Optimal Universal Programming of Unitary Gates

Yuxiang Yang [1,*] Renato Renner [1,†] and Giulio Chiribella [2,3,4,5,‡]

[1]*Institute for Theoretical Physics, Wolfgang-Pauli-Strasse 27, 8093, ETH Zürich*
[2]*QICI Quantum Information and Computation Initiative, Department of Computer Science, The University of Hong Kong,*
Pokfulam Road, Hong Kong 999077, China
[3]*Department of Computer Science, University of Oxford, Parks Road, Oxford, OX1 3QD, United Kingdom*
[4]*Perimeter Institute for Theoretical Physics, Waterloo, Ontario N2L 2Y5, Canada*
[5]*The University of Hong Kong Shenzhen Institute of Research and Innovation,*
5/F, Key Laboratory Platform Building, No.6, Yuexing 2nd Road, Nanshan, Shenzhen 518057, China

A universal quantum processor is a device that takes as input a (quantum) program, containing an encoding of an arbitrary unitary gate, and a (quantum) data register, on which the encoded gate is applied. While no perfect universal quantum processor can exist, approximate processors have been proposed in the past two decades. A fundamental open question is how the size of the smallest quantum program scales with the approximation error. Here we answer the question, by proving a bound on the size of the program and designing a concrete protocol that attains the bound in the asymptotic limit. Our result is based on a connection between optimal programming and the Heisenberg limit of quantum metrology, and establishes an asymptotic equivalence between the tasks of programming, learning, and estimating unitary gates.

Introduction.—A universal quantum processor is the desideratum of quantum computing. Ideally, one would hope to realize quantum computing in the same way as its classical counterpart, i.e., by inserting data and programs, both in the form of quantum states, into a universal quantum computer. However, the no-programming theorem [1] asserts that any universal quantum processor must be approximate, or have a nonzero probability of failure [1–3].

It has been shown that approximate universal processors with a finite-size program register do exist [1,4–9]. There one of the most important questions is to determine the cost-accuracy trade-off or, more specifically, how the program cost, i.e., the number c_P of qubits required to store the optimal program, scales with the desired accuracy of implementation, quantified by an approximation error ϵ.

Over the past two decades, many efforts have been dedicated to finding the optimal approximate universal processor [4,5,8,9] (see also Table I). The state-of-the-art result, [9], asserts that the optimal program cost c_P for a d-dimensional unitary quantum gate lies between $c_{\text{low}} := [(1 - \epsilon)K]d - (2/3)\log d$ qubits and $c_{\text{upp}} := d^2 \log (K/\epsilon)$ qubits, where K is a universal constant. Despite all efforts, the precise value for c_P remained largely unknown—especially in the small error regime, where the ratio $c_{\text{upp}}/c_{\text{low}}$ diverges.

In this Letter, we close this gap by identifying the optimal scaling of the program cost with the accuracy and therefore solving a long-standing open problem of optimal quantum programming. Specifically, our program cost scales as $[(d^2 - 1)/2] \log (1/\epsilon)$ in the small ϵ regime, which reduces the cost of the best existing protocol (see c_{upp} above) by half. The optimal scaling is achieved with a gate learning protocol, where the program is prepared by sending a quantum state through n instances of the gate to learn it [14]. The gate information is later read out by measuring the program. Our protocol achieves a diamond norm error scaling of $1/n^2$—well known as the Heisenberg limit of quantum metrology [15–18]. We thus prove the asymptotic equivalence of quantum gate programming, metrology, and learning.

Preliminaries.—We consider programming unitary gates of a system with a d-dimensional Hilbert space $\mathcal{H}$. The gates, up to an irrelevant global phase, form the special unitary group $\mathsf{SU}(\mathsf{d})$. For a pure state $|\psi\rangle$, we abbreviate its density matrix $|\psi\rangle\langle\psi|$ by ψ. Similarly, $\mathcal{U}(\cdot) := U(\cdot)U^\dagger$ denotes a unitary channel.

We will use the big-Ω notation, the big-O notation, and the big-Θ notation to characterize the asymptotic behavior of functions. For two non-negative functions $f(n)$ and $g(n)$, we write $f(n) = \Omega[g(n)]$ if there exists a constant $c_1 > 0$ so that $f(n) \geq c_1 g(n)$ for large enough n, $f(n) = O[g(n)]$ if there exists a constant $c_2 > 0$ so that $f(n) \leq c_2 g(n)$ for large enough n, and $f(n) = \Theta[g(n)]$ if $f(n) = \Omega[g(n)]$ and $f(n) = O[g(n)]$. We will also abbreviate $\log_2$ by $\log$.

Approximate universal processors.—A universal quantum processor consists of two key elements: a family of programs $\{\psi_{P,U}\}_{U\in\mathsf{SU}(\mathsf{d})}$, which are quantum states in $\mathcal{H}_P$, and the action of the processor $\mathcal{C}$, which is a quantum channel (i.e., a completely positive trace-preserving linear

TABLE I. Comparison of bounds on universal quantum gate programming. In the table, we compare our results on the programming cost with the best previous results (summarized from Table I of Ref. [9]). In the vanishing error regime $\epsilon \to 0$, both the lower bound and the upper bound are tighter than all previous results, for the first time closing the gap between the lower and upper bounds in this regime. The cost is defined as the number of qubits in the program and the error is evaluated in terms of the diamond norm (2). K denotes a universal constant.

	Upper bounds	Lower bounds
Previous works	$d^2 \log (K/\epsilon)$ [9] $4d^2 \log d/\epsilon^2$ [8,11,12]	$[(1 - \epsilon)K]d - (2/3) \log d$ [9] $\log(d^2/\varepsilon)$ [10] $[(d + 1)/2] \log (1/d) + [(d - 1)/2] \log (1/\epsilon)$ [13]
This work	$[(d^2 - 1)/2] \log [\Theta(d^3)/\epsilon]$	$\alpha \log [\Theta(d^{-4})/\epsilon]$ for any $\alpha < (d^2 - 1)/2$ and sufficiently small ϵ

map) acting on the composite Hilbert space $\mathcal{H}_S \otimes \mathcal{H}_P$ of the system and the program. Notice that all information on U should come from the program, and $\mathcal{C}$ must be independent of U. The program cost c_P is defined as $\log_2 d_P$, with the program dimension d_P being the dimension of $\mathsf{Supp}\{\psi_{P,U}\}_{U \in \mathsf{SU(d)}}$.

As shown in Fig. 1, to run any arbitrary unitary U on the system, one selects the corresponding program $\psi_{P,U}$ and plugs it into the processor, resulting in the following channel on the system:

$$\mathcal{E}_U(\cdot) := \mathrm{Tr}_P[\mathcal{C}(\cdot \otimes \psi_{P,U})]. \tag{1}$$

A pair $(\mathcal{C}, \{\psi_{P,U}\}_{U \in \mathsf{SU(d)}})$ is called an ϵ-universal processor, if

$$\frac{1}{2}\|\mathcal{U} - \mathcal{E}_U\|_\diamond \leq \epsilon \quad \forall \ U \in \mathsf{SU(d)}. \tag{2}$$

Here $\| \cdot \|_\diamond$ denotes the *diamond norm* [19], which equals the maximum trace distance between the outputs of the two channels, maximized over all input states and over all possible reference systems.

The no-programming theorem [1] rules out perfect (i.e., $\epsilon = 0$) universal processors with finite cost $c_P < \infty$. This
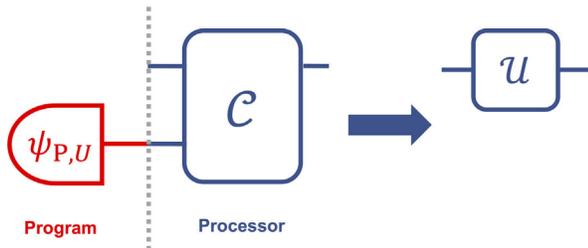
FIG. 1. An approximate universal quantum processor. An approximate universal quantum processor executes a unitary gate U on a system. It works by plugging a quantum state—the program for U—into the processor, which performs a quantum channel $\mathcal{C}$ that approximates U on the system.

impossibility result raised the question: "Given a desired accuracy $1/\epsilon$, how big does the program need to be?" This question can of course be subdivided into two, namely, to find upper and lower bounds on the program cost c_P. We summarize the best known results in Table I. Here we are providing both a new lower and a new upper bound, which match in terms of their asymptotic dependence on $1/\epsilon$.

Lower bound on the program cost.—We first establish a lower bound on the program cost. For this purpose, we exploit an alternative proof of the no-programming theorem, originally developed in the framework of general probabilistic theories [20]. The idea is that the exact implementation of a unitary gate requires the channel $\mathcal{C}$ to leave the system and the program uncorrelated. Using this fact, the program can be recycled, thereby generating multiple copies of the desired unitary gate. The approximate version of this argument was first used by us to determine the energy requirement of quantum processors [21] and is further exploited here.

To approximate a unitary quantum gate U with good precision, there should be almost no correlation between the system and the program after we apply $\mathcal{C}$. This means that the complementary channel of $\mathcal{E}_U$, defined as $\bar{\mathcal{E}}_{\rho_S}(\cdot) := \mathrm{Tr}_S[\mathcal{C}(\rho_S \otimes (\cdot))]$, is almost independent of ρ_S. It further suggests that, instead of discarding the program after one usage, we can *recycle* it: we can invert the action of $\bar{\mathcal{E}}_{\rho_S}$ on the program state by a (ρ_S)-independent operation and get back the original program. The program can be further used, generating multiple uses of U at the cost of an increased approximation error. Notice that the argument does not hold for noisy or classical processes. For instance, using a controlled unitary $|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes \sigma_z$ and an ancillary qubit $(1/\sqrt{2})(|0\rangle + |1\rangle)$, one can (perfectly) implement the channel $\rho \to (1/2)(\rho + \sigma_z\rho\sigma_z)$. However, the system and the ancillary qubit become strongly correlated after the implementation.

By the above argument, we can show (see [22] for details) that an ϵ-universal processor for a single use of U can be turned into a $(4m\sqrt{2\epsilon})$-universal processor for m

uses of U for any $m \geq 1$. This requires the original program to contain enough information for programming up to $1/\sqrt{\epsilon}$ uses of U. This fact, in turn, implies a bound on its minimum information content and therefore its size. This ultimately leads to the following theorem, which can be regarded as a quantitative version of the no-programming theorem [1]:

Theorem 1.—(Approximate no-programming theorem).— Consider any ϵ-universal processor with program cost c_P. For any (ϵ-independent) parameter $\delta > 0$, the program cost is lower bounded as

$$c_P \geq (1 - \delta - 4\sqrt{2\epsilon})(d^2 - 1)\log\left(\frac{\delta}{4\sqrt{2\epsilon}(d^2 - 1)}\right) - 1. \tag{3}$$

This immediately implies the expression for the lower bound stated in Table I. The key message from the above theorem is that, for any $\alpha < (d^2 - 1)/2$, the program dimension $d_P = 2^{c_P}$ satisfies

$$d_P = \Omega(1/\epsilon^\alpha). \tag{4}$$

Taking $\epsilon \to 0$ in Eq. (4), one gets $d_P \to \infty$, recovering the original no-programming theorem [1].

Optimal approximate universal processor.—Next we construct an approximate universal processor that achieves the bound in Theorem 1. Our processor works in a measure-and-operate (MO) fashion, as illustrated in Fig. 2. It measures the input program $\psi_{P,U}$ with a suitable positive operator-valued measurement (POVM) $\{d\hat{U}M_{\hat{U}}\}_{\hat{U}\in\mathsf{SU(d)}}$, where $d\hat{U}$ is the Haar measure. The measurement yields an estimate $\hat{U}$ of the gate U, and the processor performs the corresponding gate on the system. Explicitly, our optimal processor obeys the following procedure:

Protocol 1.   An MO-universal processor.

1: (Generating the program.) Apply $U^{\otimes n}$ to a suitable quantum state $|\psi_P\rangle$.
2: Measure $|\psi_{P,U}\rangle := U^{\otimes n}|\psi_P\rangle$ with $\{d\hat{U}M_{\hat{U}}\}_{\hat{U}\in\mathsf{SU(d)}}$.
3: Apply $\hat{U}$ to the state of the system, where $\hat{U}$ is the measurement outcome.

The program in Protocol 1 is prepared by applying n parallel uses of U on a quantum state (called the *probe state*). The performance of this processor is then determined jointly by the choice of the probe state and the choice of the POVM $\{d\hat{U}M_{\hat{U}}\}_{\hat{U}\in\mathsf{SU(d)}}$. It is known from quantum metrology [16,17,37] that the performance of the measurement is optimized using nonproduct probe states and POVMs. In [22], we identify a probe state and a POVM which, when incorporated into Protocol 1, yields an optimal processor asymptotically achieving the $[(d^2 - 1)/2]\log(1/\epsilon)$ scaling bound of Theorem 1.
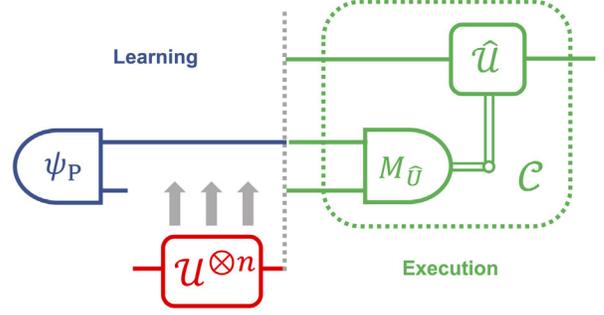
FIG. 2.   A learning protocol for unitary gates. In the learning phase, a probe state ψ_P, possibly entangled with a reference system, is prepared. It is then sent through n parallel instances of U, resulting in a program $\psi_{P,U}$. The program is later measured, and the gate corresponding to the measurement outcome $\hat{U}$ is performed on the system.

Theorem 2.—Consider the estimation of an unknown unitary gate on a d-dimensional quantum system. When $n \geq 2d(d-1)$ uses of the gate are available, the diamond norm error for the optimal estimation is bounded as

$$\epsilon \leq 2d\left(\frac{\pi(d-1)^2(3d-2)}{d \cdot n}\right)^2. \tag{5}$$

The probe state has dimension bounded as

$$d_P \leq \left(\frac{9n}{3d-2}\right)^{d^2-1}. \tag{6}$$

Reference [37] showed that the estimation of an arbitrary d-dimensional unitary given n uses can be done with an error scaling $1/n^2$. The error was measured by the entanglement gate infidelity, which is upper bounded by $1 - (1 - \epsilon)^2$. Theorem 2 refines this result by not only achieving the $1/n^2$ scaling but also identifying an explicit expression of the constant of proportionality. In addition, our result holds for the more stringent error criterion ϵ, i.e., the diamond norm error, and we also determine how the probe state dimension scales with n.

Combining Eq. (5) with Eq. (6), we get

Corollary 3.—The program cost c_P of Protocol 1 is upper bounded as

$$c_P \leq \left(\frac{d^2 - 1}{2}\right)\log\left(\frac{162\pi^2(d-1)^4}{d \cdot \epsilon}\right). \tag{7}$$

It is obvious from the above corollary that

$$c_P \leq \left(\frac{d^2 - 1}{2}\right)\log\left(\frac{162\pi^2 d^3}{\epsilon}\right), \tag{8}$$

which matches Table I and achieves a quadratic reduction compared to known results.

Asymptotic equivalence of programming, metrology, and learning.—From the previous discussion, we can see that an optimal way of programming a unitary is actually to let the processor learn and memorize it (see Fig. 2). The task of learning a unitary U from n instances [14,38,39] consists of a learning phase and an execution (or testing) phase. In the learning phase, the protocol makes n (not necessarily parallel) queries to U. In the execution phase, the protocol emulates the learned unitary on an arbitrary input state. Notice that the execution phase happens after the learning phase; thus, the protocol should be able to store the information of U.

A learning protocol induces a programmable processor in the sense that the learning phase can be used to generate a program. Nevertheless, one should keep in mind that learning and programming are not equivalent. Indeed, in the task of programming, the program does not have to be generated by learning, i.e., by applying multiple instances of U on a quantum state. As learning has this additional constraint, its resource requirement is at least as stringent as that of programming. Therefore, because Protocol 1 is an optimal processor, it is also an optimal learning protocol. The performance of optimal learning given n instances is thus given by Theorem 1, achieved by unitary gate metrology. In summary, for finite dimensional quantum gates, the performances of programming, metrology, and learning are asymptotically equal,

$$\text{programming} \approx \text{metrology} \approx \text{learning}.$$

Quantum versus classical advantage.—One may wonder if it is possible to simply use a classical program, e.g., to write down the description of the gate on a tape. Here we show, via a simple example, that our Protocol 1, which uses a quantum program, beats the best processor that uses classical programs in scaling.

Let us consider the case of programming a phase gate $U_\theta = |0\rangle\langle 0| + e^{-i\theta}|1\rangle\langle 1|$, where $\theta \in [0, 2\pi)$ is the (unknown) phase, for it allows for explicit calculations. Fixing the program dimension $d_P := 2^{c_P}$, the best classical strategy is nothing but dividing the range $[0, 2\pi)$ into d_P equal-width intervals. The tag of the interval that contains θ is used as the program, and the processor runs $U_{\hat{\theta}}$ with $\hat{\theta}$ being the middle point of the interval. Because $\max |\hat{\theta} - \theta| = \pi/d_P$, the error of this approach is $\epsilon_{\text{classical}} = \sqrt{[1 - \cos(\pi/d_P)]/2} \simeq \pi/(2d_P)$, which is inversely proportional to the program dimension.

In contrast, we can employ our Protocol 1, where we use the sine state [15]

$$|\psi\rangle = \sqrt{\frac{2}{d_P}} \sum_{m=0}^{d_P - 1} \sin \frac{\pi(m + 1/2)}{d_P} |m\rangle \qquad (9)$$

as the probe state and the covariant POVM $\{(d\hat{\theta}/2\pi)|\eta_{\hat{\theta}}\rangle\langle\eta_{\hat{\theta}}| : |\eta_{\hat{\theta}}\rangle := \sum_{m=0}^{d_P - 1} e^{-im\hat{\theta}}|m\rangle\}_{\hat{\theta}}$ as the measurement. The error can be evaluated as

$$\epsilon_{\text{quantum}} \simeq \frac{\pi^2}{2d_P^2}, \qquad (10)$$

which is inversely proportional to the square of the program dimension. In other words, the program dimension of a processor with classical programs is quadratically larger than that of our quantum processor. In the more complex case of programming a d-dimensional unitary gate, the classical strategy is to construct an ϵ mesh of the unitary gates, which was employed by Ref. [9]. The program cost was given in Table I as $d^2 \log(K/\epsilon)$, higher than twice the cost of our quantum strategy in the small ϵ regime. This proves the claimed quantum-over-classical advantage in programming.

Conclusion and further discussions.—We identified the optimal scaling of the program cost with accuracy in a universal quantum processor. The optimal scaling can be achieved with a measure-and-operate learning protocol. With this finding, we showed the asymptotic equivalence between programming, metrology, and learning.

In this work, we determined the optimal dependence of the program size on the accuracy parameter ε. An interesting extension would be to determine the optimal scaling with the dimension of the target system d. Moreover, the task we focused on is universal programming, which requires the processor to work well for every gate of a certain dimension. It is natural to expect that a smaller set of gates would lead to a smaller program cost. Observe from Eq. (8) that the prefactor $(d^2 - 1)/2$ is exactly one half the number of real parameters determining a qudit unitary gate (up to a global phase). We therefore conjecture a general formula, valid for parametric families of quantum gates with a continuous dependence on ν real parameters:

$$c_P \sim \left(\frac{\nu}{2}\right) \log\left(\frac{C_{\nu,d}}{\epsilon}\right), \qquad (11)$$

where $C_{\nu,d}$ is a parameter, possibly dependent on ν and d but independent of ϵ.

Another key reason for making this conjecture is that the ultimate performances of quantum information processing tasks share similar forms in the asymptotic limit of "many copies." In particular, one can consider the compression of identically prepared quantum systems, e.g., states of the form $\rho^{\otimes n}$ with ρ unknown and n being large. It turns out that the minimum cost of the memory, when requiring the error to be vanishing for large n, is $(\nu/2) \log n$ (qu)bits in the leading order [40–45]. Here ν, the number of variable real parameters, appears again. Further pursuit in this direction could lead to the discovery of a universality rule, which governs the behavior of optimal quantum devices in the limit of macroscopically many copies.

---

[*] yangyu@ethz.ch
[†] renner@ethz.ch
[‡] giulio@hku.hk

[1] M. A. Nielsen and I. L. Chuang, Phys. Rev. Lett. **79**, 321 (1997).

[2] M. Hillery, V. Bužek, and M. Ziman, Phys. Rev. A **65**, 022301 (2002).

[3] M. Sedlák, A. Bisio, and M. Ziman, Phys. Rev. Lett. **122**, 170502 (2019).

[4] J. Kim, Y. Cheong, J.-S. Lee, and S. Lee, Phys. Rev. A **65**, 012302 (2001).

[5] M. Hillery, V. Bužek, and M. Ziman, Fortsch. Phys. **49**, 987 (2001).

[6] G. Vidal, L. Masanes, and J. I. Cirac, Phys. Rev. Lett. **88**, 047905 (2002).

[7] A. Brazier, V. Bužek, and P. L. Knight, Phys. Rev. A **71**, 032306 (2005).

[8] S. Ishizaka and T. Hiroshima, Phys. Rev. Lett. **101**, 240501 (2008).

[9] A. M. Kubicki, C. Palazuelos, and D. Pérez-García, Phys. Rev. Lett. **122**, 080505 (2019).

[10] C. Majenz, Entropy in quantum information theory–communication and cryptography, Ph.D. thesis, Faculty of Science, University of Copenhagen, 2017.

[11] S. Beigi and R. König, New J. Phys. **13**, 093036 (2011).

[12] M. Christandl, F. Leditzky, C. Majenz, G. Smith, F. Speelman, and M. Walter, arXiv:1809.10751.

[13] D. Pérez-García, Phys. Rev. A **73**, 052315 (2006).

[14] A. Bisio, G. Chiribella, G. M. DAriano, S. Facchini, and P. Perinotti, Phys. Rev. A **81**, 032324 (2010).

[15] V. Bužek, R. Derka, and S. Massar, Phys. Rev. Lett. **82**, 2207 (1999).

[16] G. Chiribella, G. M. DAriano, P. Perinotti, and M. F. Sacchi, Phys. Rev. Lett. **93**, 180503 (2004).

[17] E. Bagan, M. Baig, and R. Munoz-Tapia, Phys. Rev. A **70**, 030301(R) (2004).

[18] M. Hayashi, Phys. Lett. A **354**, 183 (2006).

[19] A. Y. Kitaev, Russ. Math. Surv. **52**, 1191 (1997).

[20] G. Chiribella, G. M. DAriano, and P. Perinotti, Phys. Rev. A **81**, 062348 (2010).

[21] G. Chiribella, Y. Yang, and R. Renner, arXiv:1908.10884.

[22] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.125.210501 for proofs of Theorems 1 and 2, which includes Refs. [23–36].

[23] M. A. Nielsen and I. Chuang, *Quantum Information* (Cambridge University Press, Cambridge 2000).

[24] C. A. Fuchs and J. Van De Graaf, IEEE Trans. Inf. Theory **45**, 1216 (1999).

[25] A. S. Holevo, Prob. Peredachi Inf. **9**, 3 (1973).

[26] W. Fulton and J. Harris, *Representation Theory: A First Course* (Springer, New York, 1991), Vol. 129.

[27] I. Schur, Uber eine Klasse von Matrizen, die sich einer gegebenen Matrix zuordnen lassen, Ph.D. thesis (1901).

[28] R. Alicki and M. Fannes, J. Phys. A **37**, L55 (2004).

[29] A. Winter, Commun. Math. Phys. **347**, 291 (2016).

[30] M. Horodecki, P. Horodecki, and R. Horodecki, Phys. Rev. A **60**, 1888 (1999).

[31] M. A. Nielsen, Phys. Lett. A **303**, 249 (2002).

[32] M. Raginsky, Phys. Lett. A **290**, 11 (2001).

[33] G. Chiribella, G. M. Dariano, and M. F. Sacchi, Phys. Rev. A **72**, 042338 (2005).

[34] S.-E. Ekström, C. Garoni, and S. Serra-Capizzano, Exp. Math. **27**, 478 (2018).

[35] J. Watrous, *The Theory of Quantum Information* (Cambridge University Press, Cambridge, United Kingdom, 2018).

[36] C. Itzykson and M. Nauenberg, Rev. Mod. Phys. **38**, 95 (1966).

[37] J. Kahn, Phys. Rev. A **75**, 022326 (2007).

[38] S. Gammelmark and K. Mølmer, New J. Phys. **11**, 033017 (2009).

[39] Y. Mo and G. Chiribella, New J. Phys. **21**, 113003 (2019).

[40] M. Plesch and V. Bužek, Phys. Rev. A **81**, 032317 (2010).

[41] L. A. Rozema, D. H. Mahler, A. Hayat, P. S. Turner, and A. M. Steinberg, Phys. Rev. Lett. **113**, 160504 (2014).

[42] G. Chiribella, Y. Yang, and C. Huang, Phys. Rev. Lett. **114**, 120504 (2015).

[43] Y. Yang, G. Chiribella, and D. Ebler, Phys. Rev. Lett. **116**, 080501 (2016).

[44] Y. Yang, G. Chiribella, and M. Hayashi, Phys. Rev. Lett. **117**, 090502 (2016).

[45] Y. Yang, G. Bai, G. Chiribella, and M. Hayashi, IEEE Trans. Inf. Theory **64**, 4766 (2018).