Article

Halving the cost of quantum algorithms with randomization

Check for updates

John M. Martyn [1,2,3] ✉ & Patrick Rall[2,3]

Quantum signal processing (QSP) provides a systematic framework for implementing a polynomial transformation of a linear operator, and unifies nearly all known quantum algorithms. In parallel, recent works have developed *randomized compiling*, a technique that promotes a unitary gate to a quantum channel and enables a quadratic suppression of error (i.e., $\epsilon \to O(\epsilon^2)$) at little to no overhead. Here we integrate randomized compiling into QSP through *Stochastic Quantum Signal Processing*. Our algorithm implements a probabilistic mixture of polynomials, strategically chosen so that the average evolution converges to that of a target function, with an error quadratically smaller than that of an equivalent individual polynomial. Because nearly all QSP-based algorithms exhibit query complexities scaling as $O(\log(1/\epsilon))$—stemming from a result in functional analysis—this error suppression reduces their query complexity by a factor that asymptotically approaches 1/2. By the unifying capabilities of QSP, this reduction extends broadly to quantum algorithms, which we demonstrate on algorithms for real and imaginary time evolution, phase estimation, ground state preparation, and matrix inversion.

Classical randomness plays a pivotal role in the design of quantum protocols and algorithms. In the near-term, randomized benchmarking[1] is central to calibrating and assessing the quality of quantum gates, and quasi-probability methods like probabilistic error cancellation and noise twirling can help reduce noise[2]. Similarly, random circuit sampling is central to quantum supremacy experiments[3], and randomized measurements provide a powerful probe into the properties of complex quantum systems[4]. As we progress towards quantum advantage and early fault-tolerant quantum hardware, many lines of research aim to reduce the requirements of traditional quantum algorithms by incorporating classical randomness[5–7].

Randomized compiling is a key example of leveraging classical randomness to improve quantum computation[8–10]. As its name suggests, this process randomly compiles gates at execution time, or equivalently, promotes a unitary gate to a quantum channel that is a probabilistic mixture of unitaries. Remarkably, randomized compiling can quadratically suppresses gate errors without increasing the cost of circuit synthesis. Yet, applications of this technique to quantum algorithms have so far been restricted to Trotterized Hamiltonian simulation[11–15] and phase estimation[16], leaving a vacuum of applications to other algorithms.

In an effort to fill this gap of randomized quantum algorithms, we propose using quantum signal processing (QSP)[17,18] as a medium for achieving widespread advantage of randomized compiling. QSP prepares a polynomial transformation of a linear operator, and has been shown to encompass nearly all quantum algorithms, from Hamiltonian simulation and quantum search, to matrix inversion and fast integer factoring[19,20].

In this work, we achieve exactly this goal. We merge randomized compiling with QSP by developing *Stochastic Quantum Signal Processing* (Stochastic QSP). By virtue of randomized compiling, our construction quadratically suppresses the error in a QSP polynomial approximation of a target function. To study how this suppression impacts the cost of QSP-based algorithms, we show that an elementary result in the approximation of smooth functions implies that nearly all QSP-based algorithms achieve a query complexity that scales with the error ϵ as $O(\log(1/\epsilon))$, which we also empirically confirm. Hence the quadratic suppression of error afforded by stochastic QSP translates to an asymptotic halving of the cost of a QSP-based algorithm over their deterministic counterparts (asymptotic in the limit of $\log(1/\epsilon)$ dominating the cost). In realizing this cost reduction, we "combine the strengths of QSP and randomization," as hypothesized in ref. 21.

An outline of this work is as follows. We first present the notation used in this paper below. Then, in the Results section, we review QSP and other preliminary topics, and subsequently present our main result on stochastic QSP. Here we also benchmark the performance of stochastic QSP for various quantum algorithms, including real and imaginary time evolution, phase estimation, ground state preparation, and matrix inversion. The proofs and generalizations of our construction are provided in the Methods and Supplementary Information.

[1]Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA, USA. [2]IBM Quantum, MIT-IBM Watson AI Lab, Cambridge, MA, USA. [3]These authors contributed equally: John M. Martyn, Patrick Rall. ✉e-mail: jmmartyn@mit.edu

Notation:

We will study functions $F(x)$ on the domain $x \in [-1, 1]$ and use the function norm

$$\| F \|_{[-1,1]} := \max_{x \in [-1,1]} |F(x)|, \qquad (1)$$

where we will focus on functions bounded as $\|F\|_{[-1,1]} \le 1$.

A convenient set of functions on this domain are the Chebyshev polynomials. The order n Chebyshev polynomial is defined as $T_n(x) = \cos(n \arccos(x))$ for integer $n \ge 0$, and is a polynomial of degree n with parity $n \bmod 2$ (i.e., either even or odd) and bounded magnitude $\|T_n\|_{[-1,1]} = 1$. The Chebyshev polynomials furnish an orthogonal basis in which an arbitrary function on $x \in [-1, 1]$ can be expanded:

$$F(x) = \frac{c_0}{2} + \sum_{n=1}^{\infty} c_n T_n(x), \quad c_n = \frac{2}{\pi} \int_{-1}^{1} \frac{F(x) T_n(x)}{\sqrt{1 - x^2}} \, dx, \qquad (2)$$

where c_n are the Chebyshev coefficients.

In this work we will also study unitary and non-unitary transformations. We will denote an operator by a Latin character, say A, and the associated channel by the corresponding calligraphic character: $\mathcal{A}(\rho) = A\rho A^{\dagger}$. In analyzing these operators, we will consider the spectral norm $\| A \| = \sup_{|\psi\rangle} \|A|\psi\rangle\|$ and the trace norm $\| A \|_1 = \mathrm{tr}(\sqrt{A^{\dagger}A})$, which equate to the maximal singular value and the sum of singular values, respectively.

Another relevant metric is the diamond norm, defined for a channel $\mathcal{E}$ as:

$$\|\mathcal{E}\|_{\diamond} = \sup_{\rho} \left\| (\mathcal{E} \otimes \mathcal{I})(\rho) \right\|_1, \qquad (3)$$

where this supremum is taken over normalized density matrices ρ in a possibly-enlarged Hilbert space, and $\mathcal{I}$ is the identity channel. The diamond norm induces the diamond distance between two channels:

$$d_{\diamond}(\mathcal{E}, \mathcal{F}) := \frac{1}{2} \| \mathcal{E} - \mathcal{F} \|_{\diamond}. \qquad (4)$$

For channels $\mathcal{A}(\rho) = A\rho A^{\dagger}$ and $\mathcal{B}(\rho) = B\rho B^{\dagger}$ with spectral norms $\|A\|$, $\|B\| \le 1$, the diamond distance is upper bounded as (see Lemma 4 of ref. [22] for proof):

$$d_{\diamond}(\mathcal{A}, \mathcal{B}) = \frac{1}{2} \| \mathcal{A} - \mathcal{B} \|_{\diamond} \le \| A - B \| . \qquad (5)$$

Results

The goal of this work is to integrate randomized compiling into QSP, and thereby establish a framework for designing randomized quantum algorithms that achieve reduced query complexities. Our main result is the fulfillment of this goal through *Stochastic QSP*: we replace a single QSP polynomial with a channel that is a probabilistic mixture of QSP polynomials, each strategically crafted to exploit randomized compiling and quadratically suppress error. As we show, this furnishes randomized QSP-based algorithms with roughly half the cost of their deterministic counterparts. In order to develop stochastic QSP and prove our main result, we first review its essential components: QSP, polynomial approximations to smooth functions, and randomized compiling.

QSP

The quantum signal processing (QSP) algorithm is a systematic method of implementing polynomial transformations on a quantum subsystem[17,18,23]. QSP works by interleaving a signal operator U, and a signal processing operator S, both taken to be SU(2) rotations about different axes. Conventionally, U is an x-rotation through a fixed angle, and the S a z-rotation through a variable angle ϕ:

$$U(x) = \begin{bmatrix} x & i\sqrt{1 - x^2} \\ i\sqrt{1 - x^2} & x \end{bmatrix}, \qquad S(\phi) = e^{i\phi Z}. \qquad (6)$$

Then, with a set of $d + 1$ *QSP phases* $\vec{\phi} = (\phi_0, \phi_1, \ldots, \phi_d) \in \mathbb{R}^{d+1}$, the following *QSP sequence* is defined as an interleaved product of U and S, whose matrix elements are polynomials in x:

$$
\begin{aligned}
U_{\vec{\phi}}(x) &= S(\phi_0) \prod_{i=1}^{d} U(x) S(\phi_i) \\
&= \begin{bmatrix} P(x) & iQ(x)\sqrt{1 - x^2} \\ iQ^*(x)\sqrt{1 - x^2} & P^*(x) \end{bmatrix},
\end{aligned}
\qquad (7)
$$

where $P(x)$ and $Q(x)$ are polynomials parameterized by $\vec{\phi}$ that obey:

1. $\deg(P) \le d$, $\deg(Q) \le d - 1$
2. $P(x)$ has parity $d \bmod 2$, and $Q(x)$ has parity $(d - 1) \bmod 2$ (8)
3. $|P(x)|^2 + (1 - x^2)|Q(x)|^2 = 1, \ \forall x \in [-1, 1]$.

This result implies that one can prepare polynomials in x by projecting into a block of $U_{\vec{\phi}}$, e.g., $\langle 0 | U_{\vec{\phi}} | 0 \rangle = P(x)$. While this class of polynomials is limited by the conditions of Eq. (8), one can show that by projecting into other bases (e.g., the $|+\rangle\langle +|$ basis), and incorporating linear-combination-of-unitaries circuits[24–26], QSP can encode an arbitrary degree-d polynomial that need only obey $\|P\|_{[-1, 1]} \le 1$[20]. For any such polynomial, the corresponding QSP phases $\vec{\phi}$ can be efficiently determined classically[27–31], thus amounting to merely a pre-computation step. As per Eq. (7), the cost of realizing such a degree-d polynomial is d queries to $U(x)$.

Remarkably, QSP can be generalized to implement polynomial transformations of linear operators through its extension to the quantum eigenvalue transformation (QET)[18,23] and quantum singular value transformation (QSVT)[20]. This is achieved analogous to QSP: provided access to a unitary $U[A]$ that block-encodes an operator A, one can construct a sequence of $U[A]$ and parameterized rotations that encodes a polynomial $P(A)$:

$$U[A] = \begin{bmatrix} A & \cdot \\ \cdot & \cdot \end{bmatrix} \mapsto U_{\vec{\phi}}[A] = \begin{bmatrix} P(A) & \cdot \\ \cdot & \cdot \end{bmatrix}, \qquad (9)$$

where the unspecified entries ensure unitarity. In essence, this applies QSP within each eigenspace (or singular value space) of A, and outputs a degree-d polynomial $P(A)$ acting on the eigenvalues (or singular values) of A. The cost of realizing this is d queries to the block-encoding, translating to a run-time $O(d)$.

QET and QSVT are powerful algorithms, shown to unify and simplify most known quantum algorithms, while maintaining near-optimal query complexities[20]. An algorithm can be cast into the language of QET/QSVT by constructing a polynomial approximation to a matrix function that solves the problem of interest. For instance, in Hamiltonian simulation, one can design a polynomial $P(H) \approx e^{-iHt}$ to simulate time evolution[18,23]. Algorithms encompassed in this framework include the elementary algorithms of search, simulation, and phase estimation[19,22], as well as more intricate algorithms, like matrix inversion[5,32,33], and ground state preparation[6,7].

In this work, we use the term "QSP" in place of "QET" and "QSVT", following the conventional parlance. However, this should be understood to be QET/QSVT when acting on a linear operator rather than a scalar.

Polynomial approximations to smooth functions

As emphasized above, the utility of QSP lies in generating matrix functions without the need to unitarily diagonalize the underlying matrix. Specifically,

QSP enables the approximation of a matrix function $F(A) \approx P(A)$ by a polynomial $P(A)$, where the accuracy of this approximation can be tuned by increasing the degree of $P(x)$. Because the cost of QSP-based algorithms scales with the polynomial degree, their complexity rests on results in approximation theory.

To make this connection concrete, consider the decaying exponential function $e^{-\beta x}$ for a parameter $\beta > 0$. In QSP, this function is employed to prepare thermal states and estimate partition functions at inverse temperature β[20]. It is well established that $e^{-\beta x}$ can be approximated to within additive error ϵ over $x \in [-1, 1]$ by a polynomial of degree $d = O(\sqrt{\beta} \log(1/\epsilon))$[20,34]. Similarly, consider the error function $\text{erf}(kx)$ for a parameter $k > 0$, which is used in QSP to approximate the step function by selecting a large k and is applicable to ground state preparation[6,35]. Prior work has proven that $\text{erf}(kx)$ can be approximated to within additive error ϵ by a polynomial of degree $d = O(k \log(1/\epsilon))$[20,34]. In both cases, the degree grows with which decreasing error and increasing parameters β or k.

Observe that in each of these examples, the degree of the polynomial scales with the error as $O(\log(1/\epsilon))$. This scaling is a generic feature of polynomial approximations to smooth functions, which originates from the expansion of a function on $x \in [-1, 1]$ in the basis of Chebyshev polynomials (see Eq. (2)):

$$F(x) = \frac{c_0}{2} + \sum_{n=1}^{\infty} c_n T_n(x). \qquad (10)$$

As we prove in the Supplementary Information Section I, if $F(x)$ is C^∞ function (i.e., continuous and infinitely differentiable), then its Chebyshev coefficients decay super-polynomially as $|c_n| = e^{-O(n^r)}$ for some exponent $r > 0$.

For a large class of smooth functions, it is found that $r = 1$[36], such that $|c_n| = e^{-O(n)}$ decays geometrically. In this case, a truncation of the Chebyshev series at order d, $P(x) = \sum_{n=0}^{d} c_n T_n(x)$, furnishes a degree d polynomial approximation to $F(x)$ that suffers error

$$\max_{x \in [-1,1]} |P(x) - F(x)| = \max_{x \in [-1,1]} \left| \sum_{n=d+1}^{\infty} c_n T_n(x) \right| \qquad (11)$$
$$\leq \sum_{n=d+1}^{\infty} |c_n| = \sum_{n=d+1}^{\infty} e^{-O(n)} = e^{-O(d)}.$$

Hence, to guarantee an error at most ϵ, it suffices to choose a degree $d = O(\log(1/\epsilon))$.

In practice many polynomial approximations are constructed via truncated Chebyshev series. This includes polynomial approximations to a wide range of functions relevant to quantum algorithms, including the decaying exponential $e^{-\beta x}$, trigonometric functions $\sin(tx)$, $\cos(tx)$, the step function $\Theta(x)$, the inverse function $1/x$, and beyond. (Although the step function $\Theta(x)$ and the inverse function $1/x$ exhibit singularities at $x = 0$, and thus are not C^∞ functions, they can however be approximated by C^∞ functions by excluding a small region around their singularity. This strategy is used in practice, and renders these function amenable to results on polynomial approximations to smooth functions.) Accordingly, these approximations all exhibit degrees that scale with the error as $d = O(\log(1/\epsilon))$, which carries over to the complexity of their corresponding QSP-based algorithms.

Randomized compiling and the mixing lemma

In order to incorporate randomization into QSP, we will use *randomized compiling*. Formally introduced in ref. 8, randomized compiling replaces a deterministic quantum circuit with a circuit sampled from a distribution. This process can be viewed as replacing a unitary operation with a a quantum channel that is a probabilistic mixture of unitaries.

Remarkably, if this mixture is chosen strategically, randomized compiling enables a quadratic suppression of error: if an individual gate

approximates a target unitary with error ϵ, the randomly compiled channel can approximate the corresponding target channel with error $O(\epsilon^2)$. This error suppression is achieved at little to no increase in overhead, requiring only the ability to classically sample a distribution and implement gates on the fly.

The precise error suppression is quantified by the Hastings-Campbell mixing lemma:

Lemma 1. (Hastings-Campbell Mixing Lemma[9,10]). Let V be a target unitary operator, and $\mathcal{V}(\rho) = V\rho V^\dagger$ the corresponding channel. Suppose there exist m unitaries $\{U_j\}_{j=1}^{m}$ and an associated probability distribution p_j that approximate V as

$$\| U_j - V \| \leq a \text{ for all } j, \text{ and } \left\| \sum_{j=1}^{m} p_j U_j - V \right\| \leq b, \qquad (12)$$

for some $a, b > 0$. Then, the corresponding channel $\Lambda(\rho) = \sum_{j=1}^{m} p_j U_j \rho U_j^\dagger$ approximates $\mathcal{V}$ as

$$\| \Lambda - \mathcal{V} \|_\diamond \leq a^2 + 2b. \qquad (13)$$

This lemma enables a quadratic suppression of error if one can specify an ensemble of unitaries $\{U_j\}$ that each achieve spectral error $a = \epsilon$, and a distribution p_j such that $b = O(\epsilon^2)$. Then, while an individual unitary U_j suffers diamond norm error $O(\epsilon)$, the channel Λ achieves error $\leq a^2 + 2b = O(\epsilon^2)$. Importantly, because Λ is a probabilistic mixture of the unitaries $\{U_j\}$, the cost of simulating Λ is no more expensive than the cost of sampling p_j and implementing an individual unitary U_j.

The mixing lemma has been leveraged to improve a variety of quantum protocols through randomized compiling. Noteworthy examples include reducing the cost of gate synthesis[9,10,37–39], tightening fault-tolerance thresholds for general noise models[8], and enhancing the precision of state preparation[40,41]. On the algorithmic side, the mixing lemma has been merged with Trotterization to significantly reduce the complexity of chemical simulations[11,13], double the order of Trotter formulae[14], and accelerate imaginary time evolution[42,43]. Here we continue this campaign by extending randomized compiling to QSP. As QSP unifies nearly all quantum algorithms[19], this paves the way for new randomized quantum algorithms with reduced query complexities.

The mixing lemma for block-encodings

As emphasized in above, QSP polynomials are constructed as block-encodings. That is, a QSP polynomial $P(A)$ is encoded in a block of a higher dimensional unitary U, and accessed as $P(A) = \Pi U \Pi'$ for some orthogonal projectors Π, Π'. Conventionally, as in Eq. (9), the projectors are taken to be $\Pi = \Pi' = |0\rangle\langle 0| \otimes I$, such that $P(A)$ is encoded in the $|0\rangle\langle 0|$ block of the unitary.

To apply the mixing lemma to QSP, it is therefore necessary to establish a variant of the mixing lemma for operators block-encoded in unitary transformations. For the sake of simplicity, we present this theorem for an operator encoded in the $|0\rangle\langle 0|$ block of a unitary:

Lemma 2. (Mixing Lemma for Block-Encodings: $|0\rangle\langle 0|$ Block). Let V be a unitary that block-encodes a (possibly non-unitary) target operator S as $S = (\langle 0| \otimes I)V(|0\rangle \otimes I)$. Suppose there exist m unitaries $\{U_j\}_{j=1}^{m}$ that block-encode operators R_j as $R_j = (\langle 0| \otimes I)U_j(|0\rangle \otimes I)$. Also suppose there exists an associated probability distribution p_j such that

$$\| R_j - S \| \leq a \text{ for all } j,$$
$$\left\| \sum_{j=1}^{m} p_j R_j - S \right\| \leq b. \qquad (14)$$

Then, the corresponding unitary channel $\Lambda(\rho) = \sum_{j=1}^{m} p_j U_j \rho U_j^{\dagger}$ approximates the action of the channel $\mathcal{V}$ as

$$\left\| \bar{\Lambda} - \bar{\mathcal{V}} \right\|_{\diamond} \le a^2 + 2b, \tag{15}$$

where $\bar{\Lambda}$ and $\bar{\mathcal{V}}$ are channels that access the block-encodings of Λ and $\mathcal{V}$ by appending an ancilla qubit and projecting onto $|0\rangle$:

$$\begin{aligned} \bar{\Lambda}(\rho) &= (\langle 0| \otimes I) \cdot \Lambda\left(|0\rangle\langle 0| \otimes \rho\right) \cdot (|0\rangle \otimes I) \\ \bar{\mathcal{V}}(\rho) &= (\langle 0| \otimes I) \cdot \mathcal{V}\left(|0\rangle\langle 0| \otimes \rho\right) \cdot (|0\rangle \otimes I). \end{aligned} \tag{16}$$

For brevity, we defer the proof of this theorem to Supplementary Information Section II. there, we also showcase an analogous result for arbitrary block-encodings accessed by projectors Π, Π'. Lemma 2 indicates that by implementing the probabilistic mixture of block-encoding unitaries $\Lambda(\sigma) = \sum_{j=1}^{m} p_j U_j \sigma U_j^{\dagger}$ on an input state $\sigma = |0\rangle\langle 0| \otimes \rho$, and projecting the block-encoding qubit onto $|0\rangle$ (hence the projectors $|0\rangle \otimes I$ and $\langle 0| \otimes I$), one can reproduce evolution under the target operator S with diamond norm error $a^2 + 2b$. In short, this result is proven by showing that the channel $\bar{\Lambda}(\rho)$ is equal to the probabilistic mixture of block-encoded operators $\sum_{j=1}^{m} p_j R_j \rho R_j^{\dagger}$, to which the mixing lemma applies. Parallel to the usual mixing lemma, this result enables a quadratic suppression of error by selecting operators R_j and an associated probability distribution p_j such that $a = \epsilon$ and $b = O(\epsilon^2)$.

Main result: stochastic QSP

Lemma 2 very naturally applies to QSP. In this context, the target operation is a matrix function: $S = F(A)$, yet the operators we have access to are QSP polynomials: $R_j = P_j(A)$. A common goal is to simulate evolution under $F(A)$ as $\mathcal{F}_A(\rho) = F(A)\rho F(A)^{\dagger}$, which encompasses algorithms such as time evolution, linear systems solvers, and ground state preparation, among many others. Traditionally, one achieves this goal with QSP by finding a suitable polynomial approximation to $F(x)$ as $|P(x) - F(x)| \le \epsilon$, such that evolving under this polynomial with QSP as $\mathcal{P}_A(\rho) = P(A)\rho P(A)^{\dagger}$ suffers error $\| \mathcal{P}_A - \mathcal{F}_A \|_{\diamond} \le O(\epsilon)$. If $P(x)$ is a degree d polynomial, this procedure requires d queries to the block-encoding of A.

Here we exploit the mixing lemma to approximate evolution under $F(A)$ to the same level of accuracy, but at asymptotically half the number of queries to the block-encoding. We achieve this by designing an ensemble of polynomials that each approximate $F(A)$ as $\| P_j(A) - F(A) \| \le O(\sqrt{\epsilon})$, and an associated probability distribution that obeys $\|\sum_j p_j P_j(A) - F(A)\| \le O(\epsilon)$. Then, Lemma 2 readily implies that the channel $\Lambda_A(\rho) = \sum_j P_j(A)\rho P_j(A)^{\dagger}$ suffers error $\| \Lambda_A - \mathcal{F}_A \|_{\diamond} \le O(\epsilon)$. We also show that implementing Λ_A requires a number of queries to the block-encoding $\approx d/2 + O(1)$, a cost reduction stemming from the fact that polynomial approximations of smooth functions tend to have degrees that scale as $d = O(\log(1/\epsilon))$. Intuitively, this scaling implies that a polynomial that achieves error $O(\sqrt{\epsilon})$ (e.g., $P_j(x)$) has a degree that is asymptotically half that of a polynomial that achieves error $O(\epsilon)$ (e.g., $P(x)$).

Presentation. Therefore, rather than implement a degree d polynomial, one can instead sample over an ensemble of polynomials of *average degree* $\approx d/2 + O(1)$, while retaining the same level of precision. As the corresponding channel is constructed as a probabilistic mixture of QSP sequences, we term this algorithm *Stochastic Quantum Signal Processing*:

Theorem 1. (Stochastic QSP). Suppose that $F(x)$ is a bounded function $\|F\|_{[-1, 1]} \le 1$ with a Chebyshev expansion

$$F(x) = \sum_{n=0}^{\infty} c_n T_n(x) \tag{17}$$

on the domain $x \in [-1, 1]$, where for some degree $d \ge 2$ the coefficients decay as

$$|c_n| \le C e^{-qn} \text{ for all } n \ge d/2, \tag{18}$$

for some constants $C, q > 0$. Suppose furthermore that a degree-d truncation of $F(x)$, $P(x) = \sum_{n=0}^{d} c_n T_n(x)$, achieves an approximation error of ϵ as:

$$|F(x) - P(x)| \le \sum_{n=d+1}^{\infty} |c_n| =: \epsilon, \tag{19}$$

such that a QSP implementation of the channel $\mathcal{P}_A(\rho) = P(A)\rho P(A)^{\dagger}$ for some operator A deviates from the target channel $\mathcal{F}_A(\rho) = F(A)\rho F(A)^{\dagger}$ by an error

$$\| \mathcal{P}_A - \mathcal{F}_A \|_{\diamond} \le 2\epsilon = O(\epsilon), \tag{20}$$

while making d queries to the block-encoding of A.

Then there exists an ensemble of $\approx d/2$ polynomials $\{P_j(x)\}$ of degree $\deg(P_j) \le d$, and an associated probability distribution p_j such that:

$$\begin{aligned} |P_j(x) - F(x)| &\le 2\sqrt{\epsilon} \\ \left| \sum_{j=1}^{\infty} p_j P_j(x) - F(x) \right| &\le \epsilon, \end{aligned} \tag{21}$$

while the average degree of these polynomials is

$$\begin{aligned} d_{\text{avg}} &:= \sum_{j=1}^{\infty} p_j \deg(P_j) \\ &\le \frac{d}{2} + \frac{\log(C)}{2q} - \frac{\log(1 - e^{-q})}{2q} + \frac{1}{2} + \frac{1}{1 - e^{-q}} \\ &= \frac{d}{2} + O(1). \end{aligned} \tag{22}$$

Therefore, according to the mixing lemma for block-encodings (Lemma 2), the channel $\Lambda_A(\rho) = \sum_{j=1} p_j P_j(A)\rho P_j(A)^{\dagger}$ suffers error

$$\| \Lambda_A - \mathcal{F}_A \|_{\diamond} \le 6\epsilon = O(\epsilon), \tag{23}$$

while making $d_{\text{avg}} \le d/2 + O(1)$ queries to a block-encoding of A in expectation. The cost reduction realized by this channel is (here, $\lesssim$ neglects the terms independent of d and C in Eq. (22), which are less relevant than $\log(C)/2q$ in practice)

$$\frac{d_{\text{avg}}}{d} \lesssim \frac{1}{2} \left(1 + \frac{\log(C)}{qd} \right), \tag{24}$$

which approaches $1/2$ in the limit of large d.

We include the proof of this theorem in the Methods section. As a high-level overview, we construct the polynomial ensemble by selecting each polynomial to be the degree $\approx d/2$ truncation of the Chebyshev expansion of $F(x)$, plus a single higher-order term in the expansion up to degree d. Inclusion of the degree $d/2$ truncation guarantees that the first condition of the mixing lemma is satisfied with error $O(\sqrt{\epsilon})$, and sampling the higher order terms reproduces the degree d truncation in expectation, thus satisfying the second condition of the mixing lemma with error $O(\epsilon)$. Next, the probability distribution is chosen to be proportional to the coefficients of the Chebyshev expansion. Because these are assumed to decay exponentially, the corresponding probability mass is concentrated around $d/2$, and so the average degree equates to $d/2 + O(1)$. For visual intuition of this behavior, we provide an illustration of our stochastic QSP construction in Fig. 1.

Interpretation. Let us first take a minute to interpret this result. According to Theorem 3, stochastic QSP replaces a deterministic polynomial $P(x)$ with
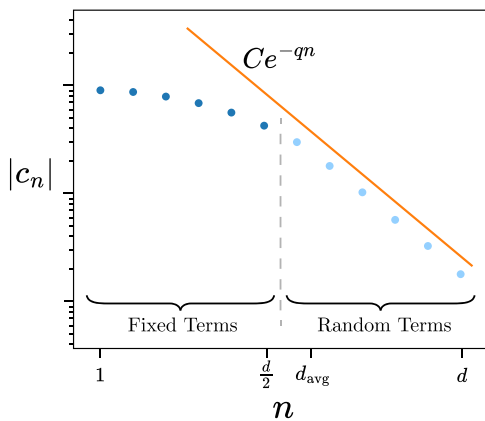
Fig. 1 | An illustration of our stochastic QSP construction. According to Theorem 3, we are provided a function whose Chebyshev coefficients c_n decay exponentially as $|c_n| \leq Ce^{-qn}$ for $n \geq d/2 + 1$. Then, by defining a cutoff degree $\approx d/2$ (see Eq. (36)), we can design an ensemble of polynomials that consists of all lower-order terms $n \leq d/2$, and a single higher-order term $d/2 < n \leq d$ that is randomly sampled (see Eq. (37)). The average degree of these polynomials is $d_{avg} = \frac{d}{2} + O(1)$.

an ensemble of polynomials $\{P_j(x)\}$ and probability distribution p_j, whose average evolution achieves the same precision up to a constant factor, but at asymptotically half the cost. Importantly, this result is agnostic to the specific polynomial, requiring only that its coefficients decay exponentially according to Eq. (18). As we showed in above, this condition is generally satisfied by polynomial approximations to smooth functions, rendering stochastic QSP applicable to a wide range of algorithms. In practice, the values of C and q in Eq. (18) can be chosen to minimize the ratio d_{avg}/d, which effectively means minimizing $\log(C)/q$.

The channel implemented by stochastic QSP is the probabilistic mixture of polynomials $\Lambda_A(\rho) = \sum_j p_j P_j(A)\rho P_j(A)$. As we discussed in the previous section, this channel may be realized by implementing an identical probabilistic mixture of unitaries $\{U_j\}$ that block-encode the polynomials $\{P_j(A)\}$, and post-selecting on successfully accessing these block-encodings. In practice, this can be achieved by independently sampling $j \sim p_j$, and implementing the QSP sequence that block-encodes the polynomial $P_j(A)$. Because $P_j(x)$ is only bounded as $\| P_j \|_{[-1,1]} \leq 1 + 2\sqrt{\epsilon}$ according to Eq. (21), this implementation may require rescaling the polynomials by $1 + 2\sqrt{\epsilon}$, which incurs a measurement overhead $\sim (1 + 2\sqrt{\epsilon})^2$ that asymptotically approaches 1.

This procedure of course requires knowledge of the QSP phases for each polynomial $P_j(A)$. These sets of phases can be determined classically using an efficient phase-finding algorithm, such as those of refs. 27–31, and the associated circuits can similarly be pre-compiled. Because the ensemble consists of $\approx d/2$ polynomials, each of degree at most d, the total storage requirement is $O(d^2)$ QSP phases. Consequently, stochastic QSP requires such a classical pre-computation step, with storage and computation cost poly(d), as in ordinary QSP.

While Theorem 3 introduces stochastic QSP specifically for polynomial approximations obtained from truncated Chebyshev series, it turns out that this result extends to more general polynomials. In Sec. IIF, we show how stochastic QSP applies to polynomials of definite and indefinite parity, Taylor series, trigonometric polynomials, generalized QSP[44], and approximations of the entire QSP unitary. These generalizations establish stochastic QSP as a versatile framework, applicable to the whole apparatus of QSP algorithms.

Lastly, it is important to note that while stochastic QSP reduces the expected cost to $d_{avg} \approx d/2 + O(1)$, it does not reduce the maximum degree of the polynomials implemented: some polynomials in the ensemble will have degree greater than d_{avg}, with one even having degree d. This however is unavoidable. In fact it is necessary for the ensemble to contain polynomials of degree $>d/2$ in order to attain a level of error equivalent to a degree-d

polynomial. To see this, observe that if all the polynomials in the ensemble had degree at most $k < d$, then their average $\sum_j p_j P_j(x)$ would also be a degree $k < d$ polynomial. This average polynomial would not be able to achieve a precision equivalent to that of a degree d polynomial, thus failing to meet the second condition of the mixing lemma. Note however that the distribution p_j concentrates around small values of j, meaning that degrees much larger than $\approx d/2$ are rare.

Through this interpretation, stochastic QSP is similar to the "semi-quantum matrix processing" algorithm of ref. 45 for estimating expectation values and matrix elements of a matrix-valued function. The authors achieve this by decomposing a degree-d polynomial approximation of this function into the Chebyshev basis, and sampling its constituent Chebsyhev polynomials, and measuring an estimator of the sampled polynomial that converges to the desired expectation value/matrix element. Like stochastic QSP, this procedure can reduce the expected cost of certain algorithms to a value $< d$ if the Chebyshev coefficients decay quickly, but it does not reduce the maximal degree because the order d Chebyshev polynomial can still be sampled. However, this method differs from stochastic QSP in two key ways. First, ref. 45 employs a quasi-probability technique where improvements in average degree are traded for additional variance in the measurements, whereas stochastic QSP always implements a normalized probability distribution. Second, the constructions in ref. 45 explicitly consider a target observable O and modify their circuit to accommodate a parity measurement $Z \otimes O$ with an ancilla register. Hence, their construction does not constitute an approximate realization of a desired matrix function, whereas stochastic QSP indeed approximates evolution under said matrix function, thus rendering our approach more modular.

Utility in quantum algorithms. Stochastic QSP can be integrated into larger quantum algorithms by replacing a deterministic QSP operation with the corresponding stochastic QSP channel. This remains true despite the stochastic QSP channel being incoherent, as it is a probabilistic mixture of operations.

As we prove in Theorem 3, the stochastic QSP channel closely approximates the target channel in the diamond norm, where the target channel is coherent. Therefore, the stochastic QSP channel can be substituted in for the target channel in a larger quantum algorithm, incurring only an $O(\epsilon)$ error. This is analogous to the use of ordinary QSP, which also approximates the target channel to within $O(\epsilon)$ error. The crucial advantage of stochastic QSP, however, is its efficiency: it achieves this approximation while requiring asymptotically half the number of queries as ordinary QSP.

This result holds even in more complex scenarios. For instance, if the target operation is controlled by an ancilla qubit (e.g., in the QSP-based phase estimation algorithm of ref. 19), then stochastic QSP can approximate the controlled target operation by an analogous mixture of controlled QSP operations. This result relies on an extension of the mixing lemma to controlled operations, which we prove in Supplementary Information Section II.

Similarly, stochastic QSP is fully compatible with algorithms that use the entire QSP unitary that block-encodes $P(A)$. We show in Supplementary Information Section III that the ensemble of unitaries used in stochastic QSP approximates not just the block-encoded operator $P(A)$, but also the full QSP unitary operator $U[P(A)]$ in diamond norm. Examples of such algorithms include the ground state energy estimation and ground state projection algorithms considered in ref. 5, as well as the use of amplitude amplification where the projector is constructed through a polynomial approximation $\Pi = P(A)$[20].

Generalizations of stochastic QSP

Above, we introduced stochastic QSP tailored specifically to polynomial approximations obtained from truncated Chebyshev expansions. However, it turns out that this specialization is not necessary, and that stochastic QSP is readily generalized to a much broader range of use cases. In this section, we show how our method applies to both definite- and indefinite-parity polynomials, Taylor series, trigonometric polynomials, generalized QSP[44],

and approximating the entire QSP unitary. Here, we present corollaries for these scenarios and defer their proofs to Supplementary Information Section III.

Definite and indefinite parity. The statement of Theorem 3 applies to functions $F(x)$ of indefinite parity (recall that a function is said to have definite parity if it is either even or odd; otherwise it has indefinite parity) and produces an ensemble of polynomials $\{P_j\}$ that also have indefinite parity. However, if $F(x)$ has definite parity, it turns out that this construction preserves the parity.

The parity of the implemented polynomials P_j is important because definite-parity polynomials admit simpler implementations via QSP. Indeed, by construction QSP can only produce polynomials of definite parity (see Eqs. (7) and (8)). In general, indefinite parity polynomials require using a linear combination of unitaries circuit, which demands an extra ancilla qubit and rescales the resulting block-encoding by 1/2. This rescaling can be undesirable in algorithm construction because it may necessitate amplitude amplification to be corrected, and can be avoided when the target function $F(x)$ has definite parity in the first place. We would like to retain this optimized performance in the context of stochastic QSP, which we can show is indeed the case.

Corollary 1. Consider the setting of Theorem 3, but also suppose that $F(x)$ has definite party. Then there exists an ensemble of polynomials $\{P_j\}$ with the same parity that satisfy the conditions of the theorem.

Note that this corollary also extends to complex-valued functions, which are usually approximated by QSP by decomposition into their real and imaginary components. Therefore, for an arbitrary target function, realizing stochastic QSP requires a circuit no more complicated than a QSP circuit that approximates the target function.

Taylor series. As we discussed in Section IIB, Chebyshev expansions of a smooth C^∞ functions admit exponentially-decaying coefficients, and thus yield polynomial approximations that meet the requisite conditions for stochastic QSP. Functions of interest in the quantum algorithms literature commonly exhibit this smoothness property (like $\cos(x)$ or $\exp(-\beta x)$), or are well-approximated by functions that do (like how the step function is approximated by $\mathrm{erf}(kx)$).

As such, we often desire a closed-form expression for the coefficients in the Chebyshev expansion, allowing us to give concrete guarantees for the values of C and q required by Theorem 3. However, for certain functions like $\sqrt{x}$ and $-x\ln(x)$, which are only smooth in certain domains, obtaining a closed-form expression for the Chebyshev coefficients is cumbersome, and the literature generally works with Taylor series instead (see for example Theorem 68 of ref. 20 and its applications). Fortunately, stochastic QSP directly generalizes to Taylor series, and expansions into bases of bounded polynomials more generally.

Corollary 2. Suppose $\{B_n(x)\}$ are a collection of basis functions of degree n respectively, which are all bounded as $\|B_n\|_{[-1,1]} \le 1$. Then the statement of Theorem 3 holds with $B_n(x)$ in place of $T_n(x)$.

Taylor series methods derive their accuracy from the analysis in Corollary 66 of ref. 20. The basic idea is the following. Suppose $F(x)$ is analytic and bounded on the interval $[-1, 1]$, and our goal is to approximate it by a polynomial on the interval $[-1+\delta, 1-\delta]$ for small $\delta < 1$. If $F(x) = \sum_{n=0}^{\infty} c_n x^n$ is the Taylor series of $F(x)$ with coefficients $|c_n| \le 1$, then the error from truncating at degree d is:

$$\sup_{|x| \le 1-\delta} \left| \sum_{n=d+1}^{\infty} c_n x^n \right| \le \sum_{n=d+1}^{\infty} (1-\delta)^n. \tag{25}$$

We immediately obtain an exponential decay in truncation error.

This is a slightly weaker condition than the one required for Corollary 2; we require an exponential bound on the coefficients themselves rather than on the truncation error. But if we are willing to approximate the stretched function $F((1-\delta)x)$ on the interval $[-1, 1]$ instead, then substituting the Taylor expansion yields

$$F((1-\delta)x) = \sum_{n=0}^{\infty} c_n (1-\delta)^n x^n \tag{26}$$

where the new coefficients $c_n(1-\delta)^n$ exhibit the desired exponential decay, thus rendering this Taylor series expansion amenable to stochastic QSP.

Trigonometric polynomials. Recent works[6,46] have considered a model of quantum computation in which a constant-size control register is strongly coupled to many qubits with an otherwise local connectivity graph. In such an architecture, controlled time evolution can be implemented through the Trotter approximation, but Hamiltonian simulation via QSP techniques remains out of reach due to the small size of the control register. Nonetheless QSP-like transformations of a Hamiltonian H can be implemented through applying QSP to a controlled time evolution operator.

A controlled time evolution operator is a block-encoding of e^{iHt}. Applying QSP to this block-encoding generates trigonometric polynomials $\sum_n c_n e^{inHt}$. If we select $t = \pi/\|H\|$ and let our variable of interest be $\theta = Ht = \pi H/\|H\|$, then we can approximate functions $F : [-\pi, \pi] \to [-1, 1]$ using degree-d trigonometric polynomials:

$$F(\theta) \approx \sum_n c_n e^{in\theta}. \tag{27}$$

Hence, by selecting $B_n(\theta) = e^{in\theta}$ for the basis functions, we see how Corollary 2 applies in this setting as well. Any method for constructing trigonometric polynomials can be used as long as the coefficients c_n decay exponentially. Conveniently, we show in Supplementary Information Section I that a C^∞ function $F(\theta)$ has a Fourier series with exponentially decaying coefficients. We see that, due to the relationship between Fourier expansions and Chebyshev expansions, stochastic QSP applies equally well to trigonometric polynomials as to regular polynomials.

Generalized QSP. Recently a technique was proposed ref. 44 for optimizing QSP implementation, specifically when the block-encoded operator U is unitary and is encoded via the controlled-U operation. In this situation the usual constraints on parity can be lifted when synthesizing complex polynomials, enabling polynomials of indefinite parity to be generated directly through QSP and avoiding rescaling from using an LCU circuit. For real polynomials it is possible to remove parity constraints by using Theorem 56 of ref. 20, but this introduces an undesirable factor of 1/2 as discussed earlier. Using the methods of ref. 44 this factor can be avoided.

By this reasoning, stochastic QSP is also compatible with the construction of ref. 47, which halves the asymptotic cost of QSP-based Hamiltonian simulation by using generalized QSP. This is achieved by designing a block encoding of both the quantum walk operator and its inverse, to which generalized QSP may be applied to approximate e^{-iHt} at roughly half the cost of ordinary QSP. Stochastic QSP could be applied on top of this algorithm to further reduce the cost by another factor of 1/2, equating to a total cost reduction of 1/4.

Approximating the QSP unitary. QSP methods typically project out a single polynomial from the QSP sequence, conventionally taken to be the polynomial $P(A)$ encoded in the $|0\rangle\langle0|$ block (see Eqs. (7) and (9)). Indeed stochastic QSP is concerned with approximating evolution under a channel by projecting out this component. However, in some situations[22,35] and in amplitude amplification, we are also interested in the other elements of the QSP sequence—particularly those that influence the measurements of the ancilla qubit(s) used to construct the block-encoding.

These more complex uses of QSP rely on the entire unitary $U[P(A)]$ that block-encodes the transformed operator $P(A)$. To show that stochastic QSP can be applied in these situations as well, we demonstrate that stochastic QSP also approximates the operator $U[P(A)]$ as a whole, rather than just the $P(A)$ sub-block.

Corollary 3. In the setting of Theorem 3, consider the ensemble of quantum circuits that implement stochastic QSP (i.e., the QSP circuits that implement $\{P_j(A)\}$), upon leaving the QSP ancilla qubit(s) unmeasured. Denoting this ensemble by $\{U_j\}$, the quantum channel $\rho \to \sum_j p_j U_j \rho U_j^\dagger$ approximates the map $\rho \to U[F(A)]\rho U[F(A)]^\dagger$ to error $O(\epsilon)$ in diamond norm.

Applications

Here we apply stochastic QSP to several common polynomials used in the quantum algorithms literature to assess its performance in practice. Our focus will be on polynomial approximations of smooth functions, such that Theorem 3 is applicable.

As we mentioned in Sec. IIB, includes polynomials that approximate functions with a finite number of singularities. This is achieved by excluding regions around the singularities, allowing the function to be replaced by a smooth interpolation that can then be approximated by a polynomial. As we show, this renders stochastic QSP applicable to commonly used discontinuous functions, like the step function $\Theta(x)$ and the inverse function $1/x$.

We study the following four polynomials:

1. The Jacobi-Anger expansion of cosine[Ref. 20, Lemma 57]:

$$\cos(tx) = J_0(t) + 2 \sum_{n=1}^{\infty} (-1)^n J_{2n}(t) T_{2n}(x), \qquad (28)$$

where the $J_{2n}(t)$ are the Bessel functions of the first kind. To achieve additive error at most ϵ, this series may be truncated at degree $d = O\left(|t| + \frac{\log(1/\epsilon)}{\log(e + \log(1/\epsilon)/|t|)}\right)$. This polynomial, in conjunction with the analogous expansion for $\sin(tx)$, furnishes an algorithm for Hamiltonian simulation with near-optimal query complexity[18].

2. The Jacobi-Anger expansion of an exponential decay [Ref. 34, Lemma 14]:

$$e^{-\beta(x+1)} = e^{-\beta}\left[I_0(\beta) + 2\sum_{n=1}^{\infty} I_n(\beta) T_b(-x)\right], \qquad (29)$$

where the $I_n(\beta)$ is the modified Bessel functions of the first kind. This expansion may be truncated at degree $d = O\left(\sqrt{(\beta + \log(1/\epsilon))\log(1/\epsilon)}\right) = O(\sqrt{\beta}\log(1/\epsilon))$ to achieve error at most ϵ. Naturally, the resulting polynomial is commonly used for imaginary time evolution[48,49], thermal state preparation[20], and partition function estimation[45].

3. A smooth approximation of $1/x$ in a domain away from the origin [Ref. 50, Lemma 14]:

$$\frac{1}{x} \approx \frac{1 - (1 - x^2)^b}{x}$$
$$= 4 \sum_{n=0}^{b-1} (-1)^n 2^{-2b} \sum_{m=n+1}^{b} \binom{2b}{b+m} T_{2n+1}(x) \qquad (30)$$

for an even integer $b \gg 1$. While this series is a degree $O(b)$ polynomial, its coefficients decay rapidly, such that it can be further truncated to degree $d = O\left(\sqrt{b \log(b/\epsilon)}\right)$ while guaranteeing error at most ϵ. The resulting polynomial is particularly useful for inverting matrices and thus solving linear systems of equations. If we take the (non-zero) eigenvalues of the matrix to be lower-bounded as $|\lambda| \geq \lambda_{\min}$, then in order to ensure that the

polynomial approximation behaves as $\approx 1/x$ over the range of eigenvalues, it suffices to choose chooses $b = O\left(\frac{1}{\lambda_{\min}^2}\log(1/\lambda_{\min}\epsilon)\right)$. This corresponds to a truncation degree $d = O(\kappa \log(\kappa/\epsilon))$, where $\kappa := 1/\lambda_{\min}$ is the condition number. For completeness, we note that algorithms with improved performance have very recently been discovered[33].

4. An approximation of erf(kx) obtained from integrating the Jacobi-Anger expansion of a Gaussian [Ref. 34, Corollary 1]:

$$\text{erf}(kx) = \frac{2ke^{-\frac{k^2}{2}}}{\sqrt{\pi}}\left[I_0\left(\frac{k^2}{2}\right)x \right.$$
$$\left. + \sum_{n=1}^{\infty} I_0\left(\frac{k^2}{2}\right)(-1)^n\left(\frac{T_{2n+1}(x)}{2n+1} - \frac{T_{2n-1}(x)}{2n-1}\right)\right]. \qquad (31)$$

To achieve error ϵ, it suffices to truncate this series at degree $d = O\left(\sqrt{(k^2 + \log(1/\epsilon))\log(1/\epsilon)}\right) = O(k\log(1/\epsilon))$. In practice, this polynomial is used to approximate the step function $\Theta(x)$ by selecting $k \gg 1$. Notable applications of this approximation include the search problem[19], phase estimation[19,22] and ground state preparation[6,35].

All four of these functions feature a *cost parameter*, namely t, β, b, and k respectively, whose value determines the truncation degree necessary to achieve an accurate approximation.

We apply stochastic QSP to these polynomials, and illustrate the cost reduction ratio d_{avg}/d as a function of d in Fig. 2. We rely on the following procedure to compute d_{avg}. First, we select an integer n_1, and then numerically determine values of C, q such that $c_n \leq Ce^{-qn}$ holds for $n \geq n_1$. This is achieved by selecting another integer $n_2 > n_1$ and computing C, q such that Ce^{-qn} goes through the points (n_1, c_{n_1}) and (n_2, c_{n_2}). In doing so, we choose n_1, n_2 to guarantee $c_n \leq Ce^{-qn}$ indeed holds for all $n \geq n_1$, and also to minimize $\log(C)/q$ so as to heuristically reduce the dominant term in the bound on d_{avg} in Eq. (22). We also select n_1, n_2 independent of the degree d; we find that n_1 naturally converges to the degree at which $|c_n|$ starts to decay exponentially. For $\cos(tx)$ and $\text{erf}(kx)$ we find that this regime sets in later as the respective cost parameter increases. Lastly, for each d, the cutoff degree d^* is computed from Eq. (36) (see "Methods"). Then d_{avg} is obtained by explicitly computing the probabilities p_j from Eq. (38) (see "Methods") and calculating the corresponding weighted average of degrees.

In Fig. 2 we observe the desired phenomenon: the cost reduction ratio d_{avg}/d approaches 1/2 as d increases, with a discrepancy scaling as $O(1/d)$. Additionally, as the cost parameter increases, the magnitude of the error terms $\sim \log(C)/qd$ can also increase, resulting in a later approach to 1/2. This is expected because a larger cost parameter requires a larger degree to maintain the same level of approximation. These results indicate that indeed, stochastic QSP reduces the query complexity of QSP by approximately 1/2 in regimes of practical interest.

A more surprising phenomenon is that for some functions and values of the cost parameter, d_{avg}/d approaches 1/2 *from below*, resulting in improved performance for small d. This arises because we determine d_{avg} by choosing values C, q so as to minimize the deviation $\log(C)/q$. In some cases, this can result in a value $C < 1$, or equivalently $\log(C) < 0$, which causes the ratio d_{avg}/d to deviate from 1/2 by a negative value as per Eq. (22). Moreover, the sawtooth behavior observed for each function in Fig. 2 is explained by the presence of the ceiling function in the definition of d^* in Eq. (36). When d is small then the effect of this rounding is more pronounced.

Lastly, we emphasize that the cost reduction realized by stochastic QSP hinges on the $\log(1/\epsilon)$ term in the degree of the polynomial approximation. In scenarios where the degree scales multiplicatively with $\log(1/\epsilon)$ (e.g., approximations to $e^{-\beta x}$, $\text{erf}(kx)$, and $1/x$), the cost reduction quickly approaches 1/2. Conversely, in cases where the degree scales additively in $\log(1/\epsilon)$ (e.g., approximations to $\cos(tx)$ and $\sin(tx)$), the cost reduction only approaches 1/2 when the $\log(1/\epsilon)$ term dominates. For instance, in Hamiltonian simulation, the $\log(1/\epsilon)$ term may not dominate in the large t limit, and consequently the practical cost reduction will instead lie between 1 and 1/2.
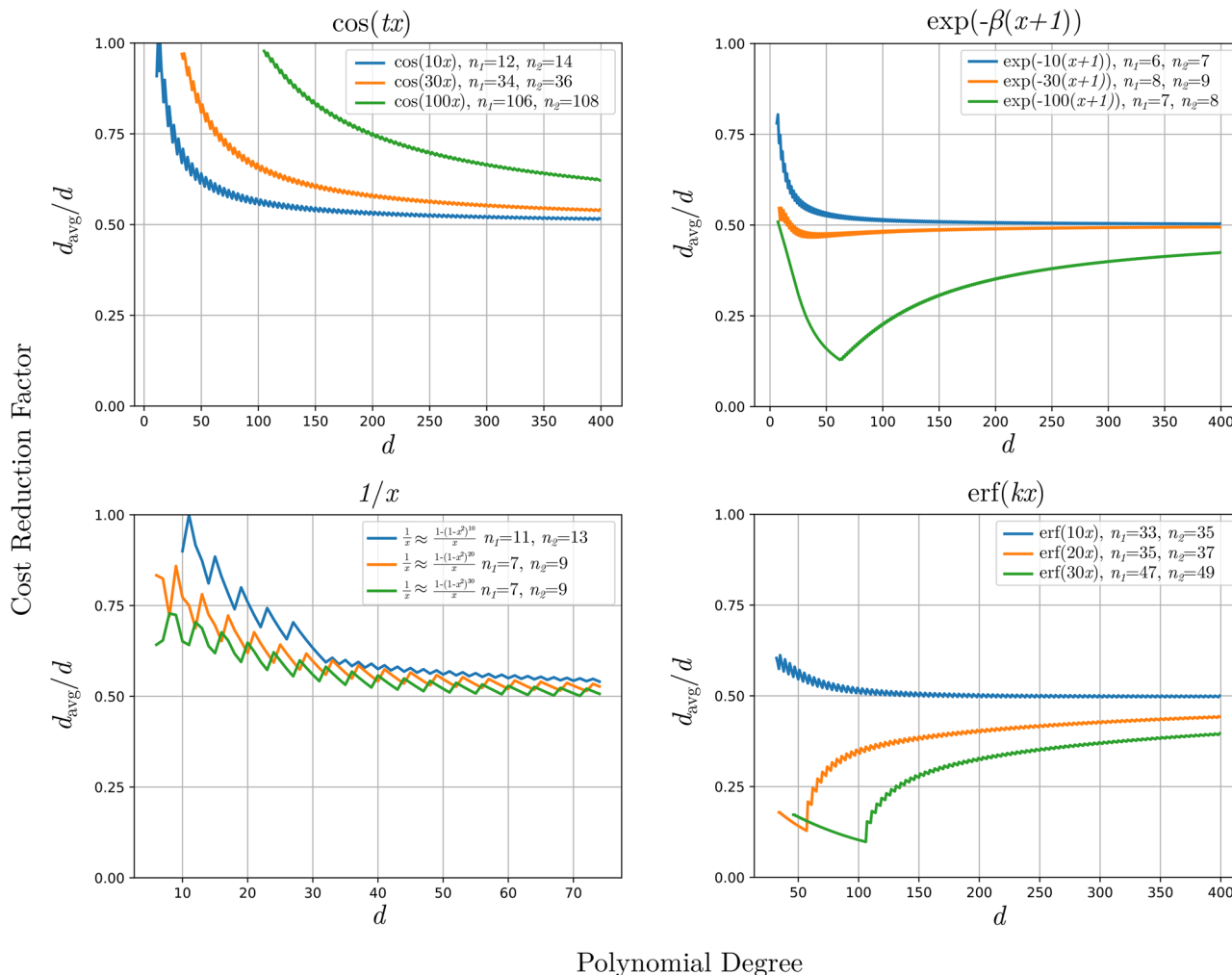
Fig. 2 | Average reduction in polynomial degree and hence circuit depth for some common polynomials in the literature. d_{avg} is computed directly from the probabilities p_i in Eq. (38), *not* from the bound in Eq. (45). The cutoff degree d^* is selected according to Eq. (36), which depends on a choice of C and q obtained from n_1, n_2—see the explanation in the main text.

Discussion

By merging quantum signal processing and randomized compiling, we have developed stochastic QSP. As we showed, stochastic QSP enables us to lift a QSP algorithm to its randomized counterpart, and simultaneously shrink the circuit complexity by a factor of 2. We empirically verified this cost reduction for various algorithms of interest, including real/imaginary time evolution, matrix inversion, phase estimation, and ground state preparation. This reduction can also be interpreted as enabling a cost parameter in the underlying function to increase (e.g., a longer time t in Hamiltonian simulation) without changing the query complexity. In aggregate, this result demonstrates that classical randomness is a useful resource for quantum computing, and can help bring quantum algorithms closer to the near-term.

Moreover, in this work we did not consider noisy gates, but rather assumed the ability to perform QSP exactly. As such, we leveraged randomized compiling to suppress error in QSP polynomial approximations to functions. Nonetheless, as randomized compiling can also suppress noise in erroneous gates[8], this suggests that a practical implementation of stochastic QSP could benefit from also randomizing over the gate set, as a sort of doubly-stochastic channel. Along these lines, it would be interesting to study the requirements and conditions for implementing stochastic QSP on near-term quantum hardware.

The performance improvement realized through randomized compiling suggests further uses of this technique in quantum information.

While here we have applied randomized compiling to quantum algorithms via QSP, it is likely that randomized compiling can confer a similar advantage to the traditional constructions of quantum algorithms (e.g., Grover search[51], or conventional phase estimation via the quantum Fourier transform[52]). Further, it would be interesting to search for problems for which randomized compiling (or a variant thereof) confers a super-quadratic suppression of error, translating to a cost reduction by a factor smaller than 1/2 in our context. With an eye toward applications, the mixing lemma could also be used to better understand and generate random unitaries and unitary designs[53–56], and perhaps even be integrated with randomized measurements[4] to improve near-term protocols for studying quantum systems. Likewise, there is an absence of randomized compiling in classical simulation algorithms, which could admit similar improvements for problems aimed to emulate quantum mechanics. Given the ubiquity of random processes in the physical world, it is only natural that we can harness randomness to gain deeper insights into quantum systems.

Note: After posting this work, we were made aware of a recent paper[57] that also mixes over polynomials to suppress error. Despite this similarity, our work makes novel contributions: we provide (1) an analytic construction of the ensemble of polynomials and associated probability distribution, and (2) a rigorous argument for the cost reduction by a factor of 1/2. These key contributions establish stochastic QSP as a versatile framework with strong performance guarantees.

Methods
Proof of Theorem 3

Proof. The underlying goal of Theorem 3 is to simulate evolution under $F(A)$ according to the channel $\mathcal{F}_A(\rho) = F(A)\rho F(A)^\dagger$. For notational convenience, let us define the degree t polynomial truncation of the Chebyshev series of $F(x)$ as

$$P^{[t]}(x) := \sum_{n=0}^{t} c_n T_n(x). \tag{32}$$

By the assumption that $|c_n| \le Ce^{-qn}$ for $n \ge d/2$, we find that for $t \ge d/2$, this polynomial truncation suffers an error over $x \in [-1, 1]$:

$$
\begin{aligned}
|P^{[t]}(x) - F(x)| &= \left| \sum_{n=t+1}^{\infty} c_n T_n(x) \right| \\
&\le \sum_{n=t+1}^{\infty} |c_n| \le \sum_{n=t+1}^{\infty} Ce^{-qn} = \frac{Ce^{-qt}}{1 - e^{-q}}.
\end{aligned} \tag{33}
$$

Therefore, the error suffered by the degree d truncation $P^{[d]}(x) = P(x)$ is at most

$$|P^{[d]}(x) - F(x)| \le \frac{Ce^{-qd}}{1 - e^{-q}} =: \epsilon. \tag{34}$$

To estimate evolution under $\mathcal{F}_A$, one can use QSP to implement the channel $\mathcal{P}_A(\rho) = P^{[d]}(A)\rho P^{[d]}(A)^\dagger$. Using the diamond norm bound of Eq. (5), we see that $\mathcal{P}_A$ suffers error

$$
\begin{aligned}
\| \mathcal{P}_A - \mathcal{F}_A \|_\diamond &\le 2 \| P^{[d]}(A) - F(A) \| \\
&\le 2 \max_{x \in [-1,1]} |P^{[d]}(x) - F(x)| \\
&\le 2\epsilon.
\end{aligned} \tag{35}
$$

As $P^{[d]}(A)$ is a degree d polynomial, the cost of implementing $\mathcal{P}_A$ is d queries to the block-encoding of A.

Our goal is to reproduce evolution under $\mathcal{F}_A$ to diamond norm error $O(\epsilon)$ by using a probabilistic mixture of QSP polynomials and invoking the mixing lemma. We construct these polynomials by truncating the original series at a cutoff degree d^*, selected as follows. Because the mixing lemma quadratically suppresses error, and the degree scales as $\log(1/\epsilon)$, there should exist an ensemble of polynomials of degree around $d^* \approx d/2$ whose corresponding channel suffers the same error as the original degree d polynomial. We can then readily determine d^* by demanding that it be the smallest integer suffering error at most $\sqrt{\epsilon}$:

$$
\left(\frac{Ce^{-qd^*}}{1 - e^{-q}} \right)^2 \le \epsilon = \frac{Ce^{-qd}}{1 - e^{-q}} \quad \Rightarrow
$$
$$
d^* = \left\lceil \frac{d}{2} + \frac{\log(C)}{2q} - \frac{\log(1 - e^{-q})}{2q} \right\rceil = \frac{d}{2} + O(1). \tag{36}
$$

Because the error suffered by a single polynomial obtained by truncating at d^* is greater than that at d, we have $d^* < d$.

Next, let us define our ensemble of polynomials as

$$
P_j(x) = \begin{cases} P^{[d^*]}(x) + \frac{c_{d^*+j}}{p_j} T_{d^*+j}(x) & |c_{d^*+j}| > 0 \\ 0 & c_{d^*+j} = 0 \end{cases} \tag{37}
$$

for $j = 1, 2, \ldots, d - d^*$, where p_j is the associated probability distribution defined as

$$p_j = \frac{|c_{d^*+j}|}{\sum_{k=1}^{d-d^*} |c_{d^*+k}|}. \tag{38}$$

Intuitively, each polynomial $P_j(x)$ consists of the degree d^* truncation $P^{[d^*]}(x)$, and an additional higher-order Chebyshev polynomial chosen such that the average polynomial is the degree d truncation:

$$\sum_{j=1}^{d-d^*} p_j P_j(x) = P^{[d]}(x). \tag{39}$$

The distribution p_j is chosen such that terms with larger Chebyshev coefficients are given more probability mass and are preferentially sampled. Each polynomial in this ensemble is guaranteed to suffer error

$$
\begin{aligned}
|P_j(x) - F(x)| &\le |P^{[d^*]}(x) - F(x)| + \frac{|c_{d^*+j}|}{p_j} \\
&\le \sum_{n=d^*+1}^{\infty} |c_n| + \sum_{k=1}^{d-d^*} |c_{d^*+k}| \\
&\le 2 \sum_{n=d^*+1}^{\infty} |c_n| \le 2 \sum_{n=d^*+1}^{\infty} Ce^{-qn} \\
&= \frac{2Ce^{-qd^*}}{1 - e^{-q}} \le 2\sqrt{\epsilon} = O(\sqrt{\epsilon}).
\end{aligned} \tag{40}
$$

This bound also implies that $P_j(x)$ is bounded as $\| P_j \|_{[-1,1]} \le 1 + 2\sqrt{\epsilon}$. On the other hand, the average polynomial suffers error

$$\left| \sum_{j=1}^{d-d^*} p_j P_j(x) - F(x) \right| = |P^{[d]}(x) - F(x)| \le \epsilon. \tag{41}$$

In the language of the mixing lemma for block-encodings (Lemma 2), this corresponds to values $a = 2\sqrt{\epsilon}$ and $b = \epsilon$. Accordingly, the channel $\Lambda_A(\rho) = \sum_{j=1}^{d-d^*} p_j P_j(A)\rho P_j(A)^\dagger$ suffers error

$$\| \Lambda_A - \mathcal{F}_A \|_\diamond \le a^2 + 2b = 6\epsilon = O(\epsilon). \tag{42}$$

Lastly, the expected cost of instantiating $\Lambda_A(\rho)$ is the average degree $d_{\text{avg}} = \sum_{j=1}^{d-d^*} p_j \deg(P_j)$, which corresponds to the average number of queries to the block-encoding. Note that the degrees of these polynomials are $\deg(P_j) = d^* + j \le d$. To evaluate the average degree, recall that this theorem assumes that

$$|c_n| \le Ce^{-qn} \text{ for all } n \ge d/2, \tag{43}$$

This implies that the mean of the distribution $p_j := \frac{|c_{d^*+j}|}{\sum_{k=1}^{d-d^*} |c_{d^*+k}|}$ is upper-bounded by the mean of the geometric distribution $\tilde{p}_j := \frac{e^{-q(d^*+j)}}{\sum_{k=1}^{d-d^*} e^{-q(d^*+k)}} = \frac{e^{-qj}}{\sum_{k=1}^{d-d^*} e^{-qk}}$. Hence, we may upper bound d_{avg} as

$$
\begin{aligned}
d_{\text{avg}} &= \sum_{j=1}^{d-d^*} p_j \deg(P_j) = d^* + \sum_{j=1}^{d-d^*} j p_j \\
&\le d^* + \sum_{j=1}^{d-d^*} j \tilde{p}_j \\
&= d^* - (d - d^*) \frac{e^{-q(d-d^*)}}{1 - e^{-q(d-d^*)}} + \frac{1}{1 - e^{-q}} \\
&\le d^* + \frac{1}{1 - e^{-q}} \\
&= \left\lceil \frac{d}{2} + \frac{\log(C)}{2q} - \frac{\log(1 - e^{-q})}{2q} \right\rceil + \frac{1}{1 - e^{-q}} \\
&\le \frac{d}{2} + \frac{\log(C)}{2q} - \frac{\log(1 - e^{-q})}{2q} + \frac{1}{2} + \frac{1}{1 - e^{-q}} \\
&= \frac{d}{2} + O(1)
\end{aligned} \tag{44}
$$

where line 3 follows from evaluating mean of the geometric distribution (i.e., evaluating a geometric series), and line 4 from $d^* < d$. Accordingly, the cost

reduction realized by Λ_A is the ratio

$$
\begin{aligned}
\frac{d_{\mathrm{avg}}}{d} &\leq \frac{1}{2}\left(1 + \frac{\log(C)}{qd} - \frac{\log(1 - e^{-q})}{qd} + \frac{1}{d} + \frac{2}{(1 - e^{-q})d}\right) \\
&\approx \frac{1}{2}\left(1 + \frac{\log(C)}{qd}\right)
\end{aligned}
\tag{45}
$$

where the last line follows from the fact that the $\log(C)$ contribution dominates in practice. In the large d limit, the cost reduction approaches 1/2 inverse-polynomially fast.

Lastly, as this construction makes no reference to the eigenvalues or singular values of A, stochastic QSP applies equally as well to QET and QSVT, where $P(A)$ acts on the eigenvalues or singular vectors of A, respectively. In addition, while the presentation of stochastic QSP here is tailored toward functions expressed in the basis of Chebyshev polynomials, we extend this result to more general functions and arbitrary arbitrary bases in Section IIF. $\square$

Observe from Eq. (37) that each polynomial in the ensemble is constructed as the degree $d^* \approx d/2$ truncation plus a higher order term in the Chebyshev expansion, up to degree d. The degree d^* truncation guarantees that the first condition of the mixing lemma is satisfied with error $O(\sqrt{\epsilon})$, and sampling the higher order terms ensures that the second condition of the mixing lemma is satisfied with error $O(\epsilon)$. The specific sampling distribution is chosen to be proportional to the coefficients of the Chebyshev expansion, which decay exponentially and lead to an average degree $d/2 + O(1)$. Through this view, the ensemble of stochastic QSP can be seen as a fixed low-order term plus higher-order correction terms, which are randomly sampled according to the QDrift protocol[11] to leverage randomized compiling. An ensemble of a similar structure, albeit in the context of Trotterization, was recently used in ref. 14 to double the order Trotter formulae through randomized compiling.

Data availability
The data in Fig. 2 are available by request from the authors.

Code availability
The code used to generate Fig. 2 is available by request from the authors.

References
1. Knill, E. et al. Randomized benchmarking of quantum gates. *Phys. Rev. A* **77**, 012307 (2008).
2. van den Berg, E., Minev, Z. K., Kandala, A. & Temme, K. Probabilistic error cancellation with sparse Pauli-Lindblad models on noisy quantum processors. *Nat. Phys.* **19**, 1116–1121 (2023).
3. Arute, F. et al. Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505–510 (2019).
4. Elben, A. et al. The randomized measurement toolbox. *Nat. Rev. Phys.* **5**, 9–24 (2022).
5. Lin, L. & Tong, Y. Optimal polynomial-based quantum eigenstate filtering with application to solving quantum linear systems. *Quantum* **4**, 361 (2020).
6. Dong, Y., Lin, L. & Tong, Y. Ground-state preparation and energy estimation on early fault-tolerant quantum computers via quantum eigenvalue transformation of unitary matrices. *PRX Quantum* **3**, 040305 (2022).
7. Lin, L. & Tong, Y. Heisenberg-limited ground-state energy estimation for early fault-tolerant quantum computers. *PRX Quantum* **3**, 010318 (2022).
8. Wallman, J. J. & Emerson, J. Noise tailoring for scalable quantum computation via randomized compiling. *Phys. Rev. A* **94**, 052325 (2016).
9. Campbell, E. Shorter gate sequences for quantum computing by mixing unitaries. *Phys. Rev. A* **95**, 042306 (2017).
10. Hastings, M. B. Turning gate synthesis errors into incoherent errors, http://arxiv.org/abs/1612.01011 arXiv:1612.01011 [quant-ph] (2016).
11. Campbell, E. Random compiler for fast Hamiltonian simulation. *Phys. Rev. Lett.* **123**, 070503 (2019).
12. Childs, A. M., Ostrander, A. & Su, Y. Faster quantum simulation by randomization. *Quantum* **3**, 182 (2019).
13. Ouyang, Y., White, D. R. & Campbell, E. T. Compilation by stochastic Hamiltonian sparsification. *Quantum* **4**, 235 (2020).
14. Cho, C.-H., Berry, D. W. & Hsieh, M.-H. Doubling the order of approximation via the randomized product formula. *Phys. Rev. A* **109**, 062431 (2024).
15. Nakaji, K., Bagherimehrab, M. & Aspuru-Guzik, A. High-order randomized compiler for Hamiltonian simulation. *PRX Quantum* **5**, 020330 (2024).
16. Wan, K., Berta, M. & Campbell, E. T. Randomized quantum algorithm for statistical phase estimation. *Phys. Rev. Lett* **129**, 030503 (2022).
17. Low, G. H., Yoder, T. J. & Chuang, I. L. Methodology of resonant equiangular composite quantum gates. *Phys. Rev. X* **6**, 041067 (2016).
18. Low, G. H. & Chuang, I. L. Optimal Hamiltonian simulation by quantum signal processing. *Phys. Rev. Lett.* **118**, 010501 (2017).
19. Martyn, J. M., Rossi, Z. M., Tan, A. K. & Chuang, I. L. Grand unification of quantum algorithms. *PRX Quantum* **2**, 040203 (2021).
20. Gilyén, A., Su, Y., Low, G.H. & Wiebe, N., Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proc. 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC '19. https://doi.org/10.1145/3313276.3316366 (ACM, 2019).
21. Berry, D. W. A random approach to quantum simulation. *Physics* **12**, 91 (2019).
22. Rall, P. Faster coherent quantum algorithms for phase, energy, and amplitude estimation. *Quantum* **5**, 566 (2021).
23. Low, GuangHao & Chuang, I. L. Hamiltonian simulation by qubitization. *Quantum* **3**, 163 (2019).
24. Berry, D. W., Childs, A. M., Cleve, R., Kothari, R. & Somma, R. D. Exponential improvement in precision for simulating sparse Hamiltonians. In: *Proc. 46th Annual ACM Symposium on Theory of Computing* (IEEE, 2014).
25. Berry, D. W., Childs, A. M., Cleve, R., Kothari, R. & Somma, R. D. Simulating Hamiltonian dynamics with a truncated Taylor series. *Phys. Rev. Lett.* **114**, 090502 (2015).
26. Berry, D. W., Childs, A. M. & Kothari, R., Hamiltonian simulation with nearly optimal dependence on all parameters. In: *Proc. IEEE 56th Annual Symposium on Foundations of Computer Science* (IEEE, 2015).
27. Haah, J. Product decomposition of periodic functions in quantum signal processing. *Quantum* **3**, 190 (2019).
28. Chao, R., Ding, D., Gilyen, A., Huang, C. & Szegedy, M. Finding angles for quantum signal processing with machine precision, http://arxiv.org/abs/2003.02831 arXiv:2003.02831 [quant-ph] (2020).
29. Dong, Y., Meng, X., Whaley, K. B. & Lin, L. Efficient phase-factor evaluation in quantum signal processing. *Phys. Rev. A* **103**, 042419 (2021).
30. Ying, L. Stable factorization for phase factors of quantum signal processing. *Quantum* **6**, 842 (2022).
31. Yamamoto, S. & Yoshioka, N. Robust angle finding for generalized quantum signal processing, http://arxiv.org/abs/2402.03016 arXiv:2402.03016 [quant-ph] (2024).
32. Tong, Y., An, D., Wiebe, N. & Lin, L. Fast inversion, preconditioned quantum linear system solvers, fast Green's-function computation, and fast evaluation of matrix functions. *Phys. Rev. A* **104**, 032422 (2021).
33. Dalzell, A. M. A shortcut to an optimal quantum linear system solver, http://arxiv.org/abs/2406.12086 arXiv:2406.12086 [quant-ph] (2024).
34. Low, G. H. & Chuang, I. L. Hamiltonian simulation by uniform spectral amplification, http://arxiv.org/abs/1707.05391 arXiv:1707.05391 [quant-ph] (2017).

35. Lin, L. & Tong, Y. Near-optimal ground state preparation. *Quantum* **4**, 372 (2020).

36. Boyd, J. P. *Chebyshev and Fourier Spectral Methods* (Courier Corporation, 2001).

37. Low, G. H. Halving the cost of quantum multiplexed rotations, http://arxiv.org/abs/2110.13439 arXiv:2110.13439 [quant-ph] (2021).

38. Akibue, S., Kato, G. & Tani, S. Probabilistic unitary synthesis with optimal accuracy. *ACM Trans. Quantum Comput.* **5**, 1–27 (2024).

39. Yoshioka, N., Akibue, S., Morisaki, H., Tsubouchi, K. & Suzuki, Y. Error crafting in probabilistic quantum gate synthesis, http://arxiv.org/abs/2405.15565 arXiv:2405.15565 [quant-ph] (2024).

40. Akibue, S., Kato, G. & Tani, S. Quadratic improvement on accuracy of approximating pure quantum states and unitary gates by probabilistic implementation, http://arxiv.org/abs/2111.05531 arXiv:2111.05531 [quant-ph] (2022).

41. Akibue, S., Kato, G. & Tani, S. Probabilistic state synthesis based on optimal convex approximation. *npj Quantum Inf.* **10**, 3 (2024).

42. Huang, Y., Shao, Y., Ren, W., Sun, J. & Lv, D. Efficient quantum imaginary time evolution by drifting real-time evolution: an approach with low gate and measurement complexity. *J. Chem. Theory Comput.* **19**, 3868–3876 (2023).

43. Pocrnic, M., Hagan, M., Carrasquilla, J., Segal, D. & Wiebe, N. Composite qdrift-product formulas for quantum and classical simulations in real and imaginary time. *Phys. Rev. Res.* **6**, 013224 (2024).

44. Motlagh, D. & Wiebe, N. Generalized quantum signal processing. *PRX Quantum* **5**, 020368 (2024).

45. Tosta, A., de Lima Silva, T., Camilo, G. & Aolita, L. Randomized semi-quantum matrix processing. *npj Quantum Inf.* **10**, 93 (2024).

46. Katabarwa, A., Gratsea, K., Caesura, A. & Johnson, P. D. Early fault-tolerant quantum computing. *PRX Quantum* **5**, 020101 (2024).

47. Berry, D. W., Motlagh, D., Pantaleoni, G. & Wiebe, N. Doubling the efficiency of Hamiltonian simulation via generalized quantum signal processing. *Phys. Rev. A* **110**, 012612 (2024).

48. Silva, Thais L., Taddei, M. M., Carrazza, S. & Aolita, L. Fragmented imaginary-time evolution for early-stage quantum signal processors. *Sci. Rep.* **13**, 18258 (2023).

49. Chan, H. H. S., Ramo, D. M. and Fitzpatrick, N. Simulating non-unitary dynamics using quantum signal processing with unitary block encoding, http://arxiv.org/abs/2303.06161 arXiv:2303.06161 [quant-ph] (2023).

50. Childs, A. M., Kothari, R. & Somma, R. D. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM J. Comput.* **46**, 1920–1950 (2017).

51. Grover, L. K. A fast quantum mechanical algorithm for database search. In: *Proc. twenty-eighth annual ACM symposium on theory of computing* https://doi.org/10.1145/237814.237866. 212–219 (1996).

52. Kitaev, A. Yu. Quantum measurements and the abelian stabilizer problem, http://arxiv.org/abs/quant-ph/9511026 arXiv:quant-ph/9511026 [quant-ph] (1995).

53. Gross, D., Audenaert, K. & Eisert, J. Evenly distributed unitaries: on the structure of unitary designs. *J. Math. Phys.* **48**, 052104 (2007).

54. Fisher, P. A. M., Khemani, V., Nahum, A., Vijay, S. & Random quantum circuits. *Annu. Rev. Condens. Matter Phys.* **14**, 335–379 (2023).

55. Harrow, A. W. & Low, R. A. Random quantum circuits are approximate 2-designs. *Commun. Math. Phys.* **291**, 257–302 (2009).

56. Brandão, FernandoG. S. L., Harrow, A. W. & Horodecki, Michał. Local random quantum circuits are approximate polynomial designs. *Commun. Math. Phys.* **346**, 397–434 (2016).

57. Wang, Y. & Zhao, Q. Faster quantum algorithms with "fractional"-truncated series, http://arxiv.org/abs/2402.05595 arXiv:2402.05595 [quant-ph] (2024).

Competing interests
The authors have no competing interests.

Additional information
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41534-025-01003-2.

Correspondence and requests for materials should be addressed to John M. Martyn.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.