Pure-state tomography with parallel unentangled measurements

François Verdeil and Yannick Deville
Université de Toulouse, UPS, Centre National de la Recherche Scientifique, CNES, OMP, IRAP, Toulouse, France

(Received 9 August 2022; accepted 9 December 2022; published 5 January 2023)

Quantum state tomography (QST) aims at estimating a quantum state from averaged quantum measurements made on copies of that 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, and 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, e.g., using one-qubit Pauli measurements. We propose two sets of quantum measurements that one can make on a pure state as well as the algorithms that use the measurement outcomes in order to identify the state. We also discuss how those estimates can be fine 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, including for mixed states that are close to being pure.

DOI: 10.1103/PhysRevA.107.012408

I. PRIOR WORK AND PROBLEM STATEMENT

Quantum state tomography (QST) aims to estimate a quantum state from averaged quantum measurements made on copies of that state. It is often necessary for quantum computation [1] and has been extensively studied for mixed states. The most basic version is detailed in [1] at the beginning of Sec. 8.4.2. It uses measurements defined by Pauli operators, often called Pauli measurements [2–7]. This version is simple and very robust but requires computing the averages of $4^{n_{qb}} - 1$ different types of multiqubit Pauli measurements where n_{qb} is the number of qubits of the state. This method suffers from poor scalability, as a consequence of the large number of measurements required, increasing exponentially with the number of qubits. That is, considering that 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), the required number of measurements is on the order of $d^2 = 4^{n_{qb}}$. In order to perform QST with fewer types of measurements, one can focus on a subset of all mixed states. The most popular assumption is that the density matrix ρ representing the state has a low rank. Reference [4] introduced a compressed sensing approach that requires the averages of $O[rd \log(d)^2]$ types of two-outcome measurements [8] to estimate the state where r is the rank of ρ. References [2,3] later built upon this idea of QST via compressed sensing. More recently, bounded rank QST was introduced [9]. 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 [10] and further studied in [11,12]. 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 measurements 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 [13] 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 two and three qubits (Pauli measurements are unentangled). Our addition to that paper is that we will address the generic case with any n_{qb}. Furthermore, we will use parallel measurements like in [13] 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 [14] Finkelstein describes a setup able to distinguish almost all pure states, with only $n_{\text{prob}} = 2d$ probabilities. n_{prob} is the number of empirical probabilities that are measured. For example, if we use d two-outcome measurements and average them (i.e., compute the empirical probabilities of all measurement outcomes), we get $2d$ empirical probabilities. There is a negligible (zero measure) set of pure states that the setup of [14] cannot recover up to a global phase. It is called the failure set. In addition to the failure set, the main problem of [14] 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 $n_{\text{prob}} = 4d$ probabilities that also has a negligible failure set. Technically [15] introduces five measurements that yield $5d$ probabilities (obtained from averaging the results of five different kinds of d-outcome measurements), but only the four measurements defined by its Eq. (2) are needed to achieve QST. The measurements of [15] are more realistic as they are performed on four 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 onto 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 [14,16–19]). A pure state $|\varphi\rangle$ of an n_{qb}-qubit system is represented by a complex unit-norm vector $\boldsymbol{v}$ with d elements. Pure-state tomography aims at estimating $\boldsymbol{v}$ from measurements. The theoretical probabilities of all outcomes of the considered types of measurements are contained in the vector $|\mathbf{A}\boldsymbol{v}|^2$ where $\mathbf{A}$ is an $n_{\text{prob}} \times d$ matrix determined by the types of measurements performed and $|.|^2$ stands for a componentwise squared modulus. Recovering $\boldsymbol{v}$ (up to a global phase) from $|\mathbf{A}\boldsymbol{v}|^2$ (generally it is $|\mathbf{A}\boldsymbol{v}|$ instead of $|\mathbf{A}\boldsymbol{v}|^2$ but both problems are essentially the same) is called phase retrieval. The first question asked in phase recovery concerns injectivity: how can one choose $\mathbf{A}$ in order to make sure that $|\mathbf{A}\boldsymbol{v}|^2$ contains enough information to recover $\boldsymbol{v}$ up to a global phase? Proving that a given $\mathbf{A}$ guarantees injectivity is a difficult question. Reference [16] gave a minimal number of measurements below which injectivity is impossible. In our case this condition is $n_{\text{prob}} > 4d - 3 - c(d)n_{qb}$ rows for some $c(d) \in [1, 2]$. Reference [17] showed that for a generic $\mathbf{A}$, having $4d - 2$ rows or more is a sufficient condition for injectivity. Reference [14] does not beat the bound of [16] as it has a non-null failure set (even though it is of zero measure), which means that the measurements are not injective.

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 [14,15] give their own closed-form algorithms to recover the phases which are adapted to their versions of $\mathbf{A}$. Reference [18] 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 $n_{\text{prob}} = 4d$ using an optimization algorithm of [18] with a number of probabilities consistent with the lower bound of [16]. The probabilities can be obtained by averaging the results of four types of parallel unentangled measurements.

Section IV describes a method with $n_{\text{prob}} = (2n_{qb} + 1)d$ 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 Secs. III or IV which it uses in order to maximize the likelihood of the measurements.

Finally, in Sec. 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 $|0\dots0\rangle$,..., $|1\dots1\rangle$. The components of $|\varphi\rangle$ in the basis can be stored in a d-element vector ($d = 2^{n_{qb}}$) $\boldsymbol{v} = [v_1 \quad \dots \quad v_d]^T$ 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. d-outcome measurement

A quantum measurement on a state in a d-dimensional Hilbert space has at most d outcomes. It is possible to define measurements with fewer than d outcomes but we consider them to be suboptimal, as will be explained in Sec. II E.

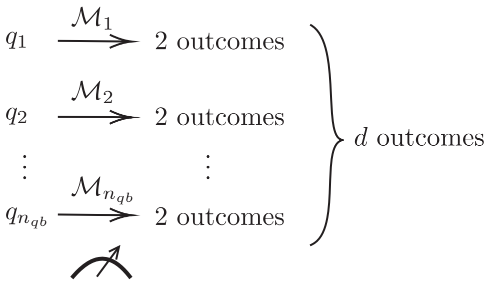
The actual values of the d outcomes are of no interest to us: we are only interested in the theoretical probabilities of each outcome when performing a measurement for a given state $\boldsymbol{v}$. For a given measurement $\mathcal{M}$, we call $\boldsymbol{p}_{\mathcal{M}}$ the d-dimensional vector that contains those probabilities. There exists a unitary matrix $\mathbf{E}_{\mathcal{M}}$ such that $\boldsymbol{p}_{\mathcal{M}} = |\mathbf{E}^*_{\mathcal{M}}\boldsymbol{v}|^2$ (Born rule) where $.^*$ is the transconjugate. In practice $\mathbf{E}_{\mathcal{M}}$ is the matrix the columns of which are unit-norm eigenvectors of the Hermitian matrix that characterizes the measurements (see Sec. 2.2.5 in [1]). We call $\mathbf{E}_{\mathcal{M}}$ the eigenvector matrix of the measurement $\mathcal{M}$.

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

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

C. Parallel unentangled measurement

We define a parallel unentangled measurement as a d-outcome measurement that can be performed with simultaneous one-qubit measurements on each one of the n_{qb} qubits:



It can be shown that the resulting eigenvector matrix $\mathbf{E}_{\mathcal{M}}$ can be written as the tensor product of the n_{qb} 2×2 eigenvector matrices of the one-qubit measurements: $\mathbf{E}_{\mathcal{M}} = \mathbf{E}_{\mathcal{M}_1} \otimes \ldots \otimes \mathbf{E}_{\mathcal{M}_{n_{qb}}}$.

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 to the three nontrivial Pauli measurements. The Hermitian measurement matrices associated with the directions X, Y, and Z are the last three Pauli matrices defined in Sec. 2.1.3 of [1]: $\boldsymbol{\sigma}_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $\boldsymbol{\sigma}_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$, and $\boldsymbol{\sigma}_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ and the corresponding eigenvector matrices may be shown to read

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

If the qubit represents the spin of an electron, those eigenvector matrices represent the measurement of the spin component along three 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. The resulting eigenvector matrix is the tensor product of the two-dimensional matrices of (5). For example for two qubits, measuring the first one along Z and the second one along X has the following eigenvector matrix:

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

This measurement is not equivalent to a two-qubit Pauli measurement: A two-qubit Pauli measurement only has two outcomes, as the considered observable is the product of the two observables associated with two one-qubit Pauli measurements. Instead we consider the "concatenation" of two one-qubit Pauli measurements, so that our measurement has d (in the example $d = 4$) outcomes and does not waste half of the information. For n_{qb} qubits there are $3^{n_{qb}}$ different measurements of this type. Both of the QST methods of Sec. III and IV as well as the fine tuning algorithms of Sec. 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., that 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.

(1) 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).

(2) 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.

(1) Reference [10] uses successive two-outcome projective measurements on nonorthogonal entangled eigenstates. And each iteration of the algorithm requires 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.

(2) Reference [14] considers projective two-outcome measurements on one-dimensional spaces. Half of those measurements can be performed using a single parallel unentangled measurement (with the identity matrix as the eigenvector matrix), but the other half cannot.

(3) Reference [15] considers two parallel unentangled measurements (called local measurements in [15]) and two d-outcome entangled measurements that can be mapped onto the other two using a gate that performs the Fourier transform. This setup is far more reasonable than the others as it requires a single known standard gate.

(4) References [2–7] 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 Sec. 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 two-outcome measurements returning either +1 or −1. There are sets of Pauli measurements the expected values of which can be deduced from the outcomes of parallel unentangled measurements without loss of information ([13] 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 papers we here make a point to only use unentangled parallel measurements. We could have chosen other matrices than (5). 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, Sec. III A describes the four types of parallel unentangled measurements that are performed, Sec. III B explains why it is reasonable to think that they are injective up to a global phase, and Sec. III C describes a first algorithm to recover the phases.

A. Types of measurements

In the QST method described here, we perform four types of measurements on the considered d-dimensional state: The first measurement measures all the qubits along Z; its eigenvector matrix, $\mathbf{E}_{Z...Z}$, is the identity matrix; the second measurement measures all the qubits along Y; the third measures all the qubits 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 $\widehat{\boldsymbol{p}_{\mathcal{M}}}$ for $\mathcal{M}$ spanning the four types of measurements. We then have an $n_s = 4d$ dimensional vector with $n_s = 4(d-1)$ degrees of freedom. We call it $\widehat{\boldsymbol{p}_s}$. The associated theoretical probability vector is $\boldsymbol{p}_s = |\mathbf{A}_s\boldsymbol{v}|^2$, where s stands for "small" because the corresponding matrix in Sec. IV has more rows. $\mathbf{A}_s$ is the concatenation of the transconjugates of the eigenvector matrices of the measurements we perform; $\mathbf{A}_s$ is defined similarly:

$$\mathbf{A}_s = \begin{bmatrix} \mathbf{E}^*_{Z...Z} \\ \mathbf{E}^*_{Y...Y} \\ \mathbf{E}^*_{X...X} \\ \mathbf{E}^*_{XYXY...} \end{bmatrix} \text{ and } \mathbf{A}_s = \begin{bmatrix} \mathbf{E}^*_{Z...Z} \\ \mathbf{E}^*_{Y...Y} \\ \mathbf{E}^*_{X...X} \\ \mathbf{E}^*_{XYXY...} \end{bmatrix} \tag{2}$$

with the nonredundant eigenvector matrices $\mathbf{E}_{.}$ defined in Sec. II D. Let us define the nonredundant probabilities $\boldsymbol{p}_s = |\mathbf{A}_s\boldsymbol{v}|^2$. Since the norm of $\boldsymbol{v}$ is 1, $\boldsymbol{p}_s$ and $\boldsymbol{p}_s$ contain the same information (see Sec. II B). In Sec. III B we will consider $\mathbf{A}_s$, n_s, and $\boldsymbol{p}_s$ 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 $\boldsymbol{p}_s$ in Sec. III C in order to recover the state from the measurements. We want to use all the measurements from $\widehat{\boldsymbol{p}_s}$ whether they are redundant or not.

B. Injectivity

$\mathbf{A}_s$ is an $n_s \times d$ matrix and $\boldsymbol{v}$ has unit norm. We want to know whether the measurements we chose are sufficient to recover any $\boldsymbol{v}$ from $|\mathbf{A}_s\boldsymbol{v}|^2$ 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 $\boldsymbol{v} \to |\mathbf{A}_s\boldsymbol{v}|^2$ is never truly injective as changing the global phase of $\boldsymbol{v}$ will not change $|\mathbf{A}_s\boldsymbol{v}|^2$. This issue of injectivity was studied before in [16,17,19] in a slightly different setup: the considered measurements are $|\mathbf{A}_s\boldsymbol{v}|$ instead of $|\mathbf{A}_s\boldsymbol{v}|^2$, but this does not change anything for the injectivity. Also $\boldsymbol{v}$ 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 $\boldsymbol{v}$ and insert the row $[0, \dots, 0, 1]$ between the $(d-1)$th row and the dth row of $\mathbf{A}_s$. This ensures that the norm of $\boldsymbol{v}$ is constrained: its square is the sum of the first d constrained measurements, because the first d rows of $\mathbf{A}_s$ are the identity matrix. With this change $\mathbf{A}_s$ has $4d-3$ rows. According to [16] the minimal number of rows for $\mathbf{A}_s$ 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 $\mathbf{A}_s$ that ensures injectivity, and proving it for a given $\mathbf{A}_s$ is a known hard problem. The closest result we found to a sufficient condition is in [17] where it is shown that for a generic $\mathbf{A}_s$, having $4d-2$ or more rows ensures injectivity. $\mathbf{A}_s$ 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 $\mathbf{A}_s$ would be in it (although it probably would because the set is of full measure), but this is a moot point because we are one row short of satisfying the $4d-2$ condition anyway. However, [19] 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 three measurement types would not be enough to achieve injectivity with $n_{qb} > 2$ as the bound of [16] 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 $n_{qb} > 2$. Four is the lowest number of measurement types for which we can hope to always achieve injectivity. In summary, we are unable to prove injectivity for the measurements defined by (2), and the validity of an associated QST algorithm will only be tested with the simulations of Sec. VI. We chose to use four measurements so that injectivity is technically possible (and likely). The types of measurements we chose are arbitrary (though we were mindful to select diverse eigenvector matrices to avoid poor conditioning issues).

C. A first quantum pure-state tomography method

In the current section, we show how the method proposed in [18] can be used in our framework to recover $\boldsymbol{v}$ from the sample probabilities $\widehat{\boldsymbol{p}_s}$, an estimate of $\boldsymbol{p}_s = |\mathbf{A}_s\boldsymbol{v}|^2$ (we only consider $\mathbf{A}_s$ from now on; $\mathbf{A}_s$ was only useful to discuss the injectivity). The optimization problem considered in [18] is the following:

$$\min_{\boldsymbol{v}} \| |\mathbf{A}_s\boldsymbol{v}| - \sqrt{\widehat{\boldsymbol{p}_s}} \| \tag{3}$$

where $\sqrt{\widehat{\boldsymbol{p}_s}}$ is the elementwise square root of $\widehat{\boldsymbol{p}_s}$ and $||.||$ is the L_2 norm. Reference [18] does not include the unit-norm constraint on $\boldsymbol{v}$ 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 $|\mathbf{A}_s\boldsymbol{v}|^2$ is the squared norm of $\boldsymbol{v}$ and the sum of the first d elements of $\widehat{\boldsymbol{p}_s}$ is 1, therefore if $|\mathbf{A}_s\boldsymbol{v}|$ is close to $\sqrt{\widehat{\boldsymbol{p}_s}}$, their squared norms will also be close, and therefore the squared norm of $\boldsymbol{v}$ will be close to 1. In [18], it is shown that (3) is equivalent to the following optimization problem (originally it came from [20]):

$$\min_{\mathbf{U} \text{ s.t. } \mathcal{C}} \operatorname{tr}(\mathbf{UM}) \tag{4}$$

where $\mathbf{M} = \operatorname{diag}(\sqrt{\widehat{\boldsymbol{p}_s}})(I - \mathbf{A}_s\mathbf{A}_s^\dagger)\operatorname{diag}(\sqrt{\widehat{\boldsymbol{p}_s}})$, † is the pseudoinverse, $\operatorname{diag}(\sqrt{\widehat{\boldsymbol{p}_s}})$ is the diagonal matrix the diagonal of which is $\sqrt{\widehat{\boldsymbol{p}_s}}$, and $\mathcal{C}$ represents the following condition on the

$n_s \times n_s$ matrix $\mathbf{U}$:

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

Reference [18] shows that if $\mathbf{U}$ is a solution of (4), then the associated $\boldsymbol{u}$ of (5) is an approximation of the phase of $\mathbf{A}_s\boldsymbol{v}$, and the resulting estimate of $\boldsymbol{v}$ defined as

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

($*$ is the elementwise product) is the solution of (3) proposed in [18]. We do not detail the proof here but the intuition behind the formulation of (4) is fairly simple: the criterion of (4) can be rewritten as $(\boldsymbol{u} * \sqrt{\widehat{\boldsymbol{p}}_s})^*(I - \mathbf{A}_s\mathbf{A}_s^\dagger)(\boldsymbol{u} * \sqrt{\widehat{\boldsymbol{p}}_s})$. $\boldsymbol{u} * \sqrt{\widehat{\boldsymbol{p}}_s}$ is our estimate of $\mathbf{A}\boldsymbol{v}$ and $I - \mathbf{A}_s\mathbf{A}_s^\dagger$ is the projection on the complement of the image of $\mathbf{A}$ (the kernel of $\mathbf{A}^*$). Therefore, we are looking for the phases ($\boldsymbol{u}$) that bring our estimate of $\mathbf{A}\boldsymbol{v}$ as close as possible to the image of $\mathbf{A}$ (more precisely they minimize the norm of the projection on the kernel of $\mathbf{A}^*$). Equation (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 [18] the criterion $\text{tr}(\mathbf{UM})$ is convex and the only constraint that makes the problem nonconvex 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:

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

In the absence of noise, the solution that we are looking for, $\mathbf{U}_0 = \boldsymbol{u}_0\boldsymbol{u}_0^*$ where $\boldsymbol{u}_0$ is the phase of $\mathbf{A}\boldsymbol{v}$, is a solution of both the relaxed problem (7) and the original problem (4), because the criteria of both of those problems are positive (as $\mathbf{M}$ is a positive matrix) and it is easy to check that (in the absence of noise) $\text{tr}(\mathbf{U}_0\mathbf{M}) = 0$. This does not guarantee that the relaxation does not change anything though, as the minimum might not be unique. Like in most of [18] we choose to ignore this issue because, as we will see further in this paper, the relaxed problem will only be used to initialize a nonconvex faster and more precise optimization algorithm, so we can tolerate small errors. Once (7) is solved using the PhaseCut algorithm of [18] (rewritten in Appendix B), the eigenvectors and eigenvalues of the solution $\mathbf{U}$ are computed. In order to get an estimate of $\boldsymbol{u}$, [18] then computes $\widehat{\boldsymbol{u}}$, the eigenvector associated with the largest eigenvalue. From $\widehat{\boldsymbol{u}}$, we get the estimate of $\boldsymbol{v}$ defined in (6):

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

In [18] this method is tested with $\mathbf{A}$ matrices which represent usual use cases in the signal or image processing community (oversampled Fourier transform, multiple random illumination filters, and 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 Sec. V.

D. Comparison with the literature

Let us sum up the main features of our first QST algorithm. (1) It uses $4d$ probabilities that can be obtained by averaging the results of four parallel unentangled measurements. (2) It is reasonable to think that the chosen measurements are injective (the failure set is most likely empty). (3) The algorithm that reconstructs the state is not explicit (optimization). Goyeneche *et al.* [15] use the same number of measurement types, have a known failure space of zero measure, and provide an explicit reconstruction algorithm. The main advantage of our approach based on PhaseCut as 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:

$$\left\{\underbrace{Z\dots Z}_{n_{qb} \text{ times}}, \left\{\underbrace{Z\dots Z}_{n_{qb}-i \text{ times}}\ S\,\underbrace{X\dots X}_{i-1 \text{ times}}, \begin{matrix} 1 \leqslant i \leqslant n_{qb} \\ S \in \{X, Y\} \end{matrix}\right\}\right\}$$

. 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:

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

Each measurement is performed several times and we compute the sample probabilities $\widehat{\boldsymbol{p}}_t$ which are estimates of the theoretical probabilities $\boldsymbol{p}_t = |\mathbf{A}_t\boldsymbol{v}|^2$. The value $2n_{qb} + 1$ sounds like a lot compared to the four measurement types of Sec. III but it is a small fraction of the $3^{n_{qb}}$ possible types of measurements defined in Sec. 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 Sec. IV B.

B. A recursive pure quantum state tomography method

Let us show how a vector $\boldsymbol{v}$ can be recovered up to a global phase from $|\mathbf{A}_t\boldsymbol{v}|^2$ 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 $\boldsymbol{v}$ from $|\mathbf{A}_t\boldsymbol{v}|^2$) 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 (5). The state vector is $\boldsymbol{v} = \binom{|v_1|}{|v_2|e^{i\theta}}$. Basic calculations show that

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

Therefore, $|\mathbf{A}_t(1)\boldsymbol{v}|^2$ 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. (1) If $|v_1| = 0$ or $|v_2| = 0$, then knowing $|v_1|$ and $|v_2|$ is enough because $\binom{|v_1|}{|v_2|}$ is the same as $\boldsymbol{v}$ up to a global phase. Thus, there is no need to compute θ. (2) 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 $\boldsymbol{v}$. 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 $\boldsymbol{w}$ with $2^{n_{qb}-1}$ elements $f_{n_{qb}-1}(|\mathbf{A}_t(n_{qb} - 1)\boldsymbol{w}|^2)$ is equal to $\boldsymbol{w}$ up to a global phase. Let $\boldsymbol{v}$ be a $d = 2^{n_{qb}}$ element vector (it does not have to be unit norm). We split $\boldsymbol{v}$ into two $2^{n_{qb}-1}$ element vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$: $\boldsymbol{v} = [\begin{smallmatrix}\boldsymbol{w}_1 \\ \boldsymbol{w}_2\end{smallmatrix}]$. Let us show how $\boldsymbol{v}$ can be recovered up to a global phase from $|\mathbf{A}_t(n_{qb})\boldsymbol{v}|^2$ using the fact that $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ can be recovered from $|\mathbf{A}_t(n_{qb} - 1)\boldsymbol{w}_1|^2$ and $|\mathbf{A}_t(n_{qb} - 1)\boldsymbol{w}_2|^2$ up to global phases using $f_{n_{qb}-1}$. We start by comparing $\mathbf{A}_t(n_{qb} - 1)$ to $\mathbf{A}_t(n_{qb})$: $\mathbf{A}_t(n_{qb} - 1) = [\begin{smallmatrix}\mathbf{E}^*_{s_1} \\ \vdots \\ \mathbf{E}^*_{s_{2n_{qb}-1}}\end{smallmatrix}]$ with (9) giving the values of the strings $s_1, \ldots, s_{2n_{qb}-1}$. We can also notice that

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

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

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

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

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

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

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

From (14) and (13), we see that all the elements of $|\mathbf{A}_t(n_{qb} - 1)\boldsymbol{w}_l|^2_{i_k}$ are in $|\mathbf{A}_t(n_{qb})\boldsymbol{v}|^2_{i_k}$ $\forall k \in \{1, \ldots, 2n_{qb} - 1\}$. Since $|\mathbf{A}_t(n_{qb} - 1)\boldsymbol{w}_l|^2_{i_k}$ $\forall k \in \{1, \ldots, 2n_{qb} - 1\}$ spans all the vector $|\mathbf{A}_t(n_{qb} - 1)\boldsymbol{w}_l|^2$ we have shown that $|\mathbf{A}_t(n_{qb} - 1)\boldsymbol{w}_l|^2$ is known from part of the measurements ($|\mathbf{A}_t(n_{qb})\boldsymbol{v}|^2$) for $l = 1$ and 2. Using the induction hypothesis we can apply $f_{n_{qb}-1}$ to the known quantities $|\mathbf{A}_t(n_{qb} - 1)\boldsymbol{w}_1|^2$ and $|\mathbf{A}_t(n_{qb} - 1)\boldsymbol{w}_2|^2$ in order to get $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ up to global phases. Let us call our estimates $\widehat{\boldsymbol{w}_1}$ and $\widehat{\boldsymbol{w}_2}$, $\boldsymbol{w}_1 = e^{i\theta_1}\widehat{\boldsymbol{w}_1}$ and $\boldsymbol{w}_2 = e^{i\theta_2}\widehat{\boldsymbol{w}_2}$. We now only need to know $\theta_2 - \theta_1$ in order to know $\boldsymbol{v}$ up to a global phase. Let us get $\theta_2 - \theta_1$ from the last $2d$ elements of $|\mathbf{A}_t(n_{qb})\boldsymbol{v}|^2$. We define $\boldsymbol{L_m}$ as the column vector containing those last $2d$ elements:

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

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

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

Let us introduce the following notations:

$$\begin{aligned} \boldsymbol{m} &= \tfrac{1}{2}|\mathbf{E}^*_{X\ldots X}\widehat{\boldsymbol{w}_1}|^2 + \tfrac{1}{2}|\mathbf{E}^*_{X\ldots X}\widehat{\boldsymbol{w}_2}|^2, \\ \boldsymbol{d_c} &= \overline{\mathbf{E}^*_{X\ldots X}\widehat{\boldsymbol{w}_1}} * \mathbf{E}^*_{X\ldots X}\widehat{\boldsymbol{w}_2}, \\ \boldsymbol{d}(\theta) &= \cos(\theta)\mathrm{Re}(\boldsymbol{d_c}) - \sin(\theta)\mathrm{Im}(\boldsymbol{d_c}) \end{aligned} \tag{15}$$

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

$$\boldsymbol{L_m}(\theta_2 - \theta_1) = \begin{bmatrix} \boldsymbol{m} + \boldsymbol{d}(\theta_2 - \theta_1) \\ \boldsymbol{m} - \boldsymbol{d}(\theta_2 - \theta_1) \\ \boldsymbol{m} + \boldsymbol{d}(\theta_2 - \theta_1 - \pi/2) \\ \boldsymbol{m} - \boldsymbol{d}(\theta_2 - \theta_1 - \pi/2) \end{bmatrix}. \tag{16}$$

We aim at deriving $\theta_2 - \theta_1$ from $\boldsymbol{L_m}$ (which is known from the measurements). We first notice from the definition of $\boldsymbol{d}(\theta)$ in (15) that if $\boldsymbol{d_c}$ is zero on every component then $\boldsymbol{d}(\theta_2 - \theta_1)$ is also zero on every component (which means it does not depend on $\theta_2 - \theta_1$) and $\boldsymbol{L_m}$ is simply $\boldsymbol{m}$ repeated four times [see (16)]. Therefore recovering $\theta_2 - \theta_1$ (and $\boldsymbol{v}$) from $\boldsymbol{L_m}$ is impossible. However, we hereafter show that this is the only case when $\theta_2 - \theta_1$ cannot be recovered from $\boldsymbol{L_m}$. And the ensemble of $\boldsymbol{v}$ which makes this occur has zero measure. Let us assume that at least a single element of $\boldsymbol{d_c}$ is not zero. Let us call k its index, d_k the corresponding nonzero element (we take the element which has the highest modulus), and $d_k(\theta_2 - \theta_1)$ and m_k the kth elements of $\boldsymbol{d}(\theta_2 - \theta_1)$ and $\boldsymbol{m}$, respectively. Then all we need is the kth and $(k + d)$th elements of $\boldsymbol{L_m}$ the expressions of which are $m_k + \cos(\theta_2 - \theta_1)\mathrm{Re}(d_k) - \sin(\theta_2 - \theta_1)\mathrm{Im}(d_k)$ and $m_k + \sin(\theta_2 - \theta_1)\mathrm{Re}(d_k) + \cos(\theta_2 - \theta_1)\mathrm{Im}(d_k)$. Those known elements can be put in a column vector and rewritten as

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

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 two elements of $\boldsymbol{L_m}$ (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 $\boldsymbol{L_m}$. But, in practice the sample probabilities give an imperfect estimate of $\boldsymbol{L_m}$ which we call $\widehat{\boldsymbol{L_m}}$. In order to be robust to the errors, we aim to find the angle $\widehat{\theta_2 - \theta_1}$ that minimizes $||\boldsymbol{L_m}(\theta_2-\theta_1) - \widehat{\boldsymbol{L_m}}||$; this way we use all sample probabilities and not just two. We use a quasi-Newton BFGS (Broyden–Fletcher–Goldfarb–Shanno) algorithm [21] (implemented with fminunc in the MATLAB numerical software [22]) 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.

(1) Recovering the state (up to a global phase) from the measurements is possible for $n_{qb} = 1$.

(2) 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 $\boldsymbol{v}$ 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 $\widehat{\boldsymbol{v}}_{\mathrm{rec}}$.

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 one. 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 one qubit. For one qubit, the state is recovered using (10) which involves six probabilities, among which only four are required [we could obtain the same result without using the fourth and sixth elements of $|\mathbf{A_t}(1)\boldsymbol{v}|^2$]. 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 two 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 $\boldsymbol{v}$ 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 errors 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.

(1) It uses $(2n_{qb} + 1)d$ probabilities that can be obtained by averaging the results of $2n_{qb} + 1$ parallel unentangled measurements.

(2) The measurements are injective outside a known failure set with zero measure.

(3) 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 (it 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 $\boldsymbol{v}$, denoted as $\widehat{\boldsymbol{v}}_{pc}$ and $\widehat{\boldsymbol{v}}_{\mathrm{rec}}$, respectively. $\widehat{\boldsymbol{v}}_{pc}$ 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 Sec. IV B that computes $\widehat{\boldsymbol{v}}_{\mathrm{rec}}$ is also imperfect. It relies heavily on the measurements along $Z\ldots Z, Z\ldots ZX$ and $Z\ldots ZY$ (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 $X\ldots X$ and $YX\ldots X$ [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 $\boldsymbol{v}$ as the measurements along $Z\ldots Z$, but the former are barely used. Therefore the estimation methods of Secs. 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:

$$(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{y}}) = \underset{\boldsymbol{x},\boldsymbol{y} \text{ s.t. } ||\boldsymbol{x}||_2 + ||\boldsymbol{y}||_2 < 1}{\arg\min} \mathscr{L}_{(\boldsymbol{x},\boldsymbol{y})}(\widehat{\boldsymbol{p}}) \tag{18}$$

where $\widehat{\boldsymbol{p}}$ is the vector that contains sample probabilities and $\mathscr{L}_{(\boldsymbol{x},\boldsymbol{y})}(\widehat{\boldsymbol{p}})$ is to be understood as the negative log likelihood of measuring the sample probabilities $\widehat{\boldsymbol{p}}$ if the true state is $\boldsymbol{v}(\boldsymbol{x},\boldsymbol{y})$, with $\boldsymbol{x}$ and $\boldsymbol{y}$ defined hereafter. In the whole paper, whenever we write "negative log likelihood" (or $\mathscr{L}$) 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 $\boldsymbol{v}(\boldsymbol{x},\boldsymbol{y})$ with respect to which $\mathscr{L}$ will be minimized is defined as [0.9]$\boldsymbol{v}(\boldsymbol{x},\boldsymbol{y}) = [\sqrt{1-||\boldsymbol{x}||_2^2-||\boldsymbol{y}||_2^2}, x_1+iy_1,\dots,x_{d-1}+iy_{d-1}]^T$. $\boldsymbol{x}$ and $\boldsymbol{y}$ are $d-1$ element vectors representing the real and imaginary parts of the last elements of $\boldsymbol{v}$. The constraint in (18) is $r^2<1$ (with $r=\sqrt{||\boldsymbol{x}||_2^2+||\boldsymbol{y}||_2^2}$) and not $r^2 \leqslant 1$ because optimization is easier on an open set. We mitigate the effect of this imperfect constraint by permuting the first component of $\boldsymbol{v}$ and the component of $\boldsymbol{v}$ 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 changes are limited to the optimization algorithm.

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

$$\boldsymbol{x}' = \frac{\tan\left(\frac{\pi}{2}r\right)}{r}\boldsymbol{x} \text{ and } \boldsymbol{x} = \frac{\frac{2}{\pi}\text{atan}(r')}{r'}\boldsymbol{x}',$$
$$\boldsymbol{y}' = \frac{\tan\left(\frac{\pi}{2}r\right)}{r}\boldsymbol{y} \text{ and } \boldsymbol{y} = \frac{\frac{2}{\pi}\text{atan}(r')}{r'}\boldsymbol{y}'$$

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

$$(\widehat{\boldsymbol{x}'},\widehat{\boldsymbol{y}'}) = \underset{\boldsymbol{x}',\boldsymbol{y}'}{\arg\min}\,\mathscr{L}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}}). \tag{19}$$

In order to solve (19) we again use the BFGS [21] 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 nonconvex optimization methods, we need a good initialization point, and we use either $\widehat{\boldsymbol{v}}_{pc}$ or $\widehat{\boldsymbol{v}}_{\text{rec}}$. The most likely $\boldsymbol{v}$ is $\widehat{\boldsymbol{v}}_{\text{ML}} = \boldsymbol{v}(\widehat{\boldsymbol{x}'},\widehat{\boldsymbol{y}'})$, with $\widehat{\boldsymbol{x}'},\widehat{\boldsymbol{y}'}$ defined in (19). All that remains now is to define the expression of the negative log likelihood $\mathscr{L}$ with respect to $\boldsymbol{v}$. In the following two subsections we will give two expressions for the normalized log likelihood: $\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}})$ and $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}})$.

B. Exact likelihood

In [23] the formula for the likelihood of a multioutput quantum measurement is given (albeit for a mixed state represented by ρ which we would have to replace by $\boldsymbol{v}\boldsymbol{v}^*$). It boils down to

$$\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}}) = -\sum_{k=1}^{n_{\text{prob}}} n_k \ln\{[|\mathbf{A}\boldsymbol{v}(\boldsymbol{x}',\boldsymbol{y}')|^2]_k\}. \tag{20}$$

$[|\mathbf{A}\boldsymbol{v}(\boldsymbol{x}',\boldsymbol{y}')|^2]_k$ is the kth element of $|\mathbf{A}\boldsymbol{v}(\boldsymbol{x}',\boldsymbol{y}')|^2$, $\mathbf{A}$ is the measurement matrix, either $\mathbf{A}_s$ or $\mathbf{A}_t$; n_k is the number of times the kth outcome occurred, i.e., the kth element of $\widehat{\boldsymbol{p}}$ (either $\widehat{\boldsymbol{p}_s}$ or $\widehat{\boldsymbol{p}_t}$) 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 $\widehat{\boldsymbol{p}}$ the likelihood of which we want to compute is the average of independent realizations of the same random variable. Its expected value is the vector of theoretical probabilities $\boldsymbol{p}(\boldsymbol{x}',\boldsymbol{y}')$ that depends on the state. Let us define $\boldsymbol{\varepsilon}(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}') = \widehat{\boldsymbol{p}} - \boldsymbol{p}(\boldsymbol{x}',\boldsymbol{y}')$ and $\boldsymbol{\varepsilon}(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}')$ is $\boldsymbol{\varepsilon}(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}')$ with the last element removed [no information is lost as the sum of the elements of $\boldsymbol{\varepsilon}(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}')$ is zero]. In Appendix A, we show that if N is the number of times the measurements have been averaged, then $\sqrt{N}\boldsymbol{\varepsilon}(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}')$ asymptotically ($N \to +\infty$) follows a zero-mean multivariate normal distribution. Its covariance matrix $\boldsymbol{\Sigma}$ is computed in Appendix A. $\boldsymbol{\Sigma}$ depends on the theoretical probabilities, and we need to remove this dependency. With that in mind, we get to the following approximation for the negative log likelihood:

$$\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}}) = N\boldsymbol{\varepsilon}(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}')^T\widetilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{\varepsilon}(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}') \tag{21}$$

where $\widetilde{\boldsymbol{\Sigma}}^{-1}$ is an approximation of the covariance matrix that uses $\widetilde{\boldsymbol{p}} = \frac{\widehat{\boldsymbol{p}}+\frac{5}{N}}{1+\frac{5d}{N}}$ as a regularized approximation of $\boldsymbol{p}$; this is justified in Appendix A. Appendix A also shows that this equation boils down to

$$\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}}) = N\sum_{k=1}^{d}\frac{\varepsilon_k(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}')^2}{\widetilde{p_k}}. \tag{22}$$

This log likelihood is the result of two approximations that are true only when $N \to +\infty$: we approximated $\boldsymbol{\varepsilon}(\widehat{\boldsymbol{p}},\boldsymbol{x}',\boldsymbol{y}')$ as the realization of a Gaussian random vector and we used an approximation for $\boldsymbol{\Sigma}$. In practice, the resulting approximation is smoother and easier to minimize than $\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}})$ if the initialization point is not good enough (as will be shown in Sec. VI C). However, with a good initialization, the state that minimizes $\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}})$ should be closer to the true state than the one that minimizes $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}})$. The smaller N, the starker the difference. This will be shown in Sec. VI B.

D. Mixed minimization

As stated above $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ is supposed to be easier to minimize but the minimum of $\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ is supposed to be a better estimate. A good way to combine the two advantages is to start the optimization process by minimizing $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ and finish it by minimizing $\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$. In practice, we here again run the BFGS algorithm [21] on $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ for 100 iterations starting from the initialization point of Secs. III or IV; this yields $\widehat{\boldsymbol{v}}_{\text{inter}}$. And then we run the BFGS algorithm on $\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ starting from $\widehat{\boldsymbol{v}}_{\text{inter}}$ 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 Sec. 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 Sec. III. We only explained PhaseCut for the setup with four different measurement types described in Sec. 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 Sec. IV A. In the current section, we test both PhaseCut and the recursive algorithm on 50 randomly generated seven-qubit pure states. Each real and imaginary part of each component of the state vector is set to a Gaussian pseudorandom number before the vector gets normalized. The two sets of measurement types of Secs. III A and IV A are considered. They contain, respectively, 4 and $2\times 7+1=15$ measurement types. We test those algorithms with two different fixed numbers of total measurements N_C: 5000 and 500 000. Thus each one of the four measurement types of the setup of Sec. III A is performed either $N_C=1250$ or 125 000 times and each one of the 15 measurement types of the setup of Sec. III A is performed either $N_C=333$ or 33 333 times. The metric used in order to quantify the proximity of $\widehat{\boldsymbol{v}}$ to the actual vector $\boldsymbol{v}$ up to a phase factor is

$$\mu=||\boldsymbol{v}-\widehat{\boldsymbol{v}}\cdot e^{-i\xi}||_2 \tag{23}$$

with ξ the angle that minimizes our metric: $e^{i\xi}=\frac{\boldsymbol{v}^*\widehat{\boldsymbol{v}}}{|\boldsymbol{v}^*\widehat{\boldsymbol{v}}|}$. 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 zero). A more widely used metric in the literature is the fidelity (see Sec. 9.2.2 in [1]) $f=|\boldsymbol{v}^*\widehat{\boldsymbol{v}}|$. It can be shown that $f=(1-\frac{\mu^2}{2})$. We do not use the fidelity because the small errors are pushed too close to 1. We also think our metric is more intuitive, because μ is the norm of the error between two unit-norm vectors, and the meaning of $\mu=0.10$ is clearer than $f=0.995$. Figure 1 shows the error of $\widehat{\boldsymbol{v}}_{pc}$ obtained by using PhaseCut with 100 to 100 000 iterations for the two setups (4 and 15 measurement types). With 15 (and not with four) measurement types, the recursive algorithm can be implemented. We display the biggest and smallest errors of $\widehat{\boldsymbol{v}}_{\text{rec}}$ obtained with the recursive algorithm with horizontal bold red (upper) and green (lower) lines, respectively. The recursive algorithm is performed with a fixed number of steps; this is why we plot its errors as horizontal lines and not as curves 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. With enough iterations ($\approx 10^4$ for $N_c=5000$ and $\approx 10^5$ for $N_c=500\,000$) PhaseCut is more precise than the recursive algorithm in the setups for which they can both be implemented, but, as will now be shown, it is much slower. Each iteration of PhaseCut is costly, because we are working on an $n_{\text{prob}}\times n_{\text{prob}}$ matrix. With MATLAB, on a 2.11-GHz four-core processor with a 32-GB RAM, each iteration of PhaseCut takes around 4 ms for the setup with four types of measurements and around 45 ms for the setup with 15 types of measurements. In that same 15 measurement type of setup, the recursive algorithm takes 200 ms. This is much faster than PhaseCut, which runs in minutes, as it requires thousands of iterations.

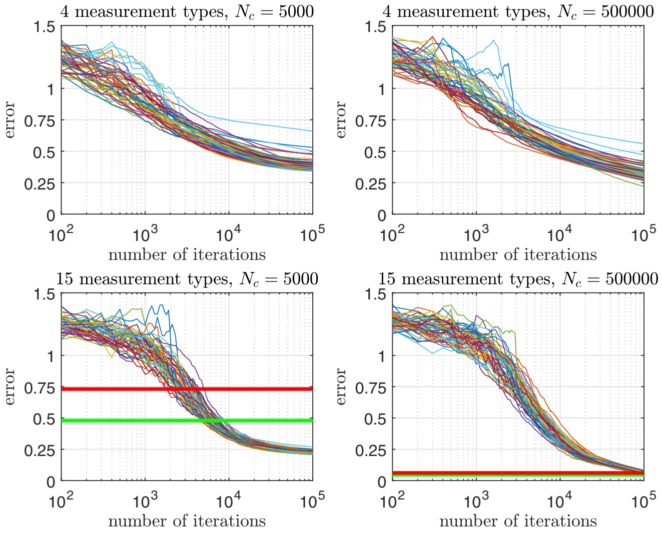


FIG. 1. Estimation errors [defined in (23)] of the initialization algorithms. The bold red (upper) and green (lower) horizontal lines are the worst and best errors for the recursive algorithm (only available with 15 measurement types) on the 50 randomly generated pure states. The other curves represent the evolution of the error on the PhaseCut estimates with the 50 states. Each iteration of PhaseCut is costly. In Sec. VI C, we set its number of iterations to 5000; we also decide to use the recursive algorithm whenever possible.

B. Likelihood estimator comparison

In Sec. V we defined two likelihood estimators, based on the likelihood maximization. The first one minimizes the true negative log likelihood $\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ and the other minimizes a version of the negative log likelihood that is supposed to be smoother, namely, $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$. We know that $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ 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 $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ to be a worse estimator than the global minimum of $\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ for a limited number of measurements. In order to check whether this is true and quantify the difference, we compute the errors of both estimators when they are initialized at the true state $\boldsymbol{v}$. 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 $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ and then minimizes $\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$. These three types of errors are computed with 1000 randomly generated initial states for the four setups described in Sec. VI A 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 derived from the 1000 errors associated with the initial states; those cdf are shown

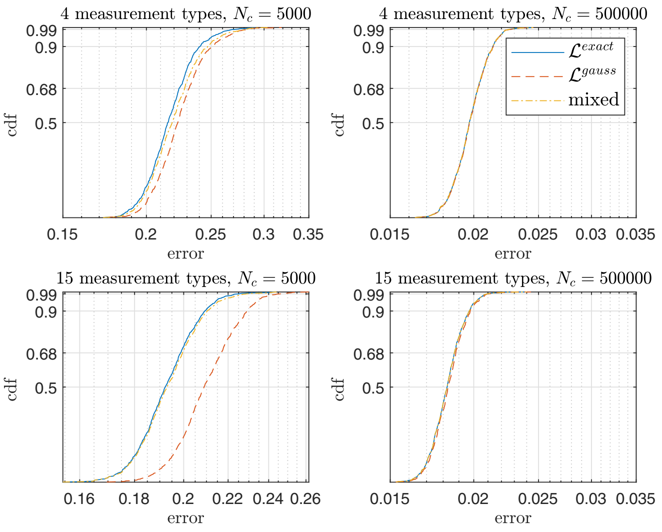


FIG. 2. Empirical cdf of the errors of the three maximum likelihood estimators.

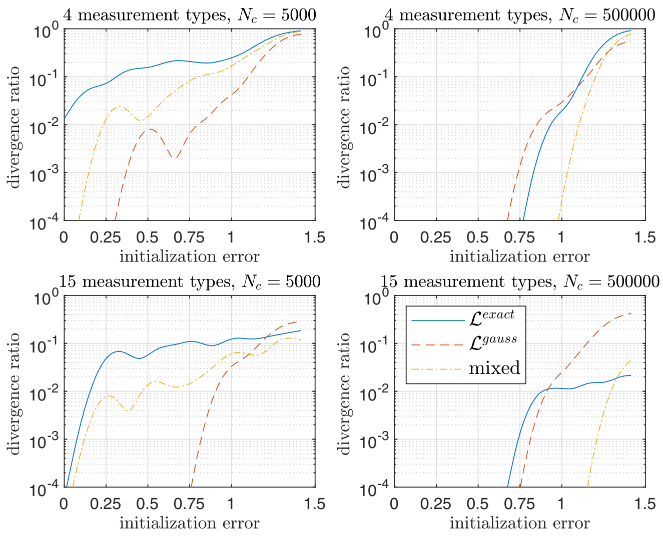


FIG. 3. Convergence of the different likelihood algorithms in the presence of initialization errors.

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 $\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$. There can be small differences however. Its turns out that they sometimes converge toward close but different minima. This is due to the fact that the small error made by the first 100 iterations of the mixed algorithm (during which $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ is minimized) can be enough to affect the final result. The differences between the three 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 we 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 $\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ and $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ as well as the mixed algorithm. We randomly generate 1000 states $\boldsymbol{v}$ to be estimated (with the same method as the one used to generate the 50 initial states of Sec. VI A). They 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 $\boldsymbol{v}$ and define $\{b_i^{\text{algo}}, i \leqslant 1000, \text{algo} \in \{\text{exact, Gauss, mixed}\}\}$ 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 them is smaller than 1% of the error between the first one (initialized without error) and the true state vector. For each of the three algorithms, we then define the rate of divergence $\delta_{\text{algo}}(\mu)$ 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:

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

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 $\delta_{\text{algo}}(\mu)$ will be close to zero. If the associated b_i^{algo} are 1 (i.e., the algorithm does not converge towards the proper minimum) then $\delta_{\text{algo}}(\mu)$ will be close to 1. The parameter α quantifies how far away from μ we look for results; we selected $\alpha = 0.1$. Figure 3 shows the rates of divergence of the three algorithms in the four setups described in Sec. VI A with 4 or 15 measurement types and 5000 or 500 000 total measurements. The two plots on the right are of limited interest to us as the rate of divergence is always very low ($\leqslant 10^{-4}$) 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 (in the two plots on the left), the best algorithm seems to be the minimization of $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$. Indeed increased robustness to the initialization error is the whole reason why we introduced $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$. The mixed algorithm does not quite reach the same robustness but it is certainly an improvement over the algorithm that minimizes $\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ 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

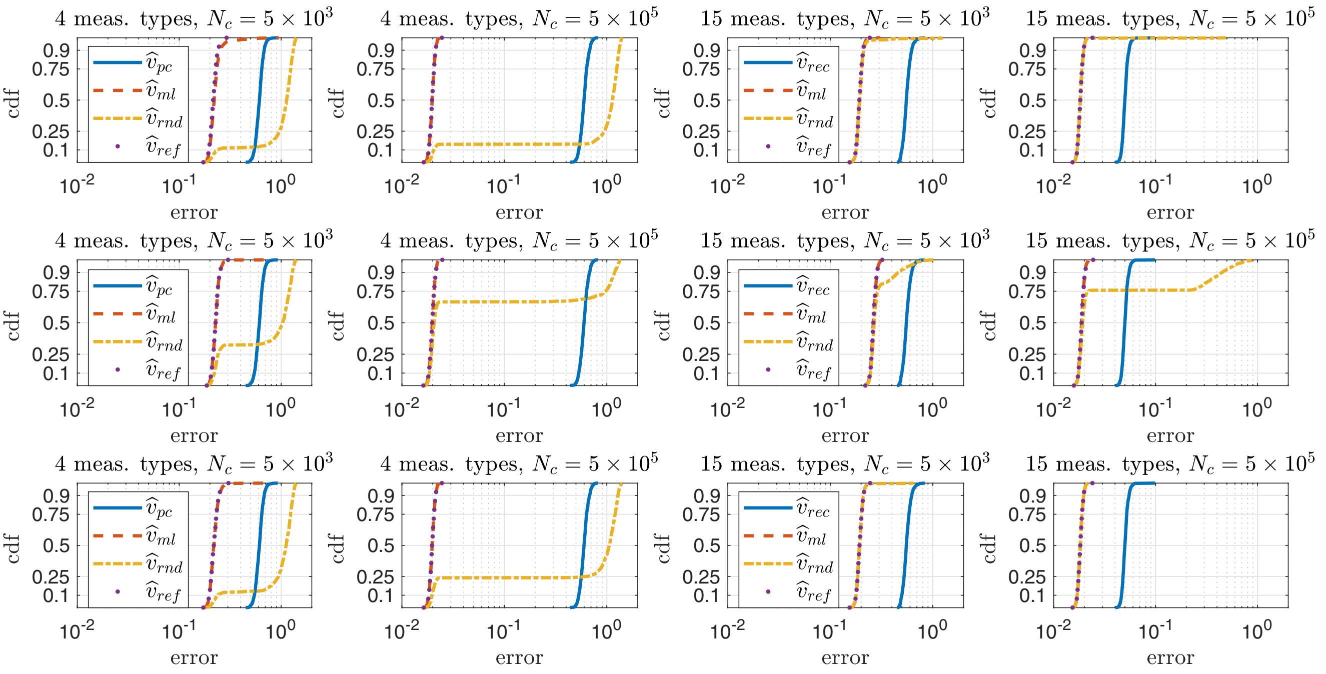


FIG. 4. Empirical cdf of the QST error. In the four plots of the top row, $\mathscr{L}^{\text{exact}}_{(x',y')}$ is minimized. For the middle row, it is $\mathscr{L}^{\text{Gauss}}_{(x',y')}$. The lowest row corresponds to the mixed algorithm. The blue solid curve (the one on the right in every plot) is the cdf of the error of the initialization algorithm (PhaseCut for four measurement types and the recursive algorithm for 15 measurement types); it does not depend on the likelihood maximization algorithm and is the same in every row. The legends of the second and fourth columns are not displayed in order to keep the curves visible; they would be the same as the legends of the first and third columns, respectively.

real global minimum. The rate of divergence δ is not useless however, and Fig. 3 shows us that, generally, with either the mixed algorithm or the algorithm that minimizes $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$, an initialization error lower than 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 Sec. IV with 15 types of measurements for seven qubits (because it is faster than PhaseCut) and when PhaseCut has to be used (so with four measurement types) we only perform 5000 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 Secs. III and IV, fine tuned with the three algorithms of Sec. V on $n_{qb} = 7$ qubits, with the four setups described in Sec. VI A. For each setup, and for each version of the ML algorithm, four estimates of $\boldsymbol{v}$ are computed.

(1) The initial estimate has $\widehat{\boldsymbol{v}}_{pc}$ for the setup with four measurement types or $\widehat{\boldsymbol{v}}_{\text{rec}}$ for the setup with 15 measurement types. It does not depend on the choice of the ML algorithm.

(2) $\widehat{\boldsymbol{v}}_{\text{ML}}$ is the result of the likelihood optimization (minimizing either $\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ or $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ or both successively) initialized at the initial estimate.

(3) $\widehat{\boldsymbol{v}}_{\text{ref}}$ is the result of the likelihood optimization initialized at the true $\boldsymbol{v}$ (not available in practice; it should be the global maximum likelihood; if $\widehat{\boldsymbol{v}}_{\text{ML}} = \widehat{\boldsymbol{v}}_{\text{ref}}$ then the initial estimate was good enough). We call $\widehat{\boldsymbol{v}}_{\text{ref}}$ the reference; it has already been defined (but not named) in Sec. VI B and represented in Fig. 2.

(4) Finally, $\widehat{\boldsymbol{v}}_{\text{rnd}}$ is the result of the likelihood optimization initialized at a randomly generated normalized vector (if $\widehat{\boldsymbol{v}}_{\text{rnd}}$ is not worse than $\widehat{\boldsymbol{v}}_{\text{ML}}$, then the initial estimate was unnecessary and one can only use the maximum likelihood algorithm initialized randomly).

For each setup, 1000 states $\boldsymbol{v}$ to be estimated are randomly generated (with the same method as the one used to generate the 50 initial states of Sec. VI A). We compute the estimates of each $\boldsymbol{v}$ with the different algorithms and display the empirical cdf of the errors in Fig. 4 (each row of plots corresponds to a different ML algorithm). The performances of the three ML algorithms are quite similar (when excluding the random initialization), but some differences can be noted.

(1) The algorithm that minimizes $\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ is supposed to be less robust to the initialization error than the others. It is only apparent for the setup with four measurements and $N_c =$ 5000. $\widehat{\boldsymbol{v}}_{\text{ML}}$ is not quite as precise as $\widehat{\boldsymbol{v}}_{\text{ref}}$.

(2) The algorithm that minimizes $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ does not have that problem; $\widehat{\boldsymbol{v}}_{\text{ML}}$ and $\widehat{\boldsymbol{v}}_{\text{ref}}$ are always indistinguishable. However, the version of $\widehat{\boldsymbol{v}}_{\text{ref}}$ computed by minimizing $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ is not as precise as the version that minimizes $\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$. This can be seen by comparing Figs. 4 and 5 but it is more visible in Fig. 2 that represents the performances of the three references in a single graph.

(3) The mixed algorithm seems to combine the advantages of those based on $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ and $\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$. $\widehat{\boldsymbol{v}}_{\text{ML}}$ is almost equal to $\widehat{\boldsymbol{v}}_{\text{ref}}$, and $\widehat{\boldsymbol{v}}_{\text{ref}}$ is almost as good with this mixed algorithm as

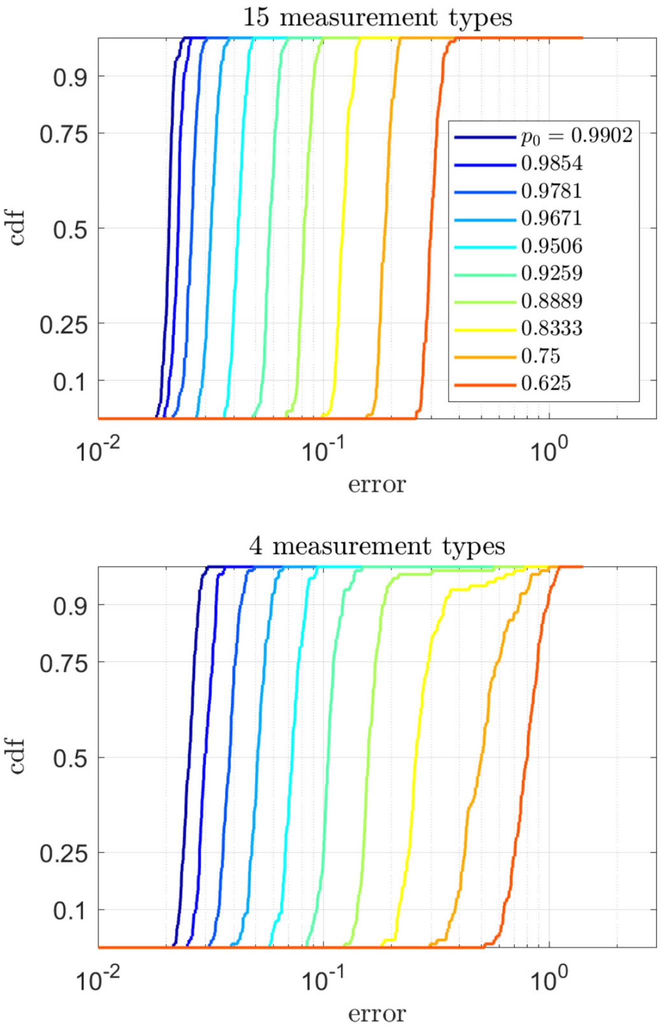


FIG. 5. Empirical cdf of the error of $\widehat{\boldsymbol{v}_{\text{ML}}}$ computed with the mixed algorithm. The input states are mixed states. The curves with the smaller p_0 are on the left; those with the highest p_0 are on the right.

with $\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ (see Fig. 2 for a clearer comparison of the two values of $\widehat{\boldsymbol{v}}_{\text{ref}}$).

The performances of $\widehat{\boldsymbol{v}}_{\text{rnd}}$, the maximum likelihood estimators initialized at a random point, are interesting. For the setup with four measurement types, it is always a much worse estimate than $\widehat{\boldsymbol{v}}_{\text{ML}}$. But with 15 measurement types it is (almost) as good as the maximum likelihood estimators initialized at $\widehat{\boldsymbol{v}}_{\text{rec}}$ (unless we use the $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ minimization). This could make us question the relevance of the recursive algorithm defined in Sec. 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 II). We can also compare the performances of the two initialization algorithms $\widehat{\boldsymbol{v}}_{pc}$ or $\widehat{\boldsymbol{v}}_{\text{rec}}$ (blue solid curve) with $\widehat{\boldsymbol{v}}_{\text{ML}}$ (red dashed curve). The error on $\widehat{\boldsymbol{v}}_{\text{ML}}$ is at least three times smaller (or a lot less for $\widehat{\boldsymbol{v}}_{pc}$ and $N_c = 500\,000$) 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 $\widehat{\boldsymbol{v}}_{\text{rnd}}$ is unwise because $\widehat{\boldsymbol{v}}_{pc}$ and $\widehat{\boldsymbol{v}}_{\text{rec}}$ can be improved with the ML algorithm whereas $\widehat{\boldsymbol{v}}_{\text{rnd}}$ cannot as it is a local minimum of the likelihood. Furthermore, with $N_c = 5000$, $\widehat{\boldsymbol{v}}_{pc}$ and $\widehat{\boldsymbol{v}}_{\text{rec}}$ have a similar accuracy (respectively for 4 and 15 measurement types). And with $N_c = 500\,000$, $\widehat{\boldsymbol{v}}_{\text{rec}}$ is a much better estimate than $\widehat{\boldsymbol{v}}_{pc}$ because the PhaseCut algorithm is limited to 5000 iterations (allowing it enough iterations to converge properly would be much slower and less accurate than the likelihood maximization). After likelihood optimization the performances of $\widehat{\boldsymbol{v}}_{\text{ML}}$ 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 ten 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 $\widehat{\boldsymbol{v}}_{\text{rec}}$ has a zero measure failure set on which phase recovery is impossible (see Sec. IV) turns out to be a nonissue. We could have expected to see some outliers in the error of $\widehat{\boldsymbol{v}}_{\text{rec}}$, and the $\widehat{\boldsymbol{v}}_{\text{ML}}$ computed from it, if the randomly generated $\boldsymbol{v}$ 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 setup with four measurement types. 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. Tables I and II give the median execution time of all the algorithms on an Intel Xeon Gold 6226R 2.9-GHz core. All the scripts ran on one thread on MATLAB. There are no significant differences between the three ML algorithms when they are not initialized at random. The random initialization is never relevant, as for the four measurement

TABLE I. Execution time for the setups with four measurement types.

	$N_c = 5000$ (s)	$N_c = 500\,000$ (s)
PhaseCut	16.9	17.4
$\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ min. from $\widehat{\boldsymbol{v}}_{pc}$	11.4	8.3
$\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ min., random init.	22	24.7
$\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ min. from $\widehat{\boldsymbol{v}}_{pc}$	8.1	5.2
$\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ min., random init.	16.6	21.3
Mixed algo. from $\widehat{\boldsymbol{v}}_{pc}$	6.8	4.7
Mixed algo., random init.	10.6	12.8

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

	$N_c = 5000$ (s)	$N_c = 500\,000$ (s)
Recursive algorithm	0.17	0.17
$\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ min. from $\widehat{\boldsymbol{v}}_{\text{rec}}$	44.4	10.9
$\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ min., random init.	272	94.4
$\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ min. from $\widehat{\boldsymbol{v}}_{\text{rec}}$	38.8	16.7
$\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ min., random init.	84.9	126.7
Mixed algo. from $\widehat{\boldsymbol{v}}_{\text{rec}}$	47.8	26.1
Mixed algo., random init.	62	38.4

type of setup it is relatively fast (as it spares us the initialization step with PhaseCut) but inaccurate, and for the setup with 15 measurement types it is always slower (sometimes much slower) than the likelihood maximization with proper initialization. In conclusion, we recommend using the mixed algorithm for the likelihood, since it is a good compromise between the $\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ minimization and the $\mathscr{L}^{\text{exact}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ minimization. The choice between the setup with four 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 Tables I and II) but the following are true. (1) It yields a slightly less precise result. The median error with the mixed algorithm and $N_c = 5000$ is 0.22 versus 0.19 with 15 measurement types. (2) 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 Tables I and II). (3) We explained (in Sec. 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.

E. Mixed states

In the current section, we test our algorithms on mixed states. All the algorithms are designed for pure states and return pure states as estimates. We will not change that but we will use mixed states that are close to pure to generate the measurements. The generated states are of rank 5 (arbitrary); the Hermitian matrix that represents the mixed state is

$$\boldsymbol{\rho} = p_0 \boldsymbol{v}_0 \boldsymbol{v}_0^* + \sum_{k=1}^{4} p_k \boldsymbol{v}_k \boldsymbol{v}_k^* \tag{24}$$

where p_0 is the highest eigenvalue of $\boldsymbol{\rho}$. The higher p_0 is, the closer $\boldsymbol{\rho}$ is from being pure. $\boldsymbol{v}_0$ is the associated state, and $\boldsymbol{v}_1$,..., $\boldsymbol{v}_4$ are chosen so that $\boldsymbol{v}_0$,..., $\boldsymbol{v}_4$ are all orthogonal to one another. If we wanted to approximate $\boldsymbol{\rho}$ with a pure state, $\boldsymbol{v}_0$ would be the best (highest fidelity) approximation. We will judge the performances of our estimators by how close they are to $\boldsymbol{v}_0$. The error is $\mu = ||\boldsymbol{v}_0 - \widehat{\boldsymbol{v}}.e^{-i\xi}||_2$ with ξ that minimizes the metric, like with (23). With this definition, the link between μ and the fidelity f established in Sec. VI A no longer holds ($f \neq 1 - \frac{\mu^2}{2}$). We perform simulations with ten different values of p_0: $\{1 - 0.325 \times (\frac{2}{3})^k\}_{k\in\{0,\dots,9\}}$ (see these values in Fig. 5). Those values are chosen in order to see to what extent the error varies linearly with respect to $1 - p_0$: Fig. 5 displays the cdf of the error with a logarithmic scale on the x axis, and if the cdf is shifted by a constant interval (in log scale) when $1 - p_0$ is multiplied by $\frac{2}{3}$, this means that the relationship between the error and $1 - p_0$ is linear. For each value of p_0, 100 vectors $\boldsymbol{v}_0, \dots, \boldsymbol{v}_4$ are randomly generated by applying the Gram-Schmidt transformation to five random (the real and imaginary parts of each component are independent centered unit-variance Gaussian) complex d-dimensional vectors. The values of $p_1, \dots, p_4$ are chosen randomly (uniform distribution between 0 and 1) and then normalized so that $\boldsymbol{\rho}$ has a unit trace. We continue to simulate a finite number of measurements, which create an error, but we choose the higher value of N_c, $N_c = 500\,000$. We only use the mixed algorithm (with such a high N_c, all three likelihood algorithms yield similar performances anyway). Figure 5 displays the cdf of the error for the two setups with the ten different values of p_0.

The approach based on our original method introduced in Sec. IV, that uses 15 measurement types, is a lot more resilient to mixed states than the approach based on the PhaseCut algorithm proposed in the literature (see Sec. III), that uses four measurement types: For $p_0 = 0.9506$, for example, the median of the error is 0.042 with 15 measurement types and 0.072 with four measurements. The error seems to vary fairly linearly with respect to $1 - p_0$ except for p_0 close to 1 because for those values, the "regular" error due to the finite number of measurements becomes more important. We also see nonlinearity for the setup with four measurement types: It starts with the green curve ($p_0 = 0.8889$) which has a slightly heavier tail than the other cdf (associated with smaller p_0). It gets more noticeable with the yellow curve ($p_0 = 0.8333$). The orange and red curves (rightmost two curves) have different shapes and slopes as compared to the curves on the left. We do not see any of these nonlinearities with the 15 measurement type of setup, but we would see them with smaller p_0.

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 with pure states 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 and works better with states that are not quite pure. The initial estimates of the considered state are then fined 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 and easy way to maximize an approximation of the likelihood. We intend to use those QST methods to perform QPT like in [24]. In [24] 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 $\sqrt{N}\boldsymbol{\varepsilon} = \sqrt{N}(\widehat{\boldsymbol{p}} - \boldsymbol{p})$ defined in Sec. V C and at simplifying the expression of the likelihood of $\boldsymbol{\varepsilon}$. We consider that $\boldsymbol{p}$ contains the probabilities of a single type of d-outcome measurement. The generalization is straightforward as the errors on different measurements are independent (see Appendix A 3). The only random vector in $\boldsymbol{\varepsilon}$ is $\widehat{\boldsymbol{p}}$ defined as the vector that contains the sample probabilities of each of the d outcomes. So $\widehat{\boldsymbol{p}} = \frac{1}{N}\boldsymbol{n}$ where each component n_i of $\boldsymbol{n}$ contains the number of times the ith outcome occurred. By definition $\boldsymbol{n}$ follows a multinomial distribution characterized by the number of trials N and the theoretical probabilities of each outcome contained in $\boldsymbol{p}$. The expected value and covariance matrix of the multinomial distribution are known: $E(\boldsymbol{n}) = N\boldsymbol{p}$ and $\mathrm{Cov}(\boldsymbol{n}) = N[\mathrm{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^T]$. We want to use the central limit theorem so let us write $\boldsymbol{n}$ as a sum: $\boldsymbol{n} = \sum_{k=1}^{N} \boldsymbol{\delta}_k$ 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 $i_k \in \{1, \dots, N\}$ the density function of which is $j \longrightarrow p_j$ (i.e., the probability that i_k takes the value $j \in \{1, \dots, N\}$ is p_j, the jth element of $\boldsymbol{p}$). $\boldsymbol{\delta}_k$ follows a multinomial distribution with a $N = 1$ trial. Its expected value is therefore $\boldsymbol{p}$ and its covariance matrix is $\mathrm{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^T$. 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 $N \to +\infty$, the distribution of $\sqrt{N}\boldsymbol{\varepsilon}$ tends to a centered multivariate normal distribution, and its covariance matrix is $\boldsymbol{\Sigma}_{\mathbf{full}} = \mathrm{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^T$. $\widehat{\boldsymbol{\Sigma}_{\mathbf{full}}}$ is an estimate of $\boldsymbol{\Sigma}_{\mathbf{full}}$; it uses $\widehat{\boldsymbol{p}}$ as we do not want to depend on the unknown vector $\boldsymbol{p}$: $\widehat{\boldsymbol{\Sigma}_{\mathbf{full}}} = \mathrm{diag}(\widehat{\boldsymbol{p}}) - \widehat{\boldsymbol{p}}\widehat{\boldsymbol{p}}^T$.

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 [25]. 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 $\sqrt{N}\boldsymbol{\varepsilon}$; 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: $\boldsymbol{\Sigma} = \mathrm{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^T$ ($\boldsymbol{p}$ is $\boldsymbol{p}$ with the last element removed). It can be estimated with the sample probabilities $\widehat{\boldsymbol{p}} = \begin{pmatrix} \widehat{p_1} \\ \vdots \\ \widehat{p_{d-1}} \end{pmatrix}$ instead of $\boldsymbol{p}$. The resulting matrix is $\widehat{\boldsymbol{\Sigma}} = \mathrm{diag}(\widehat{\boldsymbol{p}}) - \widehat{\boldsymbol{p}}\widehat{\boldsymbol{p}}^T$. Straightforward calculations show that if no element of $\widehat{\boldsymbol{p}} = \begin{pmatrix} \widehat{p_1} \\ \vdots \\ \widehat{p_d} \end{pmatrix}$ (with $\widehat{p_d} = 1 - \sum_{k=1}^{d-1} \widehat{p_k}$) is zero, then $\widehat{\boldsymbol{\Sigma}}$ is invertible and

$$\widehat{\boldsymbol{\Sigma}}^{-1} = \frac{1}{\widehat{p_d}}\mathbb{1} + \mathrm{diag}(1/\widehat{\boldsymbol{p}}) \tag{A1}$$

is its inverse. We have $1/\widehat{\boldsymbol{p}}$ as the elementwise inverse of $\widehat{\boldsymbol{p}}$ and $\mathbb{1}$ is the $d-1 \times d-1$ matrix with only ones. In practice, elements of $\widehat{\boldsymbol{p}}$ can be zeros, which 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 $\widetilde{\boldsymbol{p}}$:

$$\widetilde{\boldsymbol{p}} = \frac{\widehat{\boldsymbol{p}} + \frac{5}{N}}{1 + \frac{5d}{N}}. \tag{A2}$$

This means that we consider that each outcome has been observed five 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 [26]). The resulting estimate of the inverse of the covariance matrix is

$$\widetilde{\boldsymbol{\Sigma}}^{-1} = \frac{1}{\widetilde{p_d}}\mathbb{1} + \mathrm{diag}(1/\widetilde{\boldsymbol{p}}). \tag{A3}$$

With the inverse of $\widetilde{\boldsymbol{\Sigma}}$ and knowing that the distribution is normal and centered, we can compute the negative log likelihood of the vector (see [25]):

$$\mathscr{L}^{\mathrm{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}}) = N\boldsymbol{\varepsilon}(\widehat{\boldsymbol{p}}, \boldsymbol{x}', \boldsymbol{y}')^T \widetilde{\boldsymbol{\Sigma}}^{-1} \boldsymbol{\varepsilon}(\widehat{\boldsymbol{p}}, \boldsymbol{x}', \boldsymbol{y}'). \tag{A4}$$

We use $\widehat{\boldsymbol{p}}$ and not $\widetilde{\boldsymbol{p}}$ to compute $\boldsymbol{\varepsilon}$, otherwise the estimator that minimizes the criterion would become biased (as the minimum of $\mathscr{L}^{\mathrm{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}$ would fit $\widetilde{\boldsymbol{p}}$ which does not contain the actual sample probabilities) and the criterion would not be smoother. Let us simplify this expression using (A3) and the fact that $\sum_k \varepsilon_k = 0 \Rightarrow \varepsilon_d = -\sum_{k=1}^{d-1} \varepsilon_k$:

$$\begin{aligned} N\boldsymbol{\varepsilon}^T \widetilde{\boldsymbol{\Sigma}}^{-1} \boldsymbol{\varepsilon} &= N\boldsymbol{\varepsilon}^T \begin{pmatrix} \frac{1}{\widetilde{p}_d} \sum_{k=1}^{d-1} \varepsilon_k + \frac{\varepsilon_1}{\widetilde{p}_1} \\ \vdots \\ \frac{1}{\widetilde{p}_d} \sum_{k=1}^{d-1} \varepsilon_k + \frac{\varepsilon_{d-1}}{\widetilde{p}_{d-1}} \end{pmatrix} \\ &= N\boldsymbol{\varepsilon}^T \begin{pmatrix} \frac{\varepsilon_1}{\widetilde{p}_1} - \frac{\varepsilon_d}{\widetilde{p}_d} \\ \vdots \\ \frac{\varepsilon_{d-1}}{\widetilde{p}_{d-1}} - \frac{\varepsilon_d}{\widetilde{p}_d} \end{pmatrix} \\ &= N\Big(\sum_{k=1}^{d-1} \frac{\varepsilon_k^2}{\widetilde{p}_k} - \frac{\varepsilon_d}{\widetilde{p}_d} \sum_{k=1}^{d-1} \varepsilon_k\Big) \\ &= N\sum_{k=1}^{d} \frac{\varepsilon_k^2}{\widetilde{p}_k}. \end{aligned}$$

Therefore, the expression of the negative log likelihood is

$$\mathscr{L}^{\mathrm{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}}) = N\sum_{k=1}^{d} \frac{\varepsilon_k(\widehat{\boldsymbol{p}}, \boldsymbol{x}', \boldsymbol{y}')^2}{\widetilde{p}_k}. \tag{A5}$$

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 four (in Sec. III) or $2n_{qb} + 1$ (in Sec. IV) types of measurements. The errors between the empirical and theoretical probabilities of different measurements are independent. Therefore if $\boldsymbol{\varepsilon}(\widehat{\boldsymbol{p}}, \boldsymbol{x}', \boldsymbol{y}')$ 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:

$$\widetilde{\mathbf{\Sigma}}^{-1} = \begin{bmatrix} \widetilde{\mathbf{\Sigma}}_1^{-1} & & \\ & \ddots & \\ & & \widetilde{\mathbf{\Sigma}}_{n_t}^{-1} \end{bmatrix}. \tag{A6}$$

Each $\widetilde{\mathbf{\Sigma}}_k^{-1}$ is the regularized inverse of the covariance matrix for one measurement type defined in (A3). The negative log likelihood of $\boldsymbol{\varepsilon}(\widehat{\boldsymbol{p}}, \boldsymbol{x}', \boldsymbol{y}')$ containing $n_{\text{prob}} = n_t d$ 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:

$$\mathscr{L}^{\text{Gauss}}_{(\boldsymbol{x}',\boldsymbol{y}')}(\widehat{\boldsymbol{p}}) = N \sum_{k=1}^{n_{\text{prob}}} \frac{\varepsilon_k(\widehat{\boldsymbol{p}}, \boldsymbol{x}', \boldsymbol{y}')^2}{\widetilde{p}_k}. \tag{A7}$$

APPENDIX B: PHASECUT ALGORITHM

Input: Matrix $\mathbf{M}$ of Sec. III C

```
U^0 = I_d, N_run = 5000
for j = 1, ..., N_run
    U^k = U^{k-1}
    pick k ∈ {1, ..., d} (random uniform).
    k_c = [1, ..., k − 1, k + 1, d]^T
    compute u = U^j_{k_c,k_c} m_{k_c,k} and γ = u* m_{k_c,k}
    where m_{k_c,k} is the kth column of M with the
    kth element removed. U^j_{k_c,k_c} is U^j with the
    kth row and column removed.
    if γ > 0 set u^{k+1}_{k_c,k} = u^{k+1*}_{k,k_c} = −sqrt(1/γ) u
    else u^{k+1}_{k_c,k} = u^{k+1*}_{k,k_c} = 0
    where u^{k+1}_{k_c,k} and u^{k+1}_{k,k_c} refer to parts of U^k
    (kth column without the kth element and kth
    row without the kth element respectively)
end for
```

Output: The matrix $\mathbf{U}^{N_{\text{run}}}$ which is Hermitian positive and contains only ones on the diagonal

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University, New York, 2000).

[2] A. Kalev, R. L. Kosut, and I. H. Deutsch, npj Quantum Inf. **1**, 15018 (2015).

[3] A. Smith, C. A. Riofrío, B. E. Anderson, H. Sosa-Martinez, I. H. Deutsch, and P. S. Jessen, Phys. Rev. A **87**, 030102 (2013).

[4] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert, Phys. Rev. Lett. **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, Phys. Rev. A **93**, 032140 (2016).

[6] T. Cai, D. Kim, Y. Wang, M. Yuan, and H. H. Zhou, Ann. Statist. **44**, 682 (2016).

[7] Y. Wang, Ann. Statist. **41**, 2462 (2013).

[8] A quantum measurement on a state in a d-dimensional Hilbert space has at most d possible outcomes; a two-outcome measurement only has 2 (0 and 1 for example). Averaging a two-outcome measurement means computing the empirical probabilities of the two outcomes on a given number of trials.

[9] C. H. Baldwin, I. H. Deutsch, and A. Kalev, Phys. Rev. A **93**, 052105 (2016).

[10] C. Ferrie, Phys. Rev. Lett. **113**, 190404 (2014).

[11] R. J. Chapman, C. Ferrie, and A. Peruzzo, Phys. Rev. Lett. **117**, 040402 (2016).

[12] S. T. Ahmad, A. Farooq, and H. Shin, Sci. Rep. **12**, 5092 (2022).

[13] J. Cotler and F. Wilczek, Phys. Rev. Lett. **124**, 100401 (2020).

[14] J. Finkelstein, Phys. Rev. A **70**, 052107 (2004).

[15] D. Goyeneche, G. Cañas, S. Etcheverry, E. S. Gómez, G. B. Xavier, G. Lima, and A. Delgado, Phys. Rev. Lett. **115**, 090401 (2015).

[16] T. Heinosaari, L. Mazzarella, and M. M. Wolf, Commun. Math. Phys. **318**, 355 (2013).

[17] R. Balan, P. Casazza, and D. Edidin, Applied and Computational Harmonic Analysis **20**, 345 (2006).

[18] I. Waldspurger, A. d'Aspremont, and S. Mallat, Math. Program. **149**, 47 (2015).

[19] A. S. Bandeira, J. Cahill, D. G. Mixon, and A. A. Nelson, Applied and Computational Harmonic Analysis **37**, 106 (2014).

[20] N. Z. Shor, Sov. J. Comput. Syst. Sci. **25**, 1 (1987).

[21] C. G. Broyden, IMA J Appl Math **6**, 76 (1970).

[22] https://www.mathworks.com/products/matlab.html.

[23] Z. Hradil, J. Řeháček, J. Fiurášek, and M. Ježek, in *Quantum State Estimation* (Springer-Verlag, Berlin, 2004), pp. 59–112.

[24] F. Verdeil, Y. Deville, and A. Deville, in *Proceedings of the IEEE Statistical Signal Processing Workshop, Rio de Janeiro, Brazil* (IEEE, Piscataway, NJ, 2021), pp. 161–165.

[25] A. Gut, *An Intermediate Course in Probability* (Springer, New York, 2009).

[26] R. Blume-Kohout, Phys. Rev. Lett. **105**, 200504 (2010).