Scalable randomised benchmarking of non-Clifford gates

Andrew W Cross, Easwar Magesan, Lev S Bishop, John A Smolin and Jay M Gambetta

Randomised benchmarking is a widely used experimental technique to characterise the average error of quantum operations. Benchmarking procedures that scale to enable the characterisation of $n$-qubit circuits rely on efficient procedures for manipulating those circuits and, as such, have been limited to subgroups of the Clifford group. However, universal quantum computers require additional, non-Clifford gates to approximate arbitrary unitary transformations. We define a scalable randomised benchmarking procedure over $n$-qubit unitary matrices that correspond to protected non-Clifford gates for a class of stabiliser codes. We present efficient methods for representing and composing group elements, sampling them uniformly and synthesising corresponding poly$(n)$-sized circuits. The procedure provides experimental access to two independent parameters that together characterise the average gate fidelity of a group element.

**INTRODUCTION**

A key step to realising a large-scale universal quantum computer is demonstrating that decoherence and other realistic imperfections are small enough to be overcome by fault-tolerant quantum computing protocols. Randomised benchmarking (RB) has become a standard experimental technique for characterising the average error of quantum gates partly because of its insensitivity to state preparation and measurement errors. Benchmarking provides robust estimates of average gate fidelity and it can characterise specific interleaved gate errors, addressability errors and leakage errors.

RB techniques that efficiently scale to many qubits have been limited to subgroups of gates in the Clifford group, as computations with this group are tractable. However, the Clifford group is not enough for general quantum computations. Previous work generalises RB to groups that include non-Clifford gates, but only on single qubits, a significant limitation. Methods for bounding the average fidelity of specific types of non-Clifford gates have also been considered.

We present a scalable RB procedure that includes important non-Clifford circuits, such as circuits composed from $T = \sqrt{Z}$ and controlled-NOT (CNOT) gates that naturally occur in fault-tolerant quantum computations. The $n$-qubit matrix groups we study are a generalisation of the standard dihedral group and coincide in some cases with protected gates in stabiliser codes, such as $k$-dimensional colour codes. Circuits built from these gates cannot be universal but do constitute significant portions of magic state distillation protocols, repeat-until-success circuits and the vital quantum Fourier transform. We show that there are efficient methods for representing and composing group elements, sampling them uniformly and synthesising corresponding circuits whose size grows polynomially with the number of qubits $n$. The benchmarking procedure provides experimental access to two independent noise parameters through exponential decays of average sequence fidelities.

**RESULTS**

The quantum circuits we consider are products of CNOT gates $\Lambda_{12}(X) := |u, u \oplus v\rangle$, bit-flip gates $X(u) := |u \oplus 1\rangle$ and single-qubit $m$-phase gates $Z_m(u) := \omega_m^u |u\rangle$, where $\omega_m = e^{2\pi i/m}$. More concisely, the circuits of interest are given by the group

$$G_m := \langle \Lambda_{ij}(X(j), Z_m(j)) \rangle^\dagger/\langle \omega_m \rangle. \quad (1)$$

We call this group a CNOT-dihedral group, as it is generated by CNOTs and a single-qubit dihedral group. Although we prove certain results for general $m$, we focus mainly on the case of $m = 2^k$. This case affords efficient benchmarking and contains non-Clifford gates of interest, such as $T = \sqrt{Z}$, controlled-$\sqrt{Z}$ (defined as $\Lambda_{ij}(\sqrt{Z}) := i^m |u, v\rangle$) and controlled–controlled–$Z$ (defined as $\Lambda_{123}(Z) := (-1)^{x^m} |u, v, w\rangle$), which is locally equivalent to a Toffoli gate.

Our interest in the dihedral group was motivated by symmetries of stabiliser codes. However, another group that may have similar properties is $G_{p,m} := \langle \Lambda_{ij}^p(X(j), Z_m(j)) \rangle^\dagger/\langle \omega_m \rangle$ where $\Lambda_{ij}^p(x)$ is a $p$-controlled-NOT gate. Not all entangling gates are suitable for randomised benchmarking though. Our arguments imply that the group $\langle \Lambda_{ij}(Z), X(j), Z_m(j) \rangle$ does not yield an efficient benchmarking procedure, as twirling over this group produces a map with exponentially many parameters.

The benchmarking procedure we present here both generalises and extends naturally to interleaving gates to estimate individual gate fidelities. The procedure closely follows but we describe it in some detail for completeness. Choose a sequence of $\ell+1$ unitary gates in which the first $\ell$ gates are uniformly random elements $g_{ij}, g_{ij}, \ldots, g_{ij}$ of $G_{p,m}$ and the $(\ell+1)$th gate is $g_{i,j}^{-1} := g_{i,j}^{-1} \chi_{ij}$, where $\chi_{ij}$ denotes the $\ell$-tuple $(j_1, \ldots, j_\ell)$ labelling the sequence. We show later that elements of $G_{p,m}$ can be efficiently sampled and $g_{i,j}^{-1}$ can be efficiently computed. For each sequence, we prepare an input state $\rho$, apply $S_k := g_{k}^{-1}g_{k} \cdots g_{k}$ and measure an operator $E$. 

---

1 IBM T.J. Watson Research Center, Yorktown Heights, NY, USA.

Correspondence: AW Cross (awcross@us.ibm.com)

Received 20 October 2015; revised 17 March 2016; accepted 28 March 2016

© 2016 University of New South Wales/Macmillan Publishers Limited
Assuming each gate $g_i$ has an associated error $\epsilon_i(\rho)$, the sequence $S_j$ is implemented as

$$S_j := \epsilon_i^{-1} g_k^{-1} \circ \left( O_{\epsilon_i^{-1}} [ \epsilon_i \circ g_i ] \right)$$

$$= \epsilon_i^{-1} \circ \left( O_{\epsilon_i^{-1}} [ \epsilon_i \circ g_i ] \right)$$

where each $g_i \in G_{\mathcal{N}}$. The overlap with $E$ is $\text{Tr}[E S_j(\rho)]$. Averaging this overlap over $K$ independent sequences of errors $\epsilon_i$ gives us an estimate of the average sequence fidelity $F_{\text{seq}}(E, E, \rho) = \frac{1}{K} \sum_{j} \text{Tr}[E S_j(\rho)]$ where $S_j(\rho) := \frac{1}{K} \sum_j S_j(\rho)$ is the average quantum channel.

Defining $\epsilon$ to be the average of errors $\epsilon_i$ and assuming for all $i$ that $\delta \epsilon_i := \epsilon_i - \epsilon$ is small, the average quantum channel is

$$\hat{S}_j(\rho) = \epsilon \circ \left( E_{\epsilon} \right) + O(\delta \epsilon)$$

where $E_{\epsilon}$ is the $G_\mathcal{N}$-twirl of $E$ (see Materials and Methods). The error operator $E$ is attributed to measurement error and perturbs $E$ to a new operator $E'$. We decompose the input state and this final measurement operator in the Pauli basis to give $\rho = \sum_{i,j} e_i e_j \rho$. Neglecting the $O(\delta \epsilon)$ term, the average sequence fidelity is

$$F_{\text{seq}}(E, E, \rho) = \text{Tr} \left[ E \left( E_{\epsilon} \right) \right] = A_2 \sigma_2^2 + A_3 \sigma_3^3 + e_1$$

To see this, it is convenient to express $E_{\epsilon}$ in a corresponding Liouville representation $R^2$ (see Methods). In this representation, $R^2$ is diagonal with three distinct diagonal elements corresponding to sets of Pauli operators: the identity $I$ has value 1, the $Z$-type Pauli operators $Z_\mathcal{N}(I)$ have value $a_0$ and the remaining Pauli operators $P \setminus Z$ have value $a_0$. The Pauli operator $P$ then contributes $\rho_{\epsilon} \rho^\tau$ to $F_{\text{seq}}(E, E, \rho)$, where $\alpha$ is one of 1, $a_2$ or $a_3$ depending on $P$.

In a spirit similar to simultaneous RB, each of the two exponential decays $\sigma_2^2$ and $\sigma_3^3$ can be observed by choosing appropriate input states. For example, if we choose the input state $\rho_{i,j} = |00\ldots0\rangle \langle 00\ldots0|$, then $F_{\text{seq}} = \epsilon_0 + A_2 \sigma_2^2$ where $A_2 = \sum_{i,j} e_i e_j \rho_{i,j}$. On the other hand, if we choose $|+\ldots+\rangle := \sum_{i,j} |01\rangle \langle 10| \rho_{i,j}$, then $F_{\text{seq}} = \epsilon_0 + A_1 \sigma_3^3$ where $A_1 = \sum_{i,j} |X_{\mathcal{N}}(i,j)\rangle \langle X_{\mathcal{N}}(i,j)| \rho_{i,j}$. State preparation errors may lead to deviation from a single exponential decay, but this is detectable. The channel parameters $a_2$ and $a_3$ can be extracted by fitting the average sequence fidelity. The corresponding depolarising channel parameter $a$ is a weighted average $a = (a_2 + 2a_3)/2a + 1$, and the average gate error is given by $e = (2b^2 - 1)/(2a + 1)$ (see ref. 6).

The Materials and Methods section is devoted to proving the technical results that enable the benchmarking procedure such as a canonical decomposition of $G_{\mathcal{N}}$, efficient computation within $G_\mathcal{N}$ and twirling over $G_\mathcal{N}$ to obtain the averaged quantum channel.

**DISCUSSION**

Our results enable scalable benchmarking of a natural family of non-Clifford circuits related to quantum error-correcting codes. In principle, our procedure allows efficient benchmarking of isolated non-Clifford gates, as well as large sub-circuits for state distillation or repeat-until-success protocols. Together with standard Clifford benchmarking, our procedures enable characterisation of the full range of gates used in the leading fault-tolerant quantum computing protocols. As multi-qubit benchmarking is well within experimental reach, we expect an optimised implementation of our procedure to be quite practical.

Several natural questions arise from this work. First, one might address the asymptotically optimal cost of circuit synthesis for elements of the CNOT-dihedral groups, as well as the practical question of finding optimal circuit decompositions for elements of the smallest groups. We expect optimal circuits are computationally hard to find as $n$ grows, but experimentally it is important to minimise the number of gates. Second, unlike the Clifford group, the CNOT-dihedral group is not a 2-design. It would be interesting to find a group (or set) containing a non-Clifford gate and that is a 2-design, and in which benchmarking can be done efficiently. Third, our results show that we can efficiently perform RB. However, we have not addressed the precise sense in which quantum computations over the CNOT-dihedral group can be efficiently simulated. This may be a subtle problem. Last, there are generalised stabiliser formalisms, such as, and it is natural to ask whether one of these describes how this group acts on some set of states.

**MATERIALS AND METHODS**

This section is devoted to proving the various results used in the benchmarking procedure: canonical decomposition of $G_{\mathcal{N}}$, efficient computation in $G_{\mathcal{N}}$, and twirling over $G_{\mathcal{N}}$ each of which is interesting in its own right. Let $m$ be general and let us briefly see some notation. The matrix representation of $G_{\mathcal{N}}$ is set by identifying $g \in G_{\mathcal{N}}$ to the matrix that maps $|0^n\rangle := |00\ldots0\rangle$ to $|b\rangle := |b_1b_2\ldots b_n\rangle$ with unit phase. We define the phase-flip gates $Z_u := (-1)^u|u\rangle\langle u|$, and controlled-Z gates $\Lambda_c(Z)|u\rangle\langle u| := (-1)^{uv}|u\rangle\langle u|$. The support of a bit string $v \in \{0,1\}^n$ is $\text{supp}(v) := \{i | v_i = 1\} \subseteq \{1,2,\ldots, n\}$. We refer to $v$ and its support interchangeably, viewing $v$ as a set and vice versa. Let $U$ be a single-qubit gate and $U(v)$ denote the gate acting as $U$ only on qubits in the support of $v$. Given $J \subseteq \{1,2,\ldots, n\}$ or elements $i, j, \ldots \in \{1,2,\ldots, n\}$, we also use the shorthand $U(J)$ and $U(i, j, \ldots)$.

Canonical form of $G_{\mathcal{N}}$

Our first goal will be to put $G_{\mathcal{N}}$ in a canonical form (the main result is contained in Theorem 1). The rewriting identities shown in Figure 1 allow us to commute diagonal elements of $G_{\mathcal{N}}$ through $\Lambda(X)$ and $X(i)$ gates.

**Figure 1.** Rewriting identities. Controlled-phase gate notation carrying the label $\alpha$ denotes a controlled-$\left(\begin{smallmatrix} \alpha \\ \alpha \end{smallmatrix}\right)^n$ gate. (a) This is the only identity that increases the number of controls. (b) This identity preserves the number of controls.
The rules for bit-flip gates are a special case of the CNOT rules. The following Lemma follows directly from definitions and formalises the role of the rewriting isomorphism in understanding the group’s structure.

Lemma 1: Let $W_m$ denote the subgroup of diagonal matrices of $G_m$ and let $\Pi = \{\Lambda(X), \Phi(j)\}$ denote the subgroup of permutation matrices. Then, $G_m$ is isomorphic to the semi-direct product of groups $G_m \rtimes W_m$. The proof of Lemma 1 is given in the Supplementary Material. Note that by definition $\Pi = \mathcal{X} \rtimes \mathcal{X}'$. As $\mathcal{X}' \cong \mathbb{Z}$ and $\mathcal{X} \cong G_m$, each element $\pi \in \Pi$ can be associated with an $n$-bit string $c \in \mathbb{F}_2^n$ and an $n$-inertible $0-1$ linear transformation $B \in G_m$. Then $\Pi \cong \mathcal{X} \rtimes \mathcal{X}'$. Here $F_2$ denotes the field with two elements. Furthermore, $|\Pi| = 2^n |F_2| = 2^{2^n}$. It remains to better understand $W_m$ (see Lemma 3 for the main result). Let $D_m$ denote the group of $2^m$ by $2^m$ diagonal unitary matrices $D$ with elements $\Theta(D) = e^{i\theta}$. Here $\ell : F_2 \to D_m$ is a function that assigns $m$th roots of unity to the diagonal and $Z_m$ is the ring of integers modulo $m$. Since $G_m$ is generated by permutation matrices and products of $m$-phase gates, $W_m \subseteq D_m$.

Let $\mathcal{S} \subset Z_m[x_1, \ldots, x_{2^n}]$ denote the polynomial ring whose elements are $p(x) := p(x_1, \ldots, x_{2^n}) = \sum_{a \in \{0, 1\}^n} p_a x_a^n$ where $a = a_1 \cdots a_n$ is a multi-index, $p_a \in Z_m$ and $x_a^n = x_{a_1} x_{a_2} \cdots x_{a_n}$. A multi-index $a$ is monomial. The multi-index takes values in $\{0, 1\}^n$ as a convenient notation, as we will evaluate $p(x)$ on binary strings, so $x_a^n = x_{a_1} x_{a_2} \cdots x_{a_n}$. The degree of a monomial is denoted $|a|$. We mainly consider $\mathbb{S}$ as an additive group. The next Lemma follows from the definition of group isomorphism and the fact that each function $f(b)$ can be expressed as a polynomial in $\mathcal{S}$.

Lemma 2: Let $p(b)$ denote evaluation of $p$ on the $n$-bit binary string $b = b_1 \cdots b_n$ with operations in $Z_m$. The function $\Phi : \mathcal{S} \to D_m$ given by $\Phi(p(b)) = e^{i\theta}$. This is a group isomorphism.

The proof of Lemma 2 is given in the Supplementary Material. The rewriting identities give an action of $\Pi$ on $W_m$ by conjugation. Let $\mathcal{W}_m := \{(\Phi(p)) : p \in \mathbb{S}\}$. On the basis of a similar application of the rewriting identities as in Lemma 1, $W_m = \langle \mathcal{W}_m \rangle$. As $W_m \subseteq D_m \cong \mathcal{S}$, $\mathcal{S}$ associates a polynomial in $\mathcal{S}$ to each element of $W_m$. By our chosen convention, matrices representing elements $w \in W_m$ are given modulo a global phase factor $e^{i\theta}$ such that $w(0^n) = 0^n$. Therefore, the preimages $\mathcal{W}_m$ have zero constant term—i.e., $p_x = 0$ when $|a| = 0$. Through $\Phi$, the rewriting identities define an action of $\Pi$ on $\mathcal{S}$ that, respectively, takes $x_1 x_2 \ldots x_p x_j$ to

$$-2x_1 x_2 \ldots x_p x_j + x_1 x_2 x_3 \ldots x_p x_j + x_1 x_2 \ldots x_p x_j \ldots \ldots$$

and $x_1 x_2 \ldots x_p x_j$ to

$$-x_1 x_2 x_3 \ldots x_p x_j + x_1 x_2 \ldots x_p x_j \ldots \ldots$$

Equation (6) increments the degree of a monomial and multiplies its coefficient by $-2$, whereas Equation (7) does not change the degree. Another way to understand iterated applications of Equation (6) is to observe that

$$x_1 \oplus x_2 \oplus \ldots \oplus x_n = \sum_{a \in \mathbb{F}_2^n, |a| = 0} (-2)^{|a|} x_a.$$

This fact relates how single qubit $Z_m$ gates acting on mod 2 linear combinations of input bits are equivalent to products of certain controlled-phase gates.

There is an element of $W_m$, corresponding to each monomial term of $\Pi$, whose form is given in Lemma 3, a sequence of CNOT gates and a sequence of bit-flip gates.

Lemma 3: $W_m$ is isomorphic to the subgroup $\mathcal{W}_m \leq \mathcal{S}$ given by

$$\{ p \in \mathcal{S} : p_0 = 0 \}$$

We can now directly compute $|G_m|$. Corollary 1

$|G_m| = 2^n \prod_{k=0}^{n-1} \binom{2^n - 2^k}{2^k - 1}.$

Proof: Let $\omega_m(a) = \text{LCM}(\alpha, m)/\alpha$. Denote the order of $a$ in $Z_m$. Observe that $(-2)^{\langle a \rangle} z_m = Z_\omega_m z_m$ as additive groups. Therefore, $W_m$ is isomorphic to a direct product of additive cyclic groups $A_m := \prod_{\alpha = 1}^{\omega_m(n)} Z_\omega_m$. This shows that $|G_m| = |A_m| = |\Pi|$. Putting everything together, we have

Theorem 1: Any element of $G_m$ can be written in canonical form as the composition of a sequence of phase gates (comprising an element of $W_m$, whose form is given in Lemma 3), a sequence of CNOT gates and a sequence of bit-flip gates. Efficient computation in $G_m$.
Given a triple \((p, \beta, c)\), we synthesise a corresponding circuit from products of CNOT gates, bit-flip gates and single-qubit m-phase gates. Our goal is efficiently synthesise a circuit whose size (number of gates) is polynomial in \(n\) but not to optimise this circuit. We independently synthesise circuits coinciding with \(p, \beta,\) and \(c\) as \(c\) corresponds to \(X(c)\), and a CNOT circuit for \(\beta\) can be found by Gaussian elimination,\(^4\) the new part of the algorithm synthesises a circuit for \(p\).

We describe the circuit synthesis for \(p\) informally. The algorithm proceeds in \(k\) rounds. Begin by initialising a working polynomial \(q(x) = px(x)\), set a round counter \(t = k\) and set a quantum circuit \(U = I\). Here \(\cdot\)~ denotes assignment. In round \(t\), we synthesise a circuit corresponding to a polynomial \(p^\alpha(x)\) that coincides with \(q(x)\) on its degree \(t\) terms. For each of the \(O(n^t)\) degree-\(t\) terms \(\left(\sim \prod_{i=1}^t p_i x^i\right)\) of \(q(x)\), we apply the constant-sized circuit \(g_{\alpha} := \rho_{g} \left(\mathcal{Z}(j)^n\right)^{n^t} \rho_{g}^{-1}\) setting \(U \leftarrow g_{\alpha}U\), where \(J=\supp(a)\) as in the proof of Lemma 3. The product of the \(g_{\alpha}\) corresponds to \(p^{\alpha}(x):= \prod_{a \in J, |a| \leq t} p \in \mathbb{P}(x)\). Therefore, we update \(q(x) := q(x) - p^{\alpha}(x)\), which now has maximum degree \(r-1\), decrement the round counter and proceed to the next round. The algorithm terminates when \(q(x) = 0\) and \(t = 0\). The total algorithm run-time and circuit size of the output \(U = O(n^t)\).

Twirling over \(G_\alpha\)

A quantum channel is a completely positive trace-preserving map whose operator sum decomposition is \(\mathcal{E}(\rho) = \sum_{A} \rho A A^\dagger\) where \(\sum_{A} A^\dagger A = I\). The twist of \(\mathcal{E}\) over a finite group \(G\) (G-twirl) is given by

\[
\mathcal{E}_G(\rho) := \frac{1}{|G|} \sum_{g \in G} U^\dagger \mathcal{E}(U \rho U^\dagger)) U.
\]

(15)

In what follows, we use several facts about group twirls. If \(G = AB\) is a direct product of groups, then \(\mathcal{E}_G(\rho) = \mathcal{E}_A(\rho) \mathcal{E}_B(\rho)\), and if \(A\) is a normal subgroup of \(G\) (denoted \(A \triangleleft G\)), then \(\mathcal{E}_G(\rho) = \mathcal{E}_{G/A}(\rho)\), where the twirl over the factor group \(G/A\) is over a set of coset representatives. Twirling any map over the Pauli group produces a Pauli channel.\(^5\) Consider a Pauli channel \(\mathcal{E}(\rho) = \sum_{a \in G} \rho \mathcal{E}_G(\rho)\), then \(\mathcal{E}(\rho)\) is over the group \(G\) isover a set of coset representatives. Any element \(\rho \in \mathcal{E}(\rho)\) gives a new equivalence relation \(P = Q\) if only if \(\rho \mathcal{E}_G(\rho)\). This relation partitions \(\rho\) into a disjoint union of orbits. By the orbit-stabiliser theorem and Lagrange's theorem,\(^2\) \(|\mathcal{E}(\rho)| = |G|\). The stabiliser of the channel \(\mathcal{E}(\rho)\) is over a set of coset representatives. Therefore, the twist, (Equation 15), can be written

\[
\mathcal{E}_G(\rho) = \sum_{C \in G} \sum_{C \in \mathcal{E}(\rho)} \left(\sum_{g \in C \in \mathcal{E}(\rho)} \frac{|g|}{|C|}\right) P P^\dagger.
\]

(16)

where \(C\) is a set of representative elements, one from each orbit.

These facts allow us to compute the twirl over \(G_\alpha\) when \(k > 1\) by expressing it as a sequence of twirls. We begin by decomposing the group. Let \(W_{\alpha} := W_{\alpha} \backslash \{I\}\) and recall that \(W_{\alpha} := \{\mathcal{Z}(j)\}\), then \(W_{\alpha} = W_{\alpha} \backslash \{I\}\). As \(C \subset W_{\alpha}\), \(C \subset W_{\alpha}\) we form the corresponding factor groups. Therefore, an element \(w \in W_{\alpha}\) can be written as \(w = w_1 w_2 \mathcal{P} w_1 w_2\), where \(w_1\) labels cosets \(w_1 \mathcal{Z}\), \(w_2 \mathcal{Z}\) labels cosets \(w_1 \mathcal{Z}\), \(w_2 \mathcal{Z}\) labels cosets \(w_1 \mathcal{Z}\), \(w_2 \mathcal{Z}\). Finally, by Lemma 1, any element \(g \in G_\alpha\) factors as \(w = w_1 w_2 \mathcal{P} w_1 w_2\), where \(w_1 \mathcal{Z}\), \(w_2 \mathcal{Z}\), \(w_1 \mathcal{Z}\), \(w_2 \mathcal{Z}\). Therefore, we have \(g = w w_1 w_2 \mathcal{P} w_1 w_2\). Our strategy is to use the decomposition to express the \(G_\alpha\)-twirl as a sequential \(P\)-twirl, \(C\)-twirl, \(C\)-twirl, \(C\)-twirl, \(C\)-twirl. Each twirl can be computed in a straightforward manner using the facts we have described, and it reduces the number of independent parameters describing the channel until we have twirled over the whole of \(G_\alpha\) (See Supplementary Materials for further details.). The final twirled map is

\[
\mathcal{E}(\rho) = \beta_0 \rho + \beta_2 \sum_{P \in \mathbb{P}(\mathbb{Z})} P \rho P + \beta_3 \sum_{P \in \mathbb{P}(\mathbb{Z})} P \rho P.
\]

(17)

In the Liouville representation in the Pauli basis, which has matrix elements \(R^{(P)}_{PQ} = Tr (\mathcal{E}(Q))\) where \(P\) and \(Q\) are \(n\)-qubit Pauli operators, this map has three diagonal elements \(R^{(P)}_{PQ}\) where \(P \leq Q\) and \(P \geq Q\) with elements \(1, 2, 4, 8, 16\) respectively.
Supplementary Information accompanies the paper on the npj Quantum Information website (http://www.nature.com/npjqi)