm describe: Given the covariance matrix Γ of a Gaussian state $|\Psi\rangle$, it outputs $d \in \text{Desc}_n$ such that $|\langle \Psi(d), \Psi \rangle| = 1$.

Lemma 3.12. *The algorithm describe : $\text{Mat}_{2n \times 2n}(\mathbb{R}) \rightarrow \text{Desc}_n$ runs in time $O(n^3)$. Its output is such that for every covariance matrix Γ , the state $\Psi(\text{describe}(\Gamma))$ is a Gaussian state with covariance matrix Γ . We have*

$$|r|^2 = |\langle x, \Psi(d) \rangle|^2 \geq 2^{-n} \quad (62)$$

for $d = (\Gamma, x, r) = \Psi(\text{describe}(\Gamma))$.

Proof. Let $\Gamma \in \text{Mat}_{2n \times 2n}(\mathbb{R})$ be a covariance matrix and let Ψ be a Gaussian state with covariance matrix Γ . By definition of the algorithm findsupport, the value $y \in \{0, 1\}^n$ computed in line 3 satisfies

$$|\langle y, \Psi \rangle|^2 \geq 2^{-n} .$$

By Eq. (23), the value r computed in Line 5 satisfies

$$r = |\langle y, \Psi \rangle| .$$

In particular, there is an angle $\vartheta \in [0, 2\pi)$ such that

$$r = \sqrt{\sigma 2^{-n} \text{Pf}(\Gamma(|y\rangle) + \Gamma)} = \langle y, e^{i\vartheta} \Psi \rangle .$$

It follows immediately that $d = (\Gamma, y, r)$ is a valid description of the Gaussian state $e^{i\vartheta} \Psi = \Psi(d)$ with the required property (62). \square

4 Classical simulation of fermionic Gaussian circuits with non-Gaussian initial states

In this section, we argue that the techniques developed in Section 3 to describe fermionic Gaussian states (including relative phases) give rise to efficient classical simulation algorithms for computations composed of non-Gaussian initial states, Gaussian unitaries and occupation number measurements. Specifically, we argue that algorithms developed in the context of stabilizer circuits can immediately be translated to this fermionic setup. Furthermore, this translation maintains runtime bounds when the stabilizer extent is replaced by the fermionic Gaussian extent. Because of the generality of this adaptation procedure – it being applicable to a variety of simulation algorithms both for strong and weak simulation – we restrict our attention to the key substitutions.

Our algorithms apply to the efficient classical simulation of fermionic circuits of the following form, involving n fermions.

- (i) The initial state $\Psi^{(0)} = \Psi$ is a possibly non-Gaussian state Ψ . We assume that its fermionic Gaussian extent $\xi(\Psi)$ and a corresponding optimal decomposition into a superposition of Gaussian states is known. This is the case for example for any four-fermion state, or a tensor product of two four-fermion states, see Section 5. Alternatively, we may assume that an upper bound $\bar{\xi}(\Psi) \geq \xi(\Psi)$ and a corresponding decomposition of Ψ achieving this value is known: In this case, runtime upper bounds will depend on $\bar{\xi}(\Psi)$ instead of $\xi(\Psi)$.
- (ii) The computation proceeds in a sequence of timesteps. At each step $t \in [T]$, one of the following is performed:
 - (a) A Gaussian unitary U_R , $R \in \text{Gen}(O(2n))$ is applied to the state. Here the choice of R may depend (in an efficiently computable manner) on measurement results obtained previously. We will leave this dependence implicit and do not take it into account in our runtime estimates, as it will typically depend heavily on the circuit considered.
 - (b) An occupation number measurement, i.e., measurement of the operator $a_j^\dagger a_j$ for some $j \in [n]$ is performed, yielding a measurement outcome $s \in \{0, 1\}$ and a corresponding post-measurement state. The choice of the mode $j \in [n]$ to be measured may again depend (in an efficient manner) on the measurement outcomes already obtained.

We note that the restriction to the set of Gaussian unitaries associated with generators of $O(2n)$ in (iia) incurs no loss of generality at the cost of possibly increasing T by a factor of order $O(n^2)$ and an additive term in the runtime of order $O(n^3)$ since a decomposition of an arbitrary element $R \in O(2n)$ of the form (19) as a product of $L \leq O(n^2)$ generators can be found in time $O(n^3)$, see the discussion below Theorem 3.2.

The use of arbitrary initial states Ψ in (i) allows us to model, in particular, the application of certain “magic gates” using so-called gadgets. These can be realized by using non-Gaussian auxiliary states combined with Gaussian operations, see e.g., [12, 13]. Since all 1-, 2- and 3-fermion states are Gaussian [32], 4-fermion states provide the smallest non-trivial examples; these will also be our main focus in Section 5. We refer to e.g., [12, 13] for a discussion of these constructions.

We proceed as follows: In Section 4, we formulate in general terms how simulation algorithms for a model can be generalized to initial states that are superpositions: This follows known approaches for stabilizer circuits augmented by magic states. In Section 4.2 we review the relationship between the \mathcal{D} -extent and the \mathcal{D} -rank defined by a dictionary \mathcal{D} . In Section 4.3 we discuss fast algorithms for estimating norms of superpositions of dictionary states. In Section 4.4 we apply these constructions to our setup.

4.1 Extending simulation algorithms to superpositions

Here we discuss how to extend simulation algorithms for an efficiently simulable model $(\mathcal{D}, \mathcal{E}, \mathcal{M})$ in such a way that the resulting extended algorithms (`χ evolve`, `χ measureprob`, `χ postmeasure`) work with any initial state Ψ which is a superposition of χ elements of \mathcal{D} (i.e., has \mathcal{D} -rank bounded by $\chi_{\mathcal{D}}(\Psi) \leq \chi$). Our discussion is standard and is included only for the reader’s convenience: It follows that for stabilizer states as discussed in [18].

Recall that the dictionary \mathcal{D} is a set of states, \mathcal{E} a set of operations and \mathcal{M} a set of measurements. In addition to the subroutines `evolve`, `measureprob` and `postmeasure` for evolution and measurement associated with $(\mathcal{D}, \mathcal{E}, \mathcal{M})$, the construction discussed here requires an efficient algorithm `overlap` which computes inner products $\langle \Psi(d_1), \Psi(d_2) \rangle$ from

descriptions $(d_1, d_2) \in \text{Desc}_n^2$. This means that the description $d \in \text{Desc}_n$ of a state $\Psi(d)$ must include phase information. For Gaussian states, the covariance matrix formalism has to be extended as discussed in Section 3.

Our goal is to find classical simulation algorithms for circuits of the following form:

- (i) The initial state $\Psi^{(0)} = \Psi$ is a superposition of the form

$$\Psi = \sum_{j=1}^{\chi} \gamma_j \varphi_j .$$

of χ states $\{\varphi_j\}_{j=1}^{\chi} \subset \mathcal{D}$ with complex coefficients $\{\gamma_j\}_{j=1}^{\chi}$. We assume that this decomposition is explicitly provided as an input to the classical algorithm in the form of a χ -tuple $\{(\gamma_j, d_j)\}_{j=1}^{\chi}$, where d_j is an efficient classical descriptions of the state φ_j .

- (ii) In each timestep $t \in [T]$,

- (a) either an evolution operation $E \in \mathcal{E}$, or
(b) a measurement $M \in \mathcal{M}$

is applied to the state. We assume that corresponding efficient descriptions of E respectively M are given to the classical simulation algorithm.

The algorithms (`evolve`, `measureprob`, `postmeasure`, `overlap`) associated with the model $(\mathcal{D}, \mathcal{E}, \mathcal{M})$ then immediately give rise to algorithms (`χ evolve`, `χ measureprob`, `χ postmeasure`) for simulating a more general circuit: At each time step $t \in [T]$, the resulting algorithm maintains the data $\{\gamma_j^{(t)}, d_j^{(t)}\}_{j=1}^{\chi}$ describing the instantaneous state $\Psi^{(t)}$ after step t as a linear combination

$$\Psi^{(t)} = \sum_{j=1}^{\chi} \gamma_j^{(t)} \Psi(d_j^{(t)})$$

of vectors belonging to the dictionary \mathcal{D} , and the subroutines (`χ evolve`, `χ measureprob`, `χ postmeasure`) are used to successively update this description (respectively sample from corresponding measurement outcomes).

Before describing the extended routines `χ evolve`, `χ measureprob`, `χ postmeasure` in more detail, it is convenient to introduce a subroutine `χ norm` which takes as input a tuple $\{(\gamma_j, d_j)\}_{j=1}^{\chi} \in (\mathbb{C} \times \text{Desc}_n)^{\chi}$ and outputs the squared norm $\|\sum_{j=1}^{\chi} \gamma_j \Psi(d_j)\|^2$. It is clear that such a primitive can be realized naively by using the algorithm `overlap` for computing inner products. This naive implementation, which we refer to as `χ naivenorm`, requires time

$$\text{time}(\chi\text{naivenorm}) = \chi^2 \text{time}(\text{overlap}) .$$

Let us now describe the procedures `χ evolve`, `χ measureprob` and `χ postmeasure`, building on a (general) norm computation subroutine `χ norm`.

- (a) if an evolution operation $E \in \mathcal{E}$ with description d_E is applied at time t , then the description is updated by setting

$$\gamma_j^{(t)} = \gamma_j^{(t-1)} \quad \text{and} \quad d_j^{(t)} = \text{evolve}(d_E, d_j^{(t-1)}) \quad \text{for} \quad j \in [\chi] .$$

This defines the algorithm `χ evolve`. The runtime of this update is

$$\text{time}(\chi\text{evolve}) = \chi \cdot \text{time}(\text{evolve}) .$$

- (b) if a (projective) measurement $M = \{M_s\}_{s \in \mathcal{M}} \in \mathcal{M}$ with description d_M is applied to the state at time t , then the probability of obtaining $s \in \mathcal{M}$ is given by

$$p(s|\Psi^{(t-1)}) = \|M_s \Psi^{(t-1)}\|^2 = \left\| \sum_{j=1}^{\chi} \gamma_j^{(t-1)} \sqrt{p(s|\Psi_j^{(t-1)})} \Psi_j^{(t-1)}(M, s) \right\|^2 .$$

Here the probability $p(s|\Psi_j^{(t-1)}) = \|M_s \Psi_j^{(t-1)}\|^2 = \text{measureprob}(d_j^{(t-1)}, d_M, s)$ of obtaining outcome s when measuring $\Psi_j^{(t-1)}$ can be efficiently obtained from the description $d_j^{(t-1)}$ of $\Psi_j^{(t-1)}$ and the description d_M of M . (Summands j where the probability $p(s|\Psi_j^{(t-1)})$ vanishes can be omitted from this sum.) Similarly, a description $d_j(s) = \text{postmeasure}(d_j^{(t-1)}, d_M, s)$ of the (normalized) post-measurement state $\Psi_j^{(t-1)}(M, s) = \frac{1}{\sqrt{p(s|\Psi_j^{(t-1)})}} M_s \Psi_j^{(t-1)}$ (when measuring $\Psi_j^{(t-1)}$) can be obtained

efficiently. In particular, setting $\tilde{\gamma}_j = \gamma_j^{(t-1)} \sqrt{p(s|\Psi_j^{(t-1)})}$, we conclude that the outcome probability

$$p(s|\Psi^{(t-1)}) = \left\| \sum_{j=1}^{\chi} \tilde{\gamma}_j \Psi(d_j(s)) \right\|^2 \quad (63)$$

is the squared norm of a superposition of elements from \mathcal{D} . This expression (together with the norm computation routine χnorm) defines the algorithm $\chi\text{measureprob}$. In particular, given $\{\tilde{\gamma}_j, d_j(s), p(s|\Psi_j^{(t-1)})\}_{j=1}^{\chi}$, the probability $p(s|\Psi)$ can be evaluated (exactly) in runtime $\text{time}(\chi\text{norm})$. Since $\chi\text{measureprob}$ first has to compute the descriptions $d_j(s)$ of the post-measurement states $\Psi_j^{(t-1)}(M, s)$ and the probabilities $\{p(s|\Psi_j^{(t-1)})\}_{j=1}^{\chi}$, its runtime is

$$\text{time}(\chi\text{measureprob}) = \text{time}(\chi\text{norm}) + \chi \cdot (\text{time}(\text{measureprob}) + \text{time}(\text{postmeasure})) .$$

One can easily verify that the post-measurement state after time step t is given by

$$\Psi^{(t)} = \sum_{j=1}^{\chi} \gamma_j^{(t)} \Psi(d_j^{(t)}) ,$$

where

$$\gamma_j^{(t)} = \frac{\tilde{\gamma}_j}{\sqrt{p(s|\Psi^{(t-1)})}} \quad \text{and} \quad d_j^{(t)} = d_j(s) .$$

In particular, this means that (similarly as for $\chi\text{measureprob}$) we have an algorithm $\chi\text{postmeasure}$ which given $\{(\gamma_j, d_j^{(t-1)})\}_{j=1}^{\chi}$ and $p(s|\Psi^{(t-1)})$, computes a description of the post-measurement state in time

$$\text{time}(\chi\text{postmeasure}) = \chi \cdot (\text{time}(\text{postmeasure}) + \text{time}(\text{measureprob})) .$$

Given the ability to compute $p(s|\Psi)$ and assuming, e.g., that the number $|\mathcal{M}|$ of measurement outcomes is constant, one can then sample from this distribution (when the goal is to perform weak simulation) to get an outcome $s \in \mathcal{M}$.

Using the naive algorithm $\chi\text{naivenorm}$ for χnorm gives runtimes

$$\begin{aligned}\text{time}(\chi\text{evolve}) &= \chi \cdot \text{time}(\text{evolve}) \\ \text{time}(\chi\text{measureprob}) &= \chi^2 \cdot \text{time}(\text{overlap}) + \chi \cdot (\text{time}(\text{postmeasure}) + \text{time}(\text{measureprob})) . \\ \text{time}(\chi\text{postmeasure}) &= \chi \cdot (\text{time}(\text{postmeasure}) + \text{time}(\text{measureprob}))\end{aligned}\tag{64}$$

As a function of χ , this is dominated by the computation of the squared norm (63) in $\chi\text{measureprob}$ which takes time $O(\chi^2)$.

4.2 Sparsification: Relating \mathcal{D} -extent to approximate \mathcal{D} -rank

Algorithms whose complexity depends on the \mathcal{D} -extent $\xi_{\mathcal{D}}(\Psi)$ instead of the (exact) \mathcal{D} -rank $\chi_{\mathcal{D}}(\Psi)$ (see Eq. (3)) of the initial state Ψ can be obtained as follows. The idea consists in replacing Ψ by a state $\tilde{\Psi}$ which is δ -close to Ψ and has bounded \mathcal{D} -rank. More precisely, it relies on the following result which connects the \mathcal{D} -extent $\xi_{\mathcal{D}}(\Psi)$ to the approximate \mathcal{D} -rank $\chi_{\mathcal{D}}^{\delta}(\Psi)$ defined in Eq. (4).

Theorem 4.1 (Theorem 1 in [18]). *Suppose $\Psi = \sum_{j=1}^m \gamma_j \varphi_j$ is a decomposition of a normalized vector Ψ into a superposition of elements $\{\varphi_j\}_{j=1}^m$ belonging to the dictionary \mathcal{D} . Then*

$$\chi_{\mathcal{D}}^{\delta}(\Psi) \leq 1 + \|\gamma\|_1^2 / \delta^2$$

where $\|\gamma\|_1 = \sum_{j=1}^m |\gamma_j|$ is the 1-norm of γ . In particular, we have the relationship

$$\chi_{\mathcal{D}}^{\delta}(\Psi) \leq 1 + \xi_{\mathcal{D}}(\Psi) / \delta^2 .$$

In [18], this result was established for the dictionary $\mathcal{D} = \text{STAB}_n$ consisting of n -qubit stabilizer states. Inspection of the proof immediately shows that the statement is applicable to any dictionary \mathcal{D} (independently of, e.g., whether or not it is finite). In particular, Theorem 4.1 implies that in runtime upper bounds, the quantity $\chi_{\mathcal{D}}$ can always be replaced by (the potentially much smaller quantity) $\xi_{\mathcal{D}}(\Psi) / \delta^2$, at the cost of introducing an $O(\delta)$ -error in L^1 -distance in the sampled distribution. For example, using the naive norm estimation algorithm (i.e., inserting into (64)), this gives a quadratic scaling (for computing output probabilities) in $\xi_{\mathcal{D}}(\Psi)$. Note that here we are assuming that a decomposition of Ψ with squared L^1 -norm $\|\gamma\|_1^2$ of coefficients achieving $\xi_{\mathcal{D}}(\Psi)$ is given.

4.3 Fast norm estimation and approximate simulation

In pioneering work, Bravyi and Gosset [17] improved upon the sketched algorithm in the case of stabilizer states. This was achieved by replacing the $O(\chi^2)$ -runtime (naive) estimation algorithm $\chi\text{naivenorm}$ for the norm of a superposition of stabilizer states by a probabilistic algorithm $\chi\text{fastnorm}$. With success probability at least $1 - p_f$, the algorithm $\chi\text{fastnorm}$ provides an estimate \hat{N} of the squared norm $N = \|\Psi\|^2$ of a superposition $\Psi = \sum_{j=1}^X \gamma_j \varphi_j$ of n -qubit stabilizer states $\{\varphi_j\}_{j=1}^X$ with multiplicative error ϵ (i.e., $\hat{N} \in [(1 - \epsilon)N, (1 + \epsilon)N]$), and has runtime $O(\chi \cdot n^3 \epsilon^{-2} p_f^{-1})$. The key observation underlying the algorithm is the fact that the norm of interest can be expressed as

$$N = 2^n \mathbb{E}_{\Theta} \left[|\langle \Theta, \Psi \rangle|^2 \right] ,\tag{65}$$

i.e., it is proportional to the expected (squared) overlap of Ψ with a state $|\Theta\rangle$ drawn uniformly at random from the set of n -qubit stabilizer states. Given $\{\gamma_j, \varphi_j\}_{j=1}^\chi$, this algorithm proceeds by taking $R = \lceil p_f^{-1} \epsilon^{-2} \rceil$ stabilizer states $\Theta_1, \dots, \Theta_R$ chosen uniformly from the set of all stabilizer states, and producing the estimate

$$\hat{N} = \frac{2^n}{R} \sum_{k=1}^R |\langle \Theta_k, \Psi \rangle|^2 \quad (66)$$

of N . Importantly, expression (66) can be computed from (the descriptions of) $\{\Theta_k\}_{k=1}^R$, $\{\varphi_j\}_{j=1}^\chi$ and the coefficients $\{\gamma_j\}_{j=1}^\chi$, using $\chi \cdot R$ calls of a subroutine `overlap` which computes the overlap of two stabilizer states (including phases). This is because each summand in (66) can be written as a sum

$$|\langle \Theta_k, \Psi \rangle| = \left| \sum_{j=1}^\chi \gamma_j \langle \Theta_k, \varphi_j \rangle \right| \quad (67)$$

of χ such products. The resulting runtime of this norm estimation algorithm is thus $O(\chi \cdot R \cdot \text{time}(\text{overlap}))$ which amounts to the claimed runtime $O(\chi \cdot n^3 \epsilon^{-2} p_f^{-1})$.

We note that a similar reasoning can be applied to any situation where the norm of a superposition of dictionary elements of interest can be expressed as in Eq. (65) as the expected overlap of the inner product of Ψ with a state Θ randomly chosen according to a suitable distribution over dictionary states. Specifically, as derived in Appendix B of [60] and discussed below (see Section 4.4), this is the case for the set of fermionic Gaussian states. The corresponding norm estimation algorithm then has a runtime of the form

$$\begin{aligned} \text{time}(\chi\text{fastnorm}) &= O(\chi \cdot R \cdot \text{time}(\text{overlap}) + R \cdot \text{time}(\text{samplestate})) \\ &= O(p_f^{-1} \epsilon^{-2} (\chi \text{time}(\text{overlap}) + \text{time}(\text{samplestate}))) \end{aligned} \quad (68)$$

where `samplestate` is an algorithm producing a description of a state Θ drawn randomly from the appropriate distribution. Importantly, the runtime (68) is linear in χ , resulting in a linear dependence when replacing χ by the stabilizer extent $\xi_{\mathcal{D}}(\Psi)$ as discussed in Section 4.2.

Algorithms (`approx evolve`, `approx measureprob`, `approx postmeasure`) can now be obtained by using `χfastnorm` in place of `χnorm`. The algorithm `approx evolve` is identical to `χ evolve` since it does not involve norm computations. In contrast, `approx measureprob` is a probabilistic algorithm that can fail with probability p_f and both `approx measureprob` and `approx postmeasure` introduce an error (in the sampled distribution and the post-measurement state, respectively). This is because `χfastnorm` only estimates the norm of a superposition.

Finally, replacing χ by the \mathcal{D} -extent $\xi_{\mathcal{D}}(\Psi)$, see Section 4.2 results in a triple of approximate algorithms (`approx evolve`, `approx measureprob`, `approx postmeasure`) with parameters (ϵ, δ, p_f) describing the quality of approximation and failure probability as discussed

in Section 1.5. By construction, the runtimes of these algorithms are

$$\begin{aligned}
\text{time}(\text{approx}\text{evolve}) &= O\left(\frac{\xi_{\mathcal{D}}(\Psi)}{\delta^2} \text{time}(\text{evolve})\right) \\
\text{time}(\text{approx}\text{measure}\text{prob}) &= O\left(p_f^{-1} \epsilon^{-2} \left(\frac{\xi_{\mathcal{D}}(\Psi)}{\delta^2} \text{time}(\text{overlap}) + \text{time}(\text{sample}\text{state})\right)\right) \\
&\quad + O\left(\frac{\xi_{\mathcal{D}}(\Psi)}{\delta^2} (\text{time}(\text{post}\text{measure}) + \text{time}(\text{measure}\text{prob}))\right) \\
\text{time}(\text{approx}\text{post}\text{measure}) &= O\left(\frac{\xi_{\mathcal{D}}(\Psi)}{\delta^2} (\text{time}(\text{post}\text{measure}) + \text{time}(\text{sample}\text{state}))\right)
\end{aligned} \tag{69}$$

4.4 Fermionic linear optics with non-Gaussian initial states

The algorithms described above can be adapted in a straightforward manner to the problem of classically simulating fermionic linear optics with non-Gaussian initial states: We can simply use the efficient description of Gaussian states introduced in Section 3 and make use of the associated procedures `overlap`, `evolve` as well as `measureprob` and `postmeasure`. In particular, observe that combining Eq. (64) with the runtimes $O(n^3)$ for the algorithms `evolve`, `postmeasure` and `overlap`, and $O(1)$ for `measureprob` (see Section 3) results in the runtimes given in Table 3 for exact simulation.

To get a linear scaling in the Gaussian extent $\xi_{\mathcal{G}_n}(\Psi)$ of the initial state (for approximate simulation), the naive norm estimation needs to be replaced. A fast norm estimation scheme for superpositions of fermionic Gaussian has been described in Appendix C of Ref. [60]: Consider the following probabilistic process defined for a superposition $\Psi = \sum_{j=1}^X \gamma_j \varphi_j$, $\varphi_j \in \mathcal{G}_n$, $\gamma_j \in \mathbb{C}$ of n -mode fermionic Gaussian states:

- (i) Sample K random Gaussian states $\{\Theta_k\}_{k=1}^K$ independently and identically from the distribution induced by picking a permutation $\pi \in S_{2n}$ and a string $y \in \{0, 1\}^n$ uniformly at random and outputting

$$|\Theta(\pi, y)\rangle = U_{R_\pi} |y\rangle.$$

Here $R_\pi = O(2n)$ is a permutation matrix specified by an element $\pi \in S_{2n}$ and $y \in \{0, 1\}^n$.

- (ii) Set

$$\hat{N} = \frac{1}{K} \sum_{k=1}^K 2^n |\langle \Theta_k, \Psi \rangle|^2. \tag{70}$$

The following was shown in [60].

Lemma 4.2 (Lemma 10 in Ref. [60]). *For any $p_f \in [0, 1]$ and $\epsilon > 0$, consider the probabilistic process described above with the choice $K = \lceil 2\sqrt{n}\epsilon^{-2}p_f^{-1} \rceil$. Then the random variable \hat{N} satisfies*

$$(1 - \epsilon)\|\Psi\|^2 \leq \hat{N} \leq (1 + \epsilon)\|\Psi\|^2.$$

with probability at least $1 - p_f$.

A description of a state proportional to Θ_k can be computed from the associated pair $(\pi_k, y_k) \in S_{2n} \times \{0, 1\}^n$ using the subroutine `describe`, for each $k \in [K]$, as follows (see Fig. 15 for pseudocode for the algorithm). The covariance matrix $\Gamma_k = R_{\pi_k} \Gamma(|y\rangle) R_{\pi_k}^\dagger$ of such a state can be computed in time $O(n^3)$ from (π_k, y) , and applying `describe` to Γ_k gives the desired description. We note that any such state can be used in place of Θ_k since the expression (70) (and, in particular, (67)) does not depend on the global phase of Θ_k . With the definition (70), it follows that the probabilistic process described here can be simulated in time given by Eq. (68) using K calls to the subroutine `samplestate` shown in Fig. 15, and subsequent use of `overlap` to compute the empirical average (70).

```

1: function samplestate
2:    $y \leftarrow$  uniform random string in  $\{0, 1\}^n$ 
3:    $\pi \leftarrow$  uniform random permutation in  $S_{2n}$ 
4:    $R_\pi \in O(2n)$ 
5:   for  $j \in [2n]$  do                                 $\triangleright$  compute the permutation matrix  $R_\pi$ 
6:      $R_\pi[j] \leftarrow e_{\pi(j)}$ 
7:    $\Gamma \leftarrow R_\pi \Gamma(|y\rangle) R_\pi^\dagger$            $\triangleright$  compute the covariance matrix of  $|\Theta(\pi, y)\rangle = R_\pi |y\rangle$ 
8:   return describe( $\Gamma$ )                                $\triangleright$  output a description of  $|\Theta(\pi, y)\rangle$ 

```

Figure 15: The algorithm `samplestate` outputs a classical description of a state $|\Theta(\pi, y)\rangle = R_\pi |y\rangle$ where $\pi \in S_{2n}$ and $y \in \{0, 1\}^n$ are taken uniformly at random.

Because the runtimes of `describe` and `overlap` are both upper bounded by $O(n^3)$, this leads to an overall runtime of $O\left(n^{7/2} \epsilon^{-2} p_f^{-1} \chi\right)$ of this algorithm for computing the estimate \hat{N} of $\|\Psi\|^2$. We note this conclusion about the runtime was also reached in Ref. [60], although the issue of a potential lack of a phase reference applicable throughout the computation was not considered there. This issue is resolved by our description of Gaussian states, see Section 3.

Combing this algorithm with the runtimes given in Eq. (69) and with the runtimes $O(n^3)$ for the algorithms `evolve`, `postmeasure` and `overlap`, and $O(1)$ for `measureprob` (see Section 3) gives runtimes claimed in Table 4 for the algorithms `approx-evolve`, `approx-measureprob` and `approx-postmeasure`.

To give an idea of the feasibility of running these algorithms, let us compare to [17], where Clifford circuits on $n = 40$ qubits with up to $t = 48$ T -gates were simulated by means of an algorithm relying on the stabilizer formalism augmented with stabilizer-decompositions of magic states. These simulations were performed using MATLAB on a laptop with a 2.6GHz Intel i5 Dual Core CPU. Runtimes of individual building blocks are given in [17, Table I] and are of the order of tens of milliseconds even for 100 qubits. The algorithms of [17] have a scaling which is linear in the approximate (stabilizer) rank. Since this quantity scales exponentially in the number of copies of a single magic state, this dominates the runtime, which includes additional factors that are polynomial in the number of qubits and gates. In more detail, the authors of [17] consider the non-stabilizer state

$$|A\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\pi/4}|1\rangle).$$

A copy of this (magic) state can be used together with stabilizer operations to implement the (non-Clifford) T -gate [11]. The approximate stabilizer rank (which is defined in terms

of fidelity rather than distance in [17]) of t copies of this state is shown to be bounded by

$$\tilde{\chi}^\delta(|A\rangle^{\otimes t}) = O(2^{\gamma t} \delta^{-1}) \quad \text{where} \quad \gamma \approx 0.228 . \quad (71)$$

Now consider our simulation procedure: Here the dependence of the runtime is linear in the Gaussian extent $\xi_{\mathcal{G}_n}$ and thus again essentially linear in the δ -approximate Gaussian rank $\chi_{\mathcal{G}_n}^\delta$ (see Theorem 4.1), and this quantity dominates the runtime (which includes polynomial factors in the number of fermionic modes and gates). Thus our simulation algorithms have a runtime comparable to that of [17] when substituting the fermionic Gaussian extent of the initial state with the approximate stabilizer rank (respectively the stabilizer extent) of the initial state in the Clifford setting.

To make this more concrete, consider the state

$$|a_8\rangle = \frac{1}{\sqrt{2}} (|0000\rangle + |1111\rangle) , \quad (72)$$

which is a magic state for Gaussian computations: As shown in [12, Lemma 1], the non-Gaussian unitary (magic gate) $U_{\text{magic}} = \exp(i\pi c_1 c_2 c_3 c_4)$ can be implemented with Gaussian operations and a single copy of this state. The state $|a_8\rangle$ has Gaussian extent

$$\xi_{\mathcal{G}_n}(|a_8\rangle) = 2 \quad (73)$$

and thus we have

$$\xi_{\mathcal{G}_n}(|a_8\rangle^{\otimes t'}) \leq 2^{t'} \quad (74)$$

for t' copies of $|a_8\rangle$. (We note that in fact, the results of [49] imply that equality holds in (74).) Eq. (73) can be shown as follows: The upper bound $\xi_{\mathcal{G}_n}(|a_8\rangle) \leq 2$ is a direct consequence of the decomposition given in (72) into a sum of the Gaussian states $|0000\rangle$ and $|1111\rangle$. The dual formulation of the extent implies that

$$\xi_{\mathcal{G}_n}(|\Psi\rangle) \geq \frac{1}{F_{\mathcal{G}_n}(|\Psi\rangle)} \quad \text{for any state } \Psi \in \mathcal{H}_n , \quad (75)$$

see (98) with the choice $y = |\Psi\rangle/\sqrt{F_{\mathcal{G}_n}(|\Psi\rangle)}$. Using the fact that $F_{\mathcal{G}_n}(|a_8\rangle) = 1/2$ (see [50] – the inequality $F_{\mathcal{G}_n}(|a_8\rangle) \leq 1/2$ which is needed here was previously shown in [32]) and (75) gives the lower bound $\xi_{\mathcal{G}_n}(|a_8\rangle) \geq 2$, establishing (73).

Comparing the exponents in (71) and (74), we can give the following rough estimate: we expect that the cost (runtime) of simulating $t = 48$ T -gates in the stabilizer framework is comparable with the cost of simulation of roughly $t' \approx 10$ gates U_{magic} in the fermionic context. Similar estimates apply to other non-Gaussian operations such that the SWAP gate, a magic gate in the context of fermionic Gaussian computation. The latter can be implemented using a certain non-Gaussian state as shown in [13].

4.5 Efficient additive-error strong simulation

In a different direction of generalization, building upon the work [17] and making innovative use of a concentration inequality by Hayes [62] for vector martingales, Ref. [27] gives a randomized algorithm which, for a state Ψ obtained by applying n -qubit Clifford gates and t (non-Clifford) T -gates to $|0\rangle^{\otimes n}$, provides an additive-error estimate $\hat{p}(x)$ for the probability $p(x) = \|(\langle x| \otimes I^{\otimes(n-a)})|\Psi\rangle\|^2$ of observing a qubits in the state $|x\rangle$, with $x \in \{0, 1\}^a$. The algorithm is based on a procedure by which the probability $p(x)$ of

interest is expressed in terms of the squared norm $\left\| \left(\langle 0 |^{\otimes t-r} \otimes I^{\otimes r} \right) W |\Psi\rangle \right\|^2$ of a partially projected state, where Ψ is a tensor product of t non-stabilizer single-qubit states (arising from gadgetization), W a certain Clifford unitary, and r a circuit-dependent integer. The failure probability of the constructed algorithm is then upper bounded (see [27, Theorem 3]) by an expression depending on $p(x)$, the error ϵ , the stabilizer rank $\xi_{\text{STAB}_n}(\Psi)$ (taking the role of χ) of the product state Ψ , as well as two additional parameters than be chosen freely, but enter into the (polynomial) runtime estimate.

The described method of adapting fast algorithms for simulating Clifford circuits with non-stabilizer initial states can be applied in a similar manner to this algorithm, since this also reduces to computing inner products (including phases) between Gaussian states.

5 Multiplicativity of the Gaussian fidelity for 4 fermions

The main result of this section is a proof that the fermionic Gaussian fidelity is multiplicative for the tensor product of any two positive-parity 4-fermion states.

We begin in Section 5.1 by laying out some specific properties of 4-fermion states. We discuss a Gaussianity condition specific to 4-fermion states [58] and we write an explicit expression for any 4-fermion state as a superposition of two orthogonal (Gaussian) states. This was first introduced in Refs. [56, 58]. In Section 5.2 we establish properties of the fermionic Gaussian fidelity for 4-fermion states which are subsequently used in Section 5.3 to prove that the fermionic Gaussian fidelity is multiplicative for the tensor product of any two 4-fermion states.

5.1 Four-fermion Gaussian and non-Gaussian states

Key to our considerations is a certain antiunitary map θ acting on \mathcal{H}_+^4 , the positive-parity subspace of 4 fermions spanned by $\{|x\rangle\}_{x \in \{0,1\}_+^4}$. It is defined by its action

$$\theta |x\rangle = (-1)^{\vartheta(x)} |\bar{x}\rangle, \quad (76)$$

for $x \in \{0,1\}_+^4$, on basis states (antilinearly extended to all of \mathcal{H}_+^4), where $\vartheta(x) = x_1 + x_3 \pmod 2 = x_2 + x_4 \pmod 2 = \vartheta(\bar{x})$. Here $\bar{x} = (\bar{x}_1, \dots, \bar{x}_n)$ is obtained by flipping each bit of x . The relevant properties of this map are the following. We note that the following statement has been given in [58, Eq. (9)], along with a negative-parity version.

Lemma 5.1 ([58]). *A state $\Psi \in \mathcal{H}_+^4$ is Gaussian if and only if*

$$\langle \Psi, \theta \Psi \rangle = 0.$$

Proof. This follows from the Gaussianity criterion given in Lemma 2.3. We give the proof in Appendix A. \square

Lemma 5.2. *We have $\theta c_j c_k = c_j c_k \theta$ for all $j, k \in [8]$.*

Proof. See Appendix B. \square

The following result was first shown in Ref. [56].

Lemma 5.3 ([56]). *Let $\Psi \in \mathcal{H}_+^4$ be a unit vector. Then there are two orthogonal unit vectors $\Psi_1, \Psi_2 \in \mathcal{H}_+^4$, $\varphi \in [0, 2\pi)$ and $a \in [0, 1/\sqrt{2}]$ such that*

$$\theta \Psi_j = \Psi_j \quad \text{for } j \in [2] \quad (77)$$

and

$$\Psi = e^{i\varphi} \left(\sqrt{1-a^2} \Psi_1 + ia \Psi_2 \right) . \quad (78)$$

Proof. We first argue that it suffices to consider the case where Ψ satisfies

$$\langle \Psi, \theta \Psi \rangle \in \mathbb{R} . \quad (79)$$

This is because

$$\langle (e^{i\varphi} \Psi), \theta (e^{i\varphi} \Psi) \rangle = e^{-2i\varphi} \langle \Psi, \theta \Psi \rangle \quad \text{for every } \varphi \in [0, 2\pi) ,$$

which implies that (79) can be ensured by replacing Ψ with $e^{i\varphi} \Psi$ for a suitably chosen $\varphi \in [0, 2\pi)$.

Let Ψ be such that (79) holds. We define “real” and “imaginary” parts of Ψ by the expressions

$$\begin{aligned} \Psi_R &= \frac{1}{2} (\Psi + \theta \Psi) \\ \Psi_I &= \frac{1}{2i} (\Psi - \theta \Psi) . \end{aligned}$$

It follows immediately from this definition that

$$\Psi = \Psi_R + i \Psi_I$$

and

$$\begin{aligned} \theta \Psi_R &= \Psi_R \\ \theta \Psi_I &= \Psi_I . \end{aligned}$$

because θ is antiunitary and an involution. Furthermore, Eq. (79) implies that the vectors Ψ_R, Ψ_I are orthogonal: We have

$$\begin{aligned} 4i \langle \Psi_R, \Psi_I \rangle &= \langle \Psi + \theta \Psi, \Psi - \theta \Psi \rangle \\ &= \|\Psi\|^2 - \|\theta \Psi\|^2 + \langle \theta \Psi, \Psi \rangle - \langle \Psi, \theta \Psi \rangle \\ &= -2i \operatorname{Im} \langle \Psi, \theta \Psi \rangle \\ &= 0 \end{aligned}$$

where we used that θ is an antiunitary in the first step, and assumption (79) in the last step. The claim now follows by setting

$$(a, \Psi_1, \Psi_2) = \begin{cases} \left(\|\Psi_I\|, \frac{\Psi_R}{\|\Psi_R\|}, \frac{\Psi_I}{\|\Psi_I\|} \right) & \text{if } \|\Psi_I\| \leq 1/\sqrt{2} \\ \left(\|\Psi_R\|, \frac{\Psi_I}{\|\Psi_I\|}, \frac{\Psi_R}{\|\Psi_R\|} \right) & \text{otherwise} . \end{cases}$$

□

Theorem 5.4 ([56, 58]). *Let $\Psi \in \mathcal{H}_+^4$ be an arbitrary unit vector. Then there are a Gaussian pure state $\Psi_g \in \mathcal{G}_+^+$, $\varphi \in [0, 2\pi)$ and $f \in [1/2, 1]$ such that the state $\theta \Psi_g$ is Gaussian and orthogonal to Ψ_g and*

$$\Psi = e^{i\varphi} \left(\sqrt{f} \Psi_g + \sqrt{1-f} \theta \Psi_g \right) . \quad (80)$$

The triple (Ψ_g, φ, f) is uniquely defined by Ψ , i.e., a function of Ψ . Furthermore, the quantity $f = f(\Psi)$ is invariant under the action of Gaussian unitaries associated with special orthogonal rotations: We have

$$f(U\Psi) = f(\Psi) \quad \text{for any Gaussian unitary } U = U_R \text{ with } R \in SO(2n) . \quad (81)$$

Proof. Let $\Psi \in \mathcal{H}_+^4$ be an arbitrary unit vector. Let $\Psi_1, \Psi_2 \in \mathcal{H}_+^4$, $\varphi \in [0, 2\pi)$ and $a \in [0, 1]$ be as in Lemma 5.3. Define

$$\Psi_g^\pm = \frac{1}{\sqrt{2}} (\Psi_1 \pm i\Psi_2) .$$

Then

$$\theta\Psi_g^- = \Psi_g^+ \tag{82}$$

because of property (77). It follows that

$$\begin{aligned} \langle \Psi_g^-, \Psi_g^+ \rangle &= \langle \Psi_g^-, \theta\Psi_g^- \rangle \\ &= \frac{1}{2} \left(\|\Psi_1\|^2 + 2i\operatorname{Re}\langle \Psi_1, \Psi_2 \rangle - \|\Psi_2\|^2 \right) \\ &= 0 \end{aligned}$$

since Ψ_1 and Ψ_2 are orthogonal unit vectors.

Using (82) and the orthogonality of Ψ_g^-, Ψ_g^+ implies that

$$\langle \Psi_g^+, \theta\Psi_g^+ \rangle = \langle \Psi_g^+, \Psi_g^- \rangle = 0$$

and similarly (because θ is an involution)

$$\langle \Psi_g^-, \theta\Psi_g^- \rangle = \langle \Psi_g^-, \Psi_g^+ \rangle = 0 .$$

According to the Gaussianity criterion in Lemma 5.1, we conclude that both Ψ_g^+ and Ψ_g^- are Gaussian.

Rewriting Eq. (78) by expressing (Ψ_1, Ψ_2) in terms of (Ψ_g^+, Ψ_g^-) gives

$$\Psi = e^{i\varphi} \left(\sqrt{f}\Psi_g^- + \sqrt{1-f}\Psi_g^+ \right) \quad \text{where} \quad f = \frac{1}{2} + a\sqrt{1-a^2} .$$

The claim (80) now follows with (82).

It remains to show property (81) of the function f . This follows immediately from the fact that the antiunitary θ commutes with all quadratic monomials $c_j c_k$ of Majorana operators (see Lemma 5.2), and hence with any Gaussian unitary $U = U_R$ with $R \in SO(2n)$, i.e., $U\theta = \theta U$. Retracing the steps of the proof, it is easy to check that if (Ψ_1, Ψ_2) are the states of Lemma 5.3, and Ψ_g the state in expression (80) for Ψ , then the corresponding states (Ψ'_1, Ψ'_2) and Ψ'_g for the state $\Psi' = U\Psi$ are given by $\Psi'_j = U\Psi_j$ for $j \in [2]$ and $\Psi'_g = U\Psi_g$, respectively. This implies the claim. \square

5.2 The Gaussian fidelity for 4-fermion states

For a subset $E \subset \{0, 1\}^4$, we define $\overline{E} = \{\bar{x} \mid x \in E\}$. We also write $\Pi_E = \sum_{x \in E} |x\rangle\langle x|$ for the projection onto the span of $\{|x\rangle\}_{x \in E}$.

Lemma 5.5. *Let $E \subset \{0, 1\}_+^4$, $|E| = 4$ be a subset of even-weight strings such that $E \cup \overline{E} = \{0, 1\}_+^4$. Let $f(\Psi) \in [1/2, 1]$ be defined as in Theorem 5.4. Then*

$$\|\Pi_E \Psi\|^2 \leq f(\Psi) \quad \text{for any unit vector} \quad \Psi \in \mathcal{H}_+^4 . \tag{83}$$

Proof. Let $f = f(\Psi) \in [1/2, 1]$, $\varphi \in [0, 2\pi)$ and $\Psi_g \in \mathcal{G}_n^+$ be as in Theorem 5.4 such that

$$\Psi = e^{i\varphi} \left(\sqrt{f} \Psi_g + \sqrt{1-f} \theta \Psi_g \right). \quad (84)$$

We define

$$\begin{aligned} \alpha_x &= \langle x, \Psi_g \rangle \\ \beta_x &= \langle x, \theta \Psi_g \rangle \end{aligned} \quad \text{for every } x \in E.$$

We claim that we have the identities

$$\sum_{x \in E} (|\alpha_x|^2 + |\beta_x|^2) = 1 \quad (85)$$

$$\sum_{x \in E} \overline{\alpha_x} \beta_x = 0. \quad (86)$$

Observe that these two identities immediately imply (83): Using expression (84), we have

$$\begin{aligned} \|\Pi_E \Psi\|^2 &= \sum_{x \in E} |\langle x, \Psi \rangle|^2 \\ &= \sum_{x \in E} |\sqrt{f} \alpha_x + \sqrt{1-f} \beta_x|^2 \\ &= \left\| \sqrt{f} \vec{\alpha} + \sqrt{1-f} \vec{\beta} \right\|^2 \end{aligned} \quad (87)$$

where we defined the vectors $\vec{\alpha} = (\alpha_x)_{x \in E}$, $\vec{\beta} = (\beta_x)_{x \in E} \in \mathbb{C}^4$. Since (85) and (86) are equivalent to the statement that

$$\|\vec{\alpha}\|^2 + \|\vec{\beta}\|^2 = 1 \quad (88)$$

$$\langle \vec{\alpha}, \vec{\beta} \rangle = 0, \quad (89)$$

we obtain

$$\begin{aligned} \left\| \sqrt{f} \vec{\alpha} + \sqrt{1-f} \vec{\beta} \right\|^2 &= f \|\vec{\alpha}\|^2 + (1-f) \|\vec{\beta}\|^2 \\ &\leq \max\{f, 1-f\} \\ &= f \end{aligned} \quad (90)$$

by the Pythagorean theorem in \mathbb{C}^4 (using (89)) and by maximizing over (α, β) satisfying (88). Inserting (90) into (87) results in the upper bound (83) on $\|\Pi_E \Psi\|^2$.

It remains to prove the claimed identities (85) and (86). We argue that these are a consequence of the fact that Ψ_g is normalized and Gaussian, respectively.

Proof. Observe that by definition of the antiunitary θ , we have

$$\begin{aligned} \beta_x &= \langle x, \theta \Psi_g \rangle \\ &= (-1)^{\vartheta(\bar{x})} \overline{\langle \bar{x}, \Psi_g \rangle} \end{aligned} \quad \text{for all } x \in E.$$

In particular, this implies that

$$|\beta_x|^2 = |\langle \bar{x}, \Psi_g \rangle|^2 \quad \text{for every } x \in E. \quad (91)$$

Eq. (85) now follows from the fact that Ψ_g is normalized and positive-parity: we have

$$\begin{aligned} \sum_{x \in E} (|\alpha_x|^2 + |\beta_x|^2) &= \sum_{x \in E} (|\langle x, \Psi_g \rangle|^2 + |\langle \bar{x}, \Psi_g \rangle|^2) \\ &= \|\Psi_g\|^2 \\ &= 1 \end{aligned}$$

where we used the definition of α_x and (91) in the first step, and the assumption $E \cup \bar{E} = \{0, 1\}_+^4$ in the second identity.

Similarly, Eq. (86) is a consequence of the fact that Ψ_g is Gaussian: we have

$$\begin{aligned} \sum_{x \in \{0, 1\}_+^4} \bar{\alpha}_x \beta_x &= \sum_{x \in \{0, 1\}_+^4} \langle \Psi_g, x \rangle \langle x, \theta \Psi_g \rangle \\ &= \langle \Psi_g, \theta \Psi_g \rangle \\ &= 0 \end{aligned}$$

where we used the definition of α_x and β_x in the first step, the fact that $\Psi_g \in \mathcal{H}_+^4$ in the second step, and the characterization of Gaussianity from Lemma 5.1 in the last identity. \square

\square

Lemma 5.5 immediately implies the following expression for the fermionic Gaussian fidelity. We note that a more general expression for the ‘‘Gaussian fidelity’’ of a mixed state has previously been obtained in [58]. The proof for pure states given here is more elementary and illustrates the use of Lemma 5.5.

Theorem 5.6 (Fermionic Gaussian fidelity for 4-mode pure states [56, 58]). *Let $\Psi \in \mathcal{H}_+^4$ be a unit vector. Let $f(\Psi) \in [1/2, 1]$ be defined as in Theorem 5.4. Then*

$$F_{\mathcal{G}_4^+}(\Psi) = f(\Psi) .$$

Proof. Let $f = f(\Psi)$, $\varphi \in [0, 2\pi)$ and $\Psi_g \in \mathcal{G}_4^+$ be as in Theorem 5.4. Then we have

$$\begin{aligned} F_{\mathcal{G}_4^+}(\Psi) &\geq \left| \langle \Psi_g, e^{i\varphi} (\sqrt{f} \Psi_g + f \theta \Psi_g) \rangle \right|^2 \\ &= f \end{aligned}$$

since $\theta \Psi_g$ is orthogonal to Ψ_g . It thus suffices to show the upper bound

$$F_{\mathcal{G}_4^+}(\Psi) \leq f . \tag{92}$$

Let $\Phi_g \in \mathcal{G}_4^+$ be an arbitrary positive-parity Gaussian pure state. Then there is a Gaussian unitary $U = U_R$ with $R \in SO(2n)$ and a phase $\mu \in [0, 2\pi)$ such that $\Phi_g = e^{i\mu} U |0_F\rangle$. We will use any subset $E \subset \{0, 1\}_+^4$ of even-weight strings as in Lemma 5.5 with the additional property that $0000 \in E$, e.g., $E = \{0000, 1100, 1010, 1001\}$. Then $|0_F\rangle = \Pi_E |0_F\rangle$ is in the image of Π_E . It follows that

$$\begin{aligned} |\langle \Phi_g, \Psi \rangle| &= \left| \langle 0_F, U^\dagger \Psi \rangle \right| \\ &= \left| \langle \Pi_E 0_F, U^\dagger \Psi \rangle \right| \\ &= \left| \langle 0_F, \Pi_E U^\dagger \Psi \rangle \right| \\ &\leq \left\| \Pi_E U^\dagger \Psi \right\| \\ &\leq \sqrt{f(U^\dagger \Psi)} , \end{aligned}$$

where we used the Cauchy-Schwarz inequality in the penultimate step, and Lemma 5.5 applied to the state $U^\dagger\Psi$. Since $\Phi_g \in \mathcal{G}_4^+$ was arbitrary, the claimed inequality (92) follows by taking the square and using that $f(U^\dagger\Psi) = f(\Psi)$, see Eq. (81) of Theorem 5.4. \square

Combining Lemma 5.5 with Theorem 5.6 yields the following statement, which directly relates the weight of a state on certain subspaces to the fermionic Gaussian fidelity. It will be our main technical tool in the following.

Corollary 5.7. *Let $E \subset \{0, 1\}_+^4$, $|E| = 4$ be a subset of even-weight strings such that $E \cup \bar{E} = \{0, 1\}_+^4$. Then*

$$\|\Pi_E\Psi\|^2 \leq F_{\mathcal{G}_4^+}(\Psi) \quad \text{for any unit vector} \quad \Psi \in \mathcal{H}_+^4 .$$

5.3 Multiplicativity of the Gaussian fidelity for 4-fermion states

Here we prove that the fermionic Gaussian fidelity is multiplicative for 4-fermion states (see Theorem 5.10). We use two intermediate results stated as Lemmas 5.8 and 5.9. In Lemma 5.8, we bound the overlap of a tensor product of two (arbitrary) positive-parity pure states Ψ_A, Ψ_B with a state Φ written as a Schmidt decomposition of a bipartite fermionic pure state. In Theorem 5.10, this result is used to bound the fermionic Gaussian fidelity. More specifically, Lemma 5.9 is used to upper bound the Schmidt coefficients, giving the multiplicativity result for the Gaussian fidelity.

Lemma 5.8. *Let $\{m_x\}_{x \in \{0,1\}^4} \subset \mathbb{C}$ be arbitrary. Define*

$$|\Phi\rangle = \sum_{x \in \{0,1\}^4} m_x |x, x\rangle \in \mathcal{H}_+^8 . \quad (93)$$

Let $E \subset \{0, 1\}_+^4$, $|E| = 4$ be a subset of even-weight strings such that $E \cup \bar{E} = \{0, 1\}_+^4$. Then

$$|\langle \Phi, \Psi_A \tilde{\otimes} \Psi_B \rangle|^2 \leq F_{\mathcal{G}_4^+}(\Psi_A) F_{\mathcal{G}_4^+}(\Psi_B) \left(\max_{x \in E} |m_x| + \max_{y \in \bar{E}} |m_y| \right)^2$$

for all states $\Psi_A, \Psi_B \in \mathcal{H}_+^4$.

Proof. Because Ψ_A and Ψ_B are supported on \mathcal{H}_+^4 by assumption, we have by Eq. (31)

$$\begin{aligned} \langle \Phi, \Psi_A \tilde{\otimes} \Psi_B \rangle &= \sum_{x \in \{0,1\}_+^4} m_x (-1)^{|x|} \langle x, \Psi_A \rangle \langle x, \Psi_B \rangle \\ &= \sum_{x \in \{0,1\}_+^4} m_x e^{i\nu_x} \langle \Psi_A, x \rangle \langle x, \Psi_B \rangle \end{aligned} \quad (94)$$

where ν_x is defined by the identity

$$(-1)^{|x|} \langle x, \Psi_A \rangle = e^{i\nu_x} \langle \Psi_A, x \rangle \quad \text{for} \quad x \in \{0, 1\}_+^4 .$$

Defining the operator

$$M_\Omega = \sum_{x \in \Omega} m_x e^{i\nu_x} |x\rangle \langle x|$$

for any subset $\Omega \subset \{0, 1\}_+^4$, it follows from Eq. (94) that

$$\langle \Phi, \Psi_A \tilde{\otimes} \Psi_B \rangle = \langle \Psi_A, M_E \Psi_B \rangle + \langle \Psi_A, M_{\bar{E}} \Psi_B \rangle . \quad (95)$$

Since M_E is supported on $\text{span}\{|x\rangle\}_{x \in E}$, we have $\langle \Psi_A, M_E \Psi_B \rangle = \langle \Pi_E \Psi_A, M_E \Pi_E \Psi_B \rangle$. With the Cauchy-Schwarz inequality and the definition of the operator norm $\|M_E\|$ we thus get

$$\begin{aligned} |\langle \Psi_A, M_E \Psi_B \rangle| &\leq \|\Pi_E \Psi_A\| \cdot \|\Psi_E \Psi_B\| \cdot \|M_E\| \\ &\leq \sqrt{F_{\mathcal{G}_4^+}(\Psi_A)} \cdot \sqrt{F_{\mathcal{G}_4^+}(\Psi_B)} \cdot \|M_E\| , \end{aligned} \quad (96)$$

where we applied Corollary 5.7. Identical reasoning applies to \bar{E} and yields the inequality

$$|\langle \Psi_A, M_{\bar{E}} \Psi_B \rangle| \leq \sqrt{F_{\mathcal{G}_4^+}(\Psi_A)} \cdot \sqrt{F_{\mathcal{G}_4^+}(\Psi_B)} \cdot \|M_{\bar{E}}\| . \quad (97)$$

Combining Eqs. (96), (97) with Eq. (95), we conclude that

$$\begin{aligned} |\langle \Phi, \Psi_A \tilde{\otimes} \Psi_B \rangle| &\leq |\langle \Psi_A, M_E \Psi_B \rangle| + |\langle \Psi_A, M_{\bar{E}} \Psi_B \rangle| \\ &\leq \sqrt{F_{\mathcal{G}_4^+}(\Psi_A) F_{\mathcal{G}_4^+}(\Psi_B)} (\|M_E\| + \|M_{\bar{E}}\|) . \end{aligned}$$

Taking the square and observing that

$$\|M_\Omega\| = \max_{x \in \Omega} |m_x e^{i\nu_x}| = \max_{x \in \Omega} |m_x|$$

gives the claim. \square

The following lemma will be useful to prove the main theorem.

Lemma 5.9. *The function*

$$f(\theta, x) = \prod_{j=1}^4 (\cos \theta_j)^{1-x_j} (\sin \theta_j)^{x_j} \quad \text{for} \quad \theta = (\theta_1, \dots, \theta_4) \in \mathbb{R}^4 \text{ and } x \in \{0, 1\}^4$$

satisfies

$$|f(\theta, x)| + |f(\theta, y)| \leq 1$$

for all $\theta \in \mathbb{R}^4$ and $x, y \in \{0, 1\}^4$ with x, y even-weight and $x \neq y$.

Proof. Because x, y have even and different weight, it suffices to consider two cases, namely with $|x - y| \in \{2, 4\}$.

Consider first the case where $|x - y| = 2$. Without loss of generality, assume that $(x_1, x_2) = (y_1, y_2)$, $x_3 \neq y_3$, $x_4 \neq y_4$. Since translating θ by $-\pi/2$ interchanges $|\sin \theta|$ and $|\cos \theta|$, it suffices to show the claim for $x = (0, 0, 0, 0)$ and $y = (0, 0, 1, 1)$. In this case we have

$$\begin{aligned} |f(\theta, x)| + |f(\theta, y)| &= |\cos \theta_1 \cos \theta_2 \cos \theta_3 \cos \theta_4| + |\cos \theta_1 \cos \theta_2 \sin \theta_3 \sin \theta_4| \\ &= |\cos \theta_1 \cos \theta_2| \cdot (|\cos \theta_3 \cos \theta_4| + |\sin \theta_3 \sin \theta_4|) \\ &\leq |\cos \theta_1 \cos \theta_2| \\ &\leq 1 , \end{aligned}$$

where the first inequality follows from the Cauchy-Schwarz inequality in \mathbb{R}^2 . Since $\theta \in \mathbb{R}^4$ was arbitrary, this concludes the proof for $|x - y| = 2$.

The proof for $|x - y| = 4$, i.e., $y = \bar{x}$ proceeds similarly. Again it suffices to show the claim for $x = (0, 0, 0, 0)$. In this case

$$\begin{aligned} |f(\theta, x)| + |f(\theta, y)| &= |\cos \theta_1 \cos \theta_2 \cos \theta_3 \cos \theta_4| + |\sin \theta_1 \sin \theta_2 \sin \theta_3 \sin \theta_4| \\ &\leq |\cos \theta_1 \cos \theta_2| + |\sin \theta_1 \sin \theta_2| \\ &\leq 1, \end{aligned}$$

where the first inequality follows from $|\cos \theta_3 \cos \theta_4| \leq 1$ and $|\sin \theta_3 \sin \theta_4| \leq 1$ and the second one from the Cauchy-Schwarz inequality. \square

Theorem 5.10 (Multiplicativity of the fermionic Gaussian fidelity for 4-mode pure states.). *Let \mathcal{H}_+^n be the set of pure n -fermion states with positive parity and let \mathcal{G}_n^+ be the set of pure n -fermion Gaussian states with positive parity. We have that*

$$F_{\mathcal{G}_8}(\Psi_A \tilde{\otimes} \Psi_B) = F_{\mathcal{G}_4}(\Psi_A) F_{\mathcal{G}_4}(\Psi_B) \quad \text{for all} \quad \Psi_A, \Psi_B \in \mathcal{H}_+^4.$$

Proof. We first observe that \mathcal{H}^8 is a direct sum of the four spaces $\mathcal{H}_+^4 \tilde{\otimes} \mathcal{H}_+^4$, $\mathcal{H}_+^4 \tilde{\otimes} \mathcal{H}_-^4$, $\mathcal{H}_-^4 \tilde{\otimes} \mathcal{H}_+^4$ and $\mathcal{H}_-^4 \tilde{\otimes} \mathcal{H}_-^4$. This is because states in these subspaces have different eigenvalues with respect to the corresponding parity operators on the factors (interpreted as Majorana monomials on \mathcal{H}^8 these are the monomials $c(1^8 0^8)$ and $c(0^8 1^8)$). It follows immediately that the overlap with a state of the form $\Psi_A \tilde{\otimes} \Psi_B \in \mathcal{H}_+^4 \tilde{\otimes} \mathcal{H}_+^4$ is maximized for a decomposition into states belonging to $\mathcal{H}_+^4 \tilde{\otimes} \mathcal{H}_+^4$ only. In particular, it follows that

$$F_{\mathcal{G}_8}(\Psi_A \tilde{\otimes} \Psi_B) = F_{\mathcal{G}_4^+}(\Psi_A \tilde{\otimes} \Psi_B) \quad \text{for all} \quad \Psi_A, \Psi_B \in \mathcal{H}_+^4$$

and by the same reasoning, we have

$$F_{\mathcal{G}_4}(\Psi) = F_{\mathcal{G}_4^+}(\Psi) \quad \text{for all} \quad \Psi \in \mathcal{H}_+^4.$$

We conclude that it suffices to show that

$$F_{\mathcal{G}_8^+}(\Psi_A \tilde{\otimes} \Psi_B) = F_{\mathcal{G}_4^+}(\Psi_A) F_{\mathcal{G}_4^+}(\Psi_B) \quad \text{for all} \quad \Psi_A, \Psi_B \in \mathcal{H}_+^4.$$

Let $\Psi_A, \Psi_B \in \mathcal{H}_+^4$ be arbitrary. The inequality $F_{\mathcal{G}_8^+}(\Psi_A \tilde{\otimes} \Psi_B) \geq F_{\mathcal{G}_4^+}(\Psi_A) F_{\mathcal{G}_4^+}(\Psi_B)$ follows trivially from the definition of fermionic Gaussian fidelity in Eq. (2), because $\mathcal{G}_4^+ \tilde{\otimes} \mathcal{G}_4^+ \subseteq \mathcal{G}_8^+$.

The inequality $F_{\mathcal{G}_8^+}(\Psi_A \tilde{\otimes} \Psi_B) \leq F_{\mathcal{G}_4^+}(\Psi_A) F_{\mathcal{G}_4^+}(\Psi_B)$ is a consequence of the Schmidt decomposition for fermionic states put forward in Ref. [50] and of Lemmas 5.8 and 5.9. According to Ref. [50], an arbitrary pure fermionic state $\Phi \in \mathcal{H}^n$ admits a Schmidt decomposition of the form

$$|\Phi\rangle = \sum_{x \in \{0,1\}^n} m_x |x, x\rangle$$

with

$$m_x = \prod_{j=1}^n (\cos \theta_j)^{1-x_j} (-\sin \theta_j)^{x_j} \quad \text{with} \quad \theta_j \in \mathbb{R} \text{ for } j \in [n].$$

With this definition of m_x for $n = 4$, an arbitrary state $\Phi \in \mathcal{G}_8^+$ can be written as in Eq. (93) and the conditions for Lemma 5.8 apply. We have

$$\begin{aligned} F_{\mathcal{G}_8^+}(\Psi_A \tilde{\otimes} \Psi_B) &= \max_{\Phi \in \mathcal{G}_8^+} |\langle \Phi, \Psi_A \otimes \Psi_B \rangle| \\ &\leq F_{\mathcal{G}_4^+}(\Psi_A) F_{\mathcal{G}_4^+}(\Psi_B) \left(\max_{x \in E} |m_x| + \max_{y \in \bar{E}} |m_y| \right)^2 \end{aligned}$$

where $E \subset \{0, 1\}_+^4$ with $|E| = 4$ a subset of even weight strings such that $E \cup \bar{E} = \{0, 1\}_+^4$. Notice that x, y have even-weight and that $x \neq y$ because E and \bar{E} are disjoint sets. Identifying m_x (whose dependence on $\theta \in \mathbb{R}^4$ is implicit) with $f(\theta, x)$ in Lemma 5.9 (apart from a minus sign that is not relevant because we take the absolute value) we have

$$\begin{aligned} F_{\mathcal{G}_8^+}(\Psi_A \tilde{\otimes} \Psi_B) &\leq F_{\mathcal{G}_4^+}(\Psi_A) F_{\mathcal{G}_4^+}(\Psi_B) \left(\max_{x \in E} |m_x| + \max_{y \in \bar{E}} |m_y| \right)^2 \\ &\leq F_{\mathcal{G}_4^+}(\Psi_A) F_{\mathcal{G}_4^+}(\Psi_B) , \end{aligned}$$

giving the claim. \square

6 Multiplicativity of \mathcal{D} -fidelity implies that of \mathcal{D} -extent

In this section, we show that multiplicativity of the \mathcal{D} -fidelity implies multiplicativity of the \mathcal{D} -extent. In Section 6.1 we prove this for finite dictionaries: This follows immediately from the fact that $F_{\mathcal{D}}(\Psi)$ and $\xi_{\mathcal{D}}(\Psi)$ are related by (convex programming) duality. In Section 6.2, we extend this results for infinite, i.e., continuously parameterized dictionaries. We achieve this extension by using (finite) ϵ -nets for the set of Gaussian states. Similar approaches have been applied in the signal processing context, see e.g., the work [63], which shows how to approximately solve atomic norm minimization problems for sparse recovery when the parameters indexing the dictionary lie in a small-dimensional space.

6.1 Multiplicativity for finite dictionaries

We will restrict our attention to finite dictionaries in this section. For $|\mathcal{D}| < \infty$, the \mathcal{D} -fidelity is related to the dual formulation of the \mathcal{D} -extent as (see [19, Eq. (3.2)] and [18, Theorem 4])

$$\xi_{\mathcal{D}}(\Psi) = \max_{y \in \mathcal{H}: F_{\mathcal{D}}(y) \leq 1} |\langle \Psi, y \rangle|^2 . \quad (98)$$

Let $\mathcal{H}_1, \mathcal{H}_2$ and \mathcal{H}_3 be a triple of Hilbert spaces. Let $\{\mathcal{D}_j\}_{j \in [3]}$ be a family of dictionaries, where $\mathcal{D}_j \subset \mathcal{H}^j$ for $j \in [3]$. We assume that

$$\mathcal{D}_1 \otimes \mathcal{D}_2 \subseteq \mathcal{D}_3 . \quad (99)$$

We are interested in the following two properties:

$$\begin{aligned} \text{Mult}^{\xi}(\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3) : \quad &\xi_{\mathcal{D}_3}(\Psi_1 \otimes \Psi_2) = \xi_{\mathcal{D}_1}(\Psi_1) \xi_{\mathcal{D}_2}(\Psi_2) \quad \text{for all } \Psi_j \in \mathcal{H}_j \text{ for } j \in [2] \\ \text{Mult}^F(\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3) : \quad &F_{\mathcal{D}_3}(\Psi_1 \otimes \Psi_2) = F_{\mathcal{D}_1}(\Psi_1) F_{\mathcal{D}_2}(\Psi_2) \quad \text{for all } \Psi_j \in \mathcal{H}_j \text{ for } j \in [2] . \end{aligned}$$

As an important example, let $n_j \in \mathbb{N}$ for $j \in [2]$, $n_3 = n_1 + n_2$, $\mathcal{H}_j = (\mathbb{C}^2)^{\otimes n_j}$ and let STAB_{n_j} be the set of stabilizer states on $(\mathbb{C}^2)^{\otimes n_j}$. Then $\text{Mult}^{\xi}(\text{STAB}_{n_1}, \text{STAB}_{n_2}, \text{STAB}_{n_1+n_2})$ does

not hold for certain (large) choices of n_1 and n_2 [19]. On the other hand, for $n_1, n_2 \leq 3$ the multiplicativity property $\text{Mult}^\xi(\text{STAB}_{n_1}, \text{STAB}_{n_1}, \text{STAB}_{n_1+n_2})$ holds (see Ref. [18, Proposition 1]). This was shown using that the stabilizer fidelity is multiplicative, i.e., $\text{Mult}^F(\text{STAB}_{n_1}, \text{STAB}_{n_1}, \text{STAB}_{n_1+n_2})$.

We claim the property $\text{Mult}^F(\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3)$ implies property $\text{Mult}^\xi(\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3)$.

Theorem 6.1. *Property $\text{Mult}^F(\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3)$ implies property $\text{Mult}^\xi(\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3)$.*

Proof. We clearly have

$$\xi_{\mathcal{D}_3}(\Psi_1 \otimes \Psi_2) \leq \xi_{\mathcal{D}_1}(\Psi_1)\xi_{\mathcal{D}_2}(\Psi_2) \quad (100)$$

for all $\Psi_1 \in \mathcal{H}_1$ and $\Psi_2 \in \mathcal{H}_2$ because of property (99) of the dictionaries $\{\mathcal{D}_j\}_{j=1}^3$ and the definition (5) of $\xi_{\mathcal{D}}$. To show the converse inequality, assume that $y_1 \in \mathcal{H}_1$, $y_2 \in \mathcal{H}_2$ are such that

$$F_{\mathcal{D}_j}(y_j) \leq 1 \quad \text{and} \quad \xi_{\mathcal{D}_j}(\Psi_j) = |\langle \Psi_j, y_j \rangle|^2 \quad \text{for } j \in [2].$$

Then

$$F_{\mathcal{D}_3}(y_1 \otimes y_2) = F_{\mathcal{D}_1}(y_1)F_{\mathcal{D}_2}(y_2) \leq 1$$

where we used the assumption that property $\text{Mult}^F(\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3)$ holds to obtain the equality. This implies that $y_1 \otimes y_2$ is a feasible point of the dual program for the quantity $\xi_{\mathcal{D}_3}(\Psi_1 \otimes \Psi_2)$, see Eq. (98). Thus

$$\begin{aligned} \xi_{\mathcal{D}_3}(\Psi_1 \otimes \Psi_2) &\geq |\langle \Psi_1 \otimes \Psi_2, y_1 \otimes y_2 \rangle|^2 \\ &= |\langle \Psi_1, y_1 \rangle|^2 \cdot |\langle \Psi_2, y_2 \rangle|^2 \\ &= \xi_{\mathcal{D}_1}(\Psi_1)\xi_{\mathcal{D}_2}(\Psi_2). \end{aligned} \quad (101)$$

Expression (101) together with Eq. (100) gives the claim. \square

6.2 Multiplicativity for infinite dictionaries

In this section, we extend the results of Section 6.2 to dictionaries \mathcal{D} that may contain infinitely many elements. Our strategy is to use an ϵ -net for $\mathcal{D} \in \mathcal{H}$ with a finite number of elements, we denote by \mathcal{D}^ϵ . We relate the extent and fidelity with respect to the dictionary \mathcal{D} to the extent and fidelity with respect to its net \mathcal{D}^ϵ (see Lemmas 6.2 and 6.3) to prove that multiplicativity of the \mathcal{D} -fidelity implies multiplicativity of the \mathcal{D} -extent in Theorem 6.6. This result is a generalization of Theorem 6.1 (that considered finite dictionaries) for (possibly) infinite dictionaries.

We will make use of the notion of ϵ -net to replace our infinite set \mathcal{D} by a finite set \mathcal{D}^ϵ . Let $\|\Psi\| = \sqrt{\langle \Psi, \Psi \rangle}$ for $\Psi \in \mathcal{H}$ denote the norm on \mathcal{H} . Let $\mathcal{D} \subset \mathcal{H}$ and let $\epsilon > 0$. Then a set $\mathcal{D}^\epsilon \subset \mathcal{H}$ is called an ϵ -net for \mathcal{D} if for any $\Psi \in \mathcal{D}$ there is some $\Phi \in \mathcal{D}^\epsilon$ such that $\|\Phi - \Psi\| \leq \epsilon$.

We are interested in the case where for every $\epsilon > 0$ there is a finite ϵ -net \mathcal{D}^ϵ for \mathcal{D} , with the additional property that $\mathcal{D}^\epsilon \subset \mathcal{D}$, i.e., the net consists of elements of \mathcal{D} . A sufficient condition for this being the case is that the subset $\mathcal{D} \subset \mathcal{H}$ is compact.

Lemma 6.2. *Let $\mathcal{D} \subset \mathcal{H}$ be a set of states. Assume that there is a finite ϵ -net \mathcal{D}^ϵ for \mathcal{D} such that $\mathcal{D}^\epsilon \subset \mathcal{D}$, for some $\epsilon > 0$. Assume further that \mathcal{D}^ϵ contains an orthonormal basis of \mathcal{H} . Let d be the dimension of \mathcal{H} . Then*

$$\xi_{\mathcal{D}}(\Psi) \leq \xi_{\mathcal{D}^\epsilon}(\Psi) \leq \xi_{\mathcal{D}}(\Psi) \left(1 + \sqrt{d}\epsilon\right)^2 \quad \text{for all } \Psi \in \mathcal{H}.$$

Proof. The first inequality follows immediately from the definition of $\xi_{\mathcal{D}}$ and from the assumption that $\mathcal{D}^\epsilon \subset \mathcal{D}$.

To prove the second inequality, let $\Psi \in \mathcal{H}$ be arbitrary. By definition of $\xi_{\mathcal{D}}(\Psi)$ as an infimum, we have the following: For every $m \in \mathbb{N}$, there exist $N(m) \in \mathbb{N}$, $\{\varphi_j(m)\}_{j=1}^{N(m)} \subset \mathcal{D}$ and $\{c_j(m)\}_{j=1}^{N(m)} \subset \mathbb{C}$ such that

$$\Psi = \sum_{j=1}^{N(m)} c_j(m) \varphi_j(m)$$

and

$$\|c(m)\|_1 < \sqrt{\xi_{\mathcal{D}}(\Psi)} + \frac{1}{m}. \quad (102)$$

Furthermore, we have

$$\|c(m)\|_1^2 \geq \xi_{\mathcal{D}}(\Psi) \quad \text{for any } N(m) \in \mathbb{N}, \{s_j\}_{j=1}^{N(m)} \subset \mathcal{D} \quad \text{and} \quad \{c_j(m)\}_{j=1}^{N(m)} \subset \mathbb{C}.$$

Fix such an $m \in \mathbb{N}$. Since \mathcal{D}^ϵ is an ϵ -net for \mathcal{D} , there is, for every $j \in [N(m)]$, an element $\varphi_j^\epsilon(m) \in \mathcal{D}^\epsilon$ and $\delta_j(m) \in \mathcal{H}$ such that

$$\varphi_j(m) = \varphi_j^\epsilon(m) + \delta_j(m) \quad \text{and} \quad \|\delta_j(m)\| \leq \epsilon.$$

It follows that

$$\Psi = \left(\sum_{j=1}^{N(m)} c_j(m) \varphi_j^\epsilon(m) \right) + \delta(m) \quad \text{where} \quad \delta(m) = \sum_{j=1}^{N(m)} c_j(m) \delta_j(m).$$

By the triangle inequality we have

$$\|\delta(m)\| \leq \epsilon \sum_{j=1}^{N(m)} |c_j(m)| = \epsilon \|c(m)\|_1. \quad (103)$$

Suppose $\{e_k\}_{k=1}^d$ is an orthonormal basis contained in \mathcal{D}^ϵ . Then we can expand

$$\delta(m) = \sum_{k=1}^d \alpha_k(m) e_k,$$

and it follows from (103) and the Cauchy-Schwarz inequality that

$$\begin{aligned} \|\alpha(m)\|_1 &\leq \sqrt{d} \|\alpha(m)\|_2 \\ &= \sqrt{d} \|\delta(m)\|_2 \quad \text{by the Cauchy-Schwarz inequality in } \mathbb{C}^d \\ &\leq \sqrt{d} \epsilon \|c(m)\|_1 \quad \text{because of Eq. (103)}. \end{aligned}$$

where $\|\cdot\|_2$ is the Euclidean norm in \mathbb{C}^d . In summary, we have

$$\Psi = \sum_{j=1}^{N(m)} c_j(m) \varphi_j^\epsilon(m) + \sum_{k=1}^d \alpha_k(m) e_k,$$

with $(c, \alpha) \in \mathbb{C}^{N(m)+d}$ satisfying

$$\begin{aligned} \|(c, \alpha)\|_1 &= \sum_{j=1}^{N(m)} |c_j(m)| + \sum_{k=1}^d |\alpha_k(m)| \\ &\leq \|c(m)\|_1 \cdot (1 + \sqrt{d}\epsilon) \\ &\leq \left(\sqrt{\xi_{\mathcal{D}}(\Psi)} + \frac{1}{m} \right) \cdot (1 + \sqrt{d}\epsilon) , \end{aligned}$$

by (102). Because $\varphi_j^\epsilon(m) \in \mathcal{D}^\epsilon$ for every $j \in [N(m)]$ and $e_k \in \mathcal{D}^\epsilon$ for every $k \in [d]$, this implies that

$$\sqrt{\xi_{\mathcal{D}^\epsilon}(\Psi)} \leq \left(\sqrt{\xi_{\mathcal{D}}(\Psi)} + \frac{1}{m} \right) \cdot (1 + \sqrt{d}\epsilon) .$$

Since $m \in \mathbb{N}$ was arbitrary, we can take the limit $m \rightarrow \infty$ and obtain

$$\sqrt{\xi_{\mathcal{D}^\epsilon}(\Psi)} \leq \sqrt{\xi_{\mathcal{D}}(\Psi)} \cdot (1 + \sqrt{d}\epsilon) .$$

This gives the claim. □

Lemma 6.3. *Let $\mathcal{D}^\epsilon \subset \mathcal{D} \subset \mathcal{H}$ be as in Lemma 6.2. Then we have*

$$\sqrt{F_{\mathcal{D}^\epsilon}(\Psi)} \leq \sqrt{F_{\mathcal{D}}(\Psi)} \leq \sqrt{F_{\mathcal{D}^\epsilon}(\Psi)} + \|\Psi\| \cdot \epsilon \quad \text{for all } \Psi \in \mathcal{H} .$$

Proof. The first inequality follows trivially from the definitions using $\mathcal{D}^\epsilon \subset \mathcal{D}$. For the second inequality, let $\Psi \in \mathcal{H}$ be arbitrary. Let $\varphi \in \mathcal{D}$ be such that

$$F_{\mathcal{D}}(\Psi) = |\langle \varphi, \Psi \rangle|^2 .$$

Then (by the fact that \mathcal{D}^ϵ is an ϵ -net for \mathcal{D} and $\varphi \in \mathcal{D}$) there is an element $\varphi^\epsilon \in \mathcal{D}^\epsilon$ and $\delta \in \mathcal{H}$ such that

$$\varphi = \varphi^\epsilon + \delta \quad \text{where} \quad \|\delta\| \leq \epsilon .$$

It follows that

$$\begin{aligned} \sqrt{F_{\mathcal{D}}(\Psi)} &= |\langle \varphi, \Psi \rangle| \\ &\leq |\langle \varphi^\epsilon, \Psi \rangle| + |\langle \delta, \Psi \rangle| \\ &\leq \sqrt{F_{\mathcal{D}^\epsilon}(\Psi)} + |\langle \delta, \Psi \rangle| \\ &\leq \sqrt{F_{\mathcal{D}^\epsilon}(\Psi)} + \|\Psi\| \cdot \|\delta\| \\ &\leq \sqrt{F_{\mathcal{D}^\epsilon}(\Psi)} + \|\Psi\| \cdot \epsilon \end{aligned}$$

where we used the definition of $F_{\mathcal{D}^\epsilon}(\Psi)$ and the Cauchy-Schwarz inequality (in the penultimate step). The claim follows. □

Lemma 6.4. *We have*

$$\|y\|^2 \leq d^2 \cdot F_{\mathcal{D}^\epsilon}(y) \quad \text{for every } y \in \mathcal{H} ,$$

where $d = \dim \mathcal{H}$.

Proof. Let $\{e_k\}_{k=1}^d$ be an orthonormal basis contained in \mathcal{D}^ϵ . Then

$$|\langle e_k, y \rangle|^2 \leq F_{\mathcal{D}^\epsilon}(y) \quad \text{for every } k \in [d]$$

because $e_k \in \mathcal{D}^\epsilon$ for every $k \in [n]$. We have

$$\begin{aligned} \|y\| &= \left(\sum_{k=1}^d |\langle e_k, y \rangle|^2 \right)^{1/2} \\ &\leq \sum_{k=1}^d |\langle e_k, y \rangle| \\ &\leq d \cdot \sqrt{F_{\mathcal{D}^\epsilon}(y)} \end{aligned}$$

where we used that $\|v\|_2 \leq \|v\|_1$ for $v \in \mathbb{C}^d$. The claim follows. \square

Lemma 6.5. *Let $d_j = \dim \mathcal{H}_j$ for $j \in [2]$. Assuming that the property $\text{Mult}^F(\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3)$ holds, we have*

$$\xi_{\mathcal{D}_3}(\Psi_1 \otimes \Psi_2) \geq g(d_1, d_2, d_3, \epsilon) \xi_{\mathcal{D}_1}(\Psi_1) \xi_{\mathcal{D}_2}(\Psi_2)$$

for a function g which satisfies

$$\lim_{\epsilon \rightarrow 0} g(d_1, d_2, d_3, \epsilon) = 1 .$$

Proof. Suppose $y_j \in \mathcal{H}_j$ for $j \in [2]$ is such that

$$F_{\mathcal{D}_j^\epsilon}(y_j) \leq 1 \tag{104}$$

and

$$\xi_{\mathcal{D}_j^\epsilon}(\Psi_j) = |\langle y_j, \Psi_j \rangle|^2 \tag{105}$$

for $j \in [2]$. Such a pair (y_1, y_2) exists since \mathcal{D}_1^ϵ and \mathcal{D}_2^ϵ are finite sets and the dual definition of the extent in terms of a maximum applies (see Eq. (98)). Equation (104) implies that

$$\|y_j\| \leq d_j \quad \text{for } j \in [2] ,$$

see Lemma 6.4.

We have

$$\begin{aligned} \sqrt{F_{\mathcal{D}_3^\epsilon}(y_1 \otimes y_2)} &\leq \sqrt{F_{\mathcal{D}_3}(y_1 \otimes y_2)} && \text{by Lemma 6.3} \\ &= \sqrt{F_{\mathcal{D}_1}(y_1)} \cdot \sqrt{F_{\mathcal{D}_2}(y_2)} && \text{by the assumption that } \text{Mult}^F(\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3) \text{ holds} \\ &\leq \left(\sqrt{F_{\mathcal{D}_1^\epsilon}(y_1)} + \|y_1\| \cdot \epsilon \right) \left(\sqrt{F_{\mathcal{D}_2^\epsilon}(y_2)} + \|y_2\| \cdot \epsilon \right) && \text{by Lemma 6.3} \\ &\leq \sqrt{F_{\mathcal{D}_1^\epsilon}(y_1)} \cdot \sqrt{F_{\mathcal{D}_2^\epsilon}(y_2)} \cdot (1 + d_1 \cdot \epsilon) (1 + d_2 \cdot \epsilon) && \text{by Lemma 6.4} \\ &\leq (1 + d_1 \epsilon)(1 + d_2 \epsilon) && \text{by Eq. (104)} . \end{aligned} \tag{106}$$

Defining

$$\tilde{y}_j = (1 + d_j \epsilon)^{-1} y_j \quad \text{for } j \in [2]$$

it follows from Eq. (106) that

$$F_{\mathcal{D}_3^\epsilon}(\tilde{y}_1 \otimes \tilde{y}_2) \leq 1 .$$

By the dual formulation (98) of the quantity $\xi_{\mathcal{D}_3^\epsilon}(\Psi_1 \otimes \Psi_2)$ this implies that

$$\begin{aligned} \xi_{\mathcal{D}_3^\epsilon}(\Psi_1 \otimes \Psi_2) &\geq |\langle \tilde{y}_1 \otimes \tilde{y}_2, \Psi_1 \otimes \Psi_2 \rangle|^2 \\ &= \left(\prod_{j=1}^2 (1 + d_j \epsilon)^{-2} |\langle y_j, \Psi_j \rangle|^2 \right) \\ &= \left(\prod_{j=1}^2 (1 + d_j \epsilon)^{-2} \right) \xi_{\mathcal{D}_1^\epsilon}(\Psi_1) \xi_{\mathcal{D}_2^\epsilon}(\Psi_2) \quad \text{because of Eq. (105)} \\ &\geq \left(\prod_{j=1}^2 (1 + d_j \epsilon)^{-2} \right) \xi_{\mathcal{D}_1}(\Psi_1) \xi_{\mathcal{D}_2}(\Psi_2) \quad \text{according to Lemma 6.2.} \end{aligned} \quad (107)$$

Lemma 6.2 also implies that

$$\left(1 + \sqrt{d_3} \epsilon\right)^2 \xi_{\mathcal{D}_3}(\Psi_1 \otimes \Psi_2) \geq \xi_{\mathcal{D}_3^\epsilon}(\Psi_1 \otimes \Psi_2) . \quad (108)$$

Combining Eqs. (107) and (108) gives the claim, with

$$g(d_1, d_2, d_3 \epsilon) = \left(1 + \sqrt{d_3} \epsilon\right)^{-2} \left(\prod_{j=1}^2 (1 + d_j \epsilon)^{-2} \right) .$$

□

Theorem 6.6. *Let $\mathcal{H}_1, \mathcal{H}_2$ and \mathcal{H}_3 be a triple of Hilbert spaces and let $\{\mathcal{D}_j\}_{j \in [3]}$ be a family of dictionaries, where $\mathcal{D}_j \subset \mathcal{H}_j$. Assume that \mathcal{D}_j contains an orthonormal basis of \mathcal{H}_j , for $j \in [3]$, and that $\mathcal{D}_1 \otimes \mathcal{D}_2 \subset \mathcal{D}_3$. Assume further that for any $\epsilon > 0$ there is an ϵ -net \mathcal{D}_j^ϵ for \mathcal{D}_j such that $\mathcal{D}_j^\epsilon \subset \mathcal{D}_j$, i.e., the net consists of elements of \mathcal{D}_j . Finally, assume that*

$$F_{\mathcal{D}_3}(\Psi_1 \otimes \Psi_2) = F_{\mathcal{D}_1}(\Psi_1) F_{\mathcal{D}_2}(\Psi_2) \quad \text{for all} \quad \Psi_1 \in \mathcal{H}_1, \Psi_2 \in \mathcal{H}_2 .$$

Then

$$\xi_{\mathcal{D}_3}(\Psi_1 \otimes \Psi_2) = \xi_{\mathcal{D}_1}(\Psi_1) \xi_{\mathcal{D}_2}(\Psi_2) \quad \text{for all} \quad \Psi_1 \in \mathcal{H}_1 \text{ and } \Psi_2 \in \mathcal{H}_2 .$$

Proof. By (if necessary) replacing \mathcal{D}_j^ϵ by $\mathcal{D}_j^\epsilon \cup \{e_k^{(j)}\}_{k=1}^d$, for $j \in [3]$, where $\{e_k^{(j)}\}_{k=1}^d$ is an orthonormal basis of \mathcal{H}_j with $e_k^{(j)} \in \mathcal{D}_j$, for $n \in [3]$, we have that each \mathcal{D}_j^ϵ is finite and contains an orthonormal basis of the respective space. The inequality

$$\xi_{\mathcal{D}_3}(\Psi_1 \otimes \Psi_2) \geq \xi_{\mathcal{D}_1}(\Psi_1) \xi_{\mathcal{D}_2}(\Psi_2) \quad \text{for all} \quad \Psi_1 \in \mathcal{H}_1 \text{ and } \Psi_2 \in \mathcal{H}_2$$

now follows immediately from Lemma 6.5 by taking the limit $\epsilon \rightarrow 0$. The converse inequality is trivial because $\mathcal{D}_1 \otimes \mathcal{D}_2 \subset \mathcal{D}_3$. □

7 Multiplicativity of the Gaussian extent for four fermions

In this section we prove that the Gaussian extent is multiplicative for the tensor product of any two 4-fermion pure states with positive parity.

Theorem 7.1 (Multiplicativity of the Gaussian extent for 4-fermion pure states.). *Let \mathcal{H}_+^4 be the set of pure 4-fermion states with positive parity and let \mathcal{G}_n be the set of Gaussian states on n fermions. Then*

$$\xi_{\mathcal{G}_8}(\Psi_A \tilde{\otimes} \Psi_B) = \xi_{\mathcal{G}_4}(\Psi_A) \xi_{\mathcal{G}_4}(\Psi_B) \quad \text{for all} \quad \Psi_A, \Psi_B \in \mathcal{H}_+^4.$$

Proof. Since the metaplectic representation defines a surjective, continuous map

$$\begin{aligned} f : [0, 2\pi] \times SO(2n) &\rightarrow \mathcal{G}_n \\ (\varphi, R) &\mapsto e^{i\varphi} U_R |0_F\rangle \end{aligned}$$

from the compact set $[0, 2\pi] \times SO(2n)$ to \mathcal{G}_n , the set $\mathcal{G}_n \subset \mathcal{H}^n$ is compact. We also observe that the occupation number states

$$\{|x\rangle \mid x \in \{0, 1\}^n\}$$

form an orthonormal basis contained in \mathcal{G}_n . By compactness, we conclude that for any $\epsilon > 0$, there is a finite ϵ -net $\mathcal{G}_n^\epsilon \subset \mathcal{G}_n$ consisting of Gaussian states and containing an orthonormal basis of \mathcal{H}^n . Finally, we note that we also have the inclusion $\mathcal{G}_{n_1} \otimes \mathcal{G}_{n_2} \subset \mathcal{G}_{n_1+n_2}$ for $n_1, n_2 \in \mathbb{N}$ arbitrary.

Let us now specialize to $n_1 = n_2 = 4$. In this case, we have multiplicativity of the fermionic Gaussian fidelity by Theorem 5.10. In particular, the conditions for Theorem 6.6 apply and the claim follows. \square

Acknowledgements

BD and RK gratefully acknowledge support by the European Research Council under grant agreement no. 101001976 (project EQUIPTNT).

A Alternative Gaussianity condition for 4-fermion states

In the following, we prove Lemma 5.1.

Proof. Consider $\Psi \in \mathcal{H}_+^4$. We will show that $\langle \Psi, \theta \Psi \rangle = 0$ is equivalent to $\Lambda(|\Psi\rangle \otimes |\Psi\rangle) = 0$. According to Lemma 2.3, this is a sufficient and necessary condition for Ψ to be Gaussian.

Let $E \subset \{0, 1\}_+^4$, $|E| = 4$ be a subset of even-weight strings such that $E \cup \bar{E} = \{0, 1\}_+^4$.

We first compute $\langle \Psi, \theta \Psi \rangle$. We have

$$\begin{aligned}
\langle \Psi, \theta \Psi \rangle &= \sum_{x, y \in \{0,1\}_+^4} \langle \Psi, x \rangle \langle x | \theta (|y\rangle \langle y, \Psi \rangle) \\
&= \sum_{x, y \in \{0,1\}_+^4} \langle \Psi, x \rangle \langle x, \theta y \rangle \overline{\langle y, \Psi \rangle} \\
&= \sum_{x, y \in \{0,1\}_+^4} (-1)^{\vartheta(y)} \langle x, \bar{y} \rangle \langle \Psi, x \rangle \langle \Psi, y \rangle \\
&= \sum_{x \in \{0,1\}_+^4} (-1)^{\vartheta(x)} \langle \Psi, x \rangle \langle \Psi, \bar{x} \rangle \\
&= \sum_{x \in E} (-1)^{\vartheta(x)} \langle \Psi, x \rangle \langle \Psi, \bar{x} \rangle + (-1)^{\vartheta(\bar{x})} \langle \Psi, \bar{x} \rangle \langle \Psi, x \rangle \\
&= 2 \sum_{x \in E} (-1)^{\vartheta(x)} \langle \Psi, x \rangle \langle \Psi, \bar{x} \rangle,
\end{aligned}$$

where the second step follows from $\theta u = \bar{u}\theta$ for all $u \in \mathbb{C}$ due to the antiunitarity of θ , the third step from Eq. (76) and the fourth and final steps from $\vartheta(\bar{x}) = \vartheta(x)$ for $x \in \{0,1\}_+^4$. Thus,

$$\langle \Psi, \theta \Psi \rangle = 0 \quad \text{if and only if} \quad \sum_{x \in \{0,1\}_+^4} (-1)^{\vartheta(x)} \langle \Psi, x \rangle \langle \Psi, \bar{x} \rangle = 0. \quad (109)$$

We proceed to prove that $\Lambda(|\Psi\rangle \otimes |\Psi\rangle) = 0$ is equivalent to Eq. (109). We start by using Eq. (13) to write the operator Λ in terms of creation and annihilation operators:

$$\begin{aligned}
\Lambda &= \sum_{j=1}^8 c_j \otimes c_j \\
&= \sum_{j=1}^4 (c_{2j-1} \otimes c_{2j-1} + c_{2j} \otimes c_{2j}) \\
&= \sum_{j=1}^4 \left((a_j + a_j^\dagger) \otimes (a_j + a_j^\dagger) + i(a_j - a_j^\dagger) \otimes i(a_j - a_j^\dagger) \right) \\
&= 2 \sum_{j=1}^4 \left(a_j \otimes a_j^\dagger + a_j^\dagger \otimes a_j \right).
\end{aligned}$$

Applying this expression to $|\Psi\rangle \otimes |\Psi\rangle$ and using Eq. (11) gives

$$\begin{aligned}
\Lambda(|\Psi\rangle \otimes |\Psi\rangle) &= 2 \sum_{x, y \in \{0,1\}_+^4} \sum_{j=1}^4 \left(a_j \otimes a_j^\dagger + a_j^\dagger \otimes a_j \right) (|x\rangle \otimes |y\rangle) \langle x, \Psi \rangle \langle y, \Psi \rangle \\
&= 2 \sum_{x, y \in \{0,1\}_+^4} \sum_{j=1}^4 (-1)^{\eta_j(x+y)} (x_j \bar{y}_j + \bar{x}_j y_j) (|x \oplus e_j\rangle \otimes |y \oplus e_j\rangle) \langle x, \Psi \rangle \langle y, \Psi \rangle \\
&= 2 \sum_{x, y \in \{0,1\}_+^4} \sum_{j=1}^4 (-1)^{\eta_j(x+y)} (y_j \oplus \bar{x}_j) (|x \oplus e_j\rangle \otimes |y \oplus e_j\rangle) \langle x, \Psi \rangle \langle y, \Psi \rangle \\
&= 2 \sum_{x, y \in \{0,1\}_+^4} \left(\sum_{j=1}^4 (-1)^{\eta_j(x+y)} (y_j \oplus \bar{x}_j) \langle x \oplus e_j, \Psi \rangle \langle y \oplus e_j, \Psi \rangle \right) (|x\rangle \otimes |y\rangle)
\end{aligned} \quad (110)$$

where in the third line we used $u\bar{v} + \bar{u}v = u \oplus \bar{v}$ for all $u, v \in \{0, 1\}$ and in the last line we used $\{x \oplus e_j \mid x \in \{0, 1\}_+^4\} = \{0, 1\}_+^4$, $\eta_j(x \oplus e_j) = \eta_j(x)$ and $u \oplus v \oplus 2e_j = u \oplus v$ valid for $x \in \{0, 1\}_+^4$, $j \in [4]$ and $u, v \in \{0, 1\}$. It follows that $\Lambda(|\Psi\rangle \otimes |\Psi\rangle) = 0$ if and only if

$$\sum_{j=1}^4 (-1)^{\eta_j(x+y)} (y_j \oplus \bar{x}_j) \langle x \oplus e_j, \Psi \rangle \langle y \oplus e_j, \Psi \rangle = 0 \quad \text{for all } x, y \in \{0, 1\}_+^4. \quad (111)$$

Since x and y have the same parity, either $|x - y| = 4$ (i.e., $y = \bar{x}$), $|x - y| = 2$ or $|x - y| = 0$ (i.e., $y = x$). The expression (111) is non-zero only if $|x - y| = 4$ (we argue below why). For $|x - y| = 4$ Eq. (111) becomes

$$\sum_{j=1}^4 (-1)^{\eta_j(x+\bar{x})} \langle x \oplus e_j, \Psi \rangle \langle \bar{x} \oplus e_j, \Psi \rangle = 0 \quad \text{if and only if} \quad \sum_{z \in E} (-1)^{\vartheta(z)} \langle z, \Psi \rangle \langle z, \Psi \rangle = 0,$$

where we used $\{x \oplus e_j \mid j \in [4]\} = E$ with $E \subset \{0, 1\}_+^4$, $|E| = 4$ a subset of even-weight strings such that $E \cup \bar{E} = \{0, 1\}_+^4$. We also used that $(-1)^{\eta_j(x+\bar{x})} = (-1)^{j-1}$ can be replaced by $(-1)^{\vartheta(x)+\vartheta(z)}$ upon changing the summation over $j \in [4]$ to a summation over $z \in E$. We recovered the right hand side of Eq. (109), proving the claim.

It remains to argue that Eq. (111) is zero for $|x - y| \in \{0, 2\}$. For $|x - y| = 0$, i.e., $x = y$, Eq. (111) is zero because $x_j \otimes \bar{x}_j = 0$ for $j \in [4]$. We exemplify that terms $\propto |x\rangle \otimes |y\rangle$ with $|x - y| = 2$ are zero by considering $x = 1000$ and $y = 0100$. Starting from Eq. (110) we obtain

$$(|1000\rangle \otimes |0100\rangle) \Lambda(|\Psi\rangle \otimes |\Psi\rangle) = ((-1)^{\eta_1(1100)} + (-1)^{\eta_2(1100)}) \langle 0000, \Psi \rangle \langle 1100, \Psi \rangle = 0,$$

where $\eta_1(1100) = 0$ and $\eta_2(1100) = 1$. The remaining cases with $|x - y| = 2$ proceed similarly. □

B Commutativity of the map θ and quadratic Majorana monomials

In the following, we prove Lemma 5.2.

Proof. We start by showing that $\theta = c_1 c_3 c_5 c_7 K$, where K denotes the antiunitary given by complex conjugation in the number state basis. For this, it suffices to show that θ is antiunitary, which directly follows from unitarity of $c_1 c_3 c_5 c_7$, and that it satisfies Eq. (76). We show the later using Eqs. (13) and (11): We have

$$\begin{aligned} \theta |x\rangle &= c_1 c_3 c_5 c_7 K |x\rangle \\ &= (a_1 + a_1^\dagger)(a_2 + a_2^\dagger)(a_3 + a_3^\dagger)(a_4 + a_4^\dagger) |x\rangle \\ &= (-1)^{\eta_4(x)+\eta_3(x)+\eta_2(x)+\eta_1(x)} (x_1 + \bar{x}_1)(x_2 + \bar{x}_2)(x_3 + \bar{x}_3)(x_4 + \bar{x}_4) |\bar{x}\rangle \\ &= (-1)^{\vartheta(x)} |\bar{x}\rangle, \end{aligned}$$

where we used $(-1)^{\eta_4(x)+\eta_3(x)+\eta_2(x)+\eta_1(x)} = (-1)^{3x_1+2x_2+x_3} = (-1)^{x_1+x_3} = (-1)^{\vartheta(x)}$ and $x_j + \bar{x}_j = 1$ for $j \in [4]$.

The result $\theta c_j c_k = c_j c_k \theta$ follows from simple algebra considering $c_{2j} K = K c_{2j}$ and $c_{2j-1} K = -K c_{2j-1}$. We show these last two equalities by explicitly computing their

action on $x \in \{0, 1\}_+^4$:

$$\begin{aligned} Kc_{2j} |x\rangle &= K(a_j + a_j^\dagger) |x\rangle = (x_j + \bar{x}_j) |x \oplus e_j\rangle = |x \oplus e_j\rangle , \\ c_{2j}K |x\rangle &= (a_j + a_j^\dagger) |x\rangle = (x_j + \bar{x}_j) |x \oplus e_j\rangle = Kc_{2j} |x\rangle \end{aligned}$$

and

$$\begin{aligned} Kc_{2j+1} |x\rangle &= Ki(a_j - a_j^\dagger) |x\rangle = -i(x_j - \bar{x}_j) |x \oplus e_j\rangle , \\ c_{2j+1}K |x\rangle &= i(a_j - a_j^\dagger) |x\rangle = i(x_j - \bar{x}_j) |x \oplus e_j\rangle = -Kc_{2j+1} |x\rangle . \end{aligned}$$

□

Proof. We will prove that $\theta c_j = -c_j \theta$ for $j \in [8]$, which implies the result.

We prove this for j odd, the proof for j even proceeds similarly. We use Eq. (13) to write the Majorana operators as creation and annihilation operators which act on basis states according to Eq. (11), and we apply θ according to Eq. (76):

$$\begin{aligned} \theta c_{2j-1} |x\rangle &= \theta(a_j + a_j^\dagger) |x\rangle \\ &= \theta(-1)^{\eta_j(x)} (x_j + \bar{x}_j) |x \oplus e_j\rangle \\ &= (-1)^{\eta_j(x)} (-1)^{\vartheta(x \oplus e_j)} (x_j + \bar{x}_j) |\overline{x \oplus e_j}\rangle , \end{aligned}$$

$$\begin{aligned} c_{2j-1} \theta |x\rangle &= (-1)^{\vartheta(x)} (a_j + a_j^\dagger) |\bar{x}\rangle \\ &= (-1)^{\eta_j(\bar{x})} (-1)^{\vartheta(x)} (\bar{x}_j + x_j) |\bar{x} \oplus e_j\rangle . \end{aligned}$$

The equality $\theta c_{2j-1} = -c_{2j-1} \theta$ follows from $|\bar{x} \oplus e_j\rangle = |\overline{x \oplus e_j}\rangle$ for $j \in [4]$, from $(-1)^{\eta_j(\bar{x})} = (-1)^{j+1} (-1)^{\eta_j(x)}$ and from $(-1)^{\vartheta(x)} = (-1)^j (-1)^{\vartheta(x \oplus e_j)}$. □

References

- [1] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. Topological quantum memory. *Journal of Mathematical Physics*, 43(9):4452–4505, 2002. <https://doi.org/10.1063/1.1499754>.
- [2] Ruben S. Andrist, H. Bombin, Helmut G. Katzgraber, and M. A. Martin-Delgado. Optimal error correction in topological subsystem codes. *Physical Review A*, 85:050302, 2012. <https://doi.org/10.1103/PhysRevA.85.050302>.
- [3] Sergey Bravyi and Robert König. Disorder-assisted error correction in Majorana chains. *Communications in Mathematical Physics*, 316(3):641–692, 2012. <https://doi.org/10.1007/s00220-012-1606-9>.
- [4] Andrew S. Darmawan and David Poulin. Tensor-network simulations of the surface code under realistic noise. *Physical Review Letters*, 119:040502, 2017. <https://doi.org/10.1103/PhysRevLett.119.040502>.
- [5] Sergey Bravyi, Matthias Englbrecht, Robert König, and Nolan Peard. Correcting coherent errors with surface codes. *npj Quantum Information*, 4(1):1–6, 2018. <https://doi.org/10.1038/s41534-018-0106-y>.
- [6] David K. Tuckett, Andrew S. Darmawan, Christopher T. Chubb, Sergey Bravyi, Stephen D. Bartlett, and Steven T. Flammia. Tailoring surface codes for highly biased noise. *Physical Review X*, 9:041031, 2019. <https://doi.org/10.1103/PhysRevX.9.041031>.

- [7] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, 2004. <https://doi.org/10.1103/PhysRevA.70.052328>.
- [8] Leslie G. Valiant. Quantum computers that can be simulated classically in polynomial time. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, STOC '01*, page 114–123, New York, USA, 2001. <https://doi.org/10.1145/380752.380785>.
- [9] Barbara M. Terhal and David P. DiVincenzo. Classical simulation of noninteracting-fermion quantum circuits. *Physical Review A*, 65:032325, 2002. <https://doi.org/10.1103/PhysRevA.65.032325>.
- [10] Emanuel Knill. Fermionic linear optics and matchgates. Technical Report LAUR-01-4472, Los Alamos National Laboratory, 2001.
- [11] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Physical Review A*, 71:022316, 2005. <https://doi.org/10.1103/PhysRevA.71.022316>.
- [12] Sergey Bravyi. Universal quantum computation with the $\nu = 5/2$ fractional quantum Hall state. *Physical Review A*, 73:042313, 2006. <https://doi.org/10.1103/PhysRevA.73.042313>.
- [13] Martin Hebenstreit, Richard Jozsa, Barbara Kraus, Sergii Strelchuk, and Mithuna Yoganathan. All pure fermionic non-Gaussian states are magic states for matchgate computations. *Physical Review Letters*, 123:080503, 2019. <https://doi.org/10.1103/PhysRevLett.123.080503>.
- [14] Mark Howard and Earl Campbell. Application of a resource theory for magic states to fault-tolerant quantum computing. *Physical Review Letters*, 118:090501, 2017. <https://doi.org/10.1103/PhysRevLett.118.090501>.
- [15] Markus Heinrich and David Gross. Robustness of magic and symmetries of the stabiliser polytope. *Quantum*, 3:132, 2019. <https://doi.org/10.22331/q-2019-04-08-132>.
- [16] Sergey Bravyi, Graeme Smith, and John A. Smolin. Trading classical and quantum computational resources. *Physical Review X*, 6(2):021043, 2016. <https://doi.org/10.1103/PhysRevX.6.021043>.
- [17] Sergey Bravyi and David Gosset. Improved classical simulation of quantum circuits dominated by Clifford gates. *Physical Review Letters*, 116(25), 2016. <https://doi.org/10.1103/PhysRevLett.116.250501>.
- [18] Sergey Bravyi, Dan Browne, Padraic Calpin, Earl Campbell, David Gosset, and Mark Howard. Simulation of quantum circuits by low-rank stabilizer decompositions. *Quantum*, 3:181, 2019. <https://doi.org/10.22331/q-2019-09-02-181>.
- [19] Arne Heimendahl, Felipe Montealegre-Mora, Frank Vallentin, and David Gross. Stabilizer extent is not multiplicative. *Quantum*, 5:400, 2021. <https://doi.org/10.22331/q-2021-02-24-400>.
- [20] Michael Beverland, Earl Campbell, Mark Howard, and Vadym Kliuchnikov. Lower bounds on the non-Clifford resources for quantum computations. *Quantum Science and Technology*, 5(3):035009, 2020. <https://doi.org/10.1088/2058-9565/ab8963>.

- [21] James R. Seddon, Bartosz Regula, Hakop Pashayan, Yingkai Ouyang, and Earl T. Campbell. Quantifying quantum speedups: Improved classical simulation from tighter magic monotones. *PRX Quantum*, 2:010345, 2021. <https://doi.org/10.1103/PRXQuantum.2.010345>.
- [22] Kaifeng Bu, Weichen Gu, and Arthur Jaffe. Stabilizer testing and magic entropy. arXiv:2306.09292, 2023.
- [23] Joel A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004. <https://doi.org/10.1109/TIT.2004.834793>.
- [24] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001. <https://doi.org/10.1137/S003614450037906X>.
- [25] Jean-Jacques Fuchs. On sparse representations in arbitrary redundant bases. *IEEE Transactions on Information Theory*, 50(6):1341–1344, 2004. <https://doi.org/10.1109/TIT.2004.828141>.
- [26] Joel A. Tropp. Recovery of short, complex linear combinations via ℓ_1 minimization. *IEEE Transactions on Information Theory*, 51(4):1568–1570, 2005. <https://doi.org/10.1109/TIT.2005.844057>.
- [27] Hakop Pashayan, Oliver Reardon-Smith, Kamil Korzekwa, and Stephen D. Bartlett. Fast estimation of outcome probabilities for quantum circuits. *PRX Quantum*, 3:020361, 2022. <https://doi.org/10.1103/PRXQuantum.3.020361>.
- [28] Farid Alizadeh and Donald Goldfarb. Second-order cone programming. *Mathematical Programming*, 95(1):3–51, 2003. <https://doi.org/10.1007/s10107-002-0339-5>.
- [29] Stephen P. Boyd and Lieven Vandenbergh. *Convex optimization*. Cambridge university press, 2004. <https://doi.org/10.1017/CB09780511804441>.
- [30] Venkat Chandrasekaran, Benjamin Recht, Pablo A. Parrilo, and Alan S. Willsky. The Convex Geometry of Linear Inverse Problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012. <https://doi.org/10.1007/s10208-012-9135-7>.
- [31] Christophe Piveteau and David Sutter. Circuit knitting with classical communication. *IEEE Transactions on Information Theory*, 70(4):2734–2745, 2024. <https://doi.org/10.1109/TIT.2023.3310797>.
- [32] Fernando de Melo, Piotr Ćwikliński, and Barbara M. Terhal. The power of noisy fermionic quantum computation. *New Journal of Physics*, 15(1):013015, 2013. <https://doi.org/10.1088/1367-2630/15/1/013015>.
- [33] Richard Jozsa and Akimasa Miyake. Matchgates and classical simulation of quantum circuits. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 464(2100):3089–3106, 2008. <https://doi.org/10.1098/rspa.2008.0189>.
- [34] Stephen D. Bartlett and Barry C. Sanders. Efficient classical simulation of optical quantum information circuits. *Physical Review Letters*, 89:207903, 2002. <https://doi.org/10.1103/PhysRevLett.89.207903>.
- [35] Stephen D. Bartlett, Barry C. Sanders, Samuel L. Braunstein, and Kae Nemoto. Efficient classical simulation of continuous variable quantum information processes. *Physical Review Letters*, 88:097904, 2002. <https://doi.org/10.1103/PhysRevLett.88.097904>.

- [36] Rolando Somma, Howard Barnum, Gerardo Ortiz, and Emanuel Knill. Efficient solvability of Hamiltonians and limits on the power of some quantum computational models. *Physical Review Letters*, 97:190501, 2006. <https://doi.org/10.1103/PhysRevLett.97.190501>.
- [37] Daniel Gottesman. *Stabilizer Codes and Quantum Error Correction*. PhD thesis, California Institute of Technology, 1997. <https://doi.org/10.7907/rzr7-dt72>.
- [38] Victor Veitch, Christopher Ferrie, David Gross, and Joseph Emerson. Negative quasi-probability as a resource for quantum computation. *New Journal of Physics*, 14(11):113011, 2012. <https://doi.org/10.1088/1367-2630/14/11/113011>.
- [39] Andrea Mari and Jens Eisert. Positive wigner functions render classical simulation of quantum computation efficient. *Physical Review Letters*, 109:230503, 2012. <https://doi.org/10.1103/PhysRevLett.109.230503>.
- [40] Hakop Pashayan, Joel J. Wallman, and Stephen D. Bartlett. Estimating outcome probabilities of quantum circuits using quasiprobabilities. *Physical Review Letters*, 115:070501, 2015. <https://doi.org/10.1103/PhysRevLett.115.070501>.
- [41] Robert Raussendorf, Juani Bermejo-Vega, Emily Tyhurst, Cihan Okay, and Michael Zurel. Phase-space-simulation method for quantum computation with magic states on qubits. *Physical Review A*, 101:012350, 2020. <https://doi.org/10.1103/PhysRevA.101.012350>.
- [42] Robert Raussendorf, Dan E. Browne, Nicolas Delfosse, Cihan Okay, and Juan Bermejo-Vega. Contextuality and wigner-function negativity in qubit quantum computation. *Physical Review A*, 95:052334, 2017. <https://doi.org/10.1103/PhysRevA.95.052334>.
- [43] Juan Bermejo-Vega, Nicolas Delfosse, Dan E. Browne, Cihan Okay, and Robert Raussendorf. Contextuality as a resource for models of quantum computation with qubits. *Physical Review Letters*, 119:120505, 2017. <https://doi.org/10.1103/PhysRevLett.119.120505>.
- [44] Markus Frembs, Sam Roberts, and Stephen D. Bartlett. Contextuality as a resource for measurement-based quantum computation beyond qubits. *New Journal of Physics*, 20(10):103011, 2018. <https://doi.org/10.1088/1367-2630/aae3ad>.
- [45] Anna Vershynina. Complete criterion for convex-gaussian-state detection. *Physical Review A*, 90:062329, 2014. <https://doi.org/10.1103/PhysRevA.90.062329>.
- [46] Shigeo Hakkaku, Yuichiro Tashima, Kosuke Mitarai, Wataru Mizukami, and Keisuke Fujii. Quantifying fermionic nonlinearity of quantum circuits. *Physical Review Research*, 4:043100, 2022. <https://doi.org/10.1103/PhysRevResearch.4.043100>.
- [47] Avinash Mocherla, Lingling Lao, and Dan E. Browne. Extending matchgate simulation methods to universal quantum circuits. arXiv:2302.02654, 2023.
- [48] Oliver Reardon-Smith, Michał Oszmaniec, and Kamil Korzekwa. Improved classical simulation of quantum circuits dominated by fermionic linear optical gates. arXiv:2307.12702, 2023.
- [49] Joshua Cudby and Sergii Strelchuk. Gaussian decomposition of magic states for matchgate computations. arXiv:2307.12654, 2023.
- [50] Alonso Botero and Benni Reznik. BCS-like modewise entanglement of fermion Gaussian states. *Physics Letters A*, 331(1):39–44, 2004. <https://doi.org/10.1016/j.physleta.2004.08.037>.

- [51] Grigori G. Amosov and Sergey N. Filippov. Spectral properties of reduced fermionic density operators and parity superselection rule. Quantum Information Processing, 16(1):2, 2016. <https://doi.org/10.1007/s11128-016-1467-9>.
- [52] Wallace Givens. Computation of plain unitary rotations transforming a general matrix to triangular form. Journal of the Society for Industrial and Applied Mathematics, 6(1):26–50, 1958. <https://doi.org/10.1137/0106004>.
- [53] Gene H. Golub and Charles F. Van Loan. Matrix Computations (3rd Ed.). Johns Hopkins University Press, USA, 1996. <https://doi.org/10.56021/9781421407944>.
- [54] Sergey Bravyi. Lagrangian representation for fermionic linear optics. Quantum Information & Computation, 5(3):216–238, 2005. <https://doi.org/10.26421/qic5.3-3>.
- [55] Woody Lichtenstein. A system of quadrics describing the orbit of the highest weight vector. Proceedings of the American Mathematical Society, 84(4):605–608, 1982. <https://doi.org/10.1090/s0002-9939-1982-0643758-8>.
- [56] Marek Kuś and Ingemar Bengtsson. “Classical” quantum states. Physical Review A, 80(2):022319, 2009. <https://doi.org/10.1103/PhysRevA.80.022319>.
- [57] Michał Oszmaniec and Marek Kuś. On detection of quasiclassical states. Journal of Physics A: Mathematical and Theoretical, 45(24):244034, 2012. <https://doi.org/10.1088/1751-8113/45/24/244034>.
- [58] Michał Oszmaniec, Jan Gutt, and Marek Kuś. Classical simulation of fermionic linear optics augmented with noisy ancillas. Physical Review A, 90(2):020302, 2014. <https://doi.org/10.1103/PhysRevA.90.020302>.
- [59] Per-Olov Löwdin. Quantum theory of many-particle systems. i. physical interpretations by means of density matrices, natural spin-orbitals, and convergence problems in the method of configurational interaction. Physical Review, 97:1474–1489, 1955. <https://doi.org/10.1103/PhysRev.97.1474>.
- [60] Sergey Bravyi and David Gosset. Complexity of quantum impurity problems. Communications in Mathematical Physics, 356(2), 2017. <https://doi.org/10.1007/s00220-017-2976-9>.
- [61] Huzihiro Araki and Hajime Moriya. Joint extension of states of subsystems for a CAR system. Communications in Mathematical Physics, 237(1):105–122, 2003. <https://doi.org/10.1007/s00220-003-0832-6>.
- [62] Thomas P Hayes. A large-deviation inequality for vector-valued martingales, 2005. Unpublished manuscript. <https://www.cs.unm.edu/~hayes/papers/VectorAzuma/>.
- [63] Gongguo Tang, Badri Narayan Bhaskar, and Benjamin Recht. Sparse recovery over continuous dictionaries-just discretize. In 2013 Asilomar Conference on Signals, Systems and Computers, pages 1043–1047, 2013. <https://doi.org/10.1109/ACSSC.2013.6810450>.