

Pure State Tomography with Parallel Unentangled Measurements

François Verdeil and Yannick Deville

Université de Toulouse, UPS, CNRS, CNES, OMP, IRAP, Toulouse, France

(Dated: August 9, 2022)

Quantum state tomography (QST) aims at estimating a quantum state from averaged quantum measurements made on copies of the state. Most quantum algorithms rely on QST at some point and it is a well explored topic in the literature, mostly for mixed states. In this paper we focus on the QST of a pure quantum state using parallel unentangled measurements. Pure states are a small but useful subset of all quantum states, their tomography requires fewer measurements and is essentially a phase recovery problem. Parallel unentangled measurements are easy to implement in practice because they allow the user to measure each qubit individually. We propose two sets of quantum measurements that one can make on a pure state as well as the algorithms that use the measurements outcomes in order to identify the state. We also discuss how those estimates can be finely tuned by finding the state that maximizes the likelihood of the measurements with different variants of the likelihood. The performances of the proposed three types of QST methods are validated by means of detailed numerical tests.

I. PRIOR WORK AND PROBLEM STATEMENT

Quantum state tomography (QST) aims at estimating a quantum state from averaged quantum measurements made on copies of the state. It often is a necessary step in quantum computation [1], it has been extensively studied for mixed states. The most basic version is detailed in [1] at the beginning of Section 8.4.2, it uses measurements defined by Pauli operators, often called Pauli measurements ([2], [3], [4], [5], [6] and [7]). This version is simple and very robust but requires computing the averages of $4^{n_{qb}} - 1$ different types of 2-outcome measurements where n_{qb} is the number of qubits of the state. This scales really badly with the number of qubits but requiring so many types of measurements is not surprising because an arbitrary state is represented by a $d \times d$ Hermitian density matrix with $4^{n_{qb}}$ real parameters (where $d = 2^{n_{qb}}$ is the dimension of the Hilbert space in which the considered state evolves). In order to perform QST with fewer types of measurements, one can focus on a subset of all states. The most popular assumption is that the density matrix ρ representing the state has a low rank. [4] introduced a compressed sensing approach that requires the averages of $O(rd \log(d)^2)$ 2-outcome measurements to estimate the state where r is the rank of ρ . [3], [2], later built upon this idea of QST via compressed sensing. More recently bounded rank QST was introduced [8]. It assumes that the rank r is known and allows the explicit reconstruction of ρ using predetermined measurements (contrary to the compressed sensing approach of [4] that does not specify the measurements to be used and finds ρ by minimizing the nuclear norm of ρ under constraints).

Other approaches do not make any assumption on ρ . In 2014 Self-Guided Quantum Tomography (SGQT) was introduced [9] and further studied in [10], [11]. It makes no assumption on ρ , and the number of measurements scales reasonably with the number of qubits. The drawback of SGQT is that the measurements that need to be performed on the state are not known beforehand and are generally entangled measurements. Entangled measurements correspond to multiqubit operators that cannot be expressed as a tensor product of single-qubit measurement operators i.e. they cannot be performed by measuring each qubit independently. In 2020 [12] introduced a method to partially identify large quantum systems (more than 100 qubits) with entangled states, for which the total state cannot even be stored on a classical computer. It relies on unentangled measurements which are easier to perform than entangled measurements in practice.

The present paper focuses on the tomography of pure states using unentangled measurements. This has been studied in [5] which tried to find the minimal number of Pauli measurements for 2 and 3 qubits (Pauli measurements are unentangled). Our addition to that article is that we will address the generic case with any n_{qb} . Furthermore, we will use parallel measurements like in [12] where it is shown that all $4^{n_{qb}}$ averaged Pauli measurements can be computed from the averages of $3^{n_{qb}}$ parallel unentangled measurements. A parallel measurement has d outcomes and provides more information on the system than a Pauli measurement that only has two outcomes.

In [13] Finkelstein describes a setup able to distinguish almost all pure states, with only $n_{prob} = 2d$ probabilities. [13] does not beat the lower bound of [14], detailed below, because there is a negligi-

ble (zero measure) set of pure states that the setup of [13] cannot recover up to a global phase, it is called the failure set. In addition to the failure set, the main problem of [13] is that the measurements are not practical, they are entangled and cannot be performed in parallel (as the matrix \mathbf{A} associated with the measurements cannot be written as the vertical concatenation of unitary matrices). In [15] Goyeneche et al. introduced a set of $n_{prob} = 4d$ probabilities that also has a negligible failure set. Technically [15] introduces 5 measurements that yield 5d probabilities (obtained from averaging the results of 5 different kinds of d -outcome measurements), but only the 4 measurements defined by its Equation (2) are needed to achieve QST. The measurements of [15] are more realistic as they are performed on 4 orthonormal bases. Two of them are unentangled but the other two are entangled. Goyeneche et al. acknowledge that this is a problem and point out the fact that the two entangled bases can be mapped into the two unentangled ones by applying the quantum Fourier transform twice. In practice this would introduce additional errors, as there are no error-free circuits able to perform the quantum Fourier transform, and one would need to perform quantum process tomography (which generally relies on QST) in order to quantify the errors and improve the Fourier-transform circuit. This is a common issue with entangled measurements, the easiest way to perform them with the current version of quantum computers is to transform them into measurements in an unentangled basis, by means of a corresponding quantum gate.

The applied mathematics community also dealt with an equivalent version of the QST problem for pure states: The phase retrieval problem (see [14], [16], [17], [18], [13]). A pure state $|\varphi\rangle$ of an n_{qb} -qubit system is represented by a complex unit-norm vector \mathbf{v} with d elements. Pure state tomography aims at estimating \mathbf{v} from measurements. The theoretical probabilities of all outcomes of the considered types of measurements are contained in the vector $|\mathbf{A}\mathbf{v}|^2$ where \mathbf{A} is an $n_{prob} \times d$ matrix (n_{prob} is the total number of probabilities) determined by the types of measurements performed and $|\cdot|^2$ stands for component-wise squared modulus. Recovering \mathbf{v} (up to a global phase) from $|\mathbf{A}\mathbf{v}|^2$ (generally it is $|\mathbf{A}\mathbf{v}|$ instead of $|\mathbf{A}\mathbf{v}|^2$ but both problems are essentially the same) is called phase retrieval. The first question asked in phase recovery is injectivity: how can one choose \mathbf{A} in order to make sure that $|\mathbf{A}\mathbf{v}|^2$ contains enough information to recover \mathbf{v} up to a global phase? Proving that a given \mathbf{A} guarantees injectivity is a difficult question. [14] gave a minimal number of measurements below which injectivity is impossible.

In our case this condition is $n_{prob} > 4d - 3 - c(d)n_{qb}$ rows for some $c(d) \in [1, 2]$. [16] showed that for a generic \mathbf{A} , having $4d - 2$ rows or more is a sufficient condition for injectivity.

Beyond injectivity, finding a solution to the phase recovery problem (whether it is unique up to a global phase or not) is the main difficulty of pure state tomography. Both [13] and [15] give their own closed-form algorithms to recover the phases which are adapted to their versions of \mathbf{A} . [17] focuses on this particular problem with a generic \mathbf{A} .

Our contributions in the present paper are as follows. Section II describes the quantum state to be identified and the measurements made. In particular, we formalize the definition of a parallel unentangled measurement.

Section III describes a method to achieve QST with $n_{prob} = 4d$ using an optimization algorithm of [17] on a number of probabilities consistent with the lower bound of [14]. The probabilities can be obtained by averaging the results of 4 types of parallel unentangled measurements.

Section IV describes an original method with $n_{prob} = (2n_{qb} + 1)d$ probabilities for which phase recovery can be achieved with a closed-form recursive algorithm. Those probabilities are obtained by averaging the results of $2n_{qb} + 1$ different kinds of measurements.

Section V describes a more precise fine tuning method that works with all types of measurements, it requires an initial estimate from one of the algorithms of Sections III or IV which it uses in order to maximize the likelihood of the measurements.

Finally, in Section VI we evaluate the performance of the proposed algorithms with simulated data.

II. STATE AND MEASUREMENTS

A. Considered state

An n_{qb} -qubit pure state $|\varphi\rangle$ can be decomposed in the canonical basis $|0\dots 0\rangle, \dots, |1\dots 1\rangle$. The components of $|\varphi\rangle$ in the basis can be stored in a d -element vector ($d = 2^{n_{qb}}$) $\mathbf{v} = [v_1 \dots v_d]^T$ where T stands for transpose. The components v_j are complex and $\sum_{j=1}^d |v_j|^2 = 1$. The global phase of $|\varphi\rangle$ has no physical meaning, so we can assume that v_1 is a real non-negative number.

B. Projective measurement

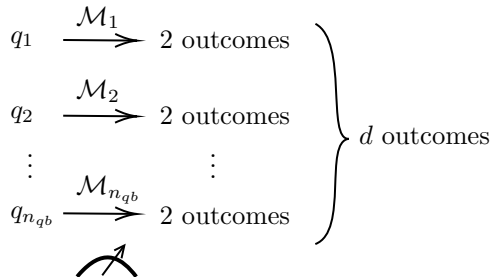
According to Section 2.2.5 in [1] a projective measurement is defined by a Hermitian matrix \mathbf{B} whose distinct eigenvalues m_k are the possible outcomes of the measurement. The probability of getting m_k when measuring a pure state represented by \mathbf{v} is $\varrho_k(\mathbf{v})$: the squared norm of the projection of \mathbf{v} on the eigenspace associated with m_k .

A projective measurement can have the following properties:

- *d*-outcome: If \mathbf{B} has d distinct eigenvalues. The eigenstates are then 1-dimensional spaces, and $\varrho_k(\mathbf{v}) = |\mathbf{e}_k^* \mathbf{v}|^2$ where \mathbf{e}_k is a d -element unit-norm vector belonging to the k -th eigenspace of \mathbf{B} ($*$ is the transconjugate). By performing the measurement several times on copies of the state, we can therefore estimate $\{|\mathbf{e}_k^* \mathbf{v}|^2\}_{1 \leq k \leq d}$ where \mathbf{e}_k spans an orthonormal basis.
- Unentangled: If the measurement can be performed with simultaneous local measurements on each qubit. For example if the matrix \mathbf{B} can be written as a tensor product of n_{qb} matrices with dimension 2×2 : $\mathbf{B} = \mathbf{B}_1 \otimes \dots \otimes \mathbf{B}_{n_{qb}}$. Then the measurement represented by \mathbf{B} can be performed by measuring simultaneously each qubit with the measurements represented by $\mathbf{B}_1, \dots, \mathbf{B}_{n_{qb}}$ and then computing the product of all the outcomes. Depending on how the 2 eigenvalues of each \mathbf{B}_k are chosen, computing the product of the outcomes can result in a loss of information (i.e. the outcomes of each \mathbf{B}_k cannot be retrieved from their product knowing their respective 2 possible values). If this is the case \mathbf{B} will have fewer than d distinct eigenvalues (see e.g. Pauli measurements).

C. Parallel unentangled measurement

If a quantum measurement represented by \mathbf{B} is unentangled and has d outcomes, we call it a parallel unentangled measurement. It can be performed on single qubits in parallel using the following setup:



Each qubit $q_1, \dots, q_{n_{qb}}$ composing the system is measured with a one-qubit measurement which has two distinct outcomes.

The eigenvalues of \mathbf{B} are not important in the present paper. Changing them without changing the eigenvectors would change \mathbf{B} but (as long as the eigenvalues remain distinct) the resulting measurement would be equivalent in the sense that it would give the same information; the possible measured values would then depend on \mathbf{B} but the associated probabilities would be the same up to a permutation, and this is what matters in the present paper.

Since the parallel unentangled measurements we consider have d outcomes (by definition), $\varrho_k(\mathbf{v}) = |\mathbf{e}_k^* \mathbf{v}|^2$ where \mathbf{e}_k is a d -element unit-norm vector belonging to the k -th eigenspace.

For a given parallel unentangled measurement \mathcal{M} let us define the eigenvector matrix: $\mathbf{E}_{\mathcal{M}} = [\mathbf{e}_1 \dots \mathbf{e}_d]$. The vector $\mathbf{p}_{\mathcal{M}}(\mathbf{v}) = |\mathbf{E}_{\mathcal{M}}^* \mathbf{v}|^2$ contains the d probabilities of the possible outcomes.

By performing several measurements on copies of the state represented by \mathbf{v} , we compute the frequencies of occurrence of each outcome, we get $\widehat{\mathbf{p}}_{\mathcal{M}}$, which we use as an approximation of $|\mathbf{E}_{\mathcal{M}}^* \mathbf{v}|^2$. We call $\widehat{\mathbf{p}}_{\mathcal{M}}$ the averaged measurements or sample probabilities. The sum of the elements of $\mathbf{p}_{\mathcal{M}}(\mathbf{v})$ is one (it is the sum of the probabilities of all possible outcomes), so no information is lost by removing one element. We define $\mathbf{E}_{\underline{\mathcal{M}}}$ the non-redundant eigenvector matrix as composed of the first $d - 1$ columns of $\mathbf{E}_{\mathcal{M}}$. Then $\mathbf{p}_{\mathcal{M}}(\mathbf{v}) = |\mathbf{E}_{\mathcal{M}}^* \mathbf{v}|^2$ is redundant but $\mathbf{p}_{\underline{\mathcal{M}}}(\mathbf{v}) = |\mathbf{E}_{\underline{\mathcal{M}}}^* \mathbf{v}|^2$ is not.

$\mathbf{E}_{\mathcal{M}}^*$, $\mathbf{E}_{\underline{\mathcal{M}}}^*$, $\mathbf{p}_{\underline{\mathcal{M}}}(\mathbf{v})$ and $\mathbf{p}_{\mathcal{M}}(\mathbf{v})$ will all be used at different points of this article with \mathcal{M} replaced by the actual measurements we will perform.

D. Considered types of measurements

We perform measurements for all qubits in parallel, with one measurement direction per qubit. For one qubit, we choose to perform measurements that are equivalent (up to a factor 1/2 on the outcomes) to the 3 non-trivial Pauli measurements. The measurement matrices \mathbf{B} associated with the directions X, Y and Z are the last three Pauli matrices defined in Section 2.1.3 of [1] and the corresponding eigenvector matrices may be shown to read:

$$\mathbf{E}_X = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \mathbf{E}_Y = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix} \quad \mathbf{E}_Z = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \quad (1)$$

If the qubit represents the spin of an electron, those eigenvector matrices represent the measurement of the spin component along 3 orthogonal directions.

There is a factor 1/2 between the outcome of the spin measurements and the Pauli measurements but it does not affect the eigenvectors.

For two or more qubits, the different qubits can be measured along X, Y or Z. It can be shown that the resulting eigenvector matrix is the tensor product of the 2-dimensional matrices of (1). For example for 2 qubits, measuring the first one along Z and the second one along X has the following eigenvector matrix

$$\mathbf{E}_{ZX} = \mathbf{E}_Z \otimes \mathbf{E}_X = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}.$$

In this example, if the qubits represent the spins of 2 electrons, then the measurement we perform is equivalent to measuring the first spin component along Z and the second along X. The spin measurement has 4 possible outcomes $(+\frac{1}{2}, +\frac{1}{2})$, $(+\frac{1}{2}, -\frac{1}{2})$, $(-\frac{1}{2}, +\frac{1}{2})$ and $(-\frac{1}{2}, -\frac{1}{2})$ and if \mathbf{v} represents the considered state, the probabilities of each outcome are in the vector $|\mathbf{E}_{ZX}^* \mathbf{v}|^2$. This measurement is not equivalent to a two-qubit Pauli measurement (even if we forget the factor 1/2), as such a Pauli measurement only has two outcomes. In fact, the Pauli measurement along ZX would return +1 for spins measurement outcomes $(+\frac{1}{2}, +\frac{1}{2})$ and $(-\frac{1}{2}, -\frac{1}{2})$ and -1 for spins measurement outcomes $(+\frac{1}{2}, -\frac{1}{2})$ and $(-\frac{1}{2}, +\frac{1}{2})$. This is inefficient as half the information is wasted.

For n_{qb} qubits there are $3^{n_{qb}}$ different measurements of this type. Both of the QST methods of Section III and IV as well as the fine tuning algorithms of Section V use a specific subset of all possible measurements.

E. Justification

We think that performing QST using a kind of measurement that is not parallel unentangled (i.e. has fewer than d outcomes or is entangled) should not be recommended in practice with the current state of quantum computers for the following reasons:

- Performing a quantum measurement that has fewer than d outcomes is suboptimal. Indeed, instead of considering a j -outcome measurement \mathcal{M}_j ($j < d$) we can use a d -outcome measurement \mathcal{M}_d that has the same eigenvectors and d distinct eigenvalues. With this definition it is strictly better to use \mathcal{M}_d than \mathcal{M}_j in all situations, as the outcomes of \mathcal{M}_d can be mapped injectively onto the outcomes of \mathcal{M}_j but the reverse is not true. Therefore \mathcal{M}_d brings us strictly more information on the system

than \mathcal{M}_j and performing either of them should be as difficult (a copy of the state is used up).

- Performing an entangled measurement requires the use of a quantum gate. This gate itself is never going to act exactly as expected and will introduce errors. In order to see if the gate works as expected, we would need to perform quantum process tomography which generally relies on QST.

But the literature on QST is full of theoretical papers that consider measurements that fall within the two types that we do not recommend. Here are some examples:

- [9] uses successive 2-outcome projective measurements on non-orthogonal entangled eigenstates. And each iteration of the algorithm would require a new type of measurement (that depends on what has been measured before and is most likely going to be entangled) and therefore a new quantum gate has to be built on the fly.
- [13] considers projective 2-outcome measurements on 1-dimensional spaces. Half of those measurements can be performed using a single parallel unentangled measurement (with the identity matrix as eigenvector matrix); but the other half cannot.
- [15] considers 2 parallel unentangled measurements (called local measurements in [15]) and 2 d -outcome entangled measurements that can be mapped on the other two using a gate that performs the Fourier transform. This setup is way more reasonable than the others as it requires a single known standard gate.
- [2], [3], [4], [5], [6] and [7] all use multiqubit Pauli measurements. Multiqubit Pauli measurements have the advantage of being unentangled and also simplify the calculation for the QST of mixed states (see the beginning of Section 8.4.2 in [1], (8.149) only works for orthogonal sets of matrices with respect to the Hilbert-Schmidt inner product, like Pauli matrices). They have the disadvantage of being 2-outcome measurements returning either +1 or -1. There are sets of Pauli measurements whose expected values can be deduced from the outcomes of parallel unentangled measurements without loss of information ([12] explains how it can be done for two qubits). But that is not the case for any set of Pauli measurements.

In contrast to those articles we here make a point to only use unentangled parallel measurements. We could have chosen other matrices than (1). We chose those matrices in order to be closer to the Pauli measurements widely used in the literature.

III. TOMOGRAPHY WITH MINIMAL NUMBER OF MEASUREMENT TYPES

The current section describes our first QST setup, Section III A describes the 4 types of parallel unentangled measurements that are performed, Section III B explains why it is reasonable to think that they are injective up to a global phase and Section III C describes a first algorithm to recover the phases.

A. Types of measurements

In the QST method described here, we perform 4 types of measurements on the considered d -dimensional state: The first measurement measures all the qubits along Z, its eigenvector matrix, $\mathbf{E}_{Z\dots Z}$ is the identity matrix, the second measurement measures all the qubits along Y, the third along X, and the fourth measures every odd-numbered qubit along X and every even-numbered qubit along Y.

After performing the measurements several times on copies of the state, we compute the sample probabilities $\widehat{\mathbf{p}}_{\mathcal{M}}$ for \mathcal{M} spanning the 4 types of measurements. We then have an $n_s = 4d$ dimensional vector with $n_s = 4(d-1)$ degrees of freedom. We call it $\widehat{\mathbf{p}}_s$. The associated theoretical probability vector is $\mathbf{p}_s = |\mathbf{A}_s \mathbf{v}|^2$, where s stands for ‘‘small’’ because the corresponding matrix in Section IV has more rows. \mathbf{A}_s is the concatenation of the transconjugates of the eigenvector matrices of the measurements we perform, $\underline{\mathbf{A}}_s$ is defined similarly

$$\mathbf{A}_s = \begin{bmatrix} \mathbf{E}_{Z\dots Z}^* \\ \mathbf{E}_{Y\dots Y}^* \\ \mathbf{E}_{X\dots X}^* \\ \mathbf{E}_{XYXY\dots}^* \end{bmatrix} \quad \text{and} \quad \underline{\mathbf{A}}_s = \begin{bmatrix} \mathbf{E}_{Z\dots Z}^* \\ \mathbf{E}_{Y\dots Y}^* \\ \mathbf{E}_{X\dots X}^* \\ \mathbf{E}_{XYXY\dots}^* \end{bmatrix}. \quad (2)$$

Let us define $\underline{\mathbf{p}}_s = |\underline{\mathbf{A}}_s \mathbf{v}|^2$. Since the norm of \mathbf{v} is 1, \mathbf{p}_s and $\underline{\mathbf{p}}_s$ contain the same information (see Section III C). In Section III B we will consider \mathbf{A}_s , n_s and $\underline{\mathbf{p}}_s$ in order to see if the measurements are injective because we do not want to introduce redundancy when counting the measurements. But, for the sake of simplicity, we will consider \mathbf{A}_s , n_s and \mathbf{p}_s in Section III C in order to recover the state from the measurements. We want to use all the measurements from $\widehat{\mathbf{p}}_s$ whether they are redundant or not.

B. Injectivity

$\underline{\mathbf{A}}_s$ is an $n_s \times d$ matrix and \mathbf{v} has unit norm. We want to know whether the measurements we chose

are sufficient to recover any \mathbf{v} from $|\underline{\mathbf{A}}_s \mathbf{v}|^2$ up to a global phase. In the rest of the paper this property will be called injectivity. It is a bit of an exaggeration because $\mathbf{v} \rightarrow |\underline{\mathbf{A}}_s \mathbf{v}|^2$ is never truly injective as changing the global phase of \mathbf{v} will not change $|\underline{\mathbf{A}}_s \mathbf{v}|^2$. This issue of injectivity was studied before in [14], [16], [18] in a slightly different setup: the considered measurements are $|\underline{\mathbf{A}}_s \mathbf{v}|$ instead of $|\underline{\mathbf{A}}_s \mathbf{v}|^2$, this does not change anything for the injectivity, also \mathbf{v} is not assumed to have unit norm, and this is important. In order to reconcile the two setups we can relax the unit-norm hypothesis for \mathbf{v} and insert the row $[0, \dots, 0, 1]$ between the $(d-1)$ -th row and the d -th row of $\underline{\mathbf{A}}_s$. This ensures that the norm of \mathbf{v} is constrained: its square is the sum of the first d constrained measurements, because the first d rows of $\underline{\mathbf{A}}_s$ are the identity matrix. With this change $\underline{\mathbf{A}}_s$ has $4d-3$ rows.

According to [14] the minimal number of rows for $\underline{\mathbf{A}}_s$ below which injectivity is impossible is $4d-3-c(d)n_{qb}$ rows for some $c(d) \in [1, 2]$. Since we have $4d-3$ rows, this necessary condition is satisfied. However there is no simple sufficient condition on $\underline{\mathbf{A}}_s$ that ensures injectivity, and proving it for a given $\underline{\mathbf{A}}_s$ is a known hard problem. The closest result we found to a sufficient condition is in [16] where it is shown that for a generic $\underline{\mathbf{A}}_s$, having $4d-2$ or more rows ensures injectivity. $\underline{\mathbf{A}}_s$ must be generic in the sense that it is part of a specific open dense set with full measure. We cannot identify this set and check that $\underline{\mathbf{A}}_s$ would be in it (although it probably would because the set is of full measure), but this is moot because we are one row short of satisfying the $4d-2$ condition anyway. However [18] explained why it is natural to think that $4d-4$ is the actual lower bound. It remains a conjecture though.

We can be sure that 3 measurement types would not be enough to achieve injectivity with $n_{qb} > 2$ as the bound of [14] would not be fulfilled: we would have $3d-2$ independent rows ($3d-3$ plus the unit-norm constraint). This is always strictly smaller than $4d-3-2n_{qb}$ for $n_{qb} > 2$. 4 is the lowest number of measurement types for which we can hope to always achieve injectivity.

C. A first quantum pure state tomography method

In the current section, we show how the method proposed in [17] can be used in our framework to recover \mathbf{v} from the sample probabilities $\widehat{\mathbf{p}}_s$, an estimate of $\mathbf{p}_s = |\mathbf{A}_s \mathbf{v}|^2$ (we only consider \mathbf{A}_s from now on, $\underline{\mathbf{A}}_s$ was only useful to discuss the injectivity). The optimization problem considered in [17] is the follow-

ing:

$$\min_{\mathbf{v}} \left\| |\mathbf{A}_s \mathbf{v}| - \sqrt{\widehat{\mathbf{p}}_s} \right\| \quad (3)$$

where $\sqrt{\widehat{\mathbf{p}}_s}$ is the element-wise square root of $\widehat{\mathbf{p}}_s$ and $\|\cdot\|$ is the L_2 norm. [17] does not include the unit-norm constraint on \mathbf{v} but, since we use \mathbf{A}_s , this constraint is implicit in the criterion to be minimized. In fact, the sum of the first d elements of $|\mathbf{A}_s \mathbf{v}|^2$ is the squared norm of \mathbf{v} and the sum of the first d elements of $\widehat{\mathbf{p}}_s$ is one, therefore if $|\mathbf{A}_s \mathbf{v}|$ is close to $\sqrt{\widehat{\mathbf{p}}_s}$, their squared norms will also be close, and therefore the squared norm of \mathbf{v} will be close to 1. In [17], it is shown that (3) is equivalent to the following optimization problem (originally it came from [19]):

$$\min_{\mathbf{U}} \text{tr}(\mathbf{U}\mathbf{M}) \quad (4)$$

s.t. \mathcal{C}

where $\mathbf{M} = \text{diag}(\widehat{\mathbf{p}}_s)(I - \mathbf{A}_s \mathbf{A}_s^\dagger) \text{diag}(\widehat{\mathbf{p}}_s)$, \dagger is the pseudo-inverse, $\text{diag}(\widehat{\mathbf{p}}_s)$ is the diagonal matrix whose diagonal is $\widehat{\mathbf{p}}_s$ and \mathcal{C} represents the following condition on the $n_s \times n_s$ matrix \mathbf{U} :

$$\exists \mathbf{u} \in \mathbb{C}^{n_s} \text{ such that } |\mathbf{u}| = [1, \dots, 1]^T \text{ and } \mathbf{U} = \mathbf{u}\mathbf{u}^*. \quad (5)$$

[17] shows that if \mathbf{U} is a solution of (4), then the associated \mathbf{u} of (5) is an approximation of the phase of $\mathbf{A}_s \mathbf{v}$, and the resulting estimate of \mathbf{v} defined as:

$$\widehat{\mathbf{v}}_0 = \mathbf{A}_s^\dagger (\mathbf{u} * \sqrt{\widehat{\mathbf{p}}_s}) \quad (6)$$

(* is the element-wise product) is the solution of (3) proposed in [17].

(4) is almost a convex optimization problem. In fact if \mathcal{C} is reformulated in an equivalent way: $\mathbf{U}_{i,i} = 1 \forall i \in [1, n_s]$, $\mathbf{U} \succeq 0$, $\text{Rank}(\mathbf{U}) = 1$ ($\mathbf{U} \succeq 0$ means that \mathbf{U} is both Hermitian and non-negative definite), according to [17] the criterion $\text{tr}(\mathbf{U}\mathbf{M})$ is convex and the only constraint that makes the problem non-convex in \mathcal{C} is $\text{Rank}(\mathbf{U}) = 1$. By relaxing it we have a convex problem that can be solved without the need for a good initialization:

$$\min_{\mathbf{U} \text{ s.t. } \mathbf{U}_{i,i}=1 \forall i, \mathbf{U} \succeq 0} \text{tr}(\mathbf{U}\mathbf{M}). \quad (7)$$

Once (7) is solved using the PhaseCut algorithm of [17], the eigenvectors and eigenvalues of the solution \mathbf{U} are computed. In order to get an estimate of \mathbf{u} , [17] then computes $\widehat{\mathbf{u}}$, the eigenvector associated with the largest eigenvalue. From $\widehat{\mathbf{u}}$, we get the estimate of \mathbf{v} defined in (6):

$$\widehat{\mathbf{v}}_{pc} = \mathbf{A}_s^\dagger (\widehat{\mathbf{u}} * \sqrt{\widehat{\mathbf{p}}_s}). \quad (8)$$

In [17] this method is tested with \mathbf{A} matrices which represent usual use-cases in the signal/image processing community (oversampled Fourier transform, multiple random illumination filters, wavelet transform) for which PhaseCut works well. However for $\mathbf{A} = \mathbf{A}_s$, PhaseCut is a good initial point but needs the fine tuning that we will detail in Section V.

D. Comparison with the literature

Let us sum up the main features of our first QST algorithm:

- It uses $4d$ probabilities that can be obtained by averaging the results of 4 parallel unentangled measurements.
- It is reasonable to think that the chosen measurements are injective (the failure set is most likely empty).
- The algorithm that reconstructs the state is not explicit (optimization).

Goyeneche et al. [15] uses the same number of measurement types, has a known failure space of zero measure and provides an explicit reconstruction algorithm. The main advantage our approach based on PhaseCut has compared to [15] is that we do not use unentangled measurements. The more general compressed sensing approach of [4] requires $O(rd \log(d)^2)$ probabilities to estimate the state where r , the rank of the density matrix, is 1 in the case of a pure state. Those probabilities could be obtained by averaging the results of $O(\log(d)^2)$ different unentangled measurements. Our method is more efficient since we use $4 = O(1)$ different unentangled measurements. Both methods have no theoretical guarantee of injectivity or closed-form solution. The validity of the solution can only be shown in simulations.

IV. CLOSED-FORM STATE TOMOGRAPHY ALGORITHM

A. Alternative types of measurements

In the alternative QST method described here, we perform the following measurements

$$\left\{ \underbrace{Z \dots Z}_{n_{qb} \text{ times}}, \left\{ \underbrace{Z \dots Z}_{n_{qb} - i \text{ times}} \quad S \quad \underbrace{X \dots X}_{i - 1 \text{ times}}, 1 \leq i \leq n_{qb} \right\}, S \in \{X, Y\} \right\}$$

The number of types of measurements is $2n_{qb} + 1$. The resulting \mathbf{A}_t (t stands for ‘‘tall’’) matrix has $n_r =$

$d(2n_{qb} + 1)$ rows:

$$\mathbf{A}_t = \begin{bmatrix} \mathbf{E}_{Z\dots Z}^* \\ \mathbf{E}_{Z\dots ZX}^* \\ \mathbf{E}_{Z\dots ZY}^* \\ \vdots \\ \mathbf{E}_{X\dots X}^* \\ \mathbf{E}_{YX\dots X}^* \end{bmatrix}. \quad (9)$$

Each measurement is performed several times and we compute the sample probabilities $\hat{\mathbf{p}}_t$ which are estimates of the theoretical probabilities $\mathbf{p}_t = |\mathbf{A}_t \mathbf{v}|^2$.

$2n_{qb} + 1$ sounds like a lot compared to the 4 measurement types of Section III but it is a small fraction of the $3^{n_{qb}}$ possible types of measurements defined in Section II D. This setup also has the advantage of coming with an attractive way to recover the state from the measurements, as will be explained in Section IV B.

B. A recursive pure quantum state tomography method

Let us show how a vector \mathbf{v} can be recovered up to a global phase from $|\mathbf{A}_t \mathbf{v}|^2$ by induction on the number of qubits.

\mathbf{A}_t depends on n_{qb} , in the rest of the current section this dependence will not be omitted and \mathbf{A}_t will be called $\mathbf{A}_t(n_{qb})$. We first show how to solve the problem (recover \mathbf{v} from $|\mathbf{A}_t \mathbf{v}|^2$) with $n_{qb} = 1$. We then explain how solving the problem for $n_{qb} - 1$ qubits yields the solution for n_{qb} qubits. From there a recursive algorithm can be implemented.

$$n_{qb} = 1: \mathbf{A}_t(1) = \begin{bmatrix} \mathbf{E}_Z^* \\ \mathbf{E}_X^* \\ \mathbf{E}_Y^* \end{bmatrix}, \text{ with the } \mathbf{E}_Z, \mathbf{E}_X, \mathbf{E}_Y$$

of (1). The state vector is $\mathbf{v} = \begin{pmatrix} |v_1| \\ |v_2|e^{i\theta} \end{pmatrix}$. Basic calculations show:

$$|\mathbf{A}_t(1)\mathbf{v}|^2 = \begin{pmatrix} |v_1|^2 \\ |v_2|^2 \\ \frac{1}{2}(|v_1|^2 + |v_2|^2 + 2|v_1||v_2|\cos(\theta)) \\ \frac{1}{2}(|v_1|^2 + |v_2|^2 - 2|v_1||v_2|\cos(\theta)) \\ \frac{1}{2}(|v_1|^2 + |v_2|^2 + 2|v_1||v_2|\sin(\theta)) \\ \frac{1}{2}(|v_1|^2 + |v_2|^2 - 2|v_1||v_2|\sin(\theta)) \end{pmatrix}. \quad (10)$$

Therefore, $|\mathbf{A}_t(1)\mathbf{v}|^2$ gives $|v_1|^2$, $|v_2|^2$, $|v_1||v_2|\cos(\theta)$ and $|v_1||v_2|\sin(\theta)$. From there, we have two cases:

- If $|v_1| = 0$ or $|v_2| = 0$, then knowing $|v_1|$ and $|v_2|$ is enough because $\begin{pmatrix} |v_1| \\ |v_2| \end{pmatrix}$ is the same as \mathbf{v} up to a global phase. Thus, there is no need to compute θ .

- If $|v_1||v_2| > 0$ then we can derive $\cos(\theta)$ and $\sin(\theta)$ from the above-defined quantities and get θ . Thus we know all parameters of \mathbf{v} .

Let us now assume that the state recovery is possible for $n_{qb} - 1$ qubits, i.e. there is a function $f_{n_{qb}-1}$ such that for a vector \mathbf{w} with $2^{n_{qb}-1}$ elements $f_{n_{qb}-1}(|\mathbf{A}_t(n_{qb}-1)\mathbf{w}|^2)$ is equal to \mathbf{w} up to a global phase. Let \mathbf{v} be a $d = 2^{n_{qb}}$ element vector (it does not have to be unit-norm). We split \mathbf{v} into two $2^{n_{qb}-1}$ element vectors \mathbf{w}_1 and \mathbf{w}_2 : $\mathbf{v} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}$. Let us show how \mathbf{v} can be recovered up to a global phase from $|\mathbf{A}_t(n_{qb})\mathbf{v}|^2$ using the fact that \mathbf{w}_1 and \mathbf{w}_2 can be recovered from $|\mathbf{A}_t(n_{qb}-1)\mathbf{w}_1|^2$ and $|\mathbf{A}_t(n_{qb}-1)\mathbf{w}_2|^2$ up to global phases using $f_{n_{qb}-1}$. We start by comparing $\mathbf{A}_t(n_{qb}-1)$ to $\mathbf{A}_t(n_{qb})$:

$$\mathbf{A}_t(n_{qb}-1) = \begin{bmatrix} \mathbf{E}_{s_1}^* \\ \vdots \\ \mathbf{E}_{s_{2^{n_{qb}-1}}}^* \end{bmatrix}$$

with (9) giving the values of the strings $s_1, \dots, s_{2^{n_{qb}-1}}$. We can also notice that:

$$\mathbf{A}_t(n_{qb}) = \begin{bmatrix} \mathbf{E}_{Zs_1}^* \\ \vdots \\ \mathbf{E}_{Zs_{2^{n_{qb}-1}}}^* \\ \mathbf{E}_{X\dots X}^* \\ \mathbf{E}_{YX\dots X}^* \end{bmatrix} \quad (11)$$

where Zs_k is the string made up of Z followed by s_1 . Using the definition of \mathbf{E} in Section II D, we have:

$$\mathbf{E}_{Zs_k}^* = \mathbf{E}_Z^* \otimes \mathbf{E}_{s_k}^* = \begin{bmatrix} \mathbf{E}_{s_k}^* & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_{s_k}^* \end{bmatrix} \forall k. \quad (12)$$

Let k be an integer ranging from 1 to $2n_{qb} - 1$, from (11) and (12), we have:

$$|\mathbf{A}_t(n_{qb})\mathbf{v}|_{i_k}^2 = \left| \begin{bmatrix} \mathbf{E}_{s_k}^* & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_{s_k}^* \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} \right|^2 = \begin{bmatrix} |\mathbf{E}_{s_k}^* \mathbf{w}_1|^2 \\ |\mathbf{E}_{s_k}^* \mathbf{w}_2|^2 \end{bmatrix} \quad (13)$$

where $|\mathbf{A}_t(n_{qb})\mathbf{v}|_{i_k}^2$ is the vector that contains the elements of $|\mathbf{A}_t(n_{qb})\mathbf{v}|^2$ indexed between $(k-1)d+1$ and kd . And using the same notation for $|\mathbf{A}_t(n_{qb}-1)\mathbf{w}_l|^2$ with l being either 1 or 2, we have

$$|\mathbf{A}_t(n_{qb}-1)\mathbf{w}_l|^2 = |\mathbf{E}_{s_k}^* \mathbf{w}_l|^2. \quad (14)$$

From (14) and (13), we see that all the elements of $|\mathbf{A}_t(n_{qb}-1)\mathbf{w}_l|^2$ are in $|\mathbf{A}_t(n_{qb})\mathbf{v}|_{i_k}^2 \forall k \in \{1, \dots, 2n_{qb} - 1\}$. Since $|\mathbf{A}_t(n_{qb}-1)\mathbf{w}_l|^2 \forall k \in \{1, \dots, 2n_{qb} - 1\}$ spans all the vector $|\mathbf{A}_t(n_{qb}-1)\mathbf{w}_l|^2$

we have shown that $|\mathbf{A}_t(n_{qb} - 1)\mathbf{w}_l|^2$ is known from part of the measurements ($|\mathbf{A}_t(n_{qb})\mathbf{v}|^2$) for $l = 1$ and $l = 2$.

Using the induction hypothesis we can apply $f_{n_{qb}-1}$ to the known quantities $|\mathbf{A}_t(n_{qb} - 1)\mathbf{w}_1|^2$ and $|\mathbf{A}_t(n_{qb} - 1)\mathbf{w}_2|^2$ in order to get \mathbf{w}_1 and \mathbf{w}_2 up to global phases. Let us call our estimates $\widehat{\mathbf{w}}_1$ and $\widehat{\mathbf{w}}_2$, $\mathbf{w}_1 = e^{i\theta_1}\widehat{\mathbf{w}}_1$ and $\mathbf{w}_2 = e^{i\theta_2}\widehat{\mathbf{w}}_2$. We now only need to know $\theta_2 - \theta_1$ in order to know \mathbf{v} up to a global phase. Let us get $\theta_2 - \theta_1$ from the last $2d$ elements of $|\mathbf{A}_t(n_{qb})\mathbf{v}|^2$. We define \mathbf{L}_m as the column vector containing those last $2d$ elements

$$\mathbf{L}_m = \left| \begin{bmatrix} \mathbf{E}_{XX\dots X}^* \\ \mathbf{E}_{YX\dots X}^* \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} \right|^2 = \left| \begin{bmatrix} \mathbf{E}_X^* \otimes \mathbf{E}_{X\dots X}^* \\ \mathbf{E}_Y^* \otimes \mathbf{E}_{X\dots X}^* \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} \right|^2$$

where on the left-hand side the strings $XX\dots X$, $YX\dots X$ have n_{qb} characters and on the right-hand side $X\dots X$ have $n_{qb} - 1$ characters. By replacing \mathbf{E}_X and \mathbf{E}_Y by their values of Section IID and calculating the tensor products, we get

$$\begin{aligned} \mathbf{L}_m &= \left| \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{E}_{X\dots X}^* \mathbf{w}_1 + \mathbf{E}_{X\dots X}^* \mathbf{w}_2 \\ \mathbf{E}_{X\dots X}^* \mathbf{w}_1 - \mathbf{E}_{X\dots X}^* \mathbf{w}_2 \\ \mathbf{E}_{X\dots X}^* \mathbf{w}_1 - i\mathbf{E}_{X\dots X}^* \mathbf{w}_2 \\ \mathbf{E}_{X\dots X}^* \mathbf{w}_1 + i\mathbf{E}_{X\dots X}^* \mathbf{w}_2 \end{bmatrix} \right|^2 \\ &= \frac{1}{2} \left| \begin{bmatrix} \mathbf{E}_{X\dots X}^* \widehat{\mathbf{w}}_1 e^{i\theta_1} + \mathbf{E}_{X\dots X}^* \widehat{\mathbf{w}}_2 e^{i\theta_2} \\ \mathbf{E}_{X\dots X}^* \widehat{\mathbf{w}}_1 e^{i\theta_1} - \mathbf{E}_{X\dots X}^* \widehat{\mathbf{w}}_2 e^{i\theta_2} \\ \mathbf{E}_{X\dots X}^* \widehat{\mathbf{w}}_1 e^{i\theta_1} - i\mathbf{E}_{X\dots X}^* \widehat{\mathbf{w}}_2 e^{i\theta_2} \\ \mathbf{E}_{X\dots X}^* \widehat{\mathbf{w}}_1 e^{i\theta_1} + i\mathbf{E}_{X\dots X}^* \widehat{\mathbf{w}}_2 e^{i\theta_2} \end{bmatrix} \right|^2. \end{aligned}$$

Let us introduce the following notations

$$\begin{aligned} \mathbf{m} &= \frac{1}{2} |\mathbf{E}_{X\dots X}^* \widehat{\mathbf{w}}_1|^2 + \frac{1}{2} |\mathbf{E}_{X\dots X}^* \widehat{\mathbf{w}}_2|^2 \\ \mathbf{d}_c &= \overline{\mathbf{E}_{X\dots X}^* \widehat{\mathbf{w}}_1} * \mathbf{E}_{X\dots X}^* \widehat{\mathbf{w}}_2 \\ \mathbf{d}(\theta) &= \cos(\theta) \text{Re}(\mathbf{d}_c) - \sin(\theta) \text{Im}(\mathbf{d}_c) \end{aligned} \quad (15)$$

where $*$ again represents the element-wise product between two vectors and $\bar{\cdot}$ is the conjugate. $\widehat{\mathbf{w}}_1$ and $\widehat{\mathbf{w}}_2$ are known quantities (from $|\mathbf{A}_t(n_{qb})\mathbf{v}|^2$) so \mathbf{m} and \mathbf{d}_c are known and $\mathbf{d}(\theta)$ can be computed for any $\theta \in [0, 2\pi]$. Let us rewrite \mathbf{L}_m as a function of $(\theta_2 - \theta_1)$ using those quantities

$$\mathbf{L}_m(\theta_2 - \theta_1) = \begin{bmatrix} \mathbf{m} + \mathbf{d}(\theta_2 - \theta_1) \\ \mathbf{m} - \mathbf{d}(\theta_2 - \theta_1) \\ \mathbf{m} + \mathbf{d}(\theta_2 - \theta_1 - \pi/2) \\ \mathbf{m} - \mathbf{d}(\theta_2 - \theta_1 - \pi/2) \end{bmatrix}. \quad (16)$$

We aim at deriving $\theta_2 - \theta_1$ from \mathbf{L}_m (which is known from the measurements). We first notice from the definition of $\mathbf{d}(\theta)$ in (15) that if \mathbf{d}_c is 0 on every component then $\mathbf{d}(\theta_2 - \theta_1)$ is also 0 on every component (which means it does not depend on $\theta_2 - \theta_1$) and \mathbf{L}_m is simply \mathbf{m} repeated 4 times (see (16)). Therefore recovering $\theta_2 - \theta_1$ (and \mathbf{v}) from \mathbf{L}_m is impossible. However, we hereafter show that this is the only case

when $\theta_2 - \theta_1$ cannot be recovered from \mathbf{L}_m . And the ensemble of \mathbf{v} which make this occur has zero measure.

Let us assume that at least a single element of \mathbf{d}_c is not zero, let us call k its index, d_k the corresponding non-zero element (we take the element which has the highest modulus), and call $d_k(\theta_2 - \theta_1)$ and m_k the k -th elements of $\mathbf{d}(\theta_2 - \theta_1)$ and \mathbf{m} respectively. Then all we need is the k -th and $(k + d)$ -th elements of \mathbf{L}_m whose expressions are $m_k + \cos(\theta_2 - \theta_1) \text{Re}(d_k) - \sin(\theta_2 - \theta_1) \text{Im}(d_k)$ and $m_k + \sin(\theta_2 - \theta_1) \text{Re}(d_k) + \cos(\theta_2 - \theta_1) \text{Im}(d_k)$. Those known elements can be put in a column vector and re-written as:

$$\begin{pmatrix} \text{Re}(d_k) & -\text{Im}(d_k) \\ \text{Im}(d_k) & \text{Re}(d_k) \end{pmatrix} \begin{pmatrix} \cos(\theta_2 - \theta_1) \\ \sin(\theta_2 - \theta_1) \end{pmatrix}. \quad (17)$$

The 2×2 matrix on the left-hand side is known (since d_k is known) and invertible (since its determinant is $|d_k|^2 > 0$). Therefore $\theta_2 - \theta_1$ can be recovered (because we have its sine and cosine) from 2 elements of \mathbf{L}_m (so two probabilities).

We could stop there and get an estimate θ_d of $\theta_2 - \theta_1$ that is computed using two elements of \mathbf{L}_m . But, in practice the sample probabilities give an imperfect estimate of \mathbf{L}_m which we call $\widehat{\mathbf{L}}_m$. In order to be robust to the errors, we aim to find the angle $\widehat{\theta_2 - \theta_1}$ that minimizes $\|\mathbf{L}_m(\theta_2 - \theta_1) - \widehat{\mathbf{L}}_m\|$, this way we use all sample probabilities and not just two. We use a quasi-Newton BFGS algorithm [20] (implemented with `fminunc` in the Matlab numerical software) initialized at θ_d , the optimization stops when the step is smaller than 10^{-30} . Technically with this optimization, the algorithm is no longer closed-form but, since it involves a single parameter, it is really fast, and improves the performances quite significantly so we choose to perform it anyway. If the readers want a real closed-form algorithm, they can use θ_d instead of computing $\widehat{\theta_2 - \theta_1}$, or use a closed-form optimization algorithm with a fixed number of steps to compute $\widehat{\theta_2 - \theta_1}$.

Let us now take a step back and summarize what we have proved in this section:

- Recovering the state (up to a global phase) from the measurements is possible for $n_{qb} = 1$.
- Assuming it is possible for $n_{qb} - 1$ we showed it is also possible for n_{qb} unless the state is in an ensemble of zero measure.

Using those previous two results, we can construct a recursive algorithm that recovers \mathbf{v} from the measurements. It will work except on the union of a finite number of failure sets of zero measure which would

also be of zero measure. The estimate given by this recursive algorithm will be called $\hat{\mathbf{v}}_{rec}$.

C. Discussion about the number of probabilities used

The recursive algorithm of the previous section calls itself twice for each reduction of the number of qubits by 1. This means that for n_{qb} , it is called once with n_{qb} qubits, twice with $n_{qb} - 1$ qubits, ..., $2^{n_{qb}-1}$ times with 1 qubit.

For 1 qubit, the state is recovered using (10) which involve 6 probabilities, among which only 4 are required (we could obtain the same result without using the fourth and sixth elements of $|\mathbf{A}_t(1)\mathbf{v}|^2$).

For $q > 1$ qubit before calling the recursive function with one fewer qubit, we compute $\theta_2 - \theta_1$ using (16). This involves 2×2^q probabilities among which only 2 are strictly required for the first estimate θ_d .

The minimum number of needed probabilities is $4 \times 2^{n_{qb}-1} + 2 \sum_{q=2}^{n_{qb}} 2^{n_{qb}-q} = 2d + 2(2^{n_{qb}-1} - 1) = 3d - 2$. Furthermore, if we take into account the fact that \mathbf{v} has unit norm, then one of the probabilities along the Z axis (which are all used) becomes redundant, and this number becomes $3(d - 1)$.

In practice all probabilities are used in order to minimize the impact of the statistical error on the probabilities. But if we wanted to remove rows from \mathbf{A}_t in (9) and only keep $3(d-1)$ of them, we could still achieve QST. However, this is a bad idea because we would no longer have a concatenation of d -outcome parallel measurements. And in practice the final estimate of the state would be less robust to the errors on the sample probabilities and the quantum setup would not be any easier to put in place, as the estimation of the $3(d-1)$ probabilities to be kept requires all $2n_{qb} + 1$ measurements to be performed anyway.

D. Comparison with the literature

Let us sum up the main features of our second QST algorithm:

- It uses $(2n_{qb} + 1)d$ probabilities that can be obtained by averaging the results of $2n_{qb} + 1$ parallel unentangled measurements.
- The measurements are injective outside a known failure set with zero measure.
- The algorithm that reconstructs the state is explicit.

Those features are very similar to those of Goyeneche et al. [15]. The advantage of our method is that the measurements it uses are unentangled. Its drawback is that it requires $2n_{qb} + 1$ measurements which is more than 4 (except for the trivial case $n_{qb} = 1$). That is the price to pay for using only unentangled measurements, we could not find a simple closed-form algorithm that works with fewer types of unentangled measurements. The more general compressed sensing approach of [4] requires $O(rd \log(d)^2)$ probabilities to estimate the state where r , the rank of the density matrix, is 1 in the case of a pure state. Those probabilities could be obtained by averaging the results of $O(\log(d)^2)$ different unentangled measurements. We do better here since we only use $2n_{qb} + 1 = O(\log(d))$ measurements. We also have the advantage of providing a closed-form algorithm contrary to the method of [4] which is very general (works for mixed states and any kind of measurement), but uses an optimization algorithm and provides no proof of injectivity.

V. LIKELIHOOD MAXIMIZATION

A. Main idea

Sections III and IV give us estimates of the state \mathbf{v} , denoted as $\hat{\mathbf{v}}_{pc}$ and $\hat{\mathbf{v}}_{rec}$ respectively. $\hat{\mathbf{v}}_{pc}$ is the solution of the QST problem with one constraint ($rank(\mathbf{U}) = 1$) relaxed, so it can be inaccurate even in the absence of errors in the sample probabilities. The algorithm of Section IV B that computes $\hat{\mathbf{v}}_{rec}$ is also imperfect. It relies heavily on the measurements along $Z\dots Z$, $Z\dots ZX$ and $Z\dots ZY$ (used $2^{n_{qb}-1}$ times for one qubit at the end of the recursive tree to compute all the moduli and half the phases differences) and it almost does not use the measurements along $X\dots X$ and $YX\dots X$ (used only once to compute one phase difference ($\theta_2 - \theta_1$) with (16)). Each of those last two measurements contains as much information on \mathbf{v} as the measurements along $Z\dots Z$, but the former are barely used.

Therefore the estimation methods of Section III and IV are hereafter supplemented by a final tuning to make them more precise. To this end, we take a maximum likelihood (ML) approach:

$$(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = \arg \min_{\mathbf{x}, \mathbf{y} \text{ s.t. } \|\mathbf{x}\|_2 + \|\mathbf{y}\|_2 < 1} \mathcal{L}_{(\mathbf{x}, \mathbf{y})}(\hat{\mathbf{p}}) \quad (18)$$

where $\hat{\mathbf{p}}$ is the vector that contains sample probabilities and $\mathcal{L}_{(\mathbf{x}, \mathbf{y})}(\hat{\mathbf{p}})$ is to be understood as the negative log-likelihood of measuring the sample probabilities $\hat{\mathbf{p}}$ if the true state is $\mathbf{v}(\mathbf{x}, \mathbf{y})$, with \mathbf{x} and \mathbf{y} defined hereafter. In the whole paper, whenever we write

“negative log-likelihood” (or \mathcal{L}) we mean “opposite of the log-likelihood up to additive and positive multiplicative constants”. These constants will not matter as the negative log-likelihood will be minimized. The vector $\mathbf{v}(\mathbf{x}, \mathbf{y})$ with respect to which \mathcal{L} will be minimized is defined as:

$$\mathbf{v}(\mathbf{x}, \mathbf{y}) = [\sqrt{1 - \|\mathbf{x}\|_2^2 - \|\mathbf{y}\|_2^2}, x_1 + iy_1, \dots, x_{d-1} + iy_{d-1}]^T.$$

\mathbf{x} and \mathbf{y} are $d - 1$ element vectors representing the real and imaginary parts of the last elements of \mathbf{v} . The constraint in (18) is $r^2 < 1$ (with $r = \sqrt{\|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2}$) and not $r^2 \leq 1$ because optimization is easier on an open set). We mitigate the effect of this imperfect constraint by permuting the first component of \mathbf{v} and the component of \mathbf{v} with the highest modulus at the initial point of the optimization. Thus, we ensure that r^2 is not going to be close to 1 unless the initial point was way off. The sample probabilities and the columns of \mathbf{A} are permuted in the same way. Those change are limited to the optimization algorithm.

Since the optimization set is open we can change the variables in order to remove the constraint altogether:

$$\mathbf{x}' = \frac{\tan(\frac{\pi}{2}r)}{r} \mathbf{x} \text{ and } \mathbf{x} = \frac{\frac{2}{\pi} \operatorname{atan}(r')}{r'} \mathbf{x}'$$

$$\mathbf{y}' = \frac{\tan(\frac{\pi}{2}r)}{r} \mathbf{y} \text{ and } \mathbf{y} = \frac{\frac{2}{\pi} \operatorname{atan}(r')}{r'} \mathbf{y}'$$

(with $r' = \sqrt{\|\mathbf{x}'\|_2^2 + \|\mathbf{y}'\|_2^2}$). The new optimization problem on \mathbf{x}' and \mathbf{y}' does not have any constraint, as when r' spans the whole space r remains strictly smaller than 1. Eq. (18) is therefore replaced by

$$(\hat{\mathbf{x}}', \hat{\mathbf{y}}') = \arg \min_{\mathbf{x}', \mathbf{y}'} \mathcal{L}_{(\mathbf{x}', \mathbf{y}')}(\hat{\mathbf{p}}). \quad (19)$$

In order to solve (19) we again use the BFGS algorithm where the analytical expressions of the gradients are provided. The algorithm stops when the norm of the optimization step is smaller than 10^{-30} . Like in most non-convex optimization methods, we need a good initialization point, we use either $\hat{\mathbf{v}}_{pc}$ or $\hat{\mathbf{v}}_{rec}$. The most likely \mathbf{v} is $\hat{\mathbf{v}}_{ml} = \mathbf{v}(\hat{\mathbf{x}}', \hat{\mathbf{y}}')$, with $\hat{\mathbf{x}}', \hat{\mathbf{y}}'$ defined in (19).

All that remains now is to define the expression of the negative log-likelihood \mathcal{L} with respect to \mathbf{v} . In the following 2 subsections we will give 2 expressions for the normalized log-likelihood: $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}(\hat{\mathbf{p}})$ and $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}(\hat{\mathbf{p}})$.

B. Exact likelihood

In [21] the formula for the likelihood of a multi-output quantum measurement is given (albeit for a

mixed state represented by ρ which we would have to replace by $\mathbf{v}\mathbf{v}^*$). It boils down to:

$$\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}(\hat{\mathbf{p}}) = - \sum_{k=1}^{n_{prob}} n_k \log(|\mathbf{A}\mathbf{v}(\mathbf{x}', \mathbf{y}')|^2)_k. \quad (20)$$

$(|\mathbf{A}\mathbf{v}(\mathbf{x}', \mathbf{y}')|^2)_k$ is the k -th element of $|\mathbf{A}\mathbf{v}(\mathbf{x}', \mathbf{y}')|^2$, \mathbf{A} is the measurement matrix, either \mathbf{A}_s or \mathbf{A}_t , n_k is the number of times the k -th outcome occurred i.e. the k -th element of $\hat{\mathbf{p}}$ (either $\hat{\mathbf{p}}_s$ or $\hat{\mathbf{p}}_t$) multiplied by the number of times the measurement is repeated, and n_{prob} is the number of rows of \mathbf{A} .

In order to get to this result we must consider the measurement counts as the realizations of a multinomial random variable. This is not an approximation, this is why we call this likelihood “exact”.

C. Gaussian approximation

In this subsection, we use the central limit theorem to approximate the scaled sample probabilities as the realization of a multivariate normal distribution. It is appropriate as the vector $\hat{\mathbf{p}}$ whose likelihood we want to compute is the average of independent realizations of the same random variable. Its expected value is the vector of theoretical probabilities $\mathbf{p}(\mathbf{x}', \mathbf{y}')$ that depends on the state. Let us define $\boldsymbol{\varepsilon}(\hat{\mathbf{p}}, \mathbf{x}', \mathbf{y}') = \hat{\mathbf{p}} - \mathbf{p}(\mathbf{x}', \mathbf{y}')$ and $\underline{\boldsymbol{\varepsilon}}(\hat{\mathbf{p}}, \mathbf{x}', \mathbf{y}')$ is $\boldsymbol{\varepsilon}(\hat{\mathbf{p}}, \mathbf{x}', \mathbf{y}')$ with the last element removed (no information is lost as the sum of the elements of $\boldsymbol{\varepsilon}(\hat{\mathbf{p}}, \mathbf{x}', \mathbf{y}')$ is 0). In Appendix A, we show that if N is the number of times the measurements have been averaged, then $\sqrt{N}\underline{\boldsymbol{\varepsilon}}(\hat{\mathbf{p}}, \mathbf{x}', \mathbf{y}')$ asymptotically ($N \rightarrow +\infty$) follows a zero-mean multivariate normal distribution. Its covariance matrix $\boldsymbol{\Sigma}$ is computed in Appendix A. $\boldsymbol{\Sigma}$ depends on the theoretical probabilities, we need to remove this dependency. With that in mind, we get to the following approximation for the negative log-likelihood:

$$\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}(\hat{\mathbf{p}}) = N \underline{\boldsymbol{\varepsilon}}(\hat{\mathbf{p}}, \mathbf{x}', \mathbf{y}')^T \tilde{\boldsymbol{\Sigma}}^{-1} \underline{\boldsymbol{\varepsilon}}(\hat{\mathbf{p}}, \mathbf{x}', \mathbf{y}') \quad (21)$$

where $\tilde{\boldsymbol{\Sigma}}^{-1}$ is an approximation of the covariance matrix that uses $\tilde{\mathbf{p}} = \frac{\hat{\mathbf{p}} + \frac{\mathbf{1}}{N}}{1 + \frac{d}{N}}$ as a regularized approximation of \mathbf{p} , this is justified in Appendix A. Appendix A also shows that this equation boils down to

$$\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}(\hat{\mathbf{p}}) = N \sum_{k=1}^d \frac{\varepsilon_k(\hat{\mathbf{p}}, \mathbf{x}', \mathbf{y}')^2}{\tilde{p}_k}. \quad (22)$$

This log-likelihood is the result of two approximations that are true only when $N \rightarrow +\infty$: we approximated $\underline{\boldsymbol{\varepsilon}}(\hat{\mathbf{p}}, \mathbf{x}', \mathbf{y}')$ as the realization of a Gaussian random vector and we used an approximation

for Σ . In practice, the resulting approximation is smoother and easier to minimize than $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}(\hat{\mathbf{p}})$ if the initialization point is not good enough (as will be shown in Section VI C). However, with a good initialization, the state that minimizes $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}(\hat{\mathbf{p}})$ should be closer to the true state than the one that minimizes $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}(\hat{\mathbf{p}})$. The smaller N , the starker the difference. This will be shown in Section VI B.

D. Mixed minimization

As stated above $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ is supposed to be easier to minimize but the minimum of $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}$ is supposed to be a better estimate. A good way to combine the two advantages is to start the optimization process by minimizing $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ and finish it by minimizing $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}$. In practice, we here again run the BFGS algorithm on $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ for 100 iterations starting from the initialization point of Sections III or IV, this yields $\hat{\mathbf{v}}_{inter}$. And then we run the BFGS algorithm on $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}$ starting from $\hat{\mathbf{v}}_{inter}$ and stopping only once a local (hopefully global) minimum has been found.

VI. NUMERICAL RESULTS

A. Performances of the two initialization algorithms

Sections III and IV detail two methods to perform QST which are used for initialization of ML algorithms. The current section aims at estimating the precision of those methods and comparing them whenever possible. The recursive algorithm of Section IV only works for a specific set of measurement types but is explicit and does not require an undefined number of iterations to converge contrary to PhaseCut defined in Section III. We only explained PhaseCut for the setup with 4 different measurement types described in Section III A, but it can be applied to any types of measurements. In particular we could apply it to the setup with $2n_{qb} + 1$ measurement types of Section IV A. In the current section, we test both PhaseCut and the recursive algorithm on 50 randomly generated 7-qubit pure states. The two sets of measurement types of Sections III A and IV A are considered. They contain respectively 4 and $2 \times 7 + 1 = 15$ measurement types. We test those algorithms with 2 different fixed numbers of total measurements N_C : 5 000 and 500 000. Thus each one of the 4 measurement types of the setup of Section III A

is performed either $N_C = 1250$ or $N_C = 125000$ times and each one of the 15 measurement types of the setup of Section III A is performed either $N_C = 333$ or $N_C = 33333$ times.

The metric used in order to quantify the proximity of $\hat{\mathbf{v}}$ to the actual vector \mathbf{v} up to a phase factor is $\mu = \|\mathbf{v} - \hat{\mathbf{v}}.e^{-i\xi}\|_2$ with ξ the angle that minimizes our metric: $e^{i\xi} = \frac{\mathbf{v}^*\hat{\mathbf{v}}}{|\mathbf{v}^*\hat{\mathbf{v}}|}$. We call μ this error in the rest of the paper. μ is maximal for orthogonal states (it is then $\sqrt{2}$), and minimal for states that differ by a global phase (it is then 0). A more widely used metric in the literature is the fidelity (see Section 9.2.2 in [1]) $f = |\mathbf{v}^*\hat{\mathbf{v}}|$. It can be shown that $f = (1 - \frac{\mu^2}{2})$. We do not use the fidelity because it can push some interesting values too close to 1.

Fig. 1 shows the error of $\hat{\mathbf{v}}_{pc}$ obtained by using PhaseCut with 100 to 100 000 iterations for the two setups (4 and 15 measurement types). With 15 (and not with 4) measurement types, the recursive algorithm can be implemented. We display biggest and smallest errors of $\hat{\mathbf{v}}_{rec}$ obtained with the recursive algorithm with horizontal bold green and red lines respectively. The recursive algorithm is performed in a fixed number of steps, this is why we plot the error on horizontal lines and not on a curve with respect to a number of iterations.

The aim of this simulation is to see how many iterations of PhaseCut are required to get a good estimate of the state and to compare the performances of the recursive algorithm with those of the more versatile PhaseCut.

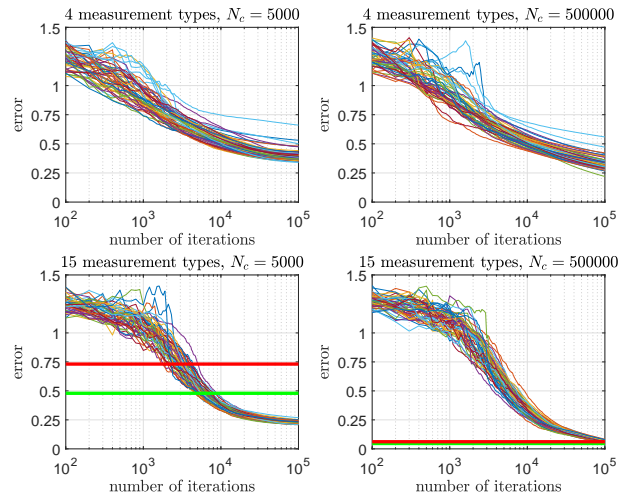


FIG. 1. Initialization algorithms. The bold red and green horizontal lines are the worst and best errors for the recursive algorithm on the 50 random states (only available with 15 measurement types). The other curves represent the evolution of the error on the PhaseCut estimates with the 50 states.

With enough iterations ($\sim 10^4$ for $N_c = 5000$ and $\sim 10^5$ for $N_c = 500000$) PhaseCut is more precise than the recursive algorithm on the setups on which they can both be implemented, but it takes way more time. Each iteration of PhaseCut is costly, because we are working on an $n_{prob} \times n_{prob}$ matrix. With Matlab, on a 2.11 GHz 4-core processor with 32 Go RAM, each iteration of PhaseCut takes around 4 ms for the setup with 4 types of measurements and around 45 ms for the setup with 15 types of measurements. In that same 15 measurement type setup, the recursive algorithm takes 200 ms. This is way faster than PhaseCut which runs in minutes, as it requires thousands of iterations.

B. Likelihood estimator comparison

In Section V we defined two likelihood estimators, based on the likelihood maximization. The first one minimizes the true negative log-likelihood $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}$ and the other minimizes a version of the negative log-likelihood that is supposed to be smoother, namely $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$. We know that $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ is an approximation of the likelihood that is accurate only if the number of measurements per measurement type is high enough. Therefore we expect the global minimum of $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ to be a worse estimator than the global minimum of $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}$ for a limited number of measurements. In order to check whether this is true and quantify the difference, we compute the errors on both estimators when they are initialized at the true state \mathbf{v} . Doing this ignores the error on the initialization point (to which the regularized Gaussian estimate is supposed to be robust). We also compute the error for the mixed algorithm which starts by minimizing $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ and then minimizes $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}$. These 3 types of errors are computed with 1000 random initial states on the four setups described in Section VIA with 4 or 15 measurement types and 5000 or 500 000 total measurements. For each of the four setups, the empirical cumulative density function (empirical cdf) is computed on the 1000 errors associated with the initial states, those cdf are shown in Fig. 2.

As predicted the error is larger with the Gaussian estimate of the likelihood, and the difference decreases when the number of measurements per measurement type increases.

The performance of the mixed minimization algorithm is very close to that of the estimator that minimizes $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}$. There can be small differences however. It turns out that they sometimes converge toward close but different minima. This is due to the

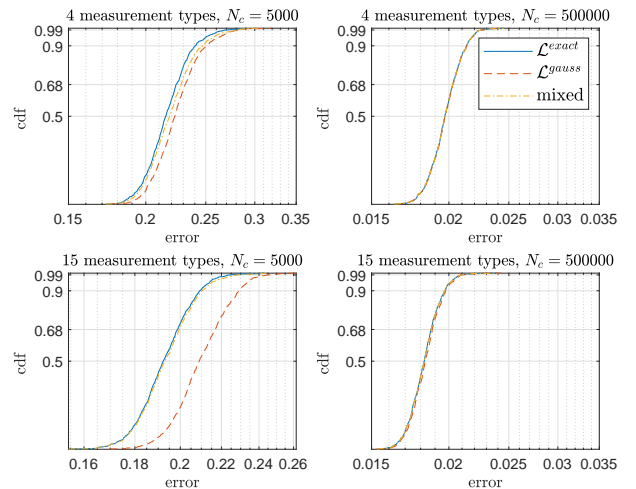


FIG. 2. empirical cdf of the errors of the 3 maximum likelihood estimators

fact that the small error made by the first 100 iterations of the mixed algorithm (during which $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ is minimized) can be enough to affect the final result.

The differences between the 3 estimators are only noticeable for $N_c = 5000$ with 15 and 4 different measurements (so 333 or 1250 measurements per measurement type).

C. Convergence of the likelihood estimators

In the current section, we intend to see what precision on the initial state is required to make sure that the likelihood optimization algorithm converges towards a reasonable solution, and compare the robustness of the three ML estimates. We compare the rates of divergence (denoted as δ and defined below) of the algorithms that minimize $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}$ and $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ as well as the mixed algorithm. 1000 random states \mathbf{v} to be estimated are considered with 1000 associated initial states of ML algorithms that have an initialization error μ linearly varied from 0 to $\sqrt{2}$ (as stated above $\sqrt{2}$ is the highest possible value for μ , it is reached if the two states are orthogonal). Let us denote as $\{\mu_i\}_{i \in \{1, \dots, 1000\}}$ the 1000 values of this initial error on states \mathbf{v} and define $\{b_i^{algo}, i \leq 1000, algo \in \{exact, Gauss, mixed\}\}$ where b_i^{algo} is -1 if the $algo$ algorithm converges towards the same minimum with the μ_i initialization error and with no error and $+1$ if it converges toward a different minimum. We say that those two minima are the same if the error μ between the two is smaller than one percent of the error between the first one

(initialized without error) and the true state vector.

For each of the 3 algorithms, we then define the rate of divergence $\delta_{algo}(\mu)$ associated with a given error μ . It takes all the b_i^{algo} into account but gives more weight to those for which the associated μ_i is close to μ :

$$\delta_{algo}(\mu) = \frac{1}{2} \left(1 + \frac{\sum_{i=1}^{1000} b_i^{algo} e^{-\left(\frac{\mu-\mu_i}{\alpha}\right)^2}}{\sum_{i=1}^{1000} e^{-\left(\frac{\mu-\mu_i}{\alpha}\right)^2}} \right).$$

Simply put, if the majority of μ_i in the vicinity of μ are associated with b_i^{algo} equal to -1 (i.e. the algorithm converges towards the proper minimum with initialization errors around μ) then, $\delta_{algo}(\mu)$ will be close to 0. If the associated b_i^{algo} are 1 (i.e. the algorithm does not converge towards the proper minimum) then, $\delta_{algo}(\mu)$ will be close to 1. The parameter α quantifies how far away from μ we look for results, we picked $\alpha = 0.1$. Fig. 3 shows the rates of divergence of the 3 algorithms in the four setups described in Section VIA with 4 or 15 measurement types and 5000 or 500 000 total measurements.

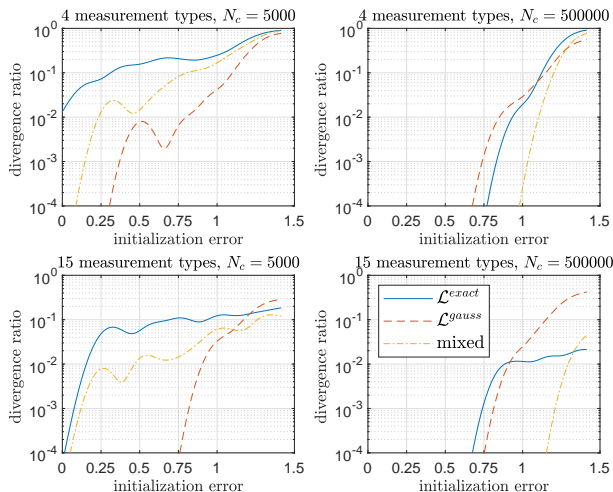


FIG. 3. Convergence of the different likelihood algorithms in the presence of initialization errors.

The two plots on the right are of limited interest to us as the rate of divergence is always very low ($\leq 10^{-4}$) for errors lower than 0.75. We are mostly interested in the rates of divergence for initialization errors μ smaller than 0.75 because according to Fig. 1, the recursive algorithm always yields an estimate that corresponds to an error lower than 0.75 and PhaseCut also does so quite quickly (for more than 5000 iterations) for every setup. For those errors (on the two plots on the left), the best algorithm seems to be the minimization of $\mathcal{L}_{(x',y')}^{gauss}$, indeed increased robustness to the initialization error is the whole reason why we introduced $\mathcal{L}_{(x',y')}^{gauss}$. The mixed algo-

rithm does not quite reach the same robustness but it is certainly an improvement over the algorithm that minimizes $\mathcal{L}_{(x',y')}^{exact}$ which has the worst performances for the relevant initialization errors. We should note that the name given to δ : “rate of divergence” is a bit severe as the likelihood algorithms never diverge in practice, they simply converge toward a false local minimum that is sometimes close to the real global minimum. δ is not useless however, and Fig. 3 shows us that, generally, with either the mixed algorithm or the algorithm that minimizes $\mathcal{L}_{(x',y')}^{gauss}$, an initialization error lower 0.75 leads to proper convergence towards the real minimum. According to Fig. 1, 5000 iterations of PhaseCut as well as the recursive algorithm generally yield an error smaller than 0.75. Therefore we choose to use the recursive algorithm when it is possible i.e. with the setup of Section IV with 15 types of measurements for 7 qubits (because it is faster than PhaseCut) and when PhaseCut has to be used (so with 4 measurement types) we only perform 5 000 iterations. We could let PhaseCut run longer but our implementation of the ML algorithm is faster.

D. Global performances

This section aims to test the algorithms of Sections III and IV, fine tuned with the 3 algorithms of Section V on $n_{qb} = 7$ qubits, with the four setups described in Section VIA. For each setup, and for each version of the ML algorithm, 4 estimates of \mathbf{v} are computed:

- The initial estimate, so $\hat{\mathbf{v}}_{pc}$ for the setup with 4 measurement types or $\hat{\mathbf{v}}_{rec}$ for the setup with 15 measurement types. It does not depend on the choice of the ML algorithm.
- $\hat{\mathbf{v}}_{ml}$ which is the result of the likelihood optimization (minimizing either $\mathcal{L}_{(x',y')}^{exact}$ or $\mathcal{L}_{(x',y')}^{gauss}$ or both successively) initialized at the initial estimate.
- $\hat{\mathbf{v}}_{ref}$ which is the result of the likelihood optimization initialized at the true \mathbf{v} (not available in practice, it should be the global maximum likelihood; if $\hat{\mathbf{v}}_{ml} = \hat{\mathbf{v}}_{ref}$ then the initial estimate was good enough). We call $\hat{\mathbf{v}}_{ref}$ the reference, it has already been defined (but not named) in Section VIB and represented in Fig. 2.
- And $\hat{\mathbf{v}}_{rnd}$ which is the result of the likelihood optimization initialized at a random normalized vector (if $\hat{\mathbf{v}}_{rnd}$ is not worse than $\hat{\mathbf{v}}_{ml}$, then the initial estimate was unnecessary and one can only use the maximum likelihood algorithm initialized randomly).

For each setup, 1000 tests are performed with 1000 randomly generated \mathbf{v} . We compute the estimates of each \mathbf{v} with the different algorithms and display the empirical cumulative density function (cdf) of the errors in Fig. 4 to Fig. 6.

The performances of the three ML algorithms are quite similar (when excluding the random initialization), but some differences can be noted:

- The algorithm that minimizes $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}$ is supposed to be less robust to the initialization error than the others. It is only apparent for the setup with 4 measurements and $N_c = 5000$. $\hat{\mathbf{v}}_{ml}$ is not quite as precise as $\hat{\mathbf{v}}_{ref}$.
- The algorithm that minimizes $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ does not have that problem, $\hat{\mathbf{v}}_{ml}$ and $\hat{\mathbf{v}}_{ref}$ are always indistinguishable. However the version of $\hat{\mathbf{v}}_{ref}$ computed by minimizing $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ is not as precise as the version that minimizes $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}$. This can be seen by comparing Fig. 4 and Fig. 5 but it is more visible on Fig. 2 that represents the performances of the 3 references on a single graph.
- The mixed algorithm seems to combine the advantages of those based on $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ and $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}$. $\hat{\mathbf{v}}_{ml}$ is almost equal to $\hat{\mathbf{v}}_{ref}$, and $\hat{\mathbf{v}}_{ref}$ is almost as good with this mixed algorithm as with $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}$ (see Fig. 2 for a clearer comparison of the two values of $\hat{\mathbf{v}}_{ref}$).

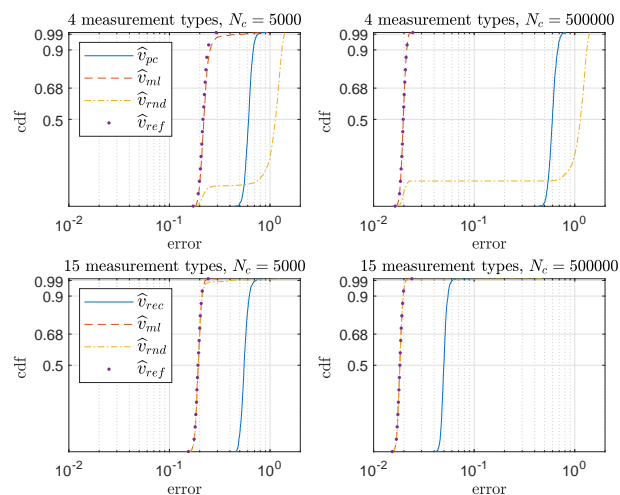


FIG. 4. Empirical cdf of QST error, $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}$ minimization.

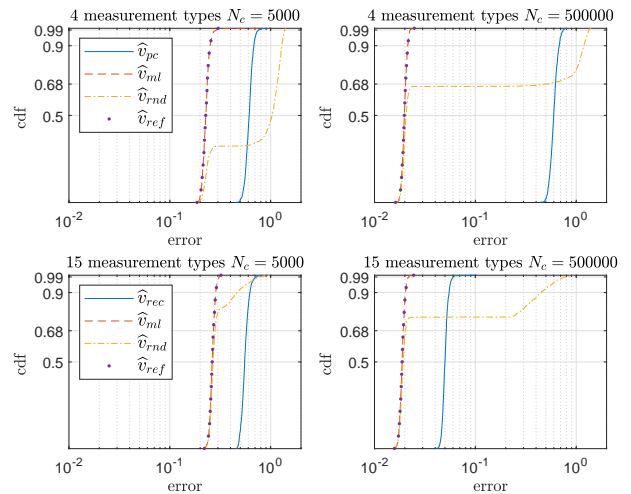


FIG. 5. Empirical cdf of QST error, $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ minimization.

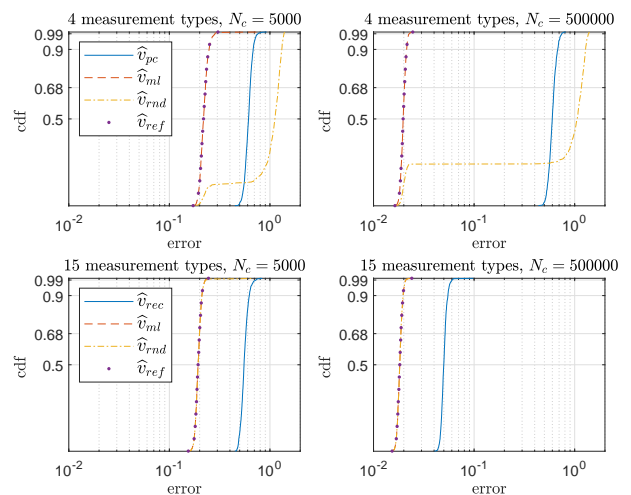


FIG. 6. Empirical cdf of QST error, mixed algorithm.

The performances of $\hat{\mathbf{v}}_{rnd}$, the maximum likelihood estimators initialized at a random point, are interesting. With the 4 measurement type setup, it is always a much worse estimate than $\hat{\mathbf{v}}_{ml}$. But with 15 measurement types it is (almost) as good as the maximum likelihood estimators initialized at $\hat{\mathbf{v}}_{rec}$ (unless we use the $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ minimization). This could make us question the relevance of the recursive algorithm defined in Section IV. It would seem that the structure of the measurement matrix \mathbf{A}_t is such that the gradient descent algorithm naturally converges towards the global minimum from any initial point. However the recursive algorithm is still useful because it is very fast and speeds up the likelihood maximization (see Table 2).

We can also compare the performances of the two

initialization algorithms $\hat{\mathbf{v}}_{pc}$ or $\hat{\mathbf{v}}_{rec}$ (blue curve) with $\hat{\mathbf{v}}_{ml}$ (dashed red curve). The error on $\hat{\mathbf{v}}_{ml}$ is at least 3 times smaller (or way less for $\hat{\mathbf{v}}_{pc}$ and $N_c = 500000$) than that of the initialization algorithms. This shows that the fine tuning with ML is very useful to reduce the error. Comparing the precision of the initialization algorithm with $\hat{\mathbf{v}}_{rnd}$ is unwise because $\hat{\mathbf{v}}_{pc}$ and $\hat{\mathbf{v}}_{rec}$ can be improved with the ML algorithm whereas $\hat{\mathbf{v}}_{rnd}$ cannot as it is a local minimum of the likelihood. Furthermore, with $N_c = 5000$, $\hat{\mathbf{v}}_{pc}$ and $\hat{\mathbf{v}}_{rec}$ have a similar accuracy (respectively on 4 and 15 measurement types). And with $N_c = 500000$, $\hat{\mathbf{v}}_{rec}$ is a way better estimate than $\hat{\mathbf{v}}_{pc}$ because the PhaseCut algorithm is limited to 5000 iterations (allowing it enough iterations to converge properly would be way slower and not as accurate as the likelihood maximization).

After likelihood optimization the performances of $\hat{\mathbf{v}}_{ml}$ with 15 and 4 measurement types are comparable (with the mixed algorithm, the 15 measurement setup is slightly better). Also the final error is roughly 10 times smaller when the number of measurements is multiplied by 100. This means that for more than 5000 measurements one can extrapolate the error (and therefore its cdf), as the error is proportional to $N_c^{-1/2}$.

The fact that the recursive algorithm used to compute $\hat{\mathbf{v}}_{rec}$ has a zero measure failure set on which phase recovery is impossible (see Section IV) turns out to be a non-issue. We could have expected to see some outliers on the error of $\hat{\mathbf{v}}_{rec}$, and the $\hat{\mathbf{v}}_{ml}$ computed from it, if the randomly generated \mathbf{v} was close enough to the failure set. It is not the case, each one of the 1000 initial states has been successfully recovered with a reasonable error. The same is true when using PhaseCut with the 4 measurement type setup. Even though we were not able to prove the injectivity, the QST goes well in practice and there are no outliers in the error if the proper algorithms are used.

Table 1 and Table 2 give the median execution time of all the algorithms on an Intel Xeon Gold 6226R 2.9 GHz core, all the scripts ran on 1 thread on Matlab. There are no significant differences between the 3 ML algorithms when they are not initialized at random. The random initialization is never relevant, as for the 4 measurement type setup it is relatively fast (as it spares us the initialization step with PhaseCut) but inaccurate; and for the 15 measurement type setup it is always slower (sometimes way slower) than the likelihood maximization with proper initialization.

	$N_c = 5000$	$N_c = 500000$
PhaseCut	16.9 s	17.4 s
$\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}$ min. from $\hat{\mathbf{v}}_{pc}$	11.4 s	8.3 s
$\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}$ min., random init.	22 s	24.7 s
$\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ min. from $\hat{\mathbf{v}}_{pc}$	8.1 s	5.2 s
$\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ min., random init.	16.6 s	21.3 s
mixed algo. from $\hat{\mathbf{v}}_{pc}$	6.8 s	4.7 s
mixed algo., random init.	10.6 s	12.8 s

TABLE I. execution time for the setups with 4 measurement types.

	$N_c = 5000$	$N_c = 500000$
recursive algorithm	0.17 s	0.17 s
$\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}$ min. from $\hat{\mathbf{v}}_{rec}$	44.4 s	10.9 s
$\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}$ min., random init.	272 s	94.4 s
$\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ min. from $\hat{\mathbf{v}}_{rec}$	38.8 s	16.7 s
$\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ min., random init.	84.9 s	126.7 s
mixed algo. from $\hat{\mathbf{v}}_{rec}$	47.8 s	26.1 s
mixed algo., random init.	62 s	38.4 s

TABLE II. execution time for the setups with 15 measurement types.

In conclusion, we recommend using the mixed algorithm for the likelihood, it is a good compromise between the $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ minimization and the $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{exact}$ minimization. The choice between the setup with 4 types of measurements and the setup with $2n_{qb} + 1$ types of measurements is less obvious. The first one is obviously simpler for the operator and the likelihood optimization is faster (see Table 1 and Table 2) but:

- It yields a slightly less precise result. The median error with the mixed algorithm and $N_c = 5000$ is 0.22 against 0.19 with 15 measurement types.
- We have no closed-form algorithm that retrieves the state from the measurements. We must rely on PhaseCut which is unprecise. PhaseCut is also slow but the time gained during the mixed ML algorithm more than makes up for it (see Table 1 and Table 2).
- We explained (in Section III B) why we think the measurements are injective, and in practice all 1000 tested states were recovered, but we were unable to prove the injectivity so far.

VII. CONCLUSION AND FUTURE WORK

In this paper we first showed how some of the work made in the applied mathematics community in the field of phase recovery can be used to define a set of four types of d -outcome measurements that should be enough to achieve QST for any pure state using the PhaseCut optimization algorithm. We also proposed a set of $(2n_{qb} + 1)$ types of d -outcome measurements as well as a recursive algorithm which allows explicit reconstruction of the state (n_{qb} is the number of qubits, $d = 2^{n_{qb}}$). Experimentally, they both give similar performances when the total number of measurements is the same (slight advantage for the second set of measurements); the first set is easier to set up and the second set is more theoretically sound.

The initial estimates of the considered state are then finely tuned with the maximum likelihood approach that is widely used in the quantum information processing literature. We introduced some refinements which make it more robust by considering a smooth an easy way to maximize an approximation of the likelihood.

We intend to use those QST methods to perform quantum process tomography (QPT) like in [22]. In [22] we introduced a QPT method that relies on measuring the state of the system after different time delays. At each time delay, we have to perform QST.

Appendix A: Covariance matrix and likelihood of the error on the sample probabilities

1. Covariance matrix

Appendix A aims at computing the asymptotic law of $\sqrt{N}\boldsymbol{\varepsilon} = \sqrt{N}(\hat{\mathbf{p}} - \mathbf{p})$ defined in Section VC and at simplifying the expression of the likelihood of $\boldsymbol{\varepsilon}$. We consider that \mathbf{p} contains the probabilities of a single type of d -outcome measurement. The generalization is straightforward as the errors on different measurements are independent (see Section A3). The only random vector in $\boldsymbol{\varepsilon}$ is $\hat{\mathbf{p}}$ defined as the vector that contains the sample probabilities of each of the d outcomes. So $\hat{\mathbf{p}} = \frac{1}{N}\mathbf{n}$ where each component n_i of \mathbf{n} contains the number of times the i -th outcome occurred. By definition \mathbf{n} follows a multinomial distribution characterized by the number of trials N and the theoretical probabilities of each outcome contained in \mathbf{p} . The expected value and covariance matrix of the multinomial distribution are known: $E(\mathbf{n}) = N\mathbf{p}$ and $Cov(\mathbf{n}) = N(diag(\mathbf{p}) - \mathbf{p}\mathbf{p}^T)$.

We want to use the central limit theorem so let us write \mathbf{n} as a sum: $\mathbf{n} = \sum_{k=1}^N \boldsymbol{\delta}_k$ where the $\{\boldsymbol{\delta}_k\}_k$

are independent and have the same distribution for different k . $\boldsymbol{\delta}_k$ contains $d - 1$ zeros and one 1 at a random index $i_k \in \{1, \dots, N\}$ whose density function is $j \rightarrow p_j$ (i.e. the probability that i_k takes the value $j \in \{1, \dots, N\}$ is p_j , the j -th element of \mathbf{p}). $\boldsymbol{\delta}_k$ follows a multinomial distribution with $N = 1$ trial. Its expected value is therefore \mathbf{p} and its covariance matrix is $diag(\mathbf{p}) - \mathbf{p}\mathbf{p}^T$. Therefore $\boldsymbol{\varepsilon}$ is the difference between the empirical average of $\boldsymbol{\delta}_k$ with N realizations and its expected value. According to the central limit theorem, when $N \rightarrow +\infty$, the distribution of $\sqrt{N}\boldsymbol{\varepsilon}$ tends to a centered multivariate normal distribution, and its covariance matrix is $\boldsymbol{\Sigma}_{\text{full}} = diag(\mathbf{p}) - \mathbf{p}\mathbf{p}^T$. $\widehat{\boldsymbol{\Sigma}}_{\text{full}}$ is an estimate of $\boldsymbol{\Sigma}_{\text{full}}$, it uses $\hat{\mathbf{p}}$ as we do not want to depend on the unknown vector \mathbf{p} : $\widehat{\boldsymbol{\Sigma}}_{\text{full}} = diag(\hat{\mathbf{p}}) - \hat{\mathbf{p}}\hat{\mathbf{p}}^T$.

2. Likelihood

The easiest way to compute the likelihood of a vector that follows a multivariate normal distribution requires us to invert the covariance matrix [23]. If the covariance matrix is not invertible, then it is not of full rank, this means that at least one component of the random vector is linearly dependent on the others and therefore it is not needed to compute the likelihood. Those components can be removed and the likelihood of the smaller vector is the same as the likelihood of the original vector. In our case, the components of $\sqrt{N}\boldsymbol{\varepsilon}$ sum to zero, therefore its covariance matrix is not invertible and any component can be removed without losing any information that could be used to compute the likelihood. Let us consider $\sqrt{N}\underline{\boldsymbol{\varepsilon}}$, it is the same vector as $\sqrt{N}\boldsymbol{\varepsilon}$ with the last component removed, and thus, its covariance matrix is the same with the last row and column removed: $\boldsymbol{\Sigma} = diag(\underline{\mathbf{p}}) - \underline{\mathbf{p}}\underline{\mathbf{p}}^T$ ($\underline{\mathbf{p}}$ is \mathbf{p} with the last element removed). It can be estimated with the sample probabilities

$\underline{\hat{\mathbf{p}}} = \begin{pmatrix} \hat{p}_1 \\ \vdots \\ \hat{p}_{d-1} \end{pmatrix}$ instead of $\hat{\mathbf{p}}$. The resulting matrix is $\widehat{\boldsymbol{\Sigma}} = diag(\underline{\hat{\mathbf{p}}}) - \underline{\hat{\mathbf{p}}}\underline{\hat{\mathbf{p}}}^T$. Straightforward calculations show that if no element of $\hat{\mathbf{p}} = \begin{pmatrix} \hat{p}_1 \\ \vdots \\ \hat{p}_d \end{pmatrix}$ (with

$\hat{p}_d = 1 - \sum_{k=1}^{d-1} \hat{p}_k$) is zero, then, $\widehat{\boldsymbol{\Sigma}}$ is invertible and

$$\widehat{\boldsymbol{\Sigma}}^{-1} = \frac{1}{\hat{p}_d} \mathbb{1} + diag(1/\underline{\hat{\mathbf{p}}}). \quad (\text{A1})$$

is its inverse. $1/\underline{\hat{\mathbf{p}}}$ is the element-wise inverse of $\underline{\hat{\mathbf{p}}}$

and $\mathbb{1}$ is the $d-1 \times d-1$ matrix with only ones. In practice, elements of $\hat{\mathbf{p}}$ can be zeros, it would make the matrix singular. In order to overcome this difficulty and avoid giving too much importance to the errors on the scarcely observed outcomes, we modify the sample probability and create a new vector $\tilde{\mathbf{p}}$:

$$\tilde{\mathbf{p}} = \frac{\hat{\mathbf{p}} + \frac{5}{N}}{1 + \frac{5d}{N}}. \quad (\text{A2})$$

This means that we consider that each outcome has been observed 5 more times than it actually was, and the total number of observations changes from N to $N + 5d$ (the choice of 5 is arbitrary). This is a standard method to make a criterion smoother (see [24]). The resulting estimate of the inverse of the covariance matrix is

$$\tilde{\Sigma}^{-1} = \frac{1}{p_d} \mathbb{1} + \text{diag}(1/\tilde{\mathbf{p}}) \quad (\text{A3})$$

With the inverse of $\tilde{\Sigma}$ and knowing that the distribution is normal and centered, we can compute the negative log-likelihood of the vector (see [23]):

$$\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}(\hat{\mathbf{p}}) = N \underline{\boldsymbol{\varepsilon}}(\hat{\mathbf{p}}, \mathbf{x}', \mathbf{y}')^T \tilde{\Sigma}^{-1} \underline{\boldsymbol{\varepsilon}}(\hat{\mathbf{p}}, \mathbf{x}', \mathbf{y}'). \quad (\text{A4})$$

We use $\hat{\mathbf{p}}$ and not $\tilde{\mathbf{p}}$ to compute $\underline{\boldsymbol{\varepsilon}}$ otherwise estimator that minimizes the criterion would become biased (as the minimum of $\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}$ would fit $\tilde{\mathbf{p}}$ which does not contains the actual sample probabilities) and the criterion would not be smoother.

Let us simplify this expression using (A3) and the fact that $\sum_k \varepsilon_k = 0 \Rightarrow \varepsilon_d = -\sum_{k=1}^{d-1} \varepsilon_k$:

$$\begin{aligned} N \underline{\boldsymbol{\varepsilon}}^T \tilde{\Sigma}^{-1} \underline{\boldsymbol{\varepsilon}} &= N \underline{\boldsymbol{\varepsilon}}^T \begin{pmatrix} \frac{1}{p_d} \sum_{k=1}^{d-1} \varepsilon_k + \frac{\varepsilon_1}{p_1} \\ \vdots \\ \frac{1}{p_d} \sum_{k=1}^{d-1} \varepsilon_k + \frac{\varepsilon_{d-1}}{p_{d-1}} \end{pmatrix} \\ &= N \underline{\boldsymbol{\varepsilon}}^T \begin{pmatrix} \frac{\varepsilon_1}{p_1} - \frac{\varepsilon_d}{p_d} \\ \vdots \\ \frac{\varepsilon_{d-1}}{p_{d-1}} - \frac{\varepsilon_d}{p_d} \end{pmatrix} \\ &= N \left(\sum_{k=1}^{d-1} \frac{\varepsilon_k^2}{p_k} - \frac{\varepsilon_d}{p_d} \sum_{k=1}^{d-1} \varepsilon_k \right) \\ &= N \sum_{k=1}^d \frac{\varepsilon_k^2}{\tilde{p}_k}. \end{aligned}$$

Therefore, the expression of the negative log-

likelihood is:

$$\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}(\hat{\mathbf{p}}) = N \sum_{k=1}^d \frac{\varepsilon_k(\hat{\mathbf{p}}, \mathbf{x}', \mathbf{y}')^2}{\tilde{p}_k}. \quad (\text{A5})$$

3. Extension to several d -outcome measurements

Since the beginning of the appendix we assumed that only one type of measurement with d outcomes was performed. In practice the methods we describe require either 4 (in Section III) or $2n_{qb} + 1$ (in Section IV) types of measurements. The errors between the empirical and theoretical probabilities of different measurements are independent. Therefore if $\underline{\boldsymbol{\varepsilon}}(\hat{\mathbf{p}}, \mathbf{x}', \mathbf{y}')$ contains $n_t > 1$ types of measurements and dn_t real components, then, its covariance matrix is a block diagonal matrix with the covariance matrix of each measurement type on the diagonal (because the measurement errors on two different measurement types are independent.). And the same goes for the inverse of its regularized covariance matrix:

$$\tilde{\Sigma}^{-1} = \begin{bmatrix} \tilde{\Sigma}_1^{-1} & & \\ & \ddots & \\ & & \tilde{\Sigma}_{n_t}^{-1} \end{bmatrix}. \quad (\text{A6})$$

Each $\tilde{\Sigma}_k^{-1}$ is the regularized inverse of the covariance matrix for one measurement type defined in (A3).

The negative log-likelihood of $\boldsymbol{\varepsilon}(\hat{\mathbf{p}}, \mathbf{x}', \mathbf{y}')$ containing $n_{prob} = n_t d$ measurements errors on n_t types of measurements is the sum of the n_t negative log-likelihoods of the error vectors of each measurement type

$$\mathcal{L}_{(\mathbf{x}', \mathbf{y}')}^{gauss}(\hat{\mathbf{p}}) = N \sum_{k=1}^{n_{prob}} \frac{\varepsilon_k(\hat{\mathbf{p}}, \mathbf{x}', \mathbf{y}')^2}{\tilde{p}_k}. \quad (\text{A7})$$

-
- [1] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information* (Cambridge University Press, 2000).
- [2] A. Kalev, R. L. Kosut, and I. H. Deutsch, npj Quantum Information **1**, 10.1038/npjqi.2015.18 (2015).
- [3] A. Smith, C. A. Riofrío, B. E. Anderson, H. Sosa-Martinez, I. H. Deutsch, and P. S. Jessen, Physical Review A **87**, 10.1103/physrev.a.87.030102 (2013).
- [4] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert, Physical Review Letters **105**, 10.1103/physrevlett.105.150401 (2010).
- [5] X. Ma, T. Jackson, H. Zhou, J. Chen, D. Lu, M. D.

- Mazurek, K. A. G. Fisher, X. Peng, D. Kribs, K. J. Resch, Z. Ji, B. Zeng, and R. Laflamme, *Physical Review A* **93**, 10.1103/physreva.93.032140 (2016).
- [6] T. Cai, D. Kim, Y. Wang, M. Yuan, and H. H. Zhou, *The Annals of Statistics* **44**, 10.1214/15-aos1382 (2016).
- [7] Y. Wang, *The Annals of Statistics* **41**, 10.1214/13-aos1156 (2013).
- [8] C. H. Baldwin, I. H. Deutsch, and A. Kalev, *Physical Review A* **93**, 10.1103/physreva.93.052105 (2016).
- [9] C. Ferrie, *Physical Review Letters* **113**, 10.1103/physrevlett.113.190404 (2014).
- [10] R. J. Chapman, C. Ferrie, and A. Peruzzo, *Physical Review Letters* **117**, 10.1103/physrevlett.117.040402 (2016).
- [11] S. T. Ahmad, A. Farooq, and H. Shin, *Scientific Reports* **12**, 10.1038/s41598-022-09143-7 (2022).
- [12] J. Cotler and F. Wilczek, *Physical Review Letters* **124**, 10.1103/physrevlett.124.100401 (2020).
- [13] J. Finkelstein, *Physical Review A* **70**, 10.1103/physreva.70.052107 (2004).
- [14] T. Heinosaari, L. Mazzarella, and M. M. Wolf, *Communications in Mathematical Physics* **318**, 355 (2013).
- [15] D. Goyeneche, G. Cañas, S. Etcheverry, E. Gómez, G. Xavier, G. Lima, and A. Delgado, *Physical Review Letters* **115**, 10.1103/physrevlett.115.090401 (2015).
- [16] R. Balan, P. Casazza, and D. Edidin, *Applied and Computational Harmonic Analysis* **20**, 345 (2006).
- [17] I. Waldspurger, A. d’Aspremont, and S. Mallat, *Mathematical Programming* **149**, 47 (2013).
- [18] A. S. Bandeira, J. Cahill, D. G. Mixon, and A. A. Nelson, *Applied and Computational Harmonic Analysis* **37**, 106 (2014).
- [19] N. Z. Shor, *Soviet Journal of Computer and Systems Sciences* **25**, 1 (1987).
- [20] C. G. Broyden, *IMA Journal of Applied Mathematics* **6**, 76 (1970).
- [21] Z. Hradil, J. Řeháček, J. Fiurášek, and M. Ježek, in *Quantum State Estimation* (Springer Berlin Heidelberg, 2004) pp. 59–112.
- [22] F. Verdeil, Y. Deville, and A. Deville, in *2021 IEEE Statistical Signal Processing Workshop (SSP)* (IEEE, Rio de Janeiro, Brazil, 2021) pp. 161–165.
- [23] A. Gut, *An Intermediate Course in Probability* (Springer, 2009).
- [24] R. Blume-Kohout, *Physical Review Letters* **105**, 200504 (2010).