Faster variational quantum algorithms with quantum kernel-based surrogate models

Quantum Science and Technology

CrossMark

PAPER

Faster variational quantum algorithms with quantum kernel-based surrogate models

Alistair W R Smith[*] , A J Paige and M S Kim

QOLS, Blackett Laboratory, Imperial College London, London SW7 2AZ, United Kingdom
[*] Author to whom any correspondence should be addressed.

E-mail: alistair.smith18@imperial.ac.uk

Abstract

We present a new optimization strategy for small-to-intermediate scale variational quantum algorithms (VQAs) on noisy near-term quantum processors which uses a Gaussian process surrogate model equipped with a classically-evaluated quantum kernel. VQAs are typically optimized using gradient-based approaches however these are difficult to implement on current noisy devices, requiring large numbers of objective function evaluations. Our approach shifts this computational burden onto the classical optimizer component of these hybrid algorithms, greatly reducing the number of quantum circuit evaluations required from the quantum processor. We focus on the variational quantum eigensolver (VQE) algorithm and demonstrate numerically that these surrogate models are particularly well suited to the algorithm's objective function. Next, we apply these models to both noiseless and noisy VQE simulations and show that they exhibit better performance than widely-used classical kernels in terms of final accuracy and convergence speed. Compared to the typically-used stochastic gradient-descent approach to VQAs, our quantum kernel-based approach is found to consistently achieve significantly higher accuracy while requiring less than an order of magnitude fewer quantum circuit executions. We analyze the performance of the quantum kernel-based models in terms of the kernels' induced feature spaces and explicitly construct their feature maps. Finally, we describe a scheme for approximating the best-performing quantum kernel using a classically-efficient tensor network representation of its input state and so provide a pathway for scaling this strategy to larger systems.

1. Introduction

Quantum computation (QC) promises an alternative computational framework for solving an array of classically intractable problems. Developments in quantum error correction and the engineering of low-noise qubits have allowed recent experiments to push into the domain of fault tolerant single and two qubit operation [1–5]. These results present us with a roadmap of how scalable fault-tolerant universal QC may be achieved however the exact timescale for this remains uncertain [6–8]. Until the threshold for fault tolerance is achieved the kinds of algorithms that can be run on-device with appreciable fidelity are limited to circuits of low depth on relatively few qubits.

These noisy intermediate scale quantum (NISQ [6]) algorithms provide us with a useful set of benchmarking tools for quantifying the performance of prototype quantum computers while being an interesting field of study in their own right [9]. These include algorithms specifically designed to showcase near-term forms of quantum advantage [10–12]. While experimental demonstrations of these 'quantum supremacy' tasks are an important step forward in demonstrating the potential of quantum computers, they are not designed to solve useful problems and the extent of their advantage over classical methods is an area of ongoing study [13–15]. More modest proposals for interesting NISQ applications include variational quantum algorithms (VQAs) and, the often related, quantum machine learning (QML) [9, 16, 17].

QML attempts to generalize the methods and results used in classical machine learning to a quantum setting. Quantum kernel methods are a promising NISQ-friendly sub-field of QML focused around the generalization of classical kernel-based algorithms. Generally speaking, kernel methods are a broad class of machine learning techniques which use kernel functions to quantify the similarity between data [18, 19]. A kernel function is a positive (semi-)definite function which quantifies the similarity between its two inputs; it is equivalent to an inner-product between these inputs mapped into a higher-dimensional feature space. Quantum analogues to classical kernel methods have primarily been applied to the former set of problems, particularly classification tasks [20–23], while applications to the latter have been largely unstudied in the literature [24].

VQAs are hybrid quantum–classical algorithms which couple the limited computational capabilities of NISQ devices with classical computational resources in an attempt to solve useful tasks [16]. Typically, a quantum state is prepared with a parameterized ansatz circuit. The parameters of the ansatz are adjusted by a classical optimizer to minimize a cost function calculated from measurements of the state. The ease of implementing low-depth ansatzes and VQAs' resilience against certain types of coherent noise [25] make these algorithms a popular choice for demonstrations on near-term devices [26–28].

While it has been suggested that scaled-up VQAs could provide a route to quantum advantage in the near-term [16, 29, 30], large hurdles exist to achieving this. In particular, barren plateaus can emerge in the optimization landscape due to device noise [31] or the use of overly expressive many-qubit ansatz circuits [32–34]. These make the optimization in VQAs exponentially expensive preventing them from being scaled to large system sizes. Any direct quantum advantage from large-scale VQE would likely require highly problem-specific ansatzes to avoid these barren plateaus and designing such ansatz circuits is an area of ongoing research [35, 36]. Despite these hurdles to scalability, VQAs algorithms have emerged as a popular tool for benchmarking current noisy devices.

There are several key factors to consider when implementing an VQA on an NISQ device. The ansatz circuit must be capable of representing a state which solves the task at hand (to a good approximation) but should not be so expressive that training becomes infeasible. Device noise must also be considered when choosing the ansatz's depth but is typically difficult to fully characterize. Large amounts of noise can deform an VQA's cost function to the extent that the obtained solution does not solve the original task while compounding trainability issues [31]. This imposes limitations on the depth of ansatz circuits which can be used and has prompted the development of a plethora of noise mitigation strategies [37–42]. Finally, a classical optimization strategy must be chosen which makes the best use of available data while taking into consideration factors such as noise and its effect on sampling costs.

In this work we combine concepts from both VQAs and quantum kernel theory to demonstrate how a surrogate model based on classically-evaluated quantum kernels allows noisy VQAs to be solved quickly and with high accuracy. We focus on the variational quantum eigensolver (VQE) algorithm in which the cost function is the energy expectation value for the ansatz's output state with respect to a Hamiltonian of interest. Our Bayesian strategy only uses quantum circuit executions to query the energy expectation value for ansatz parameters, and not the gradients of this quantity. These points of interest are chosen through classical optimization of a surrogate model built from the observed energy values. By avoiding on-device estimation of energy gradients and employing an easily-optimized surrogate model, our strategy requires fewer quantum circuit executions per-point-queried than gradient based approaches while also converging in far fewer iterations, making it much more sample-efficient in terms of total quantum circuit executions. Being a global optimization strategy, it is also resilient against local minima. Because our surrogate model is built with a classically-evaluated quantum kernel, which itself is based on the ansatz circuit, our strategy is able to effectively leverage both explicit knowledge about the ansatz circuit with implicit information about the on-device noise processes contained in the observed energy values.

As the optimization strategy outlined here uses direct classical simulations of quantum kernel functions its applicability is limited to relatively narrow and shallow ansatz circuits (up to ∼20 qubits) for which classical simulations are feasible. Even at these scales, performing VQE on-device, in order to physically prepare an approximation to a system's ground state is still a highly non-trivial problem due to device noise and the long queues often required for device access. To extend our strategy's applicability, we also outline a scheme to produce a classically tractable approximation to the most-suitable quantum kernel for VQE and so provide a framework for applying the optimization strategy to larger systems.

2. VQE on NISQ devices

The VQE algorithm is a near-term quantum algorithm that prepares approximations of the ground state and ground state energy of a Hamiltonian and has a wide range of potential applications [26, 43]. This is done by adjusting the parameters of an ansatz circuit to minimize the measured energy of its output state. For

Hamiltonian H and an ansatz state $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|0\rangle$ or $\rho(\boldsymbol{\theta}) = |\psi(\boldsymbol{\theta})\rangle\langle\psi(\boldsymbol{\theta})|$, which is produced by a parameterized unitary evolution $U(\boldsymbol{\theta})$ with a vector of parameters $\boldsymbol{\theta}$, the goal of VQE is to find $\boldsymbol{\theta}_{\text{opt}}$ such that

$$\boldsymbol{\theta}_{\text{opt}} := \underset{\boldsymbol{\theta}}{\arg\min}\, E(\boldsymbol{\theta}), \text{ where } E(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle = \text{Tr}\{H\rho(\boldsymbol{\theta}))\}, \tag{1}$$

where Tr is the trace operation. By the variational principle, the ansatz state that minimizes the energy is an approximation to the ground state and its energy expectation value is an upper bound to the ground state energy.

The presence of noise on current quantum processors provides a justification for why on-device small system VQE is still an interesting and important task. On-device VQE gives not only an approximation to the desired ground state energy but also an ansatz circuit that prepares a close approximation to the ground state on the noisy device. Obtaining such circuits with classical methods would first require accurate characterization of the exact noise processes present during the device's operation, i.e. quantum process tomography, which for more than a few qubits is an incredibly resource-intensive task [44]. Being able to physically prepare a close approximation to the ground state of a Hamiltonian is a useful first step for other quantum computing tasks and for benchmarking the device [45, 46].

Gradient descent-based optimization strategies are ubiquitous in noiseless functional minimization due to their favorable convergence guarantees and are frequently used in VQAs [16, 35, 47–51]. The gradient of the energy function with respect to each ansatz parameter can be estimated directly either through finite difference approaches or using parameter shift rules [52]. Both approaches typically require sampling the cost function at two points per parameter (or more if a parameter appears in multiple gates) meaning these parameter-wise gradients are relatively quantum resource-intensive. Current devices are sufficiently noisy that only relatively shallow circuits on few qubits produce outputs not dominated by noise. At these scales ansatz-derived barren plateaus are less of a concern however noise-induced flattening of the cost function can still increase the number of samples needed to make accurate gradient estimates [31].

To avoid the large cost of estimating noisy parameter-wise gradients more sample-efficient and noise-resilient schemes such as the simultaneous perturbation stochastic approximation (SPSA) algorithm have been employed in VQAs [26, 47, 53] at the cost of slower convergence. This allows approximate gradient descent to be performed more efficiently on noisy objective functions because its update rule requires evaluations at just two points, regardless of how many parameters are being optimized. However SPSA is often slow to converge and can become stuck in local minima, requiring multiple restarts, this means a large number of energy evaluations are needed to solve even relatively small VQE problems. Other optimization strategies avoid gradients altogether and include heuristic schemes [43, 54, 55] and surrogate model based approaches [28, 54].

3. Surrogate models and quantum kernel functions

An alternative approach to gradient descent-based VQE is to build a surrogate model for the energy function based on previously observed evaluations. This should, ideally, be differentiable such that it can be optimized more effectively than $\tilde{E}(\boldsymbol{\theta})$, the noise-corrupted energy function. This approach forms the core of Bayesian optimization (BO) [56] in which a surrogate model is trained on observed data and then optimized to choose the most promising location to query the cost function. Previous work has shown how Bayesian VQE combined with a novel information-sharing approach can drastically improve the convergence times of a set of related VQE problems [28].

Typically, a Gaussian process (GP) surrogate model is used for BO and is primarily defined in terms of a kernel function. A kernel function, $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, is a symmetric positive (semi-)definite function of two arguments in an input space $\mathcal{X}$ such that $k(\boldsymbol{x}, \boldsymbol{x}')$ quantifies the similarity between $\boldsymbol{x}$ and $\boldsymbol{x}'$. They can also be understood as an inner product in a reproducing kernel Hilbert space (RKHS) such that $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\varphi}^T(\boldsymbol{x}')\boldsymbol{\varphi}(\boldsymbol{x})$, where $\boldsymbol{\varphi} : \mathcal{X} \to \mathcal{F}$ is a feature map which takes a point $\boldsymbol{x}$ in the input space to a point $\boldsymbol{\varphi}(\boldsymbol{x})$ in the *feature space*, $\mathcal{F}$, induced by the kernel (a vector in the RKHS) [18]. We give an overview of GP surrogates and BO in sections 8.1 and 8.2.

For an GP model to make accurate predictions about a function's value at an unseen point its kernel function should properly quantify the similarity between this point and the points where the function value has already been observed. Ideally, the objective function being modeled should be a linear function in the feature spaced induced by the kernel. If this is true, the *representer theorem* [57] from classical kernel theory shows that the minimizer of any regularized empirical risk functional (the model that is 'optimal' at describing a set of training data) is given by a weighted sum of kernel evaluations with the points in the training data.

In general the exact functional form of the cost function is unknown, making it difficult to know which kernel function to use. To combat this, flexible kernel functions equipped with hyperparameters are often

used. Many of these have the universal approximation property [58], meaning an arbitrary well-behaved function at a point x^* can be approximated by a finite weighted sum of kernel evaluations between x^* and an appropriately sized 'training set' of other points. To ensure that an GP model equipped with one of these flexible, agnostic kernels yields good predictive accuracy from the available training data the kernel's hyperparameters are typically chosen by maximizing the marginal likelihood of the training data. Optimization of a kernel's hyperparameters is often the most computationally expensive stage of fitting an GP model to observed data (see section 8.1). To date, surrogate model-based approaches to VQAs have made use of these general-purpose hyperparameterized classical kernel functions [28, 59], leaving open the question of whether more appropriate kernel functions exist for VQAs.

Schuld and Petruccione [60] has demonstrated that a very general class of quantum models are linear models in the feature space induced by a quantum kernel. A quantum model is described as a quantum circuit consisting of an encoding unitary followed by a (parameterized) measurement. If an VQA's cost function has a linear dependence on the measurement outcomes of a quantum circuit then the results in [60] suggest that a quantum kernel based on the same circuit could be used to produce a powerful surrogate model for the VQA's cost function.

3.1. Quantum kernel functions

The two quantum kernels that we will consider in this work are the 'state kernel', which is equivalent to the fidelity between quantum states, and the 'unitary kernel', which quantifies the similarity between two unitary operations in terms of their Hilbert-Schmidt norm. For an ansatz circuit $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|\boldsymbol{0}\rangle$, with an n-qubit input state $|\boldsymbol{0}\rangle = |0\rangle^{\otimes n}$, the state kernel evaluated between gate angles $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ is defined as:

$$k_s(\boldsymbol{\theta}, \boldsymbol{\theta}') := \left| \langle \psi(\boldsymbol{\theta}') | \psi(\boldsymbol{\theta}) \rangle \right|^2 = \left| \langle \boldsymbol{0} | U(\boldsymbol{\theta}') U(\boldsymbol{\theta}) | \boldsymbol{0} \rangle \right|^2 \tag{2}$$

or equivalently by $k_s(\boldsymbol{\theta}, \boldsymbol{\theta}') := \mathrm{Tr}\{\rho(\boldsymbol{\theta}')\rho(\boldsymbol{\theta})\}$ (assuming a pure quantum state). As we discuss in section 3.2 and explicitly show in appendix A, the energy function is linear in the feature space induced by the state kernel provided this kernel is based on the same circuit used to calculate $E(\boldsymbol{\theta})$. As a result the state kernel is still expected to be effective when building general surrogate models for $E(\boldsymbol{\theta})$.

The unitary kernel also induces a feature space in which the $E(\boldsymbol{\theta})$ is a linear function. For an ansatz imparting a $d \times d$-dimensional unitary $U(\boldsymbol{\theta})$ the unitary kernel is defined as

$$k_u(\boldsymbol{\theta}, \boldsymbol{\theta}') := \left| \frac{\mathrm{Tr}\{U^\dagger(\boldsymbol{\theta}') U(\boldsymbol{\theta})\}}{d} \right|^2, \tag{3}$$

which is the (normalized) modulus-squared Hilbert-Schmidt norm between the unitaries $U(\boldsymbol{\theta})$ and $U(\boldsymbol{\theta}')$. We show in appendices B and C that while $E(\boldsymbol{\theta})$ is linear in the feature space induced by this kernel this space typically has a much larger dimension than that of the state kernel and so more data is required to build accurate regression models using it.

We will now explicitly construct the feature maps of these two kernels for an ansatz circuit $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|\boldsymbol{0}\rangle$. Note that while a kernel function and corresponding reproducing kernel Hilbert space are unique, the feature map is not unique and is only defined up to an isometry (which preserves the inner product and thereby the kernel). We consider n-qubit p-parameter ansatzes formed of an initial state $|\boldsymbol{0}\rangle = |0\rangle^{\otimes n}$ acted upon by p parameterized Pauli rotations (PPRs), with n-qubit Pauli operators $\{P_1, \ldots, P_p\}$, which are interleaved with $p + 1$ arbitrary fixed unitaries $\{R_1, \ldots R_{p+1}\}$. The corresponding unitary $U(\boldsymbol{\theta})$ is given by:

$$U(\boldsymbol{\theta}) := R_{p+1} e^{-iP_p \theta_p / 2} R_p \ldots R_2 e^{-iP_1 \theta_1 / 2} R_1. \tag{4}$$

'Hardware efficient' ansatzes [26] admit this description and are formed of individual PPRs with low-weight Pauli operators; these ansatzes are common in NISQ quantum computing as they often map efficiently onto a device's native gate set [16]. Ansatzes of the form given by (4) are extremely flexible, being able to implement arbitrary parameterized unitaries provided a one uses a suitably many PPRs, a sufficiently sophisticated encoding of the angles $\boldsymbol{\theta}$, and a careful choice of the $\{R_i\}$ (although these may require exponentially deep circuits [61]).

We show in appendix A that the state kernel's feature space can be constructed in terms of an ansatz-dependent 3^p-element vector of operators on $\mathcal{H} \otimes \mathcal{H}^*$ (where $\mathcal{H}$ is the qubits' Hilbert space) $\boldsymbol{s}$, given by

$$\boldsymbol{s} = \frac{1}{2^p} \left[\begin{pmatrix} (I \otimes I + P_p \otimes P_p^*) R_p \otimes R_p^* \\ (I \otimes iP_p^* - iP_p \otimes I) R_p \otimes R_p^* \\ (I \otimes I - P_p \otimes P_p^*) R_p \otimes R_p^* \end{pmatrix} \otimes_K \ldots \otimes_K \begin{pmatrix} (I \otimes I + P_1 \otimes P_1^*) R_1 \otimes R_1^* \\ (I \otimes iP_1^* - iP_1 \otimes I) R_1 \otimes R_1^* \\ (I \otimes I - P_1 \otimes P_1^*) R_1 \otimes R_1^* \end{pmatrix} \right] \tag{5}$$

and an angle-dependent vector $\boldsymbol{v}(\boldsymbol{\theta})$, given by

$$\boldsymbol{v}(\boldsymbol{\theta}) = \begin{pmatrix} 1 \\ \sin(\theta_p) \\ \cos(\theta_p) \end{pmatrix} \otimes_K \ldots \otimes_K \begin{pmatrix} 1 \\ \sin(\theta_1) \\ \cos(\theta_1) \end{pmatrix}. \tag{6}$$

Here we have made a distinction between the tensor products $\otimes$ between the physical and conjugated Hilbert spaces of the qubits and Kronecker products $\otimes_K$ which are used to construct the kernels' feature maps. The $\otimes_K$ act as tensor products between the sub-vectors in $\boldsymbol{s}$ and $\boldsymbol{vp}$ imposing matrix multiplication in the former case and ordinary multiplication in the latter. Note that the elements of $\boldsymbol{v}(\boldsymbol{\theta})$ are linearly independent Fourier components and so cannot be decomposed further.

The state kernel is then

$$k_s(\boldsymbol{\theta}, \boldsymbol{\theta}') = \boldsymbol{v}^T(\boldsymbol{\theta}')\boldsymbol{S}\boldsymbol{v}(\boldsymbol{\theta}), \tag{7}$$

where $(\boldsymbol{S})_{ij} = \langle\langle \rho_0 | s_i^\dagger s_j | \rho_0 \rangle\rangle$ is a Hermitian matrix of inner products between the (unnormalized) states $s_i | \rho_0 \rangle\rangle$ and $s_j | \rho_0 \rangle\rangle$ (s_i and s_j are the ith and j^{th} components of $\boldsymbol{s}$-, respectively), $\rho_0 = |\boldsymbol{0}\rangle\langle\boldsymbol{0}|$, and $|X\rangle\rangle$ denotes the vectorization of X so that $|\rho_0\rangle\rangle = |\boldsymbol{0}\rangle \otimes |\boldsymbol{0}^*\rangle$ [62]. As $\boldsymbol{S}$ is a Hermitian matrix of inner products it is a positive-semidefinite Gram matrix and admits a Cholesky decomposition $\boldsymbol{S} = Q^\dagger Q$ (where Q is an upper-triangular matrix). This means that the state kernel can be written as the inner-product of two 3^p-element vectors, $k_s(\boldsymbol{\theta}, \boldsymbol{\theta}') = \boldsymbol{v}^T(\boldsymbol{\theta}')Q^\dagger Q\boldsymbol{v}(\boldsymbol{\theta}) = \boldsymbol{\varphi}_s^\dagger(\boldsymbol{\theta}')\boldsymbol{\varphi}_s(\boldsymbol{\theta})$, and so its feature map $\boldsymbol{\varphi}_s : \mathcal{X} \to \mathcal{F}_s$ is

$$\boldsymbol{\varphi}_s(\boldsymbol{\theta}) = Q\boldsymbol{v}(\boldsymbol{\theta}), \text{ where } Q^\dagger Q = \boldsymbol{S}. \tag{8}$$

The construction of the feature map for the unitary kernel is similar and involves the same two vectors $\boldsymbol{v}(\boldsymbol{\theta})$ (6) and $\boldsymbol{s}$ (5). The unitary kernel is given by

$$k_u(\boldsymbol{\theta}, \boldsymbol{\theta}') = \boldsymbol{v}^T(\boldsymbol{\theta}')\boldsymbol{T}\boldsymbol{v}(\boldsymbol{\theta}), \tag{9}$$

where, $(\boldsymbol{T})_{ij} = \text{Tr}\{s_i^\dagger s_j\}/4^n$ is a Hermitian Gram matrix of Hilbert-Schmidt inner products between the operators s_i and s_j. It too is positive-semidefinite allowing it to be written in Cholesky form $\boldsymbol{T} = B^\dagger B$ (with B upper-triangular), meaning that the unitary kernel is $k_u(\boldsymbol{\theta}, \boldsymbol{\theta}') = \boldsymbol{v}^T(\boldsymbol{\theta}')B^\dagger B\boldsymbol{v}(\boldsymbol{\theta}) = \boldsymbol{\varphi}_u^\dagger(\boldsymbol{\theta}')\boldsymbol{\varphi}_u(\boldsymbol{\theta})$ and its feature map $\boldsymbol{\varphi}_u : \mathcal{X} \to \mathcal{F}_u$ is

$$\boldsymbol{\varphi}_u(\boldsymbol{\theta}) = B\boldsymbol{v}(\boldsymbol{\theta}), \text{ where } B^\dagger B = \boldsymbol{T}. \tag{10}$$

So far we have assumed that all the gate angles used in the PPRs are independent. In more sophisticated ansatzes whose gate angles are linear combinations of the ansatz parameters one can apply linear transformations to $\boldsymbol{v}(\boldsymbol{\theta})$ such that it contains a set of independent Fourier components and so apply a similar analysis. Provided the couplings between gate parameters are linear (for example if multiple parameterized gates share the same parameter), this will act to reduce the dimension of the induced feature space.

3.2. VQE in a kernel setting

For Bayesian VQE to be effective, the kernel used must result in an GP model which can well-approximate $\tilde{E}(\boldsymbol{\theta})$, the noisy energy function. An interesting question this poses is whether quantum kernels are more ideally suited to this task than typical classical kernels. To justify that these kernels are well suited, we can analyze noiseless energy landscape $E(\boldsymbol{\theta})$ for a Hamiltonian H and an ansatz unitary of the form given in (4). In appendix A we show that this is

$$E(\boldsymbol{\theta}) = \boldsymbol{h}^T \boldsymbol{v}(\boldsymbol{\theta}), \tag{11}$$

where $h_i = \langle\langle R_{p+1} H R_{p+1}^\dagger | s_i | \rho_0 \rangle\rangle$ and $\boldsymbol{s}$ is given in (5). Encouragingly, we see that the angle-dependent part of the quantum kernels' feature spaces $\boldsymbol{v}(\boldsymbol{\theta})$ makes an appearance, mirroring the decomposition of the energy function in [51] in terms of a weighted sum of Fourier components. For a suitable set of energy observations, e.g. at the points $\{(\theta_1, \cdots, \theta_p) \in \{0, \pi/2, \pi\}^{\times p}$ or $\{(\theta_1, \cdots, \theta_p) \in \{-2\pi/3, 0, 2\pi/3\}^{\times p}\}$ it is straightforward to see how the components of $\boldsymbol{h}$ (the weights of the Fourier components in $E(\boldsymbol{\theta})$) can be directly inferred (the latter set of points giving exactly the result in [51]). As the feature spaces of both the state and unitary kernels are related to $\boldsymbol{v}(\boldsymbol{\theta})$ by linear transformations, the energy is a linear function in both these spaces. More explicitly, $E(\boldsymbol{\theta}) = \boldsymbol{h}^T Q^{-1} \boldsymbol{\varphi}_s(\boldsymbol{\theta}) = \boldsymbol{h}^T B^{-1} \boldsymbol{\varphi}_u(\boldsymbol{\theta})$ (Q and B are defined in (8) and (10), respectively and are assumed to be invertible).

Given a set of training data (pairs of ansatz parameters and corresponding noiseless energy evaluations), the representer theorem [57] guarantees that the minimizer of any regularized empirical loss functional is
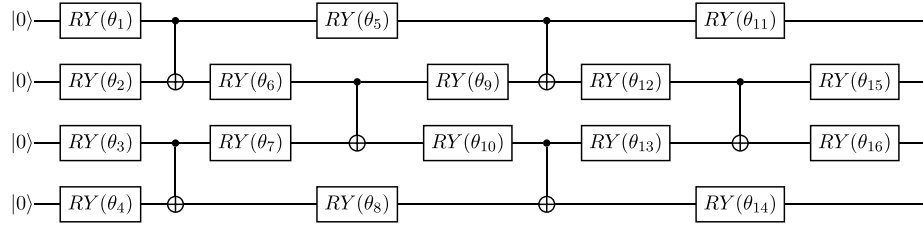
Figure 1. Ansatz used for simulations. This non-periodic ansatz yields a real wavefunction as its output state. The single-qubit gate $RY(\theta_j)$ imparts a unitary $RY(\theta_j) = e^{-i\sigma_y\theta_j/2}$. As the ansatz has four layers of *CX* gates on alternating pairs of adjacent qubits we describe this circuit as being depth 4.

given by a finite linear sum of weighted kernel evaluations evaluated at the training points (ansatz parameters whose energy values have been observed) [57]. For an l_2-regularized least-squares loss the minimizer is kernel ridge regression [63], which is mathematically equivalent to the posterior mean of an GP (21). Because these kernels have finite dimensional feature spaces, kernel ridge regression or GP models using them should be able to achieve perfect global prediction accuracy (can perfectly predict $E(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$) if the size of the training data set is greater than or equal to the feature space dimension (assuming the data is linearly independent in the feature space).

We note that linearity of $E(\boldsymbol{\theta})$ in these kernels' induced feature spaces is achieved without any hyperparameters. This removes the need for any maximum likelihood hyperparameter optimization. It also means that the state or unitary kernel between any two observed points (used to construct the Gram matrix $\mathbf{K}$) only needs to be evaluated a single time.

3.3. GPs with quantum kernels
While both quantum kernels look promising for VQE, one might suspect the state kernel to be more suitable as VQE directly concerns the energy of the state produced by the ansatz, rather than its unitary.

To evaluate the effectiveness of these quantum kernels versus typical classical kernels in kernel-based VQE regression we first consider the problem of attempting to build an GP model for observations of the noiseless $E(\boldsymbol{\theta})$. The Hamiltonian we will use throughout is the anti-ferromagnetic 1D transverse field Ising model with a longitudinal field, given by:

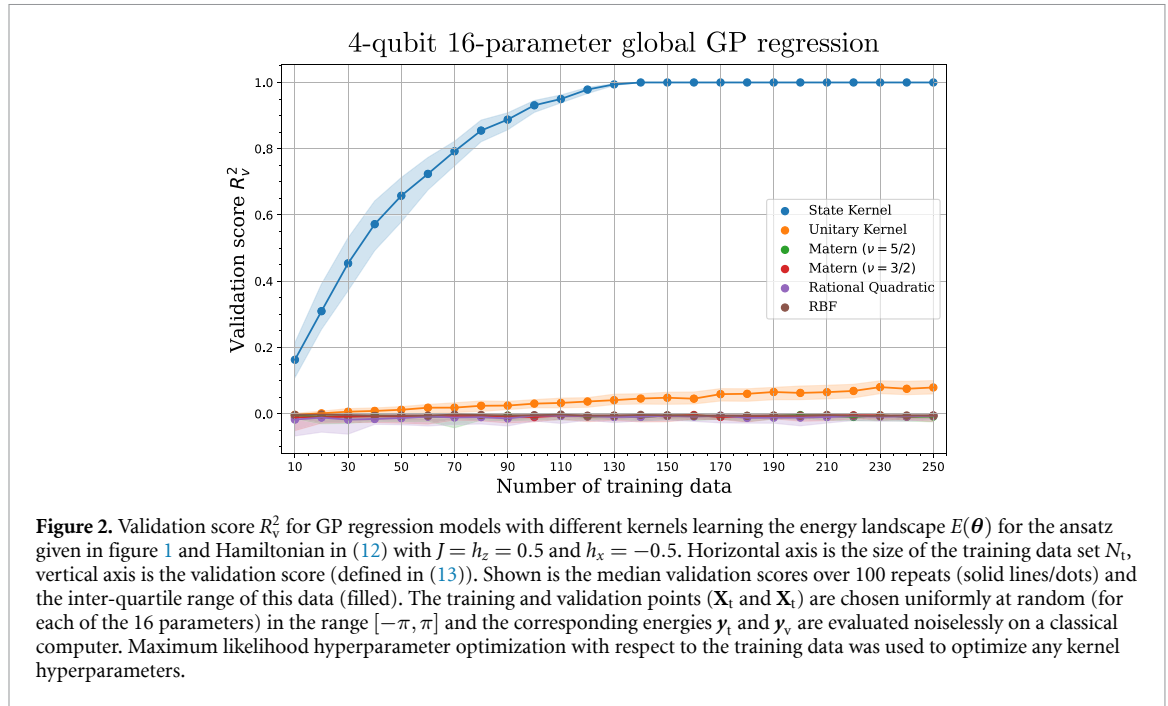$$H = J\sum_{\langle i,j\rangle} Z_i Z_j + h_x \sum_i X_i + h_z \sum_i Z_i, \tag{12}$$

where J is a coupling strength, h_x is the transverse field strength, h_z is the longitudinal field strength, X_i and Z_i are the Pauli X and Z operators acting respectively on the ith qubit, $Z_i Z_j$ denotes simultaneous Pauli Z operators on the ith and jth qubits, and the first sum is taken over nearest-neighbor pairs in a 1D spin chain with periodic boundary conditions. Throughout we use $J = h_z = 0.5$ and $h_x = -0.5$. As this is a real-valued Hamiltonian, its eigenstates can also be taken to be real. This means that for VQE we can limit our ansatz circuit to only yielding real wavefunctions. The ansatz we will use is shown in figure 1 and is designed to be efficiently implementable on a quantum processor with nearest-neighbor connectivity between qubits (which helpfully mirrors the nearest-neighbor terms in the Hamiltonian). It consists of parameterized rotations $RY(\theta_j) = e^{-i\sigma_y\theta_j/2}$ and fixed controlled-X gates to build up entanglement.

Let $\mathbf{G}[\mu(\cdot), k(\cdot, \cdot, \boldsymbol{\alpha})]$ be an GP model with covariance/kernel function k, kernel hyperparameters $\boldsymbol{\alpha}$, and mean function μ. Given a set of observed training and validation energies, $\boldsymbol{y}_t = \{E(\boldsymbol{\theta}_{t_1}), \ldots, E(\boldsymbol{\theta}_{t_{N_t}})\}$ and $\boldsymbol{y}_v = \{E(\boldsymbol{\theta}_{v_1}), \ldots, E(\boldsymbol{\theta}_{v_{N_v}})\}$ at parameters $\mathbf{X}_t = \{\boldsymbol{\theta}_{t_1}, \ldots, \boldsymbol{\theta}_{t_{N_t}}\}$ and $\mathbf{X}_v = \{\boldsymbol{\theta}_{v_1}, \ldots, \boldsymbol{\theta}_{v_{N_v}}\}$ respectively, we define validation score R_v^2 as

$$R_v^2(\mathbf{X}_v, \boldsymbol{y}_v | \mathbf{X}_t, \boldsymbol{y}_t) := 1 - \frac{\sum_{\boldsymbol{\theta} \in \mathbf{X}_v} [E(\boldsymbol{\theta}) - \hat{y}_{\mathbf{G}}(\boldsymbol{\theta})]^2}{\mathrm{Var}[\boldsymbol{y}_v]}, \tag{13}$$

where $\hat{y}_{\mathbf{G}}(\boldsymbol{\theta}) = \mathbb{E}[y^* | \boldsymbol{\theta}, \mathbf{X}_t, \boldsymbol{y}_t]$ is the prediction of model $\mathbf{G}$ at point $\boldsymbol{\theta}$ trained with data $(\mathbf{X}_t, \boldsymbol{y}_t)$ (which, if appropriate, includes hyperparameter optimization). A validation score of $R_v^2 = 0$ indicates that the GP model is performing as well (with respect to the L_2 loss) as a model that always predicts $\mathbb{E}[\boldsymbol{y}_v]$, while a score of $R_v^2 = 1$ shows perfect prediction across the unseen validation set.

Figure 2 shows the improvement in validation score as the size of the training data is increased for GP regression models equipped with different quantum and typical classical kernels. The models are presented with noiseless evaluations of $E(\boldsymbol{\theta})$ for the Hamiltonian given in (12) ($J = h_z = 0.5$ and $h_x = -0.5$). The

Figure 2. Validation score R_v^2 for GP regression models with different kernels learning the energy landscape $E(\boldsymbol{\theta})$ for the ansatz given in figure 1 and Hamiltonian in (12) with $J = h_z = 0.5$ and $h_x = -0.5$. Horizontal axis is the size of the training data set N_t, vertical axis is the validation score (defined in (13)). Shown is the median validation scores over 100 repeats (solid lines/dots) and the inter-quartile range of this data (filled). The training and validation points ($\mathbf{X}_t$ and $\mathbf{X}_t$) are chosen uniformly at random (for each of the 16 parameters) in the range $[-\pi, \pi]$ and the corresponding energies $\boldsymbol{y}_t$ and $\boldsymbol{y}_v$ are evaluated noiselessly on a classical computer. Maximum likelihood hyperparameter optimization with respect to the training data was used to optimize any kernel hyperparameters.

training and validation sets ($\mathbf{X}_t$ and $\mathbf{X}_v$) are drawn uniformly at random from the interval $\{-\pi, \pi\}$ (for each of the 16 parameters). Both the median R_v^2 over 100 repeats (solid line) and the inter-quartile range of the data are shown. The quantum kernel and energy evaluations were performed noiselessly on a classical computer. In all cases, a small diagonal offset was added to the Gram matrices used in GP model prediction $\mathbf{K} \to \mathbf{K} + 10^{-10}I$ (equivalent to $\sigma_n^2 = 10^{-10}$ in (21)) to ensure numerical stability.

We see that an GP model equipped with the state kernel greatly outperforms the other kernels and reaches the optimal validation score of $R_v^2 = 1$ once the size of the training set reaches $N_t = 136$. The validation score of the unitary kernel improves much more slowly, and for $N_t = 136$ achieves an $R_v^2 < 0.1$. However this is still much higher than the scores for the various classical kernels which perform poorly at this task. These classical kernels yield validation scores $R_v^2 < 0$ and show negligible signs of improvement as the training data set grows.

It is clear, and perhaps unsurprising, that a state kernel built from the same circuit used for a VQE problem provides a much more accurate similarity measure between data points than the other kernels we consider. This is because the energy function is linear in the finite-dimensional feature space induced by the state kernel (as discussed in section 3.2). While $E(\boldsymbol{\theta})$ is also linear in the unitary kernel's feature space, we show in section 6 that this kernel's induced feature space is typically quadratically larger than that of the state kernel. To build a globally accurate GP model, the observations must span a significant portion of this space which explains why k_s vastly outperforms k_u at this task. The classical kernels we use are universal but do not appear particularly well-suited to the complicated periodic and highly oscillatory $E(\boldsymbol{\theta})$. As this is 16-parameter problem, the density of the couple of hundred observations in the input parameter space will be far lower than what is required for these classical kernels to make good use of their universal property.

For BO we are not always interested in creating an accurate global (i.e. $\theta_i \in [-\pi, \pi]$) regression model as in figure 2. While good global accuracy is helpful, in BO only promising regions of parameter space (those that look close to the desired extremum) are explored in detail. On smaller parameter scales the energy landscape will be less dramatically oscillatory and the density of observed data will be higher meaning that the classical kernels may have a better chance of accurately interpolating between observations.

To test this we examine the validation score for GP models when performing local regression of a VQE energy landscape, restricted to a smaller region of the parameter space. In this case, the training set $\mathbf{X}_t$ and validation set $\mathbf{X}_v$ are generated by first picking a common anchor point $\boldsymbol{\theta}^{(0)}$ whose elements are chosen uniformly at random in the range $[-\pi, \pi]$. The circuit parameters for $\mathbf{X}_t$ and $\mathbf{X}_v$ are then obtained by sampling parameter vectors $\boldsymbol{\theta}$ with elements drawn uniformly at random from the interval $\theta_i \in [\theta_i^{(0)} - \pi/s, \theta_i^{(0)} + \pi/s]$, where s is a scale reduction factor.

Figure 3 illustrates this process, showing the log prediction error $\log_{10}(1 - R_v^2)$ for a repeat of the simulations in figure 2 on reduced parameter scales with scale factors $s = 10$ (3(a)) and $s = 100$ (3(b)). We see that when required to make only local predictions, GP models using the unitary and classical kernels show significant improvement although those using the state kernel are consistently the most accurate. This is to
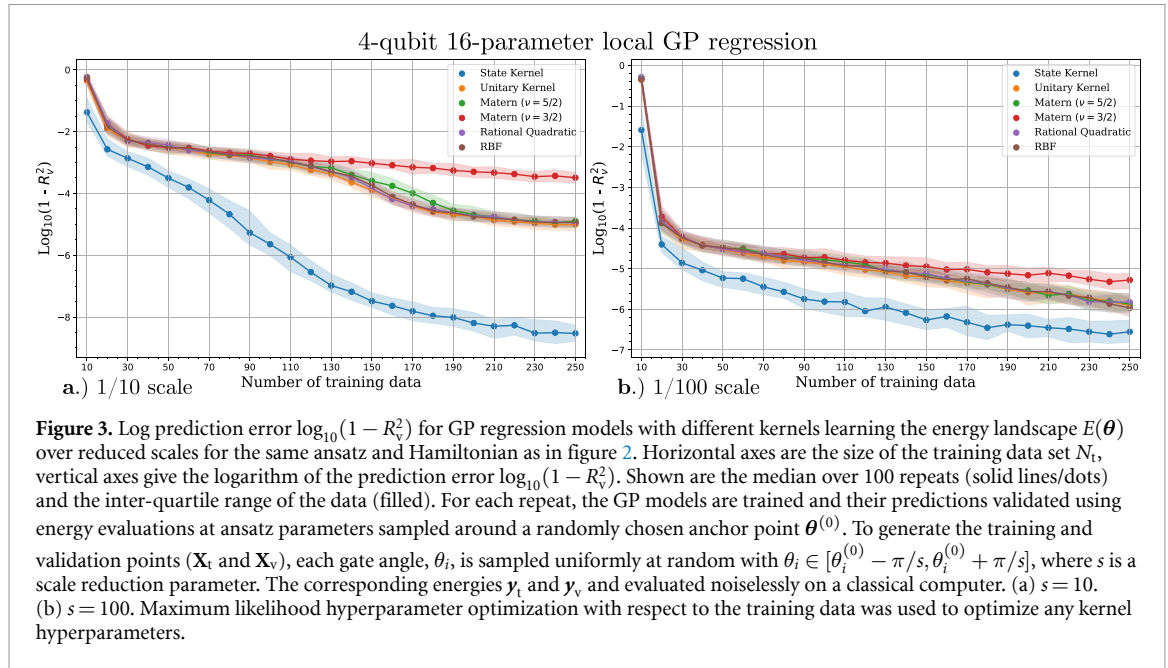
Figure 3. Log prediction error $\log_{10}(1 - R_v^2)$ for GP regression models with different kernels learning the energy landscape $E(\boldsymbol{\theta})$ over reduced scales for the same ansatz and Hamiltonian as in figure 2. Horizontal axes are the size of the training data set N_t, vertical axes give the logarithm of the prediction error $\log_{10}(1 - R_v^2)$. Shown are the median over 100 repeats (solid lines/dots) and the inter-quartile range of the data (filled). For each repeat, the GP models are trained and their predictions validated using energy evaluations at ansatz parameters sampled around a randomly chosen anchor point $\boldsymbol{\theta}^{(0)}$. To generate the training and validation points ($\mathbf{X}_t$ and $\mathbf{X}_v$), each gate angle, θ_i, is sampled uniformly at random with $\theta_i \in [\theta_i^{(0)} - \pi/s, \theta_i^{(0)} + \pi/s]$, where s is a scale reduction parameter. The corresponding energies $\boldsymbol{y}_t$ and $\boldsymbol{y}_v$ and evaluated noiselessly on a classical computer. (a) $s = 10$. (b) $s = 100$. Maximum likelihood hyperparameter optimization with respect to the training data was used to optimize any kernel hyperparameters.

be expected as observations are much denser in the input space. Denser observations allow the universal classical kernel-based models to more effectively interpolate between observations. They also reduce the portion of the state and unitary kernel's induced feature spaces which the observations must span for k_s- and k_u-based GP models to perform well.

The state kernel's advantage is most pronounced for $s = 10$, which is to be expected as this is closer to global regression. The unitary, RQ, RBF kernels and the Matern kernel with $\nu = 5/2$ all perform similarly well but the Matern kernel with $\nu = 3/2$ struggles when presented with large amounts of data (particularly for $s = 10$). This is likely because a GP equipped with a Matern kernel is $\lceil \nu \rceil - 1$ times differentiable (once differentiable for $\nu = 3/2$ and twice for $\nu = 5/2$) [63], while those using the unitary, RBF, and RQ kernels are infinitely differentiable. As the noiseless energy function (a finite weighted sum of Fourier components, see 3.2) is smooth and infinitely differentiable this may explain why the singly differentiable $\nu = 3/2$ Matern kernel performs worse than the other, more-times differentiable kernels.

4. Bayesian VQE with quantum kernels

4.1. Limitations of on-device quantum kernels
For it to be useful, the predictions from a surrogate model should be significantly easier to compute and optimize than the cost function. Given m observations, making a prediction from an GP surrogate (see section 8.1), as is done when optimizing a GP-based surrogate model in BO, costs m kernel evaluations, while fitting the model requires a Gram-matrix of $\mathcal{O}(m^2)$ kernel evaluations to be calculated.

This raises a potential issue with quantum kernel-based surrogate models, namely that both fitting the surrogate model and using it to make predictions requires repeated executions of a large number of quantum circuits. Unless the number of data points is smaller than the number of circuits required to measure the cost/energy function one could simply measure and optimize the cost directly, avoiding any inaccuracies of the approximate surrogate model. Accordingly, quantum kernels evaluated on-device are unlikely to be useful for producing the kinds of easy-to-optimize surrogate models needed for BO.

It is also worth noting that unless the ansatz used is highly structured, both the state and unitary kernels evaluated for two randomly chosen sets of gate angles will shrink exponentially with the number of qubits used [64]. While this is less of an issue for small NISQ-scale VQE problems, this means that for problems on many qubits one would require exponentially-many shots to accurately resolve these exponentially-small kernel evaluations. This is closely related to the barren-plateau problem and further reinforces our assertion that quantum kernels evaluated on-device are unlikely to be useful for producing globally-accurate surrogate models for VQE.

Device noise also greatly complicates the evaluation of quantum kernels on-device. Although using a kernel based on the noisy operation of a quantum processor may be useful in quantifying the similarity between noisy energy evaluations such kernel functions may require large numbers of samples to evaluate accurately. This would greatly complicate any surrogate model optimization and means that the usual

requirements of a kernel, for example positivity, would not necessarily be satisfied. It would also be difficult to ensure that the errors present when evaluating the kernel correspond to those encountered when estimating the cost/energy. For example, one typical implementation of the state kernel requires a circuits of twice the depth used to estimate the energy [65] and so involves more noise and decoherence than would be seen in the cost function. Alternatively, one can implement the kernel using a circuit of the same depth as the cost function but over two sets of qubits [66] which will likely have different noise characteristics.

While GPs naturally accommodate noisy observations it is normally assumed that the kernel function can be computed exactly. Circuit noise and finite sampling errors would mean an on-device quantum kernel evaluation is affected by statistical fluctuations. This would then affect the two key objects in GP modeling; the positive-definite Gram matrix **K** of pair-wise kernel evaluations for the observed points and the vector *k* of kernel evaluations between a point of interest and the observed points (see section 8.1 for a full discussion of these). Randomness in these objects can be detrimental in two ways. Firstly, if the Gram matrix **K** is composed of noisy kernel evaluations then it may not be positive definite and numerical instabilities can occur when calculating its inverse (needed to make predictions). This can be partially alleviated by increasing the noise strength hyperparameter σ_n (see section 8.1) to ensure an invertible and positive-definite **K**.

Another more serious issue comes when attempting to minimize a surrogate model built using a noisy kernel as part of a BO loop. Gradient descent methods are the standard approach for this however they require repeated accurate evaluations of $\nabla_{\boldsymbol{\theta}} \boldsymbol{k}$, the gradient of kernel evaluations between the query point and the training data. While noise-resilient methods such as SPSA noise-resilient could be used for this optimization, these could be applied directly to the objective function, bypassing the need for a surrogate model. A possible exception would be if the surrogate model is considerably cheaper to evaluate than the cost function, which would be true if the number of observations is smaller than the number of circuits required to evaluate the cost.

Due to these complications, we do not expect significant practical advantages to VQE from GPs which use device-evaluated quantum kernels. Instead we evaluate all the quantum kernels using a classical computer, which can be done tractably provided the number of qubits is relatively small and depth of the quantum circuits involved is relatively low.

For many-qubit VQE problems, in which most kernel evaluations are exponentially suppressed, one could still attempt a highly localized form of Bayesian VQE with classically-evaluated quantum kernels. In such scenarios, the Gram matrix **K** calculated for a set of randomly chosen initial points will be exponentially close to the identity. Similarly, kernel evaluations between a new point of interest $\boldsymbol{\theta}^*$ and these initially chosen points will be close to zero unless $\boldsymbol{\theta}^*$ is in the immediate vicinity of an initial point. Bayesian VQE using an GP with such an exponentially-decaying kernel will be highly localized around whichever initial point gives the lowest energy. This means that at each step in the BO, the maximum of the acquisition function (see section 8.2) will stay close to the most promising initial point (where the kernel is non-negligible). This optimization strategy would still differ from completely local methods like as gradient descent in that all previous observations (however close to each other they may be) are used to decide the next query point rather than just the most recent point seen. However for this to work, one would still need to find a useful starting point for the optimization (i.e. not in a barren-plateau) and have a method for evaluating the kernel. This could either be estimated on-device, would likely be more costly than optimizing the energy directly, or evaluated classically, which would be intractable unless the ansatz has some simple exploitable structure or admits a simplifying approximation (we discuss this in section 5).

4.2. Noiseless Bayesian VQE

We have seen that the state kernel provides a significant advantage over typical classical kernels when performing GP regression of a circuit's energy landscape and now apply these results to VQE using BO. BO is a gradient-free strategy for optimizing noisy expensive-to-evaluate objective functions [56]. As current cloud-based NISQ computers are in high demand, the problem of variationally minimizing the noisy $\tilde{E}(\boldsymbol{\theta})$ is an apt use-case for BO. We describe BO in more detail in section 8.2.

We will quantify the performance of the optimization in terms of the best seen energy error $\mathcal{E}(\boldsymbol{y}, \boldsymbol{\theta}_{\mathrm{opt}})$ which we define as the fractional difference between the lowest energy seen at the current stage in the optimization and the minimum achievable noiseless energy for the ansatz. For a set of observed energy values $\boldsymbol{y}$ and an optimal set of gate angles $\boldsymbol{\theta}_{\mathrm{opt}}$, as defined in (1), this is given by

$$\mathcal{E}(\boldsymbol{y}, \boldsymbol{\theta}_{\mathrm{opt}}) := \frac{\min(\boldsymbol{y}) - E(\boldsymbol{\theta}_{\mathrm{opt}})}{|E(\boldsymbol{\theta}_{\mathrm{opt}})|}. \tag{14}$$

The optimal gate angles $\boldsymbol{\theta}_{\mathrm{opt}}$ for the ansatz shown in figure 1 were found using 10 000 attempts of direct gradient-based minimization, giving a minimum energy of $E(\boldsymbol{\theta}_{\mathrm{opt}}) = -2.762194$.
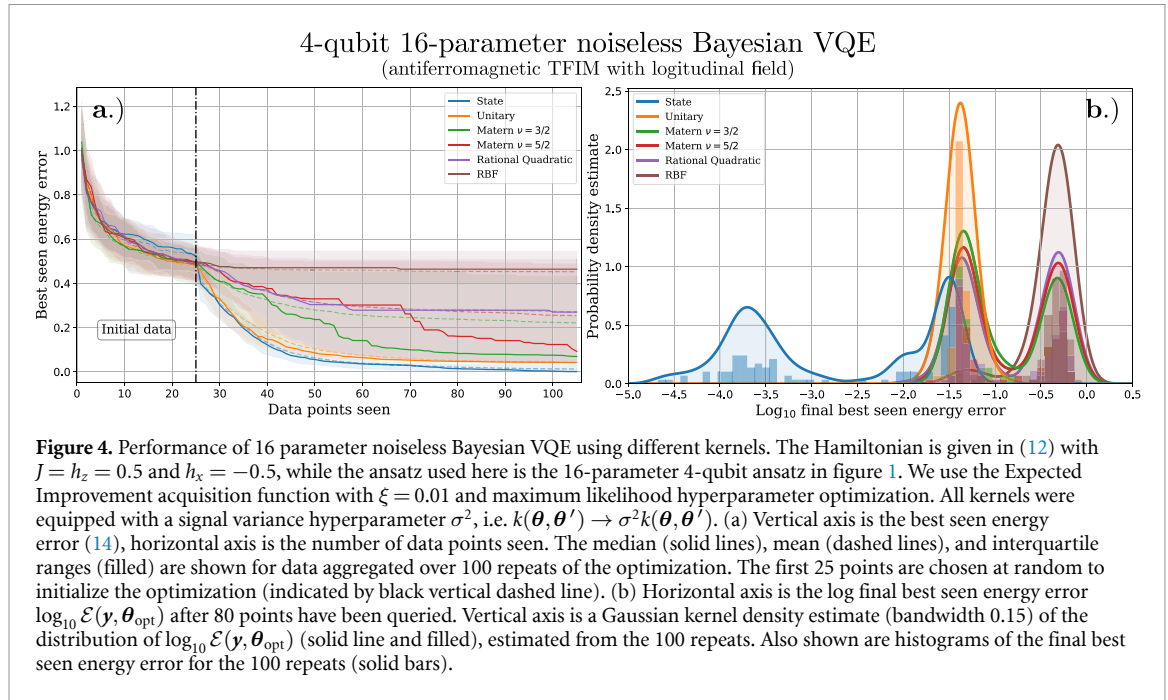
Figure 4. Performance of 16 parameter noiseless Bayesian VQE using different kernels. The Hamiltonian is given in (12) with $J = h_z = 0.5$ and $h_x = -0.5$, while the ansatz used here is the 16-parameter 4-qubit ansatz in figure 1. We use the Expected Improvement acquisition function with $\xi = 0.01$ and maximum likelihood hyperparameter optimization. All kernels were equipped with a signal variance hyperparameter σ^2, i.e. $k(\boldsymbol{\theta}, \boldsymbol{\theta}') \rightarrow \sigma^2 k(\boldsymbol{\theta}, \boldsymbol{\theta}')$. (a) Vertical axis is the best seen energy error (14), horizontal axis is the number of data points seen. The median (solid lines), mean (dashed lines), and interquartile ranges (filled) are shown for data aggregated over 100 repeats of the optimization. The first 25 points are chosen at random to initialize the optimization (indicated by black vertical dashed line). (b) Horizontal axis is the log final best seen energy error $\log_{10} \mathcal{E}(\boldsymbol{y}, \boldsymbol{\theta}_{\text{opt}})$ after 80 points have been queried. Vertical axis is a Gaussian kernel density estimate (bandwidth 0.15) of the distribution of $\log_{10} \mathcal{E}(\boldsymbol{y}, \boldsymbol{\theta}_{\text{opt}})$ (solid line and filled), estimated from the 100 repeats. Also shown are histograms of the final best seen energy error for the 100 repeats (solid bars).

Figure 4 shows the results of noiseless VQE simulations using BO with different kernels. The ansatz circuit used is shown in figure 1 and the Hamiltonian is given in (12) with $J = h_z = 0.5$ and $h_x = -0.5$. At the start of each simulation, 25 initial points were drawn uniformly at random and their energies evaluated to form an initial training data set. The Expected Improvement [67] acquisition function was used with an exploration hyperparameter $\xi = 0.01$, this value is widely used in EI-based BO [68, 69] and was found to yield the best BO performance across the classical kernels considered. For each kernel, 100 repeats of the optimization were performed. Figure 4(a) shows the median (solid lines), mean (dashed lines), and inter-quartile (filled) range of the energy error. Figure 4(b) gives histograms and Gaussian kernel density estimates (with bandwidth 0.15 [70]) of the final best seen energy errors (after 80 new points have been requested).

The state kernel outperforms the other kernels at this VQE task, both in terms of the final energy found and in the speed at which it converges. It frequently reaches an energy less than one part in $10^{-3.5} \approx 0.03\%$ away from the minimum achievable with this ansatz. Its final achieved energies form clusters around a few values, the most noticeable being one at errors between 10^{-4} and $10^{-3.5}$ and one that is close to, but still lower than, the majority of energies observed with the unitary and classical kernels. The unitary kernel also performs well, quickly and consistently converging to an error of approximately 4% and exhibits the second lowest mean and median errors overall. However, the data is strongly clustered with this error, implying that achieving a more accurate final energy may be difficult. The classical kernels vary considerably in their performance, generally having much worse mean errors than the quantum kernels. The final energies achieved by the classical kernels form two clusters; one of relatively successful runs with errors ∼6% and one with errors between 30%–60%. The errors in this latter cluster are similar to those seen in the randomly selected initialization data, implying that these optimization runs failed to achieve any significant reduction in the energy. This suggests that while these GP models using these classical kernels sometimes lead to relatively successful VQE, they often immediately and become stuck in an exploitative phase where points close to those in the initial training data are repeatedly queried. It is therefore likely that as well as their advantages in over-all performance, Bayesian VQE using the state or unitary kernel can be more resilient against initialization failures than when using the classical kernels we have considered.

For comparison, figure 5 shows a repeat of this noiseless VQE using the SPSA optimization scheme. We used the Qiskit implementation [71] which closely follows the initial proposal in [72]. We see from the experimental traces 5(a) and histograms in 5(b) that the effectiveness of this optimization strategy can vary but the average final energies typically have errors in line with those obtained for BO using the unitary kernel at around 4%. These are also comparable to the best experimental runs for BO using the various classical kernels. The data is strongly clustered around this energy error and only a handful of SPSA runs manage to compete with the low energies frequently achieved with state kernel-based BO. The SPSA runs which cluster around 4% generally require several hundred energy evaluations to achieve this level of accuracy whereas only around 100 evaluations are needed for the various kernel based models. The two SPSA runs with the lowest mean final energies had mean final energy errors on the order of 0.1% which is comparable to the
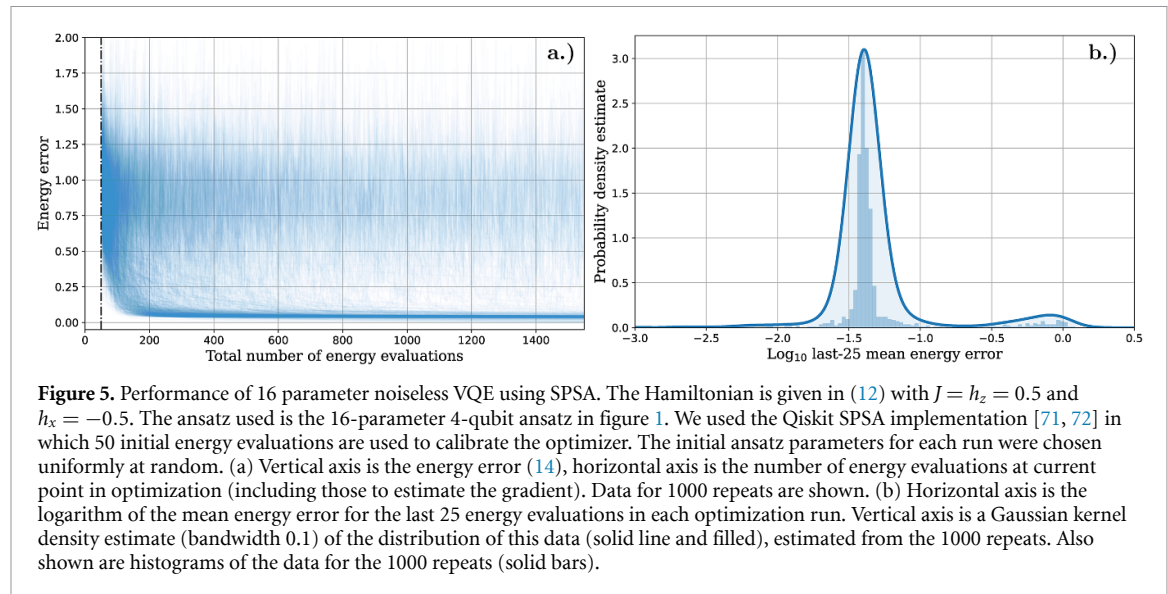
Figure 5. Performance of 16 parameter noiseless VQE using SPSA. The Hamiltonian is given in (12) with $J = h_z = 0.5$ and $h_x = -0.5$. The ansatz used is the 16-parameter 4-qubit ansatz in figure 1. We used the Qiskit SPSA implementation [71, 72] in which 50 initial energy evaluations are used to calibrate the optimizer. The initial ansatz parameters for each run were chosen uniformly at random. (a) Vertical axis is the energy error (14), horizontal axis is the number of energy evaluations at current point in optimization (including those to estimate the gradient). Data for 1000 repeats are shown. (b) Horizontal axis is the logarithm of the mean energy error for the last 25 energy evaluations in each optimization run. Vertical axis is a Gaussian kernel density estimate (bandwidth 0.1) of the distribution of this data (solid line and filled), estimated from the 1000 repeats. Also shown are histograms of the data for the 1000 repeats (solid bars).

lowest energies achieved with state kernel-based BO. However, these were only seen after 1500 total energy evaluations to be made. This level of final accuracy with SPSA appears to be extremely rare and requires greater than an order of magnitude more energy evaluations than is needed with the state kernel, highlighting the effectiveness and economy of our optimization strategy.
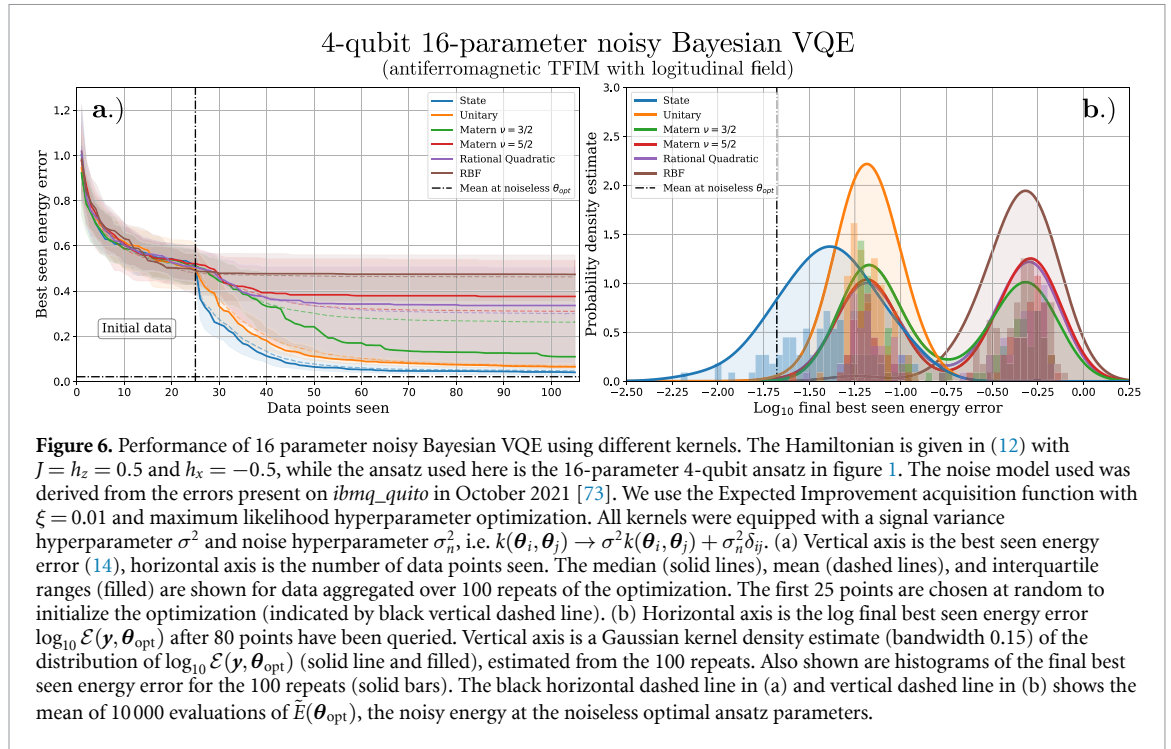
4.3. Noisy Bayesian VQE

The question remains whether a GP surrogate using a noiseless classically-evaluated quantum kernel is useful for Bayesian VQE of noisy quantum circuits. The noisy energy function is unlikely to be a linear function in the feature spaces induced by the noiseless state and unitary kernels. However, provided the device noise is not too great, these noiseless classical evaluations should provide a good approximation to the correct feature space for describing the noisy energy. By using a quantum kernel function based on the ansatz circuit we are able to leverage our prior knowledge of the (noiseless) ansatz circuit whereas standard GP surrogates can only attempt to do this through hyperparameter optimization. Because the observed energy values provide us with information about the noise on the device we are also able to take this noise into account implicitly.

Figure 6 shows the results of a repeat of the Bayesian VQE simulations illustrated in figure 4 with noisy energy evaluations. These were performed with using Qiskit's noisy quantum circuit simulation framework [71] and a noise model derived from the errors present on *ibmq_quito* in Oct 2021 [73]. The noise includes contributions from gate errors, state preparation and measurement errors (for which Qiskit's readout error mitigation was used), and finite circuit shots (10 000 per circuit per evaluation). The quantum kernel evaluations (where used) were simulated exactly using a tensor network quantum circuit implementation based on the Quimb Python package [74]. As in figure 4, figure 6(a) shows the median (solid lines), mean (dashed lines), and inter-quartile range (filled) of the best seen energy error; the difference between the lowest seen energy and the minimum possible for the ansatz. Figure 6(b) shows histograms and kernel density estimates (with bandwidth 0.15) of the final best seen energy errors (after 80 new points are queried).

The noise in the simulations makes it unlikely that the energy can reach the noiseless minimum value. Accordingly, we also show the mean energy error obtained from 10 000 repeated noisy evaluations of the energy $\tilde{E}(\boldsymbol{\theta}_{\mathrm{opt}})$, taken at $\boldsymbol{\theta}_{\mathrm{opt}}$, the noiseless optimum gate angles. While $\boldsymbol{\theta}_{\mathrm{opt}}$ may differ slightly from the gate angles that yield the true noisy optimum these evaluations serve as an indicator of the best performance to be expected from a noisy VQE implementation.

We again see the GP models based on the state kernel greatly outperform those using the other kernels. The state kernel frequently manages to achieve energies lower than the mean $\tilde{E}(\boldsymbol{\theta}_{\mathrm{opt}})$ (an error of $\sim 2\%$) while also showing the fastest convergence. The unitary kernel also performs well, consistently giving energies lower or at least as low as those found using classical kernels (around 6%) but with a significantly lower variation in its performance. When dealing with noisy evaluations, VQE using the classical kernels shows qualitatively similar performance to the noiseless case (although with different final energy errors). Almost all the simulations performed with the RBF kernel and approximately half of those with the other classical kernels give a final errors in the range 30%–60%. As in the noiseless case, these errors are similar to those seen for the randomly-selected initialization points implying that these runs failed almost immediately. The

Figure 6. Performance of 16 parameter noisy Bayesian VQE using different kernels. The Hamiltonian is given in (12) with $J = h_z = 0.5$ and $h_x = -0.5$, while the ansatz used here is the 16-parameter 4-qubit ansatz in figure 1. The noise model used was derived from the errors present on *ibmq_quito* in October 2021 [73]. We use the Expected Improvement acquisition function with $\xi = 0.01$ and maximum likelihood hyperparameter optimization. All kernels were equipped with a signal variance hyperparameter σ^2 and noise hyperparameter σ_n^2, i.e. $k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) \to \sigma^2 k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) + \sigma_n^2 \delta_{ij}$. (a) Vertical axis is the best seen energy error (14), horizontal axis is the number of data points seen. The median (solid lines), mean (dashed lines), and interquartile ranges (filled) are shown for data aggregated over 100 repeats of the optimization. The first 25 points are chosen at random to initialize the optimization (indicated by black vertical dashed line). (b) Horizontal axis is the log final best seen energy error $\log_{10} \mathcal{E}(\boldsymbol{y}, \boldsymbol{\theta}_{\mathrm{opt}})$ after 80 points have been queried. Vertical axis is a Gaussian kernel density estimate (bandwidth 0.15) of the distribution of $\log_{10} \mathcal{E}(\boldsymbol{y}, \boldsymbol{\theta}_{\mathrm{opt}})$ (solid line and filled), estimated from the 100 repeats. Also shown are histograms of the final best seen energy error for the 100 repeats (solid bars). The black horizontal dashed line in (a) and vertical dashed line in (b) shows the mean of 10 000 evaluations of $\tilde{E}(\boldsymbol{\theta}_{\mathrm{opt}})$, the noisy energy at the noiseless optimal ansatz parameters.

remainder of the classical kernel simulations achieve a much better performance which is close to the $\sim 6\%$ seen with the unitary kernel.

5. Classical simulation of quantum kernels

We have demonstrated that Bayesian VQE is significantly more effective when using quantum kernel-based GP models than when using classical kernels, both in terms of final energy accuracy and reliability. However, our simulations have concerned circuits of sufficiently few qubits and low depth that classical simulation of the quantum kernels is tractable. Once the number of qubits and/or the depth of the variational ansatz becomes too large, a GP surrogate model built on the full quantum kernel would be too computationally expensive to be practically used. However, if the number of gates to be optimized is not too great, in some instances classical evaluation of quantum kernels can remain tractable even for large numbers of qubits.

One way to ensure this is to only perform optimization on a subset of gates at any given time. Suppose we have an ansatz of the form $U(\boldsymbol{\theta}) = U_C(\boldsymbol{\theta}_C) U_B(\boldsymbol{\theta}_B) U_A(\boldsymbol{\theta}_A)$, where U_C, U_B, and U_A do not commute and any constant gates which commute with U_B have been included in U_A or U_C. If, at some stage in the VQE, we fix $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_C$ and only optimize over $\boldsymbol{\theta}_B$ then the unitary kernel is given by $k_u(\boldsymbol{\theta}_B, \boldsymbol{\theta}_B') = |\mathrm{Tr}\{U_B^\dagger(\boldsymbol{\theta}_B') U(\boldsymbol{\theta}_B)\}/d|^2$. Both the gates in the past (U_A) and future (U_C) causal light-cones of U_B cancel out due to the cyclic property of the trace meaning that the classical computational cost of evaluating $k_u(\boldsymbol{\theta}_B, \boldsymbol{\theta}_B')$ only depends on the complexity of U_B.

However, as we have seen in the previous sections, while the unitary kernel does provide some advantages over the typical classical kernels we have considered, it usually requires many more observed energy evaluations than the state kernel to produce a similarly accurate surrogate model. The state kernel for this block-wise parametrization is given by $k_s(\boldsymbol{\theta}_B, \boldsymbol{\theta}_B') = |\langle 0| U_A^\dagger(\boldsymbol{\theta}_A) U_B^\dagger(\boldsymbol{\theta}_B') U(\boldsymbol{\theta}_B) U_A(\boldsymbol{\theta}_A) |0\rangle|^2$. For this kernel, the gates in the past causal light-cone of U_B (those in U_A) must still be retained. Equivalently the kernel can be written as

$$k_s(\boldsymbol{\theta}_B, \boldsymbol{\theta}_B') = |\langle \psi_A| U_B^\dagger(\boldsymbol{\theta}_B') U(\boldsymbol{\theta}_B) |\psi_A\rangle|^2, \tag{15}$$

where $|\psi_A\rangle = U_A(\boldsymbol{\theta}_A)|0\rangle$ is the input state for an equivalent state kernel only containing U_B. If U_A is a deep circuit across many qubits such that $|\psi_A\rangle$ cannot be represented or manipulated in a classically efficient manner, then this could make classical evaluation impractical and so prevent the kernel's use in a classical GP model.

Block or layer-wise VQE as we describe above has received significant attention in the literature as a potential solution to the barren-plateau issues that prevent large-scale VQE [51, 75, 76]. If we assume that VQE can be performed effectively by selecting small blocks or layers of gates (U_B) to optimize at a time then

one could attempt to find a classically tractable approximation to $|\psi_A\rangle$ in order to produce an approximate state kernel. In section 8.3 we describe a scheme that approximates $|\psi_A\rangle$ to a matrix product state (MPS).

MPSs are a classically-efficient representation for certain types of multi-partite quantum state [77, 78]. They first saw widespread use in condensed matter physics as the underlying ansatz for the powerful density matrix renormalization group (DMRG) algorithm (and other closely related algorithms) for 1-dimensional quantum lattices [79] and are an example of the wider class of tensor network states [80]. An MPS (with periodic boundary conditions) on n sites is defined in terms of a collection of n rank-3 tensors with a single outgoing physical index (of dimension equal to that of the subsystem at the site) and one or two (depending on periodicity) virtual indices. The state is given as a contraction over pairs of virtual indices for adjacent sites, forming bonds between them and leaving only the physical indices. The amount entanglement present in the state and the complexity of representing and manipulating the MPS depends on the size of the bond indices.

Our scheme approximates the input state $|\psi_A\rangle$ by starting with an MPS representation of the input state $|\mathbf{0}\rangle$ (starting with all bond dimensions equal to 1) and contracting each constant gate from U_A into this MPS. Single qubit gates can be contracted into an MPS without changing its bond dimensions. To apply a two-qubit gate to neighboring sites/qubits in an MPS one first contracts the gate unitary into the tensors at the two involved sites. This yields a new two-site tensor which is then broken back down into two single-site tensors using singular-value-decomposition (SVD). These new MPS tensors are connected by a bond with dimension at most a factor of 4 larger than the original bond between the sites. To ensure the bond dimensions do not grow exponentially as more gates are applied, the size of the final new bond is capped at a fixed $\chi_{\max}$ by retaining only the largest $\chi_{\max}$ singular values of the two-site tensor (and the resulting state is re-normalized). Discarding singular values in this way is likely to reduce the fidelity between the truncated state and the state without truncation, degrading the accuracy of the overall approximation. To mitigate this decay in accuracy the truncated tensors are optimized to maximize the fidelity between the state with and without truncation. After all gates have been applied in this way we arrive at a new input state $|\tilde{\psi}_A^{(\chi_{\max})}\rangle$ which is an MPS with a maximal bond dimension $\chi_{\max}$ which we then use to obtain an approximation to the k_s given by

$$\tilde{k}_s(\boldsymbol{\theta}_B, \boldsymbol{\theta}_B', \chi_{\max}) := |\langle\tilde{\psi}_A^{(\chi_{\max})}|U^\dagger(\boldsymbol{\theta}_B')U(\boldsymbol{\theta}_B)|\tilde{\psi}_A^{(\chi_{\max})}\rangle|^2. \tag{16}$$

If U_B is only finitely entangling (being either low width or low depth) it can be contracted into $|\tilde{\psi}_A\rangle$ with only a finite multiplicative increase in the MPS's bond-dimension. This means that if $\chi_{\max}$ is kept relatively small we can evaluate the approximate state kernel at a cost that scales at most as $\mathcal{O}(n\chi_{\max}^3 d)$, where $d = 2$ is the individual physical dimension of the n qubits [81].

While our proposed scheme approximates the input state to a MPS this is by no means the only classically-efficient representation one could use to approximate $|\psi_A\rangle$. For simplicity we only consider nearest-neighbor entangling gates; two-qubit gates between non-neighboring qubits can also be contracted into an MPS state by means of additional SWAP gates at the cost of large increases in the bond dimensions. Depending on the situation and structure of U_A, one could instead attempt to use one of the numerous other classically-efficient tensor network states. For example, a more complicated two- (or higher-) dimensional entangling structure may motivate the use of projected entangled pair states or tree tensor-networks [82].

By approximating $|\psi_A\rangle$ gate-by-gate with an upper limit on the retained bond-dimensions, our scheme avoids explicitly calculating the potentially intractable $|\psi_A\rangle$. However, truncating the bond-dimension inevitably leads to a reduction in the fidelity between $|\psi_A\rangle$ and $|\tilde{\psi}_A^{(\chi_{\max})}\rangle$. Instead, one could choose a classically-efficient parameterization of the input state $|\psi_A\rangle$ as a kernel hyperparameter to optimize. This would allow an input state to be chosen with reference to the observed data, rather than being the result of a series of approximations. However, optimizing only a handful of kernel hyperparameters can be extremely expensive. As we discuss in section 8.1, optimizing the marginal likelihood with respect to any hyperparameter involves repeated calculation of both the Gram matrix **K**, its inverse, and its derivative (see (24)). Unless the representation of $|\psi_A\rangle$ is extremely (hyper)parameter efficient, this would likely make such a scheme impractical.

5.1. GP regression with an MPS-approximated state kernel

To test our approximation strategy we compare the predictive accuracy of a GP model built with approximated state kernels of various $\chi_{\max}$ to the accuracy when using the full state kernel. We again consider the Hamiltonian in (12) with $J = h_z = 0.5$ and $h_x = -0.5$. To ensure that the full state kernel is tractable for large numbers of observations we use a brickwork ansatz with the same structure and open boundary conditions as in (1) but with 6 qubits and 20 layers of CX gates on alternating pairs of adjacent qubits separated by RY gates (a total of 50 CX gates and 106 RYs). To engineer a situation in which a low

Figure 7. Validation scores R_v^2 and log Bayes factors $\log \mathcal{B}$ for GP regression models with approximated state kernels of various bond dimensions learning the energy landscape $E(\boldsymbol{\theta})$ for the Hamiltonian in (12) with $J = h_z = 0.5$ and $h_x = -0.5$. The ansatz had the same structure as that in figure 1 but involved six qubits and a circuit depth of 20. Only the last two layers of RY gates were varied (a U_B with 10 parameters in total) and the remainder of the circuit (U_A) was initialized with uniform random gate angles and held constant. A classically-efficient approximation to the state kernel was produced by approximating the state $|\psi_A\rangle = U_A|0\rangle$ before the parameterized U_B to an MPS with maximal bond dimension $\chi_{\max}$. Results are also shown for simulations using the full un-approximated state kernel. The training and validation points are chosen uniformly at random (for each of the 10 parameters) in the range $[-\pi, \pi]$ and the corresponding energies $\boldsymbol{y}_t$ and $\boldsymbol{y}_v$ are evaluated noiselessly on a classical computer. (a) Horizontal axis is the size of the training data set N_t, vertical axis is the validation score (defined in (13)). (b) Horizontal axis is the size of the training data set N_t, vertical axis is the log Bayes factor (defined in (17)) versus the full un-approximated state kernel for the different approximated kernels. In both plots we show the median over 100 repeats (solid lines/dots) and the inter-quartile range of the data (filled).
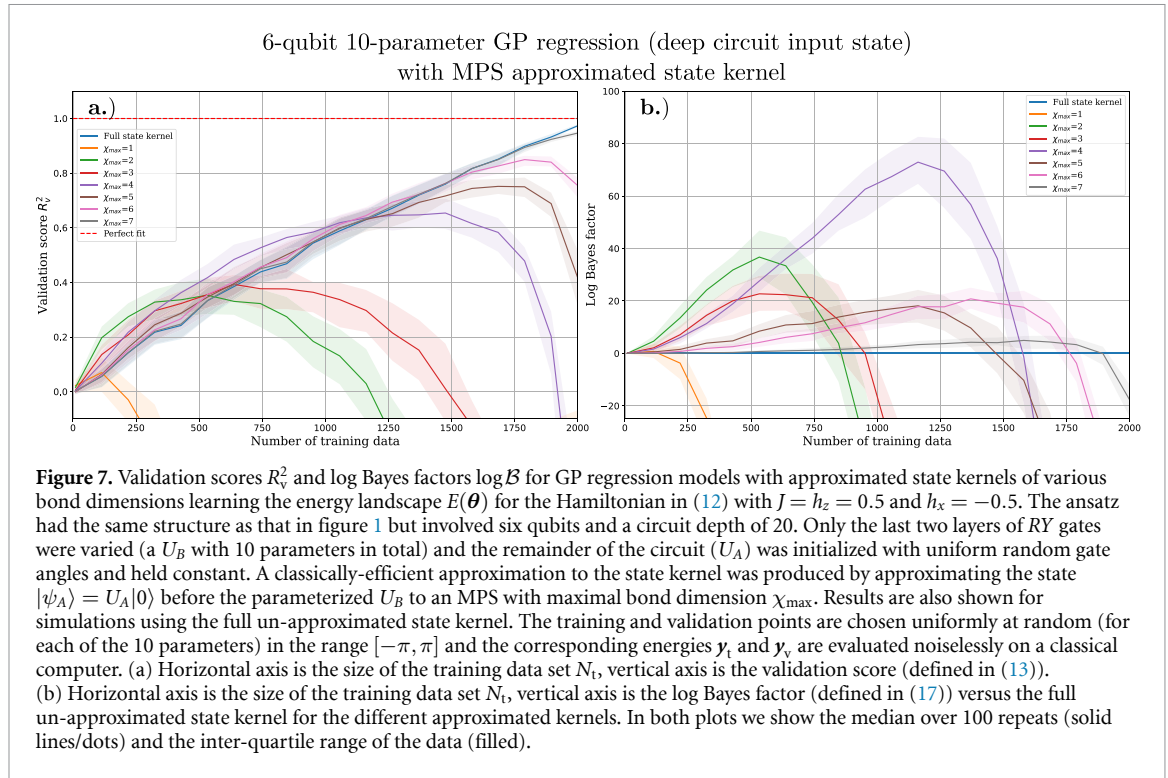
Table 1. Fidelities of MPS approximations with $|\psi_A\rangle$ used in simulations. For the maximum bond dimensions allowed in the MPS approximation, the fidelity $|\langle\psi_A|\tilde{\psi}_A^{(\chi_{\max})}\rangle|^2$ with the unapproximated $|\psi_A\rangle$ is shown.

$\chi_{\max}$	1	2	3	4	5	6	7
Fidelity	0.0054	0.2904	0.5788	0.8341	0.9382	0.9716	0.9976

$\chi_{\max}$ MPS approximation is unlikely accurate we choose the parameterized portion of the ansatz, U_B, as the last two layers of RYs (10 of which surround the final CX layer) and pick a fixed set of random gate angles for the remaining gates (which form U_A). For each simulation we generate two sets of random uniform gate angles $\mathbf{X}_t$ and $\mathbf{X}_v$ and their corresponding noiseless energies $\boldsymbol{y}_t$ and $\boldsymbol{y}_v$ as training and validation data. The size of the training data set is varied while the size of the validation set is fixed to $N_v = 100$.

Figure 7(a) shows the validation scores achieved by GP surrogates equipped with MPS approximated state kernels of different $\chi_{\max}$ compared to one equipped with the full state kernel. The fidelities of the MPS approximations with different maximum bond dimensions to $|\psi_A\rangle$ are shown in table 1. To help assess the relative suitability of the different GP models in explaining their training data, figure 7(b) shows the Log Bayes factor between the GP models with different approximated kernels and a GP model using the full state kernel. The Bayes factor, $\mathcal{B}(\mathbf{X}, \boldsymbol{y}, A, B) = p(\boldsymbol{y}|\mathbf{X}, A)/p(\boldsymbol{y}|\mathbf{X}, B)$, [83] between two statistical models A and B is defined as the ratio of the marginal likelihoods of a given set of training data under the two models. Its logarithm is

$$\log \mathcal{B}(\mathbf{X}, \boldsymbol{y}, A, B) = M_A(\mathbf{X}, \boldsymbol{y}) - M_B(\mathbf{X}, \boldsymbol{y}), \tag{17}$$

where $M_A(\mathbf{X}, \boldsymbol{y})$ and $M_B(\mathbf{X}, \boldsymbol{y})$ are the log marginal likelihoods of the data $(\mathbf{X}, \boldsymbol{y})$ with the models A and B respectively (which for the GP models we use is given by (23)). If the models A and B are assigned equal prior probabilities then the Bayes factor is equivalent to the ratio of the posterior probabilities of the two models $p(A|\boldsymbol{y}, \mathbf{X})/p(B|\boldsymbol{y}, \mathbf{X})$. This gives the Bayes factor a useful interpretation as how much more plausible one model is at explaining the data than the other [83]. A log Bayes factor $\mathcal{B}(\mathbf{X}, \boldsymbol{y}, A, B) > 1$ generally implies that model A is more strongly supported by the data than model B [83].

We see that MPS approximations to $|\psi_A\rangle$ with a higher $\chi_{\max}$ have a larger fidelity with $|\psi_A\rangle$ state and their GP models show increasingly similar performance to the GP which uses the full state kernel. Interestingly, the GP models that use approximated state kernels with lower $\chi_{\max}$ often have a higher validation score and Bayes factor (relative to the full state kernel) $\gg 1$, provided the size of the training data set is small. When dealing with small numbers of observations these models have both a higher predictive accuracy for unseen energies and are better supported by their training data than the model using k_s. This

may be because the induced feature spaces of these kernels have a smaller dimension than the full state kernel, being based on a restricted subspace of the n qubit Hilbert space. As a result, less data would be needed to span an appreciable portion of the feature space and so make accurate predictions.

For all the approximated kernels there comes a point where the validation score begins to decrease as more training data is presented. This happens when the size of the training data set becomes close to the dimension of the approximated kernel's feature space. As the energy function will not be completely linear in the approximated kernel's feature space, it becomes increasingly difficult to reconcile these different observations well with a linear model. This feature space saturation will also occur for the GP based on the full state kernel when the validation score reaches $R_v^2 = 1$ and the data spans the whole feature space. However, this is not an issue because the energy function is truly linear in the full state kernel's feature space and so energy observations can always be fully reconciled by a linear model in this space. Linear dependence will occur when the number of observations exceeds the feature space dimension at which point the Gram matrix **K** also becomes rank deficient; even before this point small eigenvalues will be present in **K** causing the GP model's predictions to become numerically unstable.

Typically a small regularizing offset, e.g. $\mathbf{K} \rightarrow \mathbf{K} + 10^{-10}\mathbf{I}$, is added to Gram matrices to ensure they are invertible by effectively adding additional dimensions to the feature vectors. Provided the offset is small, the regularization has little effect on models with well-suited kernels, like the full state kernel, as the objective function has minimal dependence on the additional components of the feature vectors introduced by the regularization. It can however help smooth out apparent inconsistencies encountered when using a kernel whose feature space is improperly aligned with the objective function by increasing the small eigenvalues present in **K** and reducing their disproportionate contribution to the predictions.
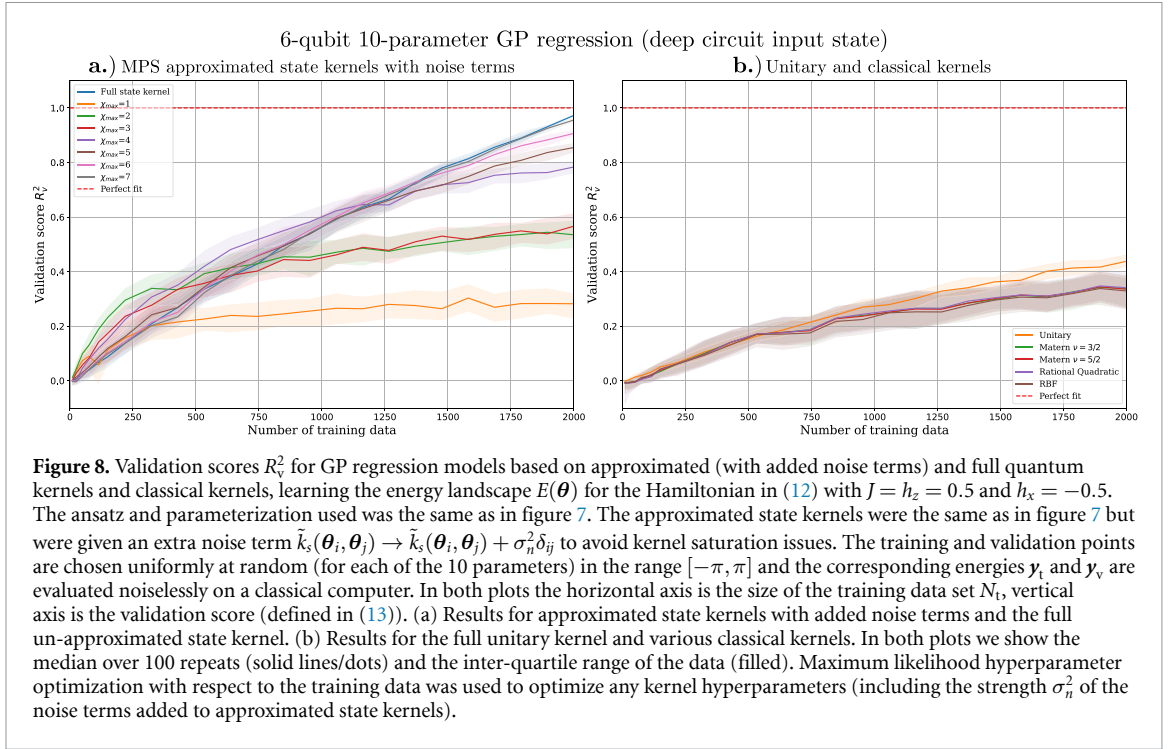
Choosing the value for this offset is key to ensuring accurate and numerically stable predictions; it must be sufficiently large to damp the contributions of small eigenvalues of **K** but not so large that it dominates the model. We see in figure 7(b) that when the feature space saturation occurs, degrading the validation score, the likelihood of the training data also drops precipitously. As this can be calculated from the training data alone, we can use this as a metric for calibrating the diagonal offset to avoid numerical instabilities. We do this by treating the weighting of the diagonal offset as a kernel hyperparameter σ_n^2 such that $\mathbf{K} \rightarrow \mathbf{K} + \sigma_n^2 I$ and optimizing this to maximize the marginal likelihood of the observations. Adding this hyperparameter is equivalent to assuming that we are dealing with observations corrupted by additive Gaussian noise with zero mean and variance σ_n^2; discrepancies between the observations and the assumption that they are drawn from a linear function in the kernel's feature space are effectively treated as an additional source of noise. When dealing with real noisy data, adding this noise term to avoid feature space saturation issues introduces negligible additional cost as a this term can be used to simultaneously deal with the observation noise.

Figure 8(a) shows the results of a repeat of the simulations in 7(a) when the approximated kernels are equipped with this additional noise term. For comparison with the classical kernels used in our other simulations, the plot in figure 8(b) shows the validation scores obtained for GP models using these classical kernels as well as the unitary kernel. Because the U_B used in these simulations only has a depth of 1 (a single layer of entangling gates) the unitary kernel can be calculated classically at low cost and so provides a benchmark against which the approximated state kernels should be compared.

When given an additional noise term, the approximated state kernels no longer suffer from an eventual drop in predictive accuracy due to feature space saturation. Instead, their performance continues to improve as more data is provided until the validation score eventually begins to saturate. For $\chi_{max} > 1$ the approximated state kernels all yield better performance than any of the classical kernels and the unitary kernel. Although the unitary kernel fares better than the classical kernels, its performance increases slowly with the number of training data due to it having a much larger feature space dimension than any of the (approximated) state kernels. Again we see for small amounts of data, the state kernel approximations with low χ_{max} have higher validation scores than the full state kernel, particularly with $\chi_{max} = 2$ for small N_t and $\chi_{max} = 4$ for intermediate N_t. This reinforces the idea that having a simpler model (with a smaller feature space) may provide accuracy benefits in the small-data regime as well as being easier to calculate. This raises the possibility that one could adaptively switch between kernels of varying complexity as more data is obtained, using comparative tools like the Bayes factor to decide which is most suitable at the current optimization stage.

6. Analysis of quantum kernel feature spaces

Many of our numerical results can be understood by analyzing the dimensions of the feature spaces induced by the state and unitary kernels. From (8), it is immediately apparent that the state kernel's feature space $\mathcal{F}_s$ is finite-dimensional, unlike the classical kernels we have considered which have infinite dimensional [63] feature spaces. This has the consequence that this kernel is *not universal* meaning that it cannot be used to

Figure 8. Validation scores R_v^2 for GP regression models based on approximated (with added noise terms) and full quantum kernels and classical kernels, learning the energy landscape $E(\boldsymbol{\theta})$ for the Hamiltonian in (12) with $J = h_z = 0.5$ and $h_x = -0.5$. The ansatz and parameterization used was the same as in figure 7. The approximated state kernels were the same as in figure 7 but were given an extra noise term $\tilde{k}_s(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) \to \tilde{k}_s(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) + \sigma_n^2 \delta_{ij}$ to avoid kernel saturation issues. The training and validation points are chosen uniformly at random (for each of the 10 parameters) in the range $[-\pi, \pi]$ and the corresponding energies $\boldsymbol{y}_t$ and $\boldsymbol{y}_v$ are evaluated noiselessly on a classical computer. In both plots the horizontal axis is the size of the training data set N_t, vertical axis is the validation score (defined in (13)). (a) Results for approximated state kernels with added noise terms and the full un-approximated state kernel. (b) Results for the full unitary kernel and various classical kernels. In both plots we show the median over 100 repeats (solid lines/dots) and the inter-quartile range of the data (filled). Maximum likelihood hyperparameter optimization with respect to the training data was used to optimize any kernel hyperparameters (including the strength σ_n^2 of the noise terms added to approximated state kernels).

approximate an unknown (well-behaved) function to an arbitrary degree of accuracy [57]. The feature map sends an input vector of p gate angles to vector with at most 3^p elements (which are linear sums of various Fourier components). Using the identity $A \otimes C^T |B\rangle\!\rangle = |ABC\rangle\!\rangle$ one can identify the set of (unnormalized) states $\{s_i |\rho_0\rangle\!\rangle\}$ whose inner products form $\boldsymbol{S}$ as a set of vectorized Hermitian operators on $\mathcal{H}$. As there can only be at most d^2 linearly independent Hermitian operators in a d-dimensional Hilbert space this puts an additional upper bound on the state kernel's feature space dimension. The maximal dimension for an n-qubit, p-parameter state kernel's feature space is therefore $\min(3^p, 4^n)$. In practice, the dimension of $\mathcal{F}_s$ may be lower than this due to redundancies or constraints on the set of Pauli rotations used in the ansatz (which reduces the span of the set of vectorized operators spanned by $s|\rho_0\rangle\!\rangle$ and so the rank of $\boldsymbol{S}$), or correlations between parameters (which would reduce the number of linearly independent components of $\boldsymbol{v}(\boldsymbol{\theta})$).

Assuming no ansatz redundancies, this implies that for an n-qubit ansatz we can hit the upper bound for $\dim \mathcal{F}_s$ imposed by the dimension of $\mathcal{H}$ (i.e. $3^p > 4^n$) using $\lceil \frac{2}{\log_2 3} n \rceil \approx \lceil 1.262n \rceil$ PPRs. However, in appendix B we show that in some cases the feature space dimension is also limited by constraints on the linear independence of the elements of $s|\rho\rangle\!\rangle$, meaning that more than $\lceil \frac{2}{\log_2 3} n \rceil$ parameterized rotations are required. We derive the following recursion relation for the maximal dimension $d_s^{(n,p)}$ of $\mathcal{F}_s$ for an n-qubit circuit with arbitrary fixed unitaries and p PPRs is given by

$$d_s^{(n,p)} := \min\left(4^n, 3d_s^{(n,p-1)}, \frac{4^n}{2} + d_s^{(n,p-1)}\right), \tag{18}$$

where for completeness we define the base-case $d_s^{(n,0)} = 1$ for all n. By finding p such that the $p + 1$th rotation first hits the linear independence bound, i.e. when $3^{p+1} > \frac{4^n}{2} + 3^p$, we find that total required number of parameterized rotations to obtain a maximal dimension feature space is equal to the upper bound $\lceil \frac{2}{\log_2 3} n \rceil$ if $\lceil \frac{2}{\log_2 3} n - \frac{1}{\log_2 3} \rceil = \lceil \frac{2}{\log_2 3} n \rceil - 1$, or $\lceil \frac{2}{\log_2 3} n \rceil + 1$ if $\lceil \frac{2}{\log_2 3} n - \frac{1}{\log_2 3} \rceil = \lceil \frac{2}{\log_2 3} n \rceil$.

The results shown in figure 2 give evidence of additional ansatz-imposed constraints on the state kernel's feature space dimension. With 16 parameters in the ansatz and 4 qubits, the absolute upper bound on the feature space dimension and thereby the number of training data points for $R_v^2 = 1$ should be $4^4 = 256$ however this instead occurs with just 136 observations. The ansatz used for these simulations (shown in figure 1) creates wavefunctions with real-valued amplitudes (in the computational basis), restricting the state space it can explore. The states it can produce have density matrices with components containing only even numbers of Y Pauli operators (and are generated by Pauli rotations with Ys acting on odd numbers of qubits) [36]. By counting the number of real-valued n-qubit Pauli operators we can obtain a tighter upper bound for the maximal feature space dimension of real ansatz circuits. As there are 3^{n-2j} combinations of $\{I, X, Z\}$ for all $\binom{n}{2j}$ placements of $2j$ Ys, this upper bound is given by $\sum_{j=0}^{\lfloor n/2 \rfloor} 3^{n-2j} \binom{n}{2j}$ which for $n = 4$ gives a dimension

of 136. This was confirmed by generating Gram matrices with this kernel for >136 uniform randomly chosen points and calculating their rank (and so the dimension of the kernel), again yielding a dimension of 136.

We also note that $I \otimes I \pm P \otimes P^*$ is a projection onto the ± 1 eigenspace of the Pauli operator $P \otimes P^*$ while $I \otimes iP^* - iP \otimes I = I \otimes iP^*(I \otimes I - P \otimes P^*)$ is a projection onto the -1 eigenspace of $P \otimes P^*$ followed by a (Clifford) unitary $I \otimes iP^*$. This means that if the initial state is a stabilizer state (such as the usual computational basis state $|0\dots0\rangle$) and all $R_1, \dots, R_p$ are Clifford operations, the elements of $s|\rho_0\rangle\rangle$ and the overlaps in S can be calculated efficiently on a classical computer as a $2n$-qubit stabilizer simulation [84, 85]. An obvious limitation however is that the feature space dimension scales exponentially in the number of parameterized rotations. This is not at all surprising as an initial stabilizer state with only a few non-Clifford rotations applied can be simulated 'efficiently' (i.e. at cost polynomial in the number of qubits n, but exponential in the number of non-Clifford gates) on a classical computer [86].

The feature map for the unitary kernel φ_u is qualitatively similar to φ_s, mapping input vectors into a finite-dimensional feature space vectors whose elements are sums of Fourier components. As $v(\theta)$ makes a reappearance here we again have an upper-bound to $\dim \mathcal{F}_u$ of 3^p for a k-Pauli rotation ansatz. We show in appendix C that there is a further upper bound due to the finite dimensionality of the n qubits' Hilbert space of $4^{2n} - 2(4^n - 1)$ meaning that the overall maximal feature space dimension for the unitary kernel with n qubits and p PPRs is

$$d_u^{(n,p)} := \min\left(4^{2n} - 2(4^n - 1), 3^p\right). \tag{19}$$

Broadly speaking, the global predictive accuracy of a kernel regression model depends on whether the objective function is linear in the kernel's feature space and, for finite-dimensional kernels, the fraction of the kernel's feature space spanned by the observed data [63]. If the (assumed noiseless) data fully spans the kernel feature space then any new point can be written as a linear combination of the feature mapped observations, meaning a GP model based on the kernel will be perfectly accurate. For both the state and unitary kernels the linearity condition holds (in the absence of noise) however the key difference between them is in the scaling of their feature spaces. In most practical applications an ansatz circuit will have a number of parameterized gates polynomial in the number of qubits $p \sim \text{poly}(n)$—(18) suggests that this would typically lead to a $\mathcal{F}_s$ of maximal dimension of 4^n (although this may be reduced by ansatz redundancies or specific circuit structure). To build a globally accurate (high predictive accuracy for all θ) state kernel-based regression model for the energy of an arbitrary n-qubit Hamiltonian we would need $\mathcal{O}(4^n)$ training data. In contrast, the unitary kernel's maximal feature space dimension scales like $O(4^{2n})$ and so one would require roughly quadratically more data to achieve the same global accuracy. This is less of an issue when considering local regression, which is equally important in BO, where the data only needs to span a subspace of the feature space local to the point of interest. This can be seen particularly in 3(b) in which the unitary kernel and classical kernels compete relatively well with the state kernel in highly localized GP regression.

7. Conclusion

The framework we have presented here allows on-device VQE of small systems to be performed with remarkably few energy evaluations. Our use of an GP surrogate model equipped with a classically-evaluated quantum kernel allows us to avoid many of the difficulties faced when implementing gradient-based optimization on NISQ devices. The two quantum kernels we have considered are based on the similarity (in terms of the fidelity) between two parameterized quantum states and two unitary operations. We have demonstrated that these kernels can be used to build very powerful GP surrogate models which are manifestly well-suited for regressing the cost function in VQE. For this regression task, these quantum kernel-based surrogate models exhibit significantly better predictive accuracy over many widely-used classical kernels. The advantage in predictive accuracy is particularly acute for the state fidelity-based kernel. Using this kernel one can build accurate regression models with far fewer samples than is needed by the unitary kernel or the classical kernels we have considered.

The advantage brought by a state kernel-based GP surrogate holds both on local and global parameter scales. Accuracy on these scales determines the effectiveness of the exploitative and exploratory phases of BO. Therefore, these results suggest that a state kernel-based GP surrogate can allow for fast Bayesian VQE with few energy evaluations. Due to the large number of kernel evaluations needed to build and optimize a quantum kernel based GP surrogate, it is impractical to evaluate these kernels on-device. Instead we propose that VQE problems at a scales where quantum kernel evaluation remains classically tractable can be solved quickly and with high accuracy using classically-evaluated quantum kernels. This allows one to make use of prior knowledge of the ansatz circuit while implicitly obtaining and making use of knowledge about a device's noise processes through the observed energy values. Through numerical experiments we have verified that this Bayesian approach to VQE is effective with both noiseless and realistic noisy energy observations.

The suitability of the state kernel for VQE stems from its induced feature space. We have explicitly constructed feature maps for the state and unitary kernels and demonstrated that the energy function in VQE is linear in both feature spaces. The dimension of the state kernel is roughly quadratically smaller than that of the unitary kernel (for typical ansatz circuits) and we believe this is the origin of the state kernel's advantage over the unitary kernel in GP regression and BO of VQE.

Finally, while we have shown our method for VQE is highly effective on small numbers of qubits, classical evaluation of the quantum kernels for more qubits quickly becomes intractable. By letting only individual blocks/layers of parameterized gates vary at any one time one can ensure that the unitary kernel remains tractable even for large numbers of qubits. For the state kernel, which is more useful for VQE, as the input state before varied gates must be considered to evaluate the kernel which may be infeasible. To remedy this we presented a scheme for approximating the state kernel so that it may be evaluated classically for large numbers of qubits. We ensure that the computational cost of evaluating this approximated state kernel remains bounded by approximating the input state to an MPS with a limited bond-dimension. Our simulations suggest that these approximated state kernels exhibit similar performance to the full state kernel in global GP regression provided the maximum allowed bond dimension is sufficiently large. Our results also suggest that GP models using an approximated state kernel with a lower maximum bond dimension (and so a smaller feature space dimension) can outperform the full state kernel in situations where few energy observations have been made.

8. Methods

8.1. GPs

The kernel-based surrogate model used in our Bayesian VQE simulations is an GP. An GP is a collection of random variables with the property that any finite subset has a joint multivariate normal distribution [63]. It is defined in terms of a covariance (a kernel) function $k(\cdot, \cdot)$ and a mean function $\mu(\cdot)$. Given a vector of m observed values $\boldsymbol{y} = (y_1, \ldots, y_m)^T$ of some unknown function at points $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)^T$ (expressed as an $m \times p$ matrix, where p in the input space dimension) then the joint distribution with a new point y^* at location $\boldsymbol{x}^*$ is [63]

$$\begin{bmatrix} \boldsymbol{y} \\ y^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \mu(\boldsymbol{x}^*) \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \boldsymbol{k} \\ \boldsymbol{k}^T & k(\boldsymbol{x}^*, \boldsymbol{x}^*) \end{bmatrix} \right), \tag{20}$$

where $\boldsymbol{\mu}$ is an m element mean vector with $\mu_i = \mu(\boldsymbol{x}_i)$, $\mathbf{K}$ is an $m \times m$ Gram matrix of pair-wise kernel evaluations between the observed inputs, $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and $\boldsymbol{k}$ is an m element vector of kernel evaluations between the new point $\boldsymbol{x}^*$ and the observed inputs $\mathbf{X}$, $k_i = k(\boldsymbol{x}^*, \boldsymbol{x}_i)$. If noise is present in the observations this is usually approximated to be normally distributed with zero mean. One can show analytically that for observations with normally distributed noise the kernel should be altered to $k(x_i, x_j) \to k(x_i, x_j) + \sigma_n^2 \delta_{ij}$, where σ_n^2 is a noise variance hyperparameter and the indices of δ_{ij} correspond to indices in the training data meaning that $\mathbf{K} \to \mathbf{K} + \sigma_n^2 I$ but $\boldsymbol{k}$ and $k(\boldsymbol{x}^*, \boldsymbol{x}^*)$ are unchanged. Marginalizing over the observed data, the resulting posterior distribution is itself a normal distribution with a posterior mean (a prediction) $\mathbb{E}[y^* | \boldsymbol{x}^*, \mathbf{X}, \boldsymbol{y}] = \hat{y}(\boldsymbol{x}^*)$ given by

$$\hat{y}(\boldsymbol{x}^*) = \mu(\boldsymbol{x}^*) + \boldsymbol{k}^T (\mathbf{K} + \sigma_n^2 I)^{-1} (\boldsymbol{y} - \boldsymbol{\mu}) \tag{21}$$

and a posterior variance $\mathrm{Var}[y^*(\boldsymbol{x}^*)] = \Delta y(\boldsymbol{x}^*)^2$ of

$$\Delta y(\boldsymbol{x}^*)^2 = k(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{k}^T (\mathbf{K} + \sigma_n^2 I)^{-1} \boldsymbol{k}. \tag{22}$$

The performance of GP regression depends on the suitability of the kernel function in describing the unknown function. Unless one has significant prior knowledge of the problem at hand, choosing a suitable kernel is often difficult. As a result, problem-agnostic approaches to GP regression often make use of flexible kernels with internal hyperparameters which are varied to fit observed data. A common approach is to find hyperparameter values that maximize the marginal likelihood of the observed data (often the log-likelihood) [63]. Thanks to the normality of the GP's distributions, the marginal log-likelihood $M(\mathbf{X}, \boldsymbol{y}) = \log p(\boldsymbol{y} | \mathbf{X}, \alpha)$ can be calculated analytically for a kernel with hyperparameter α as

$$M(\mathbf{X}, \boldsymbol{y}) = -\frac{1}{2} (\boldsymbol{y} - \boldsymbol{\mu})^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} (\boldsymbol{y} - \boldsymbol{\mu}) - \frac{1}{2} \log \det[\mathbf{K} + \sigma_n^2 \mathbf{I}] - \frac{n}{2} \log 2\pi. \tag{23}$$

The gradient of this quantity with respect to a kernel hyperparameter α is given by

$$\frac{\partial M(\mathbf{X},\boldsymbol{y})}{\partial \alpha} = \frac{1}{2}\mathrm{Tr}\{(\boldsymbol{\beta}\boldsymbol{\beta}^T - \mathbf{K}_{\sigma_n}^{-1})\frac{\partial \mathbf{K}_{\sigma_n}}{\partial \alpha}\},$$

$$\text{where } \boldsymbol{\beta} = \mathbf{K}_{\sigma_n}^{-1}\boldsymbol{y}, \ \mathbf{K}_{\sigma_n} = \mathbf{K} + \sigma_n^2\mathbf{I}, \tag{24}$$

and so this quantity can be maximized through gradient ascent at a cost primarily dictated by the calculation of $K_{\sigma_n}^{-1}$ if the kernel evaluations and their gradients are inexpensive to calculate.

The posterior variance (22) depends only on kernel evaluations and not the observations $\boldsymbol{y}$. If the kernel is fixed, a re-scaling of the training data of the problem $\boldsymbol{y} \to \alpha\boldsymbol{y}$ will leave the posterior variance unchanged whereas it should be a factor of α^2 larger. To allow the posterior variance to scale with the size of the observations an external 'signal variance' hyperparameter σ^2 is often added to the kernel $k(\boldsymbol{x},\boldsymbol{x}') \to \sigma^2 k(\boldsymbol{x},\boldsymbol{x}')$. We see from (21) and (22) that this does not change posterior mean prediction (up to a re-scaling of σ_n^2) but the posterior variance is scaled by σ^2. If the base kernel has no hyperparameters (which is true for the quantum kernels we consider) and the observed data is assumed to be noiseless then it is simple to show the marginal likelihood of the training data is maximized when $\sigma_{\text{noiseless}}^2 = (\boldsymbol{y} - \boldsymbol{\mu})\mathbf{K}^{-1}(\boldsymbol{y} - \boldsymbol{\mu})^T/m$ (where $\mathbf{K}$ is the Gram matrix for the kernel without σ^2 and m is the number of observations) – this often serves as a good starting point for maximum likelihood optimization even if noise is present.

The mean function μ is often set to a constant value, most commonly $\mu(\boldsymbol{x}) = 0$ for all $\boldsymbol{x}$, as it is primarily the covariance function which defines a GP's properties; in most cases μ only has a significant impact at points that are far away (with respect to the kernel) from any observed data. However (as shown in section 6) for many ansatz circuits the expected value (taken over the gate angles) of the noiseless energy $\mathbb{E}_{\boldsymbol{\theta}}[E(\boldsymbol{\theta})]$ is given by $\mathrm{Tr}\{H\}$ meaning that one should set $\mu(\boldsymbol{x}) = \mathrm{Tr}\{H\}$. Alternatively, one can remove this diagonal offset from the Hamiltonian by using $H' = H - \mathrm{Tr}\{H\}I$ and set $\mu = 0$. We suggest that this should be done generally as calculating or estimating the average energy for a complicated ansatz will be difficult, especially if device noise is present. Removing this offset is good practice in general as a large diagonal offset in H can lead to improper conclusions on the effectiveness of an optimization scheme.

In our simulations we build the GP models using two quantum kernels (described in section 3) and a set of classical kernel functions. These classical kernels are as follows; Matern kernels with smoothness hyperparameter $\nu = 5/2$ and $\nu = 3/2$, the radial basis function (RBF) kernel, and the rational quadratic (RQ) kernel. Their kernel functions are:

$$k_{Matern}(\boldsymbol{x},\boldsymbol{x}';\nu = 5/2, l) = \left(1 + \frac{\sqrt{5}}{l}d(\boldsymbol{x},\boldsymbol{x}') + \frac{5}{3l}d(\boldsymbol{x},\boldsymbol{x}')^2\right)\exp\left\{-\frac{\sqrt{5}}{l}d(\boldsymbol{x},\boldsymbol{x}')\right\}, \tag{25}$$

$$k_{Matern}(\boldsymbol{x},\boldsymbol{x}';\nu = 3/2, l) = \left(1 + \frac{\sqrt{3}}{l}d(\boldsymbol{x},\boldsymbol{x}')\right)\exp\left\{-\frac{\sqrt{3}}{l}d(\boldsymbol{x},\boldsymbol{x}')\right\}, \tag{26}$$

$$k_{RBF}(\boldsymbol{x},\boldsymbol{x}';l) = \exp\left\{-\frac{d(\boldsymbol{x},\boldsymbol{x}')^2}{2l^2}\right\}, \text{ and} \tag{27}$$

$$k_{RQ}(\boldsymbol{x},\boldsymbol{x}';l,\alpha) = \left(1 + \frac{d(\boldsymbol{x},\boldsymbol{x}')^2}{2\alpha l^2}\right)^{-\alpha}, \tag{28}$$

where $d(\boldsymbol{x},\boldsymbol{x}')$ is the squared Euclidean distance between points $\boldsymbol{x}$ and $\boldsymbol{x}'$, l are length scale hyperparameters, and α is a scale mixture hyperparameter for the RQ kernel.

These classical kernels require optimization of internal hyperparameters (a length scale l and, for the RQ kernel, a mixing parameter α) to ensure that the kernel's infinite-dimensional feature space most-succinctly describes the objective function. By contrast, the state and unitary kernel have finite-dimensional feature spaces in which the (noiseless) energy function is manifestly linear which means no hyperparameter optimization is required for a GP model to make sensible predictions about the energy. Hyperparameter optimization is often the most computationally intensive part of BO as it requires repeated calculation of the Gram matrix $\mathbf{K}$ (see (23) and (24)) and its inverse. If we have a kernel with no hyperparameters other than those for the signal variance and noise $k(\boldsymbol{x}_i,\boldsymbol{x}_j) = \sigma^2 k_0(\boldsymbol{x}_i,\boldsymbol{x}_j) + \sigma_n^2\delta_{ij}$, where k_0 is fixed, then the Gram matrix for the fixed part of the kernel $\mathbf{K}_0$ only needs to be calculated once. This simplifies the hyperparameter optimization as the total Gram matrix $\mathbf{K} = \sigma^2\mathbf{K}_0 + \sigma_n^2 I$ can be calculated easily for σ^2 and σ_n^2 from $\mathbf{K}_0$. Additionally, the eigenvalues/vectors of the full $\mathbf{K}$ can be found analytically in terms of those for $\mathbf{K}_0$, simplifying calculation of the inverse Gram matrices needed to both optimize the hyperparameters and make predictions from the model. This ultimately means that the state and unitary kernels only need to be evaluated once between any two data points within a training data (up to a re-scaling by σ^2 and a noise offset), in contrast to the classical kernels whose Gram matrices must be entirely recalculated whenever an internal hyperparameter is adjusted.

8.2. BO

In BO one attempts to minimize an unknown expensive-to-evaluate objective function by assuming it is randomly drawn from some family of functions [67]. A prior distribution is chosen to encode any initial beliefs about this function, usually that it is sampled from an GP. From observations of the function's values and the prior, a posterior distribution over the chosen family of functions is constructed (the posterior distribution of the GP), quantifying how likely it is that a given function could have produced the observations. An acquisition function is then calculated from the posterior distribution and is maximized/minimized to determine the next point to query. Once the new point has been queried, the posterior distribution is reconstructed with this new data and the process is repeated until convergence to the objective function global minima.

The acquisition function quantifies how 'promising' the querying of a given an unseen point appears for minimizing the objective function. These functions are often designed to balance the exploitation of regions of the parameter space that have already shown good objective function values with the exploration of areas in the parameter space where observations are sparse. By only querying parameter points that appear likely to yield a significant improvement in the objective function, BO can allow global minima to be found with remarkably few iterations. In our simulations we will use the Expected Improvement acquisition function [67] given (for minimization) by $\text{EI}(\boldsymbol{x}) = \mathbb{E}[\max(y_{\text{best}} - \hat{y}(\boldsymbol{x}) + \xi, 0)|\boldsymbol{x}^*, \mathbf{X}, \boldsymbol{y}]$, where the expectation is taken over the surrogate model's posterior distribution, $y_{\text{best}} = \min\{\boldsymbol{y}\}$ is the lowest observation seen so far, and ξ is an exploration hyperparameter. Using of the analytically tractable GP posterior distribution (i.e. normally distributed with mean/variance given by (21) and (22)) we can derive an explicit formula for this [67]:

$$\text{EI}(\boldsymbol{x}) := (y_{\text{best}} - \hat{y}(\boldsymbol{x}) + \xi)\Phi(Z) + \Delta y(\boldsymbol{x})\phi(Z), \text{ where } Z = \frac{y_{\text{best}} - \hat{y}(\boldsymbol{x}) + \xi}{\Delta y(\boldsymbol{x})}, \tag{29}$$

Φ and ϕ are the cumulative distribution and probability density functions for the standard normal distribution respectively. Expected improvement-based optimization can sometimes be overly greedy and exploitative and so a hyperparameter ξ has been added to control the ratio of exploration to exploitation. The ξ parameter allows more positive observations (up to where $y^*(\boldsymbol{x}) = y_{\text{best}} + \xi$) to count as an improvement, meaning the objective function encourages exploration further away from the point where y_{best} was observed. In our simulations we used $\xi = 0.01$ and optimized the surrogate model using the L-BFGS-B gradient-descent algorithm [87].
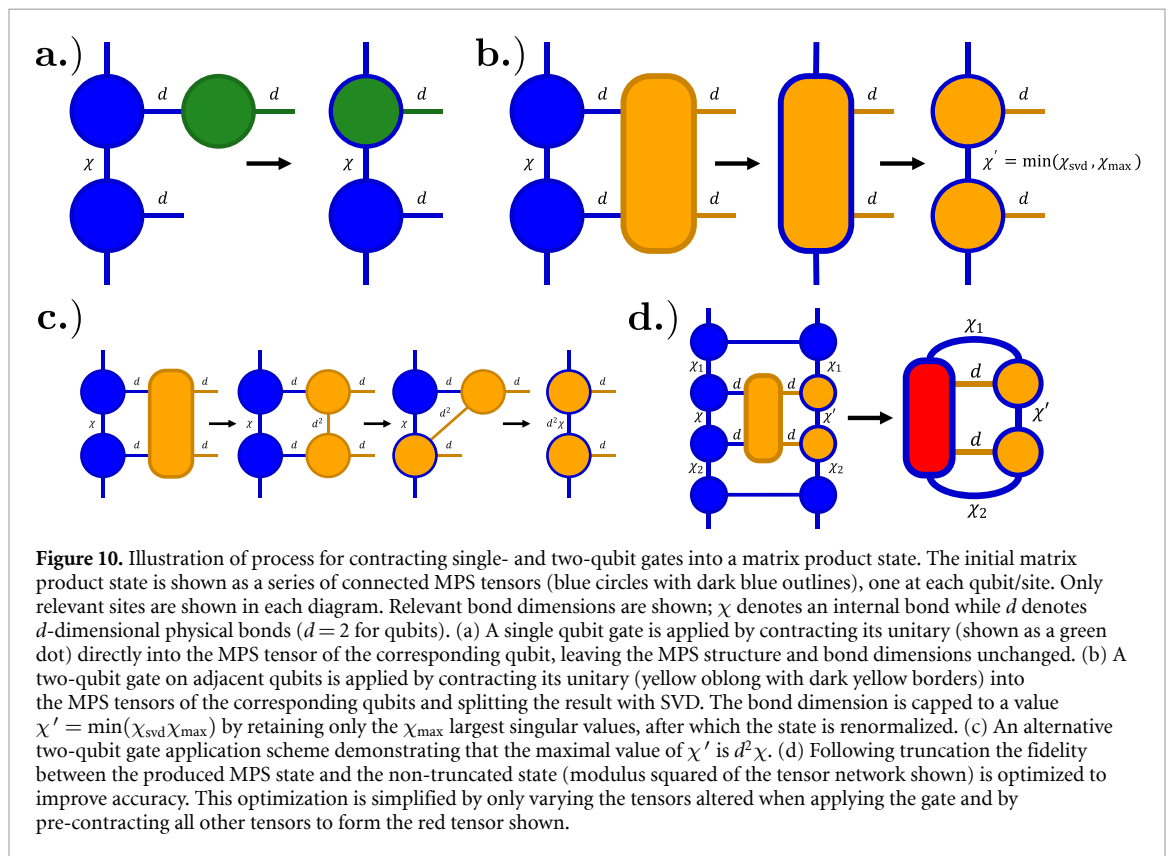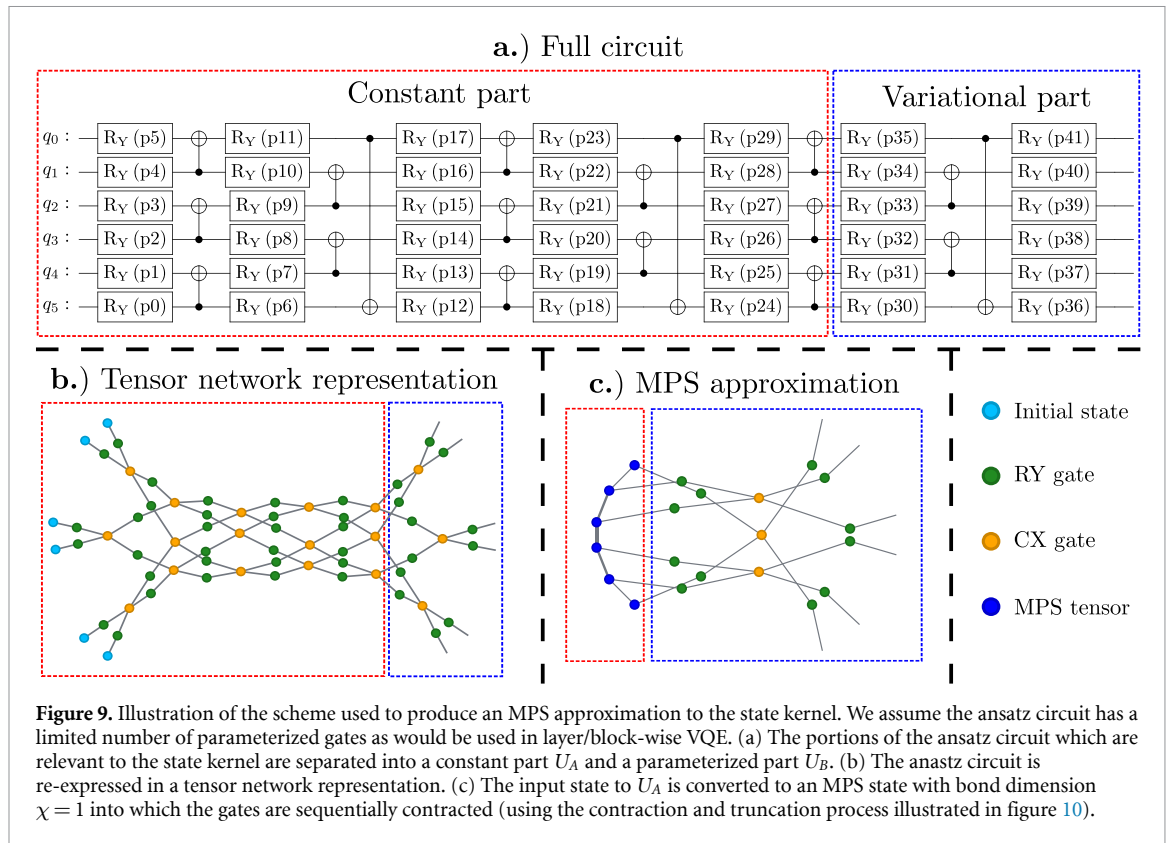
While it is possible to run BO starting with an observation at a single randomly chosen parameter point, more typically a collection of initial points are used to ensure an initial explorative phase. While there are many designs for these points [88] we use the simplest; random uniform sampling across the parameter space.

8.3. MPS approximation to the state kernel

Our approximation scheme for the state kernel relies on the parameterized circuit being of the form $|\psi(\boldsymbol{\theta}_B)\rangle = U_A(\boldsymbol{\theta}_A)U(\boldsymbol{\theta}_B)U(\boldsymbol{\theta}_C)|\mathbf{0}\rangle$ where $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_C$ are fixed, so that the state kernel can be written in the form (15). We also require that U_B is simple enough that its action on an MPS state of relatively low bond dimension can be simulated classically. If this is the case then by approximating $|\psi_A\rangle$ to an MPS $|\tilde{\psi}_A^{(\chi_{\max})}\rangle$, we can achieve a classically tractable approximation to the state kernel.

Figure 9 illustrates our approximation scheme. We begin (9(a)) with a quantum circuit composed of a constant part (U_A) and a variational part (U_B) whose parameters are to be optimized. For simplicity we omit any U_C that lies in the future light-cone of the variational part of the circuit as this has no effect on the state kernel. The circuit is then converted into a tensor-network representation, which can be done at a cost proportional to the number of gates (an example is shown in 9(b)). The initial state $|\mathbf{0}\rangle$ is then converted to an MPS with bond dimension $\chi = 1$ and each gate in the circuit is contracted into the MPS to produce $|\tilde{\psi}_A^{(\chi_{\max})}\rangle$ (shown in 9(c)).

Figure 10 illustrates the process of contracting a gate G into an MPS $|\psi_{\text{MPS}}\rangle$ (with physical dimension d). As we only consider single- and two-qubit gates (outlined in green and yellow respectively) only two MPS tensors with shared bond dimension χ are shown (outlined in blue). Vertical open bonds indicate connections to the rest of the MPS state. In 10(a) a single qubit unitary is contracted directly into the MPS tensor at the corresponding site/qubit, altering this tensor but leaving the bond dimensions unchanged. Figure 10(b) shows the contraction of a two-qubit unitary which acts on adjacent sites. The two-qubit unitary is fully contracted into the MPS tensors yielding a joint tensor for the two sites. Singular value decomposition (SVD) is used to split this joint tensor into two new MPS tensors with a new bond dimension χ'. The size of χ' depends on the specifics of the unitary and the state to which it is applied but takes a maximum value of $d^2\chi$. Figure 10(c) shows this diagrammatically by giving an alternative scheme for contracting the two-qubit

Figure 9. Illustration of the scheme used to produce an MPS approximation to the state kernel. We assume the ansatz circuit has a limited number of parameterized gates as would be used in layer/block-wise VQE. (a) The portions of the ansatz circuit which are relevant to the state kernel are separated into a constant part U_A and a parameterized part U_B. (b) The anastz circuit is re-expressed in a tensor network representation. (c) The input state to U_A is converted to an MPS state with bond dimension $\chi = 1$ into which the gates are sequentially contracted (using the contraction and truncation process illustrated in figure 10).

Figure 10. Illustration of process for contracting single- and two-qubit gates into a matrix product state. The initial matrix product state is shown as a series of connected MPS tensors (blue circles with dark blue outlines), one at each qubit/site. Only relevant sites are shown in each diagram. Relevant bond dimensions are shown; χ denotes an internal bond while d denotes d-dimensional physical bonds ($d = 2$ for qubits). (a) A single qubit gate is applied by contracting its unitary (shown as a green dot) directly into the MPS tensor of the corresponding qubit, leaving the MPS structure and bond dimensions unchanged. (b) A two-qubit gate on adjacent qubits is applied by contracting its unitary (yellow oblong with dark yellow borders) into the MPS tensors of the corresponding qubits and splitting the result with SVD. The bond dimension is capped to a value $\chi' = \min(\chi_{\text{svd}}\chi_{\text{max}})$ by retaining only the χ_{max} largest singular values, after which the state is renormalized. (c) An alternative two-qubit gate application scheme demonstrating that the maximal value of χ' is $d^2\chi$. (d) Following truncation the fidelity between the produced MPS state and the non-truncated state (modulus squared of the tensor network shown) is optimized to improve accuracy. This optimization is simplified by only varying the tensors altered when applying the gate and by pre-contracting all other tensors to form the red tensor shown.

gate. Here the unitary is first decomposed using SVD into two tensors connected by a bond of dimension d^2. Each resulting tensor is contracted into their connected MPS tensor yielding an MPS state with a two bonds between the new MPS tensors (of size χ and d^2). These bonds are combined to form a single bond of dimension $d^2\chi$. Two qubit gates on non-neighboring sites can be contracted using the same scheme by applying a chain of SWAP gates (as described in [89]) but can significantly increase the bond dimensions.

Applying many two-qubit gates will lead to exponential growth in the MPS's bond dimensions. To avoid this we truncate χ' by only retaining the χ_{max} largest singular values obtained from the SVD in the final stage of 10(b). Following truncation the MPS is re-normalized, yielding an approximation $|\tilde{\psi}_{MPS}\rangle$ to the transformed state $G|\psi_{MPS}\rangle$ with a maximum bond dimension of χ_{max}. If significant singular values of the joint tensor are discarded, the fidelity between $|\tilde{\psi}_{MPS}\rangle$ and the target state $G|\psi_{MPS}\rangle$ will be degraded. We partially mitigate this by maximizing the fidelity $|\langle\tilde{\psi}_{MPS}|G|\psi_{MPS}\rangle|^2$ between these two states, optimizing over $|\tilde{\psi}_{MPS}\rangle$. This procedure can be greatly simplified by only optimizing over the two MPS that are altered when producing $|\tilde{\psi}_{MPS}\rangle$ (shown as yellow dots with a blue outline in figures 10(b) and (d)). By first contracting over the tensors which are not optimized (creating the tensor shown with red fill figure 10(d) this optimization can be performed extremely efficiently, involving contractions over just 4 tensors. These initial contractions can also be further simplified by first canonicalizing $|\psi_{MPS}\rangle$ around the two qubits acted upon by the unitary [82] (provided the MPS has open boundaries to allow this). Once all gates in the circuit have been contracted into the MPS using the scheme we have outlined (with the necessary truncation and fidelity optimization) the resulting MPS state $|\tilde{\psi}_A^{(\chi_{max})}\rangle$ is used as the input state for the approximated state kernel given in (15).

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

Acknowledgments

Funding

Appendix A. Derivation of feature spaces

The state kernel for the ansatz in (4) is given by:

$$
\begin{aligned}
k_s(\boldsymbol{\theta}, \boldsymbol{\theta}') &= \text{Tr}\{U(\boldsymbol{\theta}')\rho_0 U^\dagger(\boldsymbol{\theta}')U(\boldsymbol{\theta})\rho_0 U^\dagger(\boldsymbol{\theta})\} \\
&= \langle\langle U(\boldsymbol{\theta}')\rho_0 U^\dagger(\boldsymbol{\theta}')|U(\boldsymbol{\theta})\rho_0 U^\dagger(\boldsymbol{\theta})\rangle\rangle \\
&= \langle\langle \rho_0|(U^\dagger(\boldsymbol{\theta}')\otimes U^T(\boldsymbol{\theta}'))(U(\boldsymbol{\theta})\otimes U^*(\boldsymbol{\theta}))|\rho_0\rangle\rangle ,
\end{aligned}
\tag{A1}
$$

where $\rho_0 = |\mathbf{0}\rangle\langle\mathbf{0}|$, $|A\rangle\rangle$ denotes the vectorization of operator A and we have made use of the identities $\langle\langle A|B\rangle\rangle = \text{Tr}\{A^\dagger B\}$ and $|ABC\rangle\rangle = A\otimes C^T|B\rangle\rangle$. At this point, we could stop and identify the state kernel's feature vectors as quantum states on $\mathcal{H}\otimes\mathcal{H}^*$ (where $\mathcal{H}$ is the Hilbert space of the qubits) of the form $U(\boldsymbol{\theta})\otimes U^*(\boldsymbol{\theta})|\rho_0\rangle\rangle$. However in doing so would have to construct these states explicitly, making analysis and comparison to classical kernels difficult. By expanding each Pauli rotation as $e^{-iP_i\theta_i/2} = \cos(\frac{\theta_i}{2})I - i\sin(\frac{\theta_i}{2})P_i$ we can express $U(\boldsymbol{\theta})$ in a convenient form as the product of k inner-products of 2-dimensional vectors $\boldsymbol{u}_q = (R_q, -iP_qR_q)^T$ and $\boldsymbol{a}(\theta) = (\cos\theta, \sin\theta)^T$

$$
U(\boldsymbol{\theta}) = R_{p+1}(\boldsymbol{u}_p^T\boldsymbol{a}(\theta_p/2))\ldots(\boldsymbol{u}_1^T\boldsymbol{a}(\theta_1/2)).
\tag{A2}
$$

This allows us to separate out the parameter-dependent part of the kernel from the ansatz-dependent part. We can then write the vectorized unitary $U(\boldsymbol{\theta})\otimes U^*(\boldsymbol{\theta})$ as

$$
U(\boldsymbol{\theta})\otimes U^*(\boldsymbol{\theta}) = R_{p+1}\otimes R_{p+1}^*(\boldsymbol{u}_p^T\boldsymbol{a}(\theta_p/2)\otimes(\boldsymbol{u}_p^*)^T\boldsymbol{a}(\theta_p/2))\ldots(\boldsymbol{u}_1^T\boldsymbol{a}(\theta_1/2)\otimes(\boldsymbol{u}_1^*)^T\boldsymbol{a}(\theta_1/2)).
\tag{A3}
$$

The tensor products of the form $\boldsymbol{u}_q^T\boldsymbol{a}(\theta_q/2)\otimes(\boldsymbol{u}_q^*)^T\boldsymbol{a}(\theta_q/2)$ can be expanded and simplified into the following inner product

$$
\boldsymbol{u}_q^T\boldsymbol{a}(\theta_q/2)\otimes(\boldsymbol{u}_q^*)^T\boldsymbol{a}(\theta_q/2) = \frac{1}{2}\begin{pmatrix}(I\otimes I + P_q\otimes P_q^*)R_q\otimes R_q^* \\ (I\otimes iP_q^* - iP_q\otimes I)R_q\otimes R_q^* \\ (I\otimes I - P_q\otimes P_q^*)R_q\otimes R_q^*\end{pmatrix}^T\begin{pmatrix}1 \\ \sin\theta_q \\ \cos\theta_q\end{pmatrix} = \boldsymbol{s}_q^T\boldsymbol{v}_q(\theta_q),
\tag{A4}
$$

where $\boldsymbol{v}_q = (1, \sin\theta_q, \cos\theta_q)^T$ and $\boldsymbol{s}_q$ is the vector of sums of unitaries above. The overall vectorized unitary is

$$U(\boldsymbol{\theta}) \otimes U^*(\boldsymbol{\theta}) = R_{p+1} \otimes R_{p+1}^* (\boldsymbol{s}_p^T \boldsymbol{v}_p(\theta_p)) \dots (\boldsymbol{s}_1^T \boldsymbol{v}_1(\theta_1)) = R_{p+1} \otimes R_{p+1}^* \boldsymbol{s}^T \boldsymbol{v}(\boldsymbol{\theta}), \tag{A5}$$

where $\boldsymbol{s}$ is given by

$$\boldsymbol{s} = \frac{1}{2^p} \left[\begin{pmatrix} (I \otimes I + P_p \otimes P_p^*) R_p \otimes R_p^* \\ (I \otimes iP_p^* - iP_p \otimes I) R_p \otimes R_p^* \\ (I \otimes I - P_p \otimes P_p^*) R_p \otimes R_p^* \end{pmatrix} \otimes_K \dots \otimes_K \begin{pmatrix} (I \otimes I + P_1 \otimes P_1^*) R_1 \otimes R_1^* \\ (I \otimes iP_1^* - iP_1 \otimes I) R_1 \otimes R_1^* \\ (I \otimes I - P_1 \otimes P_1^*) R_1 \otimes R_1^* \end{pmatrix} \right] \tag{A6}$$

and $\boldsymbol{v}(\boldsymbol{\theta})$ by

$$\boldsymbol{v}(\boldsymbol{\theta}) = \begin{pmatrix} 1 \\ \sin(\theta_p) \\ \cos(\theta_p) \end{pmatrix} \otimes_K \dots \otimes_K \begin{pmatrix} 1 \\ \sin(\theta_1) \\ \cos(\theta_1) \end{pmatrix}. \tag{A7}$$

Here we have made a distinction between the tensor products $\otimes$ that denote operators on the physical and conjugated multi-qubit Hilbert spaces and the Kronecker products (for vectors with matrix entries) $\otimes_K$ between the p 'sub-vectors' which determine the dimension of the resulting kernels' feature spaces. The state kernel is then

$$\begin{aligned} k_s(\boldsymbol{\theta}, \boldsymbol{\theta}') &= \langle\!\langle \rho_0 | \boldsymbol{v}^T(\boldsymbol{\theta}') \boldsymbol{s}^* \boldsymbol{s}^T \boldsymbol{v}(\boldsymbol{\theta}) | \rho_0 \rangle\!\rangle \\ &= \boldsymbol{v}^T(\boldsymbol{\theta}') \langle\!\langle \rho_0 | \boldsymbol{s}^* \boldsymbol{s}^T | \rho_0 \rangle\!\rangle \, \boldsymbol{v}(\boldsymbol{\theta}) \\ &= \boldsymbol{v}^T(\boldsymbol{\theta}') S \boldsymbol{v}(\boldsymbol{\theta}), \end{aligned} \tag{A8}$$

where each element of $\boldsymbol{s}^*$ is the Hermitian conjugate of the corresponding element in $\boldsymbol{s}$ so that $(\boldsymbol{S})_{ij} = \langle\!\langle \rho_0 | s_i^\dagger s_j | \rho_0 \rangle\!\rangle$. (Note that the unitary $R_{p+1} \otimes R_{p+1}^*$ has been cancelled by its Hermitian conjugate.)

The unitary kernel evaluated for this ansatz takes the following form:

$$\begin{aligned} k_u(\boldsymbol{\theta}, \boldsymbol{\theta}') &= \frac{1}{d^2} |\mathrm{Tr}\{U^\dagger(\boldsymbol{\theta}') U(\boldsymbol{\theta})\}|^2 \\ &= \frac{1}{d^2} \mathrm{Tr}\{U^\dagger(\boldsymbol{\theta}') U(\boldsymbol{\theta})\} \mathrm{Tr}\{U^T(\boldsymbol{\theta}') U^*(\boldsymbol{\theta})\} \\ &= \frac{1}{d^2} \mathrm{Tr}\{[U^\dagger(\boldsymbol{\theta}') \otimes U^T(\boldsymbol{\theta}')][U(\boldsymbol{\theta}) \otimes U^*(\boldsymbol{\theta})]\}. \end{aligned} \tag{A9}$$

Using the decomposition of this unitary given in (A5), we can write the unitary kernel as

$$\begin{aligned} k_u(\boldsymbol{\theta}, \boldsymbol{\theta}') &= \frac{1}{d^2} \mathrm{Tr}\{\boldsymbol{v}^T \boldsymbol{s}^* \boldsymbol{s}^T \boldsymbol{v}(\boldsymbol{\theta})\} \\ &= \boldsymbol{v}^T(\boldsymbol{\theta}') \frac{\mathrm{Tr}\{\boldsymbol{s}^* \boldsymbol{s}^T\}}{d^2} \boldsymbol{v}(\boldsymbol{\theta}) \\ &= \boldsymbol{v}^T(\boldsymbol{\theta}') T \boldsymbol{v}(\boldsymbol{\theta}), \end{aligned} \tag{A10}$$

where again each element of $(\boldsymbol{s}^*)_i = s_i^\dagger$ so that $\boldsymbol{T} = \mathrm{Tr}\{\boldsymbol{s}^* \boldsymbol{s}^T\}/d^2$. (Again note that the R_{p+1} terms have cancelled.)

Using the form for $U(\boldsymbol{\theta}) \otimes U^*(\boldsymbol{\theta})$ given in (A5) we also can now write the noiseless energy function $E(\boldsymbol{\theta})$ in terms of the $\boldsymbol{v}(\boldsymbol{\theta})$ vector, and thereby the feature space vectors, as

$$\begin{aligned} E(\boldsymbol{\theta}) &= \mathrm{Tr}\{H U(\boldsymbol{\theta}) \rho_0 U^\dagger(\boldsymbol{\theta})\} = \langle\!\langle H | U(\boldsymbol{\theta}) \otimes U^*(\boldsymbol{\theta}) | \rho_0 \rangle\!\rangle \\ &= \langle\!\langle R_{p+1}^\dagger H R_{p+1} | \boldsymbol{s}^T \boldsymbol{v}(\boldsymbol{\theta}) | \rho_0 \rangle\!\rangle \\ &= \langle\!\langle R_{p+1}^\dagger H R_{p+1} | \boldsymbol{s}^T | \rho_0 \rangle\!\rangle \, \boldsymbol{v}(\boldsymbol{\theta}) \\ &= \boldsymbol{h}^T \boldsymbol{v}(\boldsymbol{\theta}), \text{ where } \boldsymbol{h} = \langle\!\langle R_{p+1}^\dagger H R_{p+1} | \boldsymbol{s} | \rho_0 \rangle\!\rangle \,. \end{aligned} \tag{A11}$$

We can also use (A11) to calculate the mean value of the energy function $\mathbb{E}_{\boldsymbol{\theta}}[E(\boldsymbol{\theta})]$ with respect to the gate angles, which is required to properly set the GP' mean. Because $E(\boldsymbol{\theta})$ is a linear function of $\boldsymbol{v}(\boldsymbol{\theta})$ we have

$$\mathbb{E}_{\boldsymbol{\theta}}[E(\boldsymbol{\theta})] = \boldsymbol{h}^T \mathbb{E}_{\boldsymbol{\theta}}[\boldsymbol{v}(\boldsymbol{\theta})]. \tag{A12}$$

$v(\theta) = v^{(k)} \otimes_K \ldots \otimes_K v^{(1)}$ is a Kronecker product of k subvectors with $v^{(j)} = (1, \sin(\theta_j), \cos(\theta_j))^T$ each of which only depends on a single gate angle (we assume all gate angles are independent). This means that

$$\mathbb{E}_{\theta}[v(\theta)] = \frac{1}{(2\pi)^p} \int d^p \theta \, v(\theta) = \frac{1}{(2\pi)^p} \left(\int_0^{2\pi} v^{(k)} d\theta_p \right) \otimes_K \ldots \otimes_K \left(\int_0^{2\pi} v^{(1)} d\theta_1 \right)$$

$$= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \otimes_K \ldots \otimes_K \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} . \tag{A13}$$

So the only the first element in $\mathbb{E}_{\theta}[v(\theta)]$ is nonzero, greatly simplifying the expected energy to

$$\mathbb{E}_{\theta}[E(\theta)] = h_0 = \langle\!\langle \, R_{p+1}^{\dagger} H R_{p+1} | \frac{(R_p \otimes R_p^* + P_p R_p \otimes P_p^* R_p^*)}{2} \cdots \frac{(R_1 \otimes R_1^* + P_1 R_1 \otimes P_1^* R_1^*)}{2} | \rho_0 \rangle\!\rangle \, . \tag{A14}$$

One way to interpret $\frac{1}{2}(R_q \otimes R_q^* + P_q R_q \otimes P_q^* R_q^*)$ is as a projector onto the positive eigenspace of the Pauli operator $P_q \otimes P_q^*$. Alternatively it can be seen as taking the part of its unvectorized input (following the application of R_q) that commutes with P_q. We can decompose an input $\rho' = R_q \rho R_q^{\dagger}$ into parts (sums of Pauli operators) that commute and anticommute with P, i.e. $\rho' = c_q + a_q$ with $[P_q, c_q] = 0$ and $\{P_q, a_q\} = 0$ then $\frac{1}{2}(R_q \otimes R_q^* + P_q R_q \otimes P_q^* R_q^*)|\rho\rangle\!\rangle = \frac{1}{2}|R_q \rho R_q^{\dagger} + P_q R_q \rho R_q^{\dagger} P_q\rangle\!\rangle = |c_q\rangle\!\rangle$. Applying each of these projectors in this way we see that

$$\mathbb{E}_{\theta}[E(\theta)] = h_0 = \mathrm{Tr}\{R_{p+1}^{\dagger} H R_{p+1} c_{1,\ldots,k}\} \tag{A15}$$

where $c_{1,\ldots,k}$ is what remains of ρ_0 once the process of applying projector (applying each R_q and removing the part of the result that anticommutes with P_q). In general calculating this is complicated as it depends on the fixed unitaries $\{R_q\}$ present in the ansatz however we suggest that for most ansatzes of interest the remainder will be $c_{1,\ldots,k} = I/2^n$ as this factor is present in all density matrices (ensuring $\mathrm{Tr}\{\rho\} = 1$) and commutes with any P_q regardless of the fixed unitaries applied. This leads to our suggested approximation to the mean energy of $\mathbb{E}_{\theta}[E(\theta)] = h_0 \approx \mathrm{Tr}\{R_{p+1}^{\dagger} H R_{p+1} I/2^n\} = \mathrm{Tr}\{H\}$. It takes only a few anti-commuting PPRs on each qubit (e.g. two sequential non-commuting single-qubit PPRs on each qubit) for this approximation to be exact.

Appendix B. Limits on state kernel feature space dimension

In section 6 we identified the unnormalized states $\{s_i | \rho_0 \rangle\!\rangle\}$ as vectorizations of Hermitian operators on $\mathcal{H}$. We can show this explicitly by considering the action of the qth subvector of s (5), $s^{(q)}$, on a vectorized input $|\rho\rangle\!\rangle$ (with Hermitian ρ):

$$s^{(q)}|\rho\rangle\!\rangle = \frac{1}{2} \begin{pmatrix} (I \otimes I + P_q \otimes P_q^*) R_q \otimes R_q^* \\ (I \otimes iP_q^* - iP_q \otimes I) R_q \otimes R_q^* \\ (I \otimes I - P_q \otimes P_q^*) R_q \otimes R_q^* \end{pmatrix} |\rho\rangle\!\rangle = \frac{1}{2} \begin{pmatrix} |R_q \rho R_q^{\dagger} + P_q R_q \rho R_q^{\dagger} P_q\rangle\!\rangle \\ |iR_q \rho R_q^{\dagger} P - iP_q R_q \rho R_q^{\dagger}\rangle\!\rangle \\ |R_q \rho R_q^{\dagger} - P_q R_q \rho R_q^{\dagger} P_q\rangle\!\rangle \end{pmatrix} . \tag{B1}$$

As in the discussion given in given in appendix A on the mean energy, if we write $\rho' = R_q \rho R_q^{\dagger}$ as a sum of a commuting and an anti-commuting part with P_q, i.e. $\rho' = c_q + a_q$ with $[P_q, c_q] = 0$ and $\{P_q, a_q\} = 0$, then we can rewrite the resulting vector as

$$s^{(q)}|\rho\rangle\!\rangle = \begin{pmatrix} |c_q\rangle\!\rangle \\ |ia_q P_q\rangle\!\rangle \\ |a_q\rangle\!\rangle \end{pmatrix} . \tag{B2}$$

So the qth subvector of s acts to separate the input ρ' (transformed by R_q) into the parts which commute (first entry) and anti-commute (last entry) with P_q while the middle entry contains the anti-commuting part post-multiplied by iP_q. Like the input, all of these operators are Hermitian. The first and last entries are formed by separating Pauli operators in the decomposition of the input into commuting and anti-commuting subsets; this means that if ρ' has support on (can be decomposed into) a finite set of m Pauli operators, $\rho' = \sum_{i=1}^m P_i$, then Pauli operators that do not feature in this decomposition can only be introduced in the middle entry. We also note that c_q, $ia_q P_q$, and a_q are necessarily linearly independent as it can be shown using the cylic property of the trace, $P^2 = I$, and $P_q a_q = -a_q P_q$, that they are mutually orthogonal under the Hilbert-Schmidt norm $\langle A, B \rangle_{HS} = \mathrm{Tr}\{A^{\dagger} B\}$; $\langle a_q, c_q \rangle_{HS} = -\langle a_q, c_q \rangle_{HS} = 0$, $\langle ia_q P_q, c_q \rangle_{HS} = -\langle ia_q P_q, c_q \rangle_{HS} = 0$, $\langle ia_q P_q, a_q \rangle_{HS} = -\langle ia_q P_q, a_q \rangle_{HS} = 0$.

    The dimension of the feature space is determined by the number of parameterized rotations (increasing the dimension of $v(\theta)$) and the rank of the Gram matrix S. Each unnormalized state $s_i | \rho_0 \rangle\!\rangle = |O_i\rangle\!\rangle$ is a

vectorization of a Hermitian operator O_i meaning the inner products forming $S_{ij} = \langle\!\langle \rho_0 | s_i^\dagger s_j | \rho_0 \rangle\!\rangle$ can be written $S_{ij} = \text{Tr}\{O_i O_j\}$, the Hilbert-Schmidt inner-product between O_i and O_j. To maximize the dimension of $\mathcal{F}_s$ we must ensure that the $\{O_i\}$ for different entries of $s|\rho_0\rangle\!\rangle$ are linearly independent (spanning a subspace of Hermitian operators of dimension $\geqslant 3^p$, the number of entries).

Suppose after the jth subvector has been applied, the vector $s^{(j)} \otimes_K \ldots \otimes_K s^{(1)} | \rho_0 \rangle\!\rangle$ has m linearly independent elements/operators. We can write each element as a sum of an operator that commutes with the next PPR P_{j+1} and one that anti-commutes: $s^{(j)} \otimes_K \ldots \otimes_K s^{(1)} | \rho_0 \rangle\!\rangle = (|a_1 + c_1\rangle\!\rangle, \ldots, |a_m + c_m\rangle\!\rangle)^T$. Applying the subvector for P_{j+1} splits each element into 3 new elements, e.g. $|a_1 + c_1\rangle\!\rangle \to (|c_1\rangle\!\rangle, |ia_1 P_{j+1}\rangle\!\rangle, |a_1\rangle\!\rangle)$, increasing the feature space dimension by a factor of 3. However, these elements will not always be linearly independent. Clearly, the contribution to the number of linearly independent elements from the commuting terms $c_1 \ldots c_m$ is at most of size m. From the anti-commuting terms we separate out introduce the elements $a_1, \ldots, a_m$ and $ia_1 P_{j+1}, \ldots, ia_m P_{j+1}$, which also anti-commute with P_{j+1}. An n-qubit Pauli operator anti-commutes with $4^n/2$ other elements of the n-qubit Pauli group (ignoring phases) and commutes with the other remaining $4^n/2$ elements. As there are only $4^n/2$ Pauli operators that anti-commute with P_{j+1}, the span of the combined set of $a_1, \ldots, a_m, ia_1 P_{j+1}, \ldots, ia_m P_{j+1}$ is at most $4^n/2$ dimensional. This means that the maximal increase in the feature space dimension possible when adding another PPR to the circuit is $4^n/2$ (as we can go from m independent elements to at most $m + 4^n/2$). Denoting the maximal feature space dimension of $\mathcal{F}_s$ for n qubits and a p PPR ansatz as $d_s^{(n,p)}$ we have the recursion relation

$$d_s^{(n,p)} = \min(4^n, 3d_s^{(n,p-1)}, 4^n/2 + d_s^{(n,p-1)}).\tag{B3}$$

We define the base-case $d_s^{n,0} = 1$ for all n—this is purely for completeness as a kernel on a zero-dimensional input (no parameterized rotations) has no meaning.

Appendix C. Limits on unitary kernel feature space dimension

Like the state kernel, the unitary kernel's feature space can be expressed as a quadratic form involving a parameter-dependent vector $\boldsymbol{v}(\boldsymbol{\theta})$ and a Gram matrix $\boldsymbol{T}$ with elements $T_{ij} = \text{Tr}\{s_i^\dagger s_j\}$. Like with the state kernel, the dimension of the feature space $\mathcal{F}_u$ is determined by the number of parameters (determines the size of $\boldsymbol{v}$) and the rank of the $\boldsymbol{T}$ matrix.

The T matrix is comprised of Hilbert-Schmidt inner products between the elements of the vector $\boldsymbol{s}$. If the elements of $\boldsymbol{s}$ were completely generic $2^{2n} \times 2^{2n}$ matrices then the maximal rank of $\boldsymbol{T}$ would be 4^{2n} as there are this many orthogonal (under the Hilbert-Schmidt inner product) basis matrices, e.g. the $2n$-qubit Pauli matrices $\{I, X, Y, Z\}^{\otimes 2n}$ or the standard basis matrices $e_{ij} = |i\rangle\langle j|$. However there is additional structure present in the elements of $\boldsymbol{s}$ that reduces the maximal rank of $\boldsymbol{T}$. For example, all elements in $\boldsymbol{s}$ are symmetric under interchange of the physical ($\mathcal{H}$) and conjugated ($\mathcal{H}^*$) subsystems followed by complex conjugation and involve sums and products of various unitary and Hermitian operators. To derive the maximal scaling of the unitary kernel's feature space dimension we will first examine weighted sums of unitary matrices of the form $A \otimes A^*$, matrices $\boldsymbol{M}(\boldsymbol{w}, \boldsymbol{A})$ given by

$$\boldsymbol{M}(\boldsymbol{w}, \boldsymbol{A}) = \sum_k w_k A^{(k)} \otimes A^{(k)*}\tag{C1}$$

with $A^{(k)\dagger} A^{(k)} = I_d$ (and each A has dimension $d \times d$). In the following we will demonstrate that the maximal dimension of $\mathcal{F}_u$ is ultimately set by the number of linearly independent entries of such matrices.

For a single $A \otimes A^*$, the unitarity constraint implies that $\sum_k A_{ki} A_{kj}^* = \sum_k A_{ik} A_{jk}^* = \delta_{ij}$—these constraints can be easily related to $A \otimes A^*$ as they correspond to contractions over pairs of indices. A general element of this matrix is $(A \otimes A^*)_{ijkl} = A_{ij} A_{kl}^*$ so contraction over the first and third indices gives $\sum_p (A \otimes A^*)_{kikj} = \sum_p A_{ki} A_{kj}^* = \delta_{ij}$ (the first set of unitarity constraints) while contraction over the second and fourth indices gives $\sum_p (A \otimes A^*)_{ikjk} = \sum_p A_{ik} A_{jk}^* = \delta_{ij}$ (the second set of constraints). This appears to give us $2d^2$ equations, each of which will reduce the number of linearly independent elements by 1 giving a total of $d^4 - 2d^2$ independent elements. However, because the constraints for $i = j$ all sum to the same value, $\sum_p A_{ik} A_{ik}^* = \sum_p A_{ki} A_{ki}^* = 1$ for all i, this fixed value acts as another free parameter in each set of constraints, increasing the number of linearly independent elements by 2 to a total of $d^4 - 2(d^2 - 1)$. When considering weighted sums of these matrices as given in (C1) these constraints become

$$\sum_p \sum_p (w_p A^{(p)} \otimes A^{(p)*})_{ikjk} = \sum_p \sum_p (w_p A^{(p)} \otimes A^{(p)*})_{kikj} = \left(\sum_p w_p\right) \delta_{ij}.\tag{C2}$$

The linear dependence of the terms in the constraints with $i = j$ is only defined up to specification of the sum of the weights meaning that the quantity $\sum_p w_p$ is an additional free parameter, giving the same result.

The initial fixed unitary $R_1 \otimes R_1^*$ is in the form we have just discussed and for an n-qubit ansatz will in general have $d^4 - 2(d^2 - 1) = 4^{2n} - 2(4^n - 1)$ linearly independent elements. For unitary operators B and C, the product $(B \otimes B^*)(C \otimes C^*) = A \otimes A^*$ (with $A = BC$) is also a matrix of the form discussed above. As a result, the Pauli-containing part of the top/bottom subvector elements $I \otimes I \pm P_1 \otimes P_1^*$ acts on $R_1 \otimes R_1^*$ to give matrices that admit a decomposition as in (C1) and so have the same number of linearly independent elements as before (as will the result from applying the next fixed unitary $R_2 \otimes R_2^*$ and the subsequent top/bottom elements of the next subvector). All that is left is to consider how the application of $i(I \otimes P^* - P \otimes I)$, from the middle elements of each subvector, to a general unitary $A \otimes A^*$ (which could be part of a sum as in (C1)) affects the unitarity constraints (C2) that determine the number of linearly independent elements. We have:

$$
\sum_k (i(I \otimes P^* - P \otimes I)(A \otimes A^*))_{ikjk} = i \sum_p \sum_m (A_{ik} P_{jm}^* A_{mk}^* - P_{im} A_{mk} A_{jk}^*)
$$

$$
= i \sum_m (\delta_{im} P_{jm}^* - P_{im} \delta_{mj}) = i(P_{ji}^* - P_{ij}) \tag{C3}
$$

$$
= i(P^\dagger - P)_{ij} = 0,
$$

while for the other constraint we get the same result

$$
\sum_p (i(I \otimes P^* - P \otimes I)(A \otimes A^*))_{kikj} = i \sum_p \sum_m (A_{ki} P_{km}^* A_{mj}^* - P_{km} A_{mi} A_{kj}^*)
$$

$$
= i \sum_p \sum_m (P_{mk} A_{ki} A_{mj}^*) - i \sum_m \sum_p (P_{km} A_{mi} A_{kj}^*) \tag{C4}
$$

$$
= 0
$$

(in both cases we have used that $P^\dagger = P$). By linearity, the same results will be obtained for an arbitrary weighted sum as in (C1). These are identical to the constraints in (C2) for $i \neq j$ but give 0 rather than a fixed value (1 for a single $A \otimes A$ or $\sum_p w_p$ for a weighted sum) if $i = j$. Having the $i = j$ case evaluate to 0 means that the number of linearly independent elements in matrices formed of weighted sums of terms like this is $d^4 - 2(d^2) = 4^{2n} - 2(4^n)$. However, because these new constraints are a subset of the original constraints (C2) with $\sum_p w_p = 0$, the over-all number of linearly independent elements in the matrices that form the entries of $\boldsymbol{s}$ and therefore the maximal rank of $\boldsymbol{T}$ is given by $4^{2n} - 2(4^n - 1)$. We verified this scaling with simulations for $n = 1, 2$, and 3 qubits using Haar random unitaries for $\{R_q\}$ to ensure minimal structure in the ansatz that might limit the feature space dimension.

ORCID iD

Alistair W R Smith ⓘ https://orcid.org/0000-0003-1047-726X

References

[1] Egan L *et al* 2021 Fault-tolerant control of an error-corrected qubit *Nature* **598** 281–6
[2] Xue X, Russ M, Samkharadze N, Undseth B, Sammak A, Scappucci G and Vandersypen L M K 2022 Quantum logic with spin qubits crossing the surface code threshold *Nature* **601** 343–7
[3] Abobeih M H, Wang Y, Randall J, Loenen S J H, Bradley C E, Markham M, Twitchen D J, Terhal B M and Taminiau T H 2022 Fault-tolerant operation of a logical qubit in a diamond quantum processor *Nature* **606** 884–9
[4] Postler L *et al* 2022 Demonstration of fault-tolerant universal quantum gate operations *Nature* **605** 675–80
[5] Noiri A, Takeda K, Nakajima T, Kobayashi T, Sammak A, Scappucci G and Tarucha S 2022 Fast universal quantum gate above the fault-tolerance threshold in silicon *Nature* **601** 338–42
[6] Preskill J 2018 Quantum computing in the NISQ era and beyond *Quantum* **2** 79
[7] Sevilla J and Jess Riedel C 2020 Forecasting timelines of quantum computing (arXiv:2009.05045)
[8] Preskill J 2022 The physics of quantum information (arXiv:2208.08064)
[9] Bharti K *et al* 2022 Noisy intermediate-scale quantum algorithms *Rev. Mod. Phys.* **94** 015004
[10] Arute F *et al* (Google Quantum AI) 2019 Quantum supremacy using a programmable superconducting processor *Nature* **574** 505–10
[11] Zhong H-S *et al* 2020 Quantum computational advantage using photons *Science* **370** 1460–3
[12] Wu Y *et al* 2021 Strong quantum computational advantage using a superconducting quantum processor *Phys. Rev. Lett.* **127** 180501
[13] Bulmer J F F *et al* 2022 The boundary for quantum advantage in Gaussian Boson sampling *Sci. Adv.* **8** eabl9236
[14] Gray J and Kourtis S 2021 Hyper-optimized tensor network contraction *Quantum* **5** 410
[15] Pan F and Zhang P 2022 Simulation of quantum circuits using the big-batch tensor network method *Phys. Rev. Lett.* **128** 030501
[16] Cerezo M *et al* 2021 Variational quantum algorithms *Nat. Rev. Phys.* **3** 625–44

[17] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 Quantum machine learning *Nature* **549** 195–202

[18] Hofmann T, Schölkopf B and Smola A J 2008 Kernel methods in machine learning *Ann. Stat.* **36** 1171–220

[19] Shawe-Taylor J *et al* 2004 *Kernel Methods for Pattern Analysis* (Cambridge University Press)

[20] Rebentrost P, Mohseni M and Lloyd S 2014 Quantum support vector machine for big data classification *Phys. Rev. Lett.* **113** 130503

[21] Blank C, Park D K, Kevin Rhee J-K and Petruccione F 2020 Quantum classifier with tailored quantum kernel *npj Quantum Inf.* **6** 41

[22] Park D K, Blank C and Petruccione F 2020 The theory of the quantum kernel-based binary classifier *Phys. Lett.* A **384** 126422

[23] Mengoni R and Di Pierro A 2019 Kernel methods in quantum machine learning *Quantum Mach. Intell.* **1** 65–71

[24] Otten M, Goumiri I R, Priest B W, Chapline G F and Schneider M D 2020 Quantum machine learning using Gaussian processes with performant quantum kernels (arXiv:2004.11280)

[25] Fontana E, Fitzpatrick N, Ramo D M, Duncan R and Rungger I 2021 Evaluating the noise resilience of variational quantum algorithms *Phys. Rev.* A **104** 022403

[26] Kandala A, Mezzacapo A, Temme K, Takita M, Brink M, Chow J M and Gambetta J M 2017 Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets *Nature* **549** 242–6

[27] Sung K J, Yao J, Harrigan M P, Rubin N C, Jiang Z, Lin L, Babbush R and McClean J R 2020 Using models to improve optimizers for variational quantum algorithms *Quantum Sci. Technol.* **5** 044008

[28] Self C N, Khosla K E, Smith A W R, Sauvage F, Haynes P D, Knolle J, Mintert F and Kim M S 2021 Variational quantum algorithm with information sharing *npj Quantum Inf.* **7** 116

[29] Farhi E and Harrow A W 2016 Quantum supremacy through the quantum approximate optimization algorithm (arXiv:1602.07674)

[30] Guerreschi G G and Matsuura A Y 2019 Qaoa for max-cut requires hundreds of qubits for quantum speed-up *Sci. Rep.* **9** 6903

[31] Wang S, Fontana E, Cerezo M, Sharma K, Sone A, Cincio L and Coles P J 2021 Noise-induced barren plateaus in variational quantum algorithms *Nat. Commun.* **12** 6961

[32] Bittel L and Kliesch M 2021 Training variational quantum algorithms is NP-hard *Phys. Rev. Lett.* **127** 120502

[33] McClean J R, Boixo S, Smelyanskiy V N, Babbush R and Neven H 2018 Barren plateaus in quantum neural network training landscapes *Nat. Commun.* **9** 4812

[34] Larocca M, Czarnik P, Sharma K, Muraleedharan G, Coles P J and Cerezo M 2021 Diagnosing barren plateaus with tools from quantum optimal control (arXiv:2105.14377)

[35] Grimsley H R, Economou S E, Barnes E and Mayhall N J 2019 An adaptive variational algorithm for exact molecular simulations on a quantum computer *Nat. Commun.* **10** 1–9

[36] Tang H L, Shkolnikov V O, Barron G S, Grimsley H R, Mayhall N J, Barnes E and Economou S E 2021 Qubit-ADAPT-VQE: an adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor *PRX Quantum* **2** 020310

[37] Temme K, Bravyi S and Gambetta J M 2017 Error mitigation for short-depth quantum circuits *Phys. Rev. Lett.* **119** 180509

[38] Kandala A, Temme K, Córcoles A D, Mezzacapo A, Chow J M and Gambetta J M 2019 Error mitigation extends the computational reach of a noisy quantum processor *Nature* **567** 491–5

[39] Barron G S and Wood C J 2020 Measurement error mitigation for variational quantum algorithms (arXiv:2010.08520)

[40] Czarnik P, Arrasmith A, Cincio L and Coles P J 2021 Qubit-efficient exponential suppression of errors (arXiv:2102.06056)

[41] Funcke L, Hartung T, Jansen K, Kühn S, Stornati P and Wang X 2022 Measurement error mitigation in quantum computers through classical bit-flip correction *Phys. Rev.* A **105** 062404

[42] Maciejewski F B, Zimborás Z and Oszmaniec M 2020 Mitigation of readout noise in near-term quantum devices by classical post-processing based on detector tomography *Quantum* **4** 257

[43] Peruzzo A, McClean J, Shadbolt P, Yung M-H, Zhou X-Q, Love P J, Aspuru-Guzik A and O'Brien J L 2014 A variational eigenvalue solver on a photonic quantum processor *Nat. Commun.* **5** 4213

[44] Mohseni M, Rezakhani A T and Lidar D A 2008 Quantum-process tomography: resource analysis of different strategies *Phys. Rev.* A **77** 032322

[45] Lubinski T, Johri S, Varosy P, Coleman J, Zhao L, Necaise J, Baldwin C H, Mayer K and Proctor T 2021 Application-oriented performance benchmarks for quantum computing (arXiv:2110.03137)

[46] McCaskey A J, Parks Z P, Jakowski J, Moore S V, Morris T D, Humble T S and Pooser R C 2019 Quantum chemistry as a benchmark for near-term quantum computers *npj Quantum Inf.* **5** 99

[47] Sweke R, Wilde F, Meyer J, Schuld M, Faehrmann P K, Meynard-Piganeau B and Eisert J 2020 Stochastic gradient descent for hybrid quantum-classical optimization *Quantum* **4** 314

[48] Stokes J, Izaac J, Killoran N and Carleo G 2020 Quantum natural gradient *Quantum* **4** 269

[49] Nakanishi K M, Mitarai K and Fujii K 2019 Subspace-search variational quantum eigensolver for excited states *Phys. Rev. Res.* **1** 033062

[50] Parrish R M, Hohenstein E G, McMahon P L and Martínez T J 2019 Quantum computation of electronic transitions using a variational quantum eigensolver *Phys. Rev. Lett.* **122** 230401

[51] Parrish R M, Iosue J T, Ozaeta A and McMahon P L 2019 A Jacobi diagonalization and Anderson acceleration algorithm for variational quantum algorithm parameter optimization (arXiv:1904.03206)

[52] Wierichs D, Izaac J, Wang C and Lin C Y-Y 2022 General parameter-shift rules for quantum gradients *Quantum* **6** 677

[53] Gacon J, Zoufal C, Carleo G and Woerner S 2021 Simultaneous perturbation stochastic approximation of the quantum Fisher information *Quantum* **5** 567

[54] Zhu D *et al* 2019 Training of quantum circuits on a hybrid quantum computer *Sci. Adv.* **5** eaaw9918

[55] Shen Y, Zhang X, Zhang S, Zhang J-N, Yung M-H and Kim K 2017 Quantum implementation of the unitary coupled cluster for simulating molecular electronic structure *Phys. Rev.* A **95** 020501

[56] Snoek J, Larochelle H and Adams R P 2012 Practical Bayesian optimization of machine learning algorithms *Advances in Neural Information Processing Systems* vol 25

[57] Schölkopf B, Herbrich R and Smola A J 2001 A generalized representer theorem *Int. Conf. on Computational Learning Theory* (Springer) pp 416–26

[58] Micchelli C A, Xu Y and Zhang H 2006 Universal kernels *J. Mach. Learn. Res.* **7** 2651–67

[59] Shaffer R, Kocia L and Sarovar M 2022 Surrogate-based optimization for variational quantum algorithms (arXiv:2204.05451)

[60] Schuld M and Petruccione F 2021 *Quantum Models as Kernel Methods* (Springer International Publishing) pp 217–45

[61] Barenco A, Bennett C H, Cleve R, DiVincenzo D P, Margolus N, Shor P, Sleator T, Smolin J A and Weinfurter H 1995 Elementary gates for quantum computation *Phys. Rev.* A **52** 3457–67

[62] Gilchrist A, Terno D R and Wood C J 2011 Vectorization of quantum operations and its use (arXiv:0911.2539)

[63] Williams C K and Rasmussen C E 2006 *Gaussian Processes for Machine Learning* vol 2 (MIT Press)

[64] Życzkowski C K and Sommers H-J 2005 Average fidelity between random quantum states *Phys. Rev.* A **71** 032313

[65] Liu Y, Arunachalam S and Temme K 2021 A rigorous and robust quantum speed-up in supervised machine learning *Nat. Phys.* **17** 1013–7

[66] Cincio L, Subaşı Y, Sornborger A T and Coles P J 2018 Learning the quantum algorithm for state overlap *New J. Phys.* **20** 113022

[67] Jones D R, Schonlau M and Welch W J 1998 Efficient global optimization of expensive black-box functions *J. Glob. Optim.* **13** 455–92

[68] Lizotte D J 2008 Practical Bayesian optimization *PhD Thesis* University of Alberta

[69] Brochu E, Cora V M and de Freitas N 2010 A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning (arXiv:1012.2599)

[70] Pedregosa F *et al* 2011 Scikit-learn: machine learning in Python *J. Mach. Learn. Res.* **12** 2825–30

[71] IBM Quantum 2021 Qiskit: an open-source framework for quantum computing Zenodo https://doi.org/10.5281/zenodo.2573505

[72] Spall J C 1992 Multivariate stochastic approximation using a simultaneous perturbation gradient approximation *IEEE Trans. Autom. Control* **37** 332–41

[73] IBM Quantum Team 2021 *ibmq_quito* (available at: https://quantum-computing.ibm.com)

[74] Gray J 2018 quimb: a Python library for quantum information and many-body calculations *J. Open Source Softw.* **3** 819

[75] Slattery L, Villalonga B and Clark B K 2021 Unitary block optimization for variational quantum algorithms (arXiv:2102.08403)

[76] Skolik A, McClean J R, Mohseni M, van der Smagt P and Leib M 2021 Layerwise learning for quantum neural networks *Quantum Mach. Intell.* **3** 5

[77] Perez-Garcia D, Verstraete F, Wolf M M and Cirac J I 2007 Matrix product state representations *Quantum Inf. Comput.* **7** 401–30

[78] Verstraete F, Murg V and Cirac J I 2008 Matrix product states, projected entangled pair states and variational renormalization group methods for quantum spin systems *Adv. Phys.* **57** 143–224

[79] Schollwöck U 2011 The density-matrix renormalization group in the age of matrix product states *Ann. Phys., NY* **326** 96–192

[80] Orús R 2014 A practical introduction to tensor networks: matrix product states and projected entangled pair states *Ann. Phys., NY* **349** 117–58

[81] Paeckel S, Köhler T, Swoboda A, Manmana S R, Schollwöck U and Hubig C 2019 Time-evolution methods for matrix-product states *Ann. Phys., NY* **411** 167998

[82] Bridgeman J C and Chubb C T 2017 Hand-waving and interpretive dance: an introductory course on tensor networks *J. Phys. A: Math. Theor.* **50** 223001

[83] Kass R E and Raftery A E 1995 Bayes factors *J. Am. Stat. Assoc.* **90** 773–95

[84] Garcia H J, Markov I L and Cross A W 2012 Efficient inner-product algorithm for stabilizer states (arXiv:1210.6646)

[85] Aaronson S and Gottesman D 2004 Improved simulation of stabilizer circuits *Phys. Rev.* A **70** 052328

[86] Bravyi S, Browne D, Calpin P, Campbell E, Gosset D and Howard M 2019 Simulation of quantum circuits by low-rank stabilizer decompositions *Quantum* **3** 181

[87] Byrd R H, Lu P, Nocedal J and Zhu C 1995 A limited memory algorithm for bound constrained optimization *SIAM J. Sci. Comput.* **16** 1190–208

[88] Bossek J, Doerr C and Kerschke P 2020 Initial design strategies and their effects on sequential model-based optimization: an exploratory case study based on BBOB *Proc. 2020 Genetic and Evolutionary Computation Conf. (GECCO '20)* (Association for Computing Machinery) pp 778–86

[89] Vidal G 2003 Efficient classical simulation of slightly entangled quantum computations *Phys. Rev. Lett.* **91** 147902