Efficient Algorithm for Optimizing Adaptive Quantum Metrology Processes

Alexander Hentschel* and Barry C. Sanders

Institute for Quantum Information Science, University of Calgary, Calgary, Alberta, Canada T2N 1N4

(Received 19 April 2011; published 30 November 2011)

Quantum-enhanced metrology infers an unknown quantity with accuracy beyond the standard quantum limit (SQL). Feedback-based metrological techniques are promising for beating the SQL but devising the feedback procedures is difficult and inefficient. Here we introduce an efficient self-learning swarm-intelligence algorithm for devising feedback-based quantum metrological procedures. Our algorithm can be trained with simulated or real-world trials and accommodates experimental imperfections, losses, and decoherence.

DOI: 10.1103/PhysRevLett.107.233601 PACS numbers: 42.50.Dv, 03.65.Wj, 03.67.−a, 07.05.Mh

Precise metrology underpins modern science and engineering. However, the "standard quantum limit" (SQL) restricts achievable precision, beyond which measurement must be treated on a quantum level. Quantum-enhanced metrology (QEM) aims to beat the SQL by exploiting entangled or squeezed input states and a sophisticated detection strategy [1–3]. Feedback-based QEM is most effective as accumulated measurement data are exploited to maximize information gain in subsequent measurements, but finding an optimal QEM policy for a given measurement device is computationally intractable even for pure input states, unitary evolution U, and projective measurements. Typically, policies have been devised by clever guessing [4,5] or brute-force numerical optimization [5]. Recently, we introduced swarm-intelligence reinforcement learning to devise optimal policies for measuring an interferometric phase shift [6]. Our algorithm is space efficient; i.e., the memory requirement is a polynomial function of the number of times N that U is effected, in contrast to the exponentially expensive brute-force algorithm. Although our result demonstrated the power of reinforcement learning, our algorithm requires a run-time that is exponential in N and a perfect interferometer, thereby effectively restricting its applicability to proofs of principle. Here we report a space- *and* time-efficient algorithm (based on new heuristics) for devising QEM policies. Our algorithm works for noisy evolution and loss, thus making reinforcement learning viable for autonomous design of feedback-based QEM in a real-world setting. The devised QEM policies belong to the class of self-protected algorithms [7], as they do not employ quantum error correction to achieve robustness against quantum noise and errors.

We restrict our focus to single-parameter QEM. Interferometric phase estimation is the canonical quantum metrology problem and is applicable to measurements of time, displacements, and imaging. Therefore, we develop and benchmark our algorithm for autonomous policy design in this context. To beat the SQL, we employ an entangled sequence of N input photons, feedback control, and direct measurements of the interferometer output. For adaptive phase estimation, the interferometer processes one photon at a time. Each input photon can be in two modes, labeled $\{|0\rangle, |1\rangle\}$, corresponding to the interferometer's two paths. Thus, a time-ordered sequence of N photons implements an N-qubit state.

We assume that the interferometric transformation (Fig. 1) can be expressed as a tensor product of quantum channels (i.e., completely positive trace-nonincreasing maps [8]) $\mathcal{C}(\varphi;\Phi_m)$ for φ the unknown phase shift being estimated and Φ_m a controllable phase with $m = 0, 1, \ldots, N-1$. The channel $\mathcal{C}$ is a noisy version of the restrictive single-qubit unitary process U normally considered in QEM. Our tensor-product description corresponds to the assumption that the interferometric process, other than the control, is unchanging during the measurement procedure. Photons of the N-qubit input state $|\Psi_N\rangle$ enter the interferometer one-by-one and are transformed by $\mathcal{C}$. Detectors measure where each photon exits, thereby implementing a projective-valued measure with elements $\{|0\rangle\langle 0|, |1\rangle\langle 1|\}$ that yield 1 bit $u \in \{0,1\}$ if the photon is not lost. The processing unit (PU) modifies the interferometric phase shift by Φ_m, according to the measurement history $h_m = u_m u_{m-1} \ldots u_1 \in \{0,1\}^m$ up to the mth photon, prior to the next photon being processed. After all N input qubits have passed through the interferometer, the PU estimates the interferometric phase shift φ as $\tilde{\varphi}$. A policy ϱ is a "behavior pattern" for the PU, i.e., a collection of rules

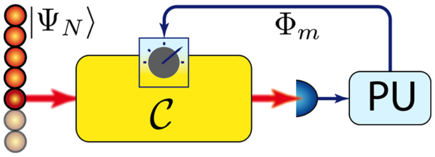

FIG. 1 (color online). Adaptive-feedback scheme for estimating an interferometric phase φ. The input state $|\Psi_N\rangle$ is fed into the unital quantum channel $\mathcal{C}$ one qubit at a time and the output qubit is measured or lost. The processing unit (PU) shifts the interferometric phase by Φ after each successful measurement prior to processing the next qubit.

that tell the PU how to set Φ_m given h_m and which phase estimate to report at the end.

The error probability distribution $P(\varsigma|\varrho)$ of the policy ϱ yields the standard error $\Delta\varphi(\varrho)$ of the estimate $\tilde{\varphi}$ for $\varsigma := \varphi - \tilde{\varphi}$. As ς is cyclic over 2π, $\Delta\varphi(\varrho)$ is given by the Holevo variance $V_H(\varrho) = \Delta\varphi(\varrho)^2 := S(\varrho)^{-2} - 1$, for $S(\varrho) := |\int_{-\pi}^{\pi} P(\varsigma|\varrho) e^{i\varsigma} d\varsigma|$ the sharpness of $P(\varsigma|\varrho)$ [9]. Evaluating $S(\varrho)$ requires exponential computing time with respect to N and thus is computationally intractable. However, from K trial runs of ϱ with randomly chosen phases $\varphi_1, \dots, \varphi_K$, we can infer a sharpness estimate $\tilde{S} := |\sum_{k=1}^{K} \exp(i\varsigma_k)|/K$ for ς_k the error of the kth phase estimate. For QEM, $\Delta\varphi(\varrho)$ should scale better than the SQL $\Delta\varphi \propto 1/\sqrt{N}$ and as close as possible to the ultimate Heisenberg limit $\Delta\varphi \propto 1/N$ [1–3].

For unitary evolution, the interferometer transforms each input qubit by $U_{\boldsymbol{n}}(\theta) = \exp\{-i\theta\hat{\boldsymbol{\sigma}} \cdot \boldsymbol{n}\}$ for $\hat{\boldsymbol{\sigma}} := (\hat{\sigma}_x, \hat{\sigma}_y, \hat{\sigma}_z)$ the Pauli matrices, $\boldsymbol{n}$ a unit vector, and $\varphi - \Phi = 2\theta$ the interferometric phase difference. Without loss of generality, we can restrict our analysis to $\boldsymbol{n} = (0, 1, 0)$. However, because of imperfections, a real-world interferometer is represented by a nonunitary quantum channel $\mathcal{C}$. We assume an unbiased interferometer; i.e., a random input qubit $\mathbb{1} = |0\rangle\langle 0| + |1\rangle\langle 1|$ is mapped to itself ($\mathcal{C}(\mathbb{1}) = \mathbb{1}$), corresponding to a *unital* channel. Hence, for continuous or discrete and countable $\boldsymbol{n}$ and θ [10],

$$\mathcal{C}(\bullet) = \sum\!\!\!\!\!\!\int_{\boldsymbol{n},\theta} w_{\boldsymbol{n}}(\theta) U_{\boldsymbol{n}}(\theta) \bullet U_{\boldsymbol{n}}^{\dagger}(\theta), \qquad w_{\boldsymbol{n}}(\theta) \in \mathbb{R}, \quad (1)$$

with $\sum\!\!\!\!\int_{\boldsymbol{n},\theta} w_{\boldsymbol{n}}(\theta) = 1$ and $w_{\boldsymbol{n}}(\theta) = \delta_{\theta,\varphi-\Phi}\delta_{\boldsymbol{n},(0,1,0)}$ for an ideal interferometer. In contrast, $\sum\!\!\!\!\int_{\boldsymbol{n},\theta} w_{\boldsymbol{n}}(\theta) = 1 - \eta$ corresponds to an input state-independent loss rate η, and quantum noise is incorporated by $w_{\boldsymbol{n}}(\theta)$ being a general distribution with $\langle\theta\rangle = (\varphi - \Phi)/2$ and $\langle\boldsymbol{n}\rangle = (0, 1, 0)$. We simulate noise using normal distributions with the aforementioned means and small standard deviations σ_θ, $\sigma_{\boldsymbol{n}} \ll 1$, corresponding to visibility $1/(2e^{2\sigma_\theta^2} - 1)$. For an optical interferometer, θ noise corresponds to path-length difference fluctuations and $\boldsymbol{n}$ to beam splitter reflectivity fluctuations. We utilize the input state

$$|\Psi_N\rangle = \sum_{n,k=0}^{N} \frac{\sin(\frac{k+1}{N+2}\pi)}{\sqrt{1 + N/2}} e^{(i/2)\pi(k-n)} d^{N/2}_{n-(N/2),k-(N/2)}\left(\frac{\pi}{2}\right)|n\rangle_{[N]}$$

from [4–6], with $d^{j}_{\nu,\mu}(\beta)$ Wigner's d matrix [11]. $|n\rangle_{[N]}$ is a permutationally symmetric state with n qubits in $|1\rangle$ and $N - n$ in $|0\rangle$ [12]. The state $|\Psi_N\rangle$ is appealing because it allows precision close to the Heisenberg limit [4,5] and is robust against loss [6]. The creation of $|\Psi_N\rangle$ is currently an open question, but our algorithm works for any other state as well.

The control flow graph of any deterministic policy for a lossless conditions and a fixed N-qubit input state can be represented as a binary decision tree of depth N with an example shown in Fig. 2(a). Each of the $\sum_{\ell=0}^{N} 2^{\ell} = 2^{N+1} - 1$ nodes of the tree corresponds to one specific state of the experiment and represents the resultant action of the policy. Numeric optimization is computationally intractable due to the exponentially large number of nodes. Therefore, we restrict our search to policies that implement a "generalized logarithmic search" (GLS) heuristic as described below, because the set of all GLS policies can be parametrized by only N parameters and contains phase-estimation policies with optimal precision scaling [6] with respect to N.

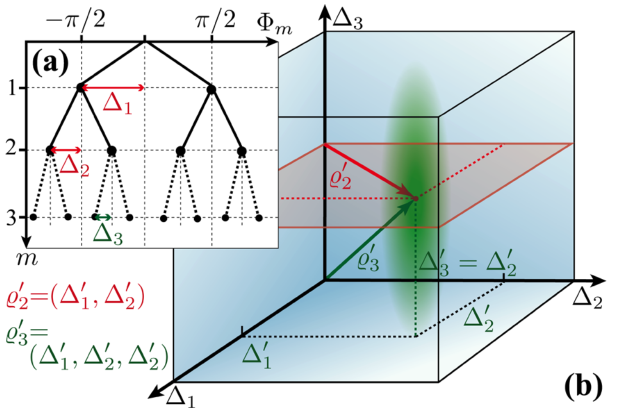

FIG. 2 (color online). (a) Decision tree representation of a GLS policy for $N = 2$ (solid) and $N = 3$ (entire tree). For each path in the tree, the inner nodes represent the applied feedback phases Φ_m and the leaf shows the final phase estimate $\tilde{\varphi}$. At depth m, a measurement $u_{m+1} = 0$ directs the path to the left and $u_{m+1} = 1$ to the right. (b) Embedding the best policy $\varrho_N' \in \mathcal{P}_N$ in the policy space $\mathcal{P}_{N+1}$, shown for $N = 2$. From the best two-qubit policy ϱ_2', the policy $\varrho_3' \in \mathcal{P}_3$ is generated as a guideline. The initial candidate policies for three input qubits are chosen according to probability density (4), indicated by the shaded area around ϱ_3'. [For clarity, the $N = 2$ case is depicted, although only candidate policies for $N > 10$ are chosen according to (4).]

For a uniform prior of $\varphi \in [0, 2\pi)$, the GLS heuristic commences with the initial feedback $\Phi_0 = 0$. After the mth measurement result $u_m \in \{0, 1\}$, the feedback phase is

$$\Phi_m = \Phi_{m-1} - (-1)^{u_m}\Delta_m. \quad (2)$$

If the qubit is lost, Φ remains unchanged. After all N input qubits are processed, there are $M \leq N$ measurement results $u_M, \dots, u_1$, and the GLS heuristic reports the phase estimate $\tilde{\varphi} = \Phi_{M-1} - (-1)^{u_M}\Delta_M$. According to this parametrization, every GLS policy for an N-qubit input state is represented by a vector $\varrho = (\Delta_1, \dots, \Delta_N)$ in the policy space $\mathcal{P}_N = [-\pi, \pi)^N$, and any such vector $\varrho \in \mathcal{P}_N$ is a valid policy. As any policy $\varrho \in \mathcal{P}_N$ utilizes a string of N input qubits, we refer to it as an N-qubit policy. Every $\varrho \in \mathcal{P}_N$ implements a GLS because ϱ has variable entries compared to logarithmic search (LS) for which $\Delta_m = \frac{1}{2}\Delta_{m-1}$ [13]. The N-qubit LS policy $(\pi/2, \pi/4, \dots, \pi/2^N) \in \mathcal{P}_N$ but does not surpass the

SQL. The duality between GLS policies and points in $\mathcal{P}_N \subset \mathbb{R}^N$ allows the use of function optimization techniques to search for an optimal $\boldsymbol{\varrho}_{\rm opt} \in \mathcal{P}_N$ with minimum $\Delta\varphi$, i.e., $\boldsymbol{\varrho}_{\rm opt} \in \operatorname{argmin}_{\boldsymbol{\varrho}\in\mathcal{P}_N} V(\boldsymbol{\varrho}) = \operatorname{argmax}_{\boldsymbol{\varrho}\in\mathcal{P}_N} S(\boldsymbol{\varrho})$. Unfortunately, this optimization problem is nonconvex and hence difficult [6].

Particle swarm optimization (PSO) algorithms [14,15] are outstandingly successful for nonconvex optimization. PSO is a "collective intelligence" strategy from the field of machine learning that learns via trial-and-error and performs as well as or better than simulated annealing and genetic algorithms [16–18]. We have shown that PSO also delivers an autonomous approach to devising adaptive phase-estimation policies for ideal interferometry [6,19].

To search for $\boldsymbol{\varrho}_{\rm opt}$, the PSO algorithm models a "swarm" of Ξ "particles" $\{p^{(1)}, p^{(2)}, \ldots, p^{(\Xi)}\}$ that move in the search space $\mathcal{P}_N$. A particle's position $\boldsymbol{\varrho}^{(i)} \in \mathcal{P}_N$ represents a candidate policy for estimating φ, which is initially chosen at random. Furthermore, $p^{(i)}$ remembers the best position, $\hat{\boldsymbol{\varrho}}^{(i)}$, it has visited so far (including its current position). In addition, $p^{(i)}$ communicates with other particles in its neighborhood $\mathcal{N}^{(i)} \subseteq \{1, 2, \ldots, \Xi\}$. We adopt the common approach to set each $\mathcal{N}^{(i)}$ in a predefined way regardless of the particles' positions by arranging them in a ring topology: for $p^{(i)}$, all particles with maximum distance r on the ring are in $\mathcal{N}^{(i)}$. In iteration t, the PSO algorithm updates the position of all particles in a round-based manner as follows. (i) Each particle $p^{(i)}$ samples $\tilde{S}(\boldsymbol{\varrho}^{(i)})$ of its current position with K trial runs. (ii) $p^{(i)}$ resamples $\tilde{S}(\hat{\boldsymbol{\varrho}}^{(i)})$ of its personal-best policy $\hat{\boldsymbol{\varrho}}^{(i)}$, and the performance of $\hat{\boldsymbol{\varrho}}^{(i)}$ is taken to be the arithmetic mean $\bar{S}(\hat{\boldsymbol{\varrho}}^{(i)})$ of all sharpness evaluations. (iii) Each $p^{(i)}$ updates $\hat{\boldsymbol{\varrho}}^{(i)}$ if $\tilde{S}(\boldsymbol{\varrho}^{(i)}) > \bar{S}(\hat{\boldsymbol{\varrho}}^{(i)})$, and (iv) communicates $\hat{\boldsymbol{\varrho}}^{(i)}$ and $\bar{S}(\hat{\boldsymbol{\varrho}}^{(i)})$ to all members of $\mathcal{N}^{(i)}$. (v) Each particle $p^{(i)}$ determines the sharpest policy $\boldsymbol{\Lambda}^{(i)} = \max_{j\in\mathcal{N}^{(i)}} \hat{\boldsymbol{\varrho}}^{(j)}$ found so far by any one particle in $\mathcal{N}^{(i)}$ (including itself) and (vi) moves to

$$\begin{aligned} \boldsymbol{\varrho}^{(i)} &\leftarrow \boldsymbol{\varrho}^{(i)} + \omega\boldsymbol{\delta}^{(i)}, \\ \boldsymbol{\delta}^{(i)} &\leftarrow \boldsymbol{\delta}^{(i)} + \beta_1\xi_1(\hat{\boldsymbol{\varrho}}^{(i)} - \boldsymbol{\varrho}^{(i)}) + \beta_2\xi_2(\boldsymbol{\Lambda}^{(i)} - \boldsymbol{\varrho}^{(i)}). \end{aligned} \tag{3}$$

The arrows indicate that the right value is assigned to the left variable. The damping factor ω assists convergence, and ξ_1, ξ_2 are uniformly-distributed random numbers from the interval [0,1] that are regenerated each time Eq. (3) is evaluated. The "exploitation weight" β_1 parametrizes the attraction of a particle to its personal-best position $\hat{\boldsymbol{\varrho}}^{(i)}$, and the "exploration weight" β_2 describes attraction to the best position $\boldsymbol{\Lambda}^{(i)}$ in the neighborhood. To improve convergence, we bound each component of $\omega\boldsymbol{\delta}^{(i)}$ by a maximum value of $v_{\max}$. The user-specified parameters ω, β_1, β_2, and $v_{\max}$ determine the swarm's behavior. Tests indicate that $\omega = 0.8$, $\beta_1 = 0.5$, $\beta_2 = 1$, and $v_{\max} = 0.2$ result in the highest probability to find an optimal policy.

The K trial runs for assessing sharpness can be simulated or performed with a real-world experiment. For finite K, the sampled sharpness has statistical errors that can prevent the PSO algorithm from learning optimal solutions [20]. We reduce sharpness errors by averaging over multiple samples in step (ii) [21]. However, for $N > 12$, the PSO algorithm fails to learn good policies from scratch due to sharpness errors [19]. Therefore, we maintain our earlier strategy of running the learning algorithm for each N independently when $N \leq 10$. For $N > 10$, our new heuristic bootstraps a starting point for the optimization of an N-qubit policy from the best $(N-1)$-qubit policy $\boldsymbol{\varrho}' = (\Delta'_1, \ldots, \Delta'_{N-1})$. Our heuristic exploits the fact that an $(N-1)$-qubit policy can be used as an N-qubit policy by ignoring the Nth measurement result. For $N \geq 10$, the optimal $(N-1)$-qubit policy estimates phases with only 10% less accuracy compared to an optimal N-qubit policy when used with the N-qubit input $|\Psi_N\rangle$ (Fig. 1 in the supplemental material [22]). Furthermore, the performance difference between the optimal N-qubit policy and the $(N-1)$-qubit policy decreases with increasing N because the relative change in qubit number decreases with increasing N. Therefore, a good $(N-1)$-qubit policy is a valuable starting point for optimizing an N-qubit policy. Exploiting this discovery is the key for devising the first algorithm that learns optimal policies efficiently; i.e., the required number of computational steps and qubits for learning an optimal policy are polynomial functions of N. In contrast, all other known methods for finding optimal phase-estimation policies are computationally intractable.

When devising N-qubit policies for $N > 10$, our new PSO algorithm applies the knowledge about previously learned policies during its initialization step. The initial policy $\boldsymbol{\varrho} \in \mathcal{P}_N$ is selected as the particle's starting position with probability

$$P(\boldsymbol{\varrho}_N) = \left(\prod_{k=1}^{N-1} \mathcal{N}_{\Delta'_k,\sigma_1}(\Delta_k)\right)\mathcal{N}_{\Delta'_N,\sigma_2}(\Delta_N), \tag{4}$$

$$\mathcal{N}_{\mu,\sigma}(x) := \begin{cases} \kappa^{-1}_{\mu,\sigma}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}, & x \in [0,\pi) \\ 0, & x \notin [0,\pi), \end{cases} \tag{5}$$

$$\kappa^{-1}_{\mu,\sigma} = \sqrt{\pi/2}\sigma\left[\operatorname{erf}\left(\frac{\pi-\mu}{\sqrt{2}\sigma}\right) + \operatorname{erf}\left(\frac{\mu}{\sqrt{2}\sigma}\right)\right], \tag{6}$$

with $\mathcal{N}_{\mu,\sigma}(x)$ a truncated normal distribution. See Fig. 2(b) for an illustration of this strategy. The standard deviation σ_1 determines the similarity of the first N actions of the newly generated policies compared to the template policy $\boldsymbol{\varrho}'$. σ_2 determines the extent to which the action for the new Nth qubit agrees with the previous action of $\boldsymbol{\varrho}'$. We found that $\sigma_1 = 0.01\pi$ and $\sigma_2 = 0.25\pi$ yields a high success rate for our PSO heuristic.

For $4 \leq N \leq 14$ and perfect interferometry, we verified that our new PSO algorithm with swarm size $\Xi = 20N$

learns optimal N-qubit policies regardless of whether each policy's sharpness is evaluated exactly (requires time $\propto 2^N$) or sampled from $K = 10N^2$ trial runs (requires polynomial run-time in N when simulated). Therefore, we sample the sharpness of each particle's current position and personal-best position in each PSO iteration. As we run the PSO algorithm for a constant 300 iterations, the entire optimization process requires $\mathcal{O}(K\Xi)$ trials. However, to obtain an N-qubit policy, we have to optimize policies for $10, 11, \ldots, N-1$ input qubits beforehand, as our algorithm requires an $(N-1)$ qubit policy for devising an N-qubit policy for any $N > 10$. Therefore, learning an N-qubit policy requires $\mathcal{O}(NK\Xi) = \mathcal{O}(N^4)$ trial runs. When the trials are simulated, the computational complexity of our PSO heuristic is $\mathcal{O}(N^6)$ (hence efficient) as a single trial run can be simulated in time $\mathcal{O}(N^2)$ [12]. Once learned, the execution of an N-qubit policy requires N entangled input qubits.

We trained our PSO algorithm with simulated trial runs for various noise and loss rates. In each case, our PSO algorithm tries to find the sharpest policy $\boldsymbol{\varrho}_N$ for given N. As the algorithm uses stochastic optimization, it is not guaranteed to learn the optimal policy every time and must be run several times independently for each N. Nevertheless, within the limits of available computational resources, the PSO algorithm succeeded in at least 25% of the runs, independently of N. We compared the policies generated by our new machine-learning algorithm to our previous numerically optimized policies [6], the Berry-Wiseman (BW) policy [4], and policies obtained by brute-force numerical optimization [5].

We first discuss policies for a noiseless, lossless setup, i.e., for unitary evolution. Figure 3(a) shows that our new method, tested to the limits of available computational resources, outperforms the BW-policy. We estimate the performance difference by calculating the scaling α of the Holevo variance V_H. Our policies yield $V_H \propto N^{-\alpha}$ with $\alpha_{\text{PSO}} = 1.494 \pm 0.003$, compared to the inferred scaling $\alpha_{\text{BW}} = 1.415 \pm 0.003$ for $N \leq 50$. Furthermore, our new *efficient* method greatly surpasses our previous optimization scheme [6] by more than tripling the domain of N for developing policies while maintaining the same precision. The inefficient brute-force optimization was carried out in the full policy space, i.e., without restriction to GLS policies. However, the resulting globally optimal policies perform better only by a constant factor of 0.88 ± 0.01 compared to our PSO-optimized policies but do not yield better scaling α. As expected the PSO algorithm yields policies approaching the SQL $V_H \propto 1/N$ for separable input states [Fig. 3(b)] [1–3].

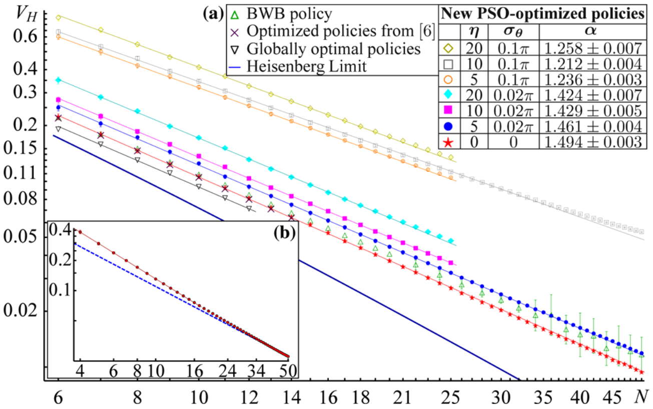

FIG. 3 (color online). Holevo phase variance V_H of PSO-optimized policies compared to other schemes vs the number of input qubits N for (a) $|\Psi_N\rangle$ and (b) $|0\cdots 00\rangle$ as input states, respectively. The dashed line shows the SQL. Because of limited computational resources, some simulations are carried out only to $N \leq 25$. (Loss rate η is in percent; $\sigma_{n_x} = \sigma_{n_y} = \sigma_{n_z} = 0.2\sigma_\theta$.)

Our new algorithm delivers the first QEM policies optimized for a simulated imperfect interferometer with loss and Gaussian quantum noisy. When applied to noisy conditions, policies generated by our new algorithm have significantly improved performances compared to policies optimized for perfect interferometry. As expected, the performance difference increases with the noise level (Fig. 3 in the supplemental material [22]). We verify that our algorithm successfully devises superior policies also for non-Gaussian noise by using skew-normal distributions with skewness $\gamma = 0.667$ for P_θ and $P_{\boldsymbol{n}}$ [23]. We find that a nonzero third standardized moment with variances kept as before does not reduce the performance of the policies learned by our new PSO algorithm (Fig. 3 in the supplemental material [22]).

In summary, we have devised an *efficient* machine-learning algorithm to construct adaptive-feedback measurement policies autonomously for time-independent, single-parameter estimation problems. Our one prerequisite is a training-phase comparison criterion to evaluate the success of candidate policies. Within the limits of available computational resources, our PSO-generated policies outperform all known schemes for adaptive single-shot phase estimation with direct measurement of the channel output. Our algorithm learns to account for experimental errors and loss thereby making time-consuming error modeling and extensive calibration dispensable.

We thank B. Bunk and Humboldt-Universität zu Berlin for computational resources, and D. W. Berry, L. Maccone, and H. M. Wiseman for comments on an earlier draft. This project has been supported by *iCORE*, AITF, NSERC, and CIFAR. B. C. S. is supported by a CIFAR Fellowship.

*A.Hentschel@ucalgary.ca

[1] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. Lett. **96**, 010401 (2006).
[2] L. Pezzé and A. Smerzi, Phys. Rev. Lett. **102**, 100401 (2009).

[3] S. Luo, Lett. Math. Phys. **53**, 243 (2000).

[4] D. W. Berry and H. M. Wiseman, Phys. Rev. Lett. **85**, 5098 (2000).

[5] D. W. Berry, H. M. Wiseman, and J. K. Breslin, Phys. Rev. A **63**, 053804 (2001).

[6] A. Hentschel and B. C. Sanders, Phys. Rev. Lett. **104**, 063603 (2010).

[7] L.-A. Wu and M. S. Byrd, Quant. Info. Proc. **8**, 1 (2008).

[8] J. Hou, J. Phys. A **43**, 385201 (2010).

[9] A. S. Holevo, in *Quantum Probability and Applications to the Quantum Theory of Irreversible Processes*, Lecture Notes in Mathematics Vol. 1055 (Springer, Berlin/ Heidelberg, 1984), p. 153.

[10] C. Mendl and M. Wolf, Commun. Math. Phys. **289**, 1057 (2009).

[11] E. P. Wigner, *Group Theory and its Application to the Quantum Mechanics of Atomic Spectra* (Academic Press, New York, 1971).

[12] A. Hentschel and B. C. Sanders, J. Phys. A **44**, 115301 (2011).

[13] R. D. Nowak, in *Proc. 46th Allerton Conference on Communications, Control, and Computing, Illinois, USA, 2008* (IEEE, New York, 2008), p. 568.

[14] R. Eberhart and J. Kennedy, in *Proc. Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995* (IEEE, New York, 1995), p. 39.

[15] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence* (John Wiley & Sons, England, 2006).

[16] S. Ethni, B. Zahawi, D. Giaouris, and P. Acarnley, in *Proc. 7th IEEE Int. Conf. on Industrial Informatics, Cardiff, England, 2009* (IEEE, New York, 2009).

[17] J. Kennedy and W. M. Spears, in *Proc. IEEE Congress on Evolutionary Computation, Anchorage, Alaska* (IEEE, New York, 1998), p. 78.

[18] P. Fourie and A. Groenwold, Structural Multidisciplinary Optimization **23**, 259 (2002).

[19] A. Hentschel and B. C. Sanders, in *Proc. 7th Int. Conf. Information Technology: New Generations, Las Vegas, 2010* (IEEE, New York, 2010), p. 506.

[20] T. Bartz-Beielstein, D. Blum, and J. Branke, in *Metaheuristics*, Operations Research/Computer Science Interfaces Series, Vol. 39 (Springer, US, 2007), p. 261.

[21] J. Pugh, Y. Zhang, and A. Martinoli, in *Proc. Swarm Intelligence Symposium, Pasadena, 2005* (IEEE, New York, 2005), p. 92.

[22] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.107.233601 for additional figures.

[23] A. Azzalini, Scand. J. Stat. Theory Appl. **32**, 159 (2005).