
Article:
Williams, C.A., Paine, A.E., Gentile, A.A. et al. (2025) Vortex detection from quantum data. Physical Review A, 112 (6). 062409. ISSN: 2469-9926

https://doi.org/10.1103/mn3x-8ygh

Vortex detection from quantum data

Chelsea A. Williams [1,2] Annie E. Paine [2] Antonio A. Gentile [2] Daniel Berger [3] and Oleksandr Kyriienko [4]

[1]*Department of Physics and Astronomy, University of Exeter, Stocker Road, Exeter EX4 4QL, United Kingdom*

[2]*PASQAL, 7 Rue Léonard de Vinci, 91300 Massy, France*

[3]*Siemens AG, Gleiwitzer Strasse 555, 90475 Nürnberg, Germany*

[4]*School of Mathematical and Physical Sciences, University of Sheffield, Sheffield S10 2TN, United Kingdom*

(Received 4 August 2025; accepted 22 October 2025; published 5 December 2025)

Quantum solutions to differential equations represent quantum data—states that contain relevant information about the system's behavior, yet are difficult to analyze. We propose an algorithm for reading out information from such data, where customized quantum circuits enable efficient extraction of flow properties. We concentrate on the process referred to as quantum vortex detection, where specialized operators are developed for pooling relevant features related to vorticity. Specifically, we propose approaches based on sliding windows and quantum Fourier analysis that provide a separation between patches of the flow field with vortex-type profiles. First, we show how contour-shaped windows can be applied, trained, and analyzed sequentially, providing a clear signal to flag the location of vortices in the flow. Second, we develop a parallel window extraction technique, such that signals from different contour positions are coherently processed to avoid looping over the entire solution mesh. We show that Fourier features can be extracted from the flow field, leading to classification of datasets with vortex-free solutions against those exhibiting Lamb-Oseen vortices. Our work exemplifies a successful case of efficiently extracting value from quantum data, and it points to the need for developing appropriate models for quantum data analysis that can be trained on them.

DOI: 10.1103/mn3x-8ygh

I. INTRODUCTION

Turbulent flows are inherently nonlinear and exhibit a rich hierarchy of spatiotemporal structures across scales, ranging from large-scale eddies down to dissipative microvortices [1–4]. Vortex detection is a critical process in computational fluid dynamics (CFD) [5] and is essential for understanding the physics of flow fields [6–8]. Accurate detection enables better turbulence modeling and multiscale analysis, leading to improved predictions in the areas of magnetohydrodynamics [9], astrophysics [10,11], oceanography [12], aerospace [13,14], and atmospheric science [15]. For industrial and engineering applications, vorticity structures provide key insights to optimize designs with respect to energy efficiency, noise, and safety [16–20]. However, it is often challenging to resolve such vortical structures unless fine-mesh direct numerical simulation is performed [21–23].

Quantum computing methods have the potential to implement direct numerical simulation–type methods using a distinct physical hardware and large operational space [24,25]. Quantum algorithms can encode solutions on a fine mesh or function basis into the Hilbert space of a quantum system [26–31]. Recent advances in quantum differential equation solvers include methods based on quantum signal processing [32–34], eigenstate filtering-based approaches [32,35], linear combinations of unitaries [36–40], Schrödingerization [41–44], Fourier transform–based solvers [31,45–47], digitized quantum adiabatic methods [48–50], and quantum iterative solvers [43,51–53]. Nonlinearity can be treated with Carleman linearization [34,35,54–60], Chebyshev-based models [61,62], and quantum nonlinear processing units [63–65]. Turbulence from the quantum encoding perspective was studied in Ref. [58]. However, the drawback of representing solutions in an amplitude-encoded way corresponds to a large readout cost—we cannot readily access information from quantum states and need to rely on sampling (repeated measurements) [66]. This calls for specialized feature extraction methods for quantum data [67].

Vortex detection from a calculated flow profile can combine local and global approaches [68,69], where the latter is computationally intensive and relies on identifying topological features. In the domain of data-driven science and engineering, machine learning (ML) demonstrated success in flow analysis [70,71] through advances in pattern recognition, classification, and segmentation. Convolutional neural networks (CNNs) and U-Net architectures were adapted to vortex detection tasks by utilizing convolutional filters [72–76]. However, these methods act on classical flow field images or simulation data arrays, and they are not applicable for quantum states. Quantum machine learning (QML) developed as a field where quantum datasets can be processed with parametrized quantum circuits [77–79], spotting patterns that are difficult to analyze classically [77,80–85], as well as developing physics-informed quantum processing methods

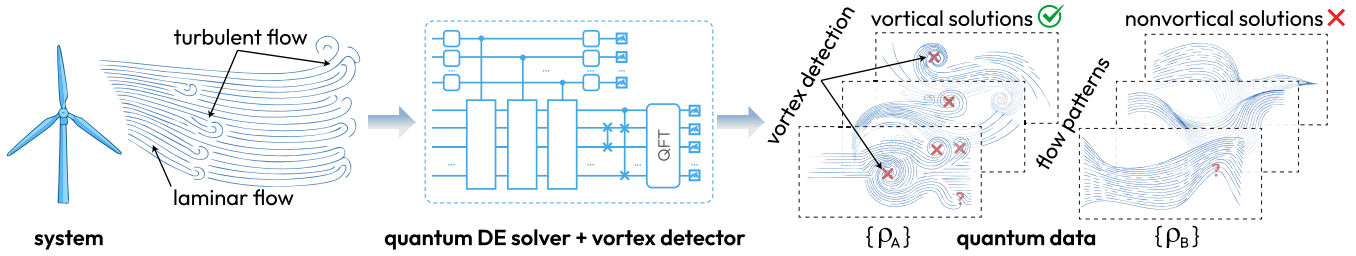

FIG. 1. Workflow for vortex detection from quantum data. A nonlinear system is modeled using a quantum differential equation solver, producing states that contain solutions with vortices (turbulent flow) or without vortices (laminar flow). Vortex detection is applied with sliding window–based spectral analysis, compressing relevant features of the flow for consequent detection. The detection circuit is trained on a few examples and only requires several tuning parameters.

[62,86–95]. From the feature selection perspective, quantum convolutional neural networks became a tool for efficient readout [96–104]. At the same time, their utility largely depends on data, competition with measure-first approaches [105–107], and generally suffers from being physics-agnostic. Our recent work introduced a quantum scientific machine learning framework for feature extraction from quantum partial differential equation (PDE) solvers [67], highlighting the need for specialized postprocessing pipelines that operate within the quantum domain. These tools must not only preserve the computational advantages of quantum encoding but also support the efficient retrieval of physically meaningful observables.

In this work, we address the challenge of extracting flow features, and specifically vortex patterns, from quantum data produced by quantum differential equation solvers. This workflow is illustrated in Fig. 1. The modeled CFD system (e.g., a wind turbine) can give rise to solutions with nontrivial patterns. These solutions can be measured experimentally, or modeled with numerical simulation tools (classical or quantum). In the case of quantum algorithms, the corresponding flow fields are stored as quantum states, and they require a complicated feature extraction procedure at the readout. These states are referred to as quantum data, which we load into a QML framework that performs quantum feature detection to distinguish between vortical and nonvortical flow field solutions. This quantum vortex detection approach enables the identification of localized vorticity features while using coherent ("parallel") processing. Specifically, our proposed quantum vortex detection framework is based on sliding window operators paired with quantum Fourier analysis. Our method enables spatially localized probes of the vorticity field and parallelized processing of solution patches, avoiding full quantum state tomography. In this work, we show that a general strategy for extracting structured information from quantum PDE solutions requires physics-aware approaches and models with only a few tunable parameters.

II. ALGORITHMS

We proceed to propose and develop an algorithm for vortex detection from quantum data, referred to as quantum vortex detection (QVD). The methodology employed for the QVD framework takes inspiration from the ideas behind convolutional neural networks (CNNs). CNNs rely on pattern recognition using a sliding window that traverses a data sample (array of values) to identify local features and hierarchical patterns within the data [108]. Similarly, we design a QVD framework for quantum feature extraction to detect vortical structures in local patches of field profiles that are analyzed in superposition. The designed models are able to learn on patterns inferred from available simulations or data. By working explicitly with quantum simulations (data), the QVD model bypasses classical-to-quantum conversion for postprocessing. Unlike a general quantum convolutional neural network (QCNN) [96,97,102,107] that simply mimic the convolution-pooling structure, the QVD framework is designed to analyze the Fourier representation of quantum data, as well as the underlying power spectrum. It is a specialized model that embeds principles from CFD into the automated detection of vortices. Such a physics-informed methodology and superposition-based processing cannot be replicated within a traditional CNN architecture.

Let us proceed to describe the input for quantum vortex detection. This consists of quantum states as solutions of quantum differential equations. At the physical level they contain local multidimensional patches of vorticity field solutions that are amplitude-encoded by construction. As a particular example, we consider the Lamb-Oseen vortex model [68], generating solutions of the type shown in Eq. (1) for varying parameters. As a solution to the Navier-Stokes equations in cylindrical coordinates, Lamb-Oseen exhibits vortical motion with a concentrated vorticity that diffuses over time [109]. The model characterizes a two-dimensional vortex structure with a Gaussian distribution of vorticity. The azimuthal velocity profile v_θ is given by

$$v_\theta = v_{\max}\left(1+\frac{1}{2\Delta}\right)\frac{r}{R}\left[1-\exp\left(-\Delta\frac{R^2}{r^2}\right)\right], \qquad (1)$$

where Δ is the vorticity diffusion parameter, $v_{\max}$ is the peak tangential velocity, R is the radial distance from the vortex center, and r is the radius of the vortex core where the vorticity is concentrated. We take this as an illustrative example and describe the approach to study solutions.

A. Quantum vortex detection: Tools

The input corresponds to the flow field as a quantum state, $|\psi_f\rangle$. We consider this a starting point for QVD. This is followed by a sequence of operations that process states (Fig. 2), which we introduce below. The goal is to perform either a regression or classification task, e.g., assigning a degree of

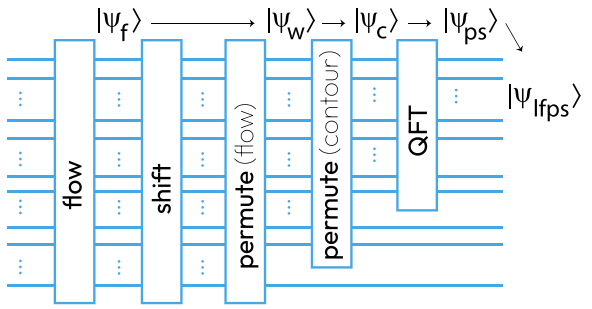

FIG. 2. Quantum circuit used to perform feature extraction in QVD. Quantum data in the form of a multidimensional flow field are encoded into state $|\psi_f\rangle$. A shift gate followed by a permutation gate are applied to select a window from this flow whose elements are encoded into the top register of state $|\psi_w\rangle$. Another permutation gate is applied to select a circular contour from the window whose ordered elements are encoded into the top register of state $|\psi_c\rangle$. The QFT is applied to calculate the power spectrum of the contour, resulting in state $|\psi_{ps}\rangle$. The low-frequency band of this power spectrum is extracted from state $|\psi_{lfps}\rangle$ by measuring the bottom register of the circuit. This circuit assumes that $n_f \geqslant n_w > n_c > n_{lfps}$, where n_f, n_w, n_c, and n_{lfps}, respectively, refer to the number of qubits in the subregisters encoding the flow field, selected window, extracted contour, and processed low-frequency power spectrum.

vorticity or classifying turbulent versus laminar flow. Given that we sequentially study patches, we can typically infer the number of vortices, alongside their locations, at the expense of repeated circuit evaluations.

The quantum vortex detection circuit requires the following operations represented by individual blocks in Fig. 2. Starting from an n-qubit input state, a shift block corresponds to

$$S(n,d) = \sum_{j=0}^{2^n-1} |(j+d) \bmod 2^n\rangle\langle j|, \tag{2}$$

which cyclically shifts the computational basis states by a fixed offset d. Mathematically, operations in Eq. (2) correspond to generalized Pauli matrices [110]. These are closely related to Toeplitz matrices that can be block-encoded efficiently [111]. In our setting, the shift operator performs a translation of the discretized flow field by one or a few grid units. This operator is inherently sparse and possesses the structure. As shown in Ref. [112], such sparse structured operators can be efficiently block-encoded, with circuit depth scaling logarithmically in the sparsity and requiring $O[\mathrm{poly}(n)]$ two-qubit gates coming from the decomposition of multicontrolled (Toffoli-like) gates. These operators were shown to compile efficiently (linearly in system size) with parallel implementation strategies [113]. Note that powers of shift operators also can be compiled efficiently with $O(n^2)$ complexity [112].

The second step of the QVD circuit corresponds to permutation blocks that structure our quantum data into a suitable form. A permutation operation can be written as

$$P(n,l) = \sum_{j=0}^{2^n-1} |j\rangle\langle \mathrm{o}_l[j]|, \tag{3}$$

which permutes the set of computational basis states $\mathcal{H} = \{|j\rangle\}_{j=0}^{2^n-1}$ according to the array o_l, such that order $\mathrm{o}_l[j]$ specifies the index of the state mapped to $|j\rangle$. This represents a bijective map and correspondingly a unitary operator. Exact compilation of $P(n,l)$ depends on the system and the required order. While the general complexity of permuting bitstrings is exponential in the worst case [114], in our setting the permutation acts only on a subset of basis states $\mathcal{S} = \{|j\rangle^{(s)}\}$ corresponding to localized regions of the discretized flow field while ignoring the $\mathcal{H}$-$\mathcal{S}$ subset. The cardinality $d_{\mathcal{S}} = |\mathcal{S}|$, therefore, scales with the number of active subregions rather than with the total Hilbert space dimension. For practical discretizations used in the QVD workflow, permutations scale with $d_{\mathcal{S}}$ and remain polynomial in the problem size, requiring $O(d_{\mathcal{S}} n)$ operations [115]. As our goal is to bring the computational space into a block-type structure, the complexity can also be inferred from the quantum Schur transform, which is polylogarithmic in system size [116]. At extremely fine spatial resolutions, however, the subset size could grow exponentially, in which case additional coarse-graining or hierarchical encoding would be required.

Finally, the quantum Fourier transform (QFT) is added to enable processing over the low-frequency spectrum rather than a wide-support quasiprobability distribution. The corresponding bijective operator reads

$$\mathrm{QFT}(n) = (1/\sqrt{2^n}) \sum_{j,k=0}^{2^n-1} \exp\left[-2\pi i (jk/2^n)\right] |k\rangle\langle j|, \tag{4}$$

with $\Omega = \{|k\rangle\}_{k=0}^{2^n-1}$ denoting the frequency basis states. The corresponding map $\mathrm{QFT} : \Omega \mapsto \mathcal{H}$ and its reverse can be implemented with $O(n^2)$ two-qubit gates in its textbook form [117] and compiled efficiently to various gatesets. Approximate versions of QFT can also be run in linear gate complexity and logarithmic depth [118].

One more ingredient is the k-frequency projector $\Pi(m) = \sum_{k=0}^{2^m-1} |k\rangle\langle k|$. This is a rank-$2^m$ projector from the full 2^n-dimensional Hilbert space. It can be implemented by measuring the first n-m qubits and postselecting on $|0\rangle^{\otimes(n-m)}$ [119], or using a coherent ancilla-based projection.

The QVD workflow is designed with fault-tolerant implementations in mind, as the circuit depth and use of multicontrolled operations make execution on near-term devices challenging. However, advances in high-coherence qubit platforms indicate that implementation of medium-scale instances could become feasible in the future.

B. Sequential QVD

We proceed to consider a step-by-step example for the QVD approach, and we visualize its element. We start with a two-dimensional vorticity flow field $\psi_{i,j}$ in the xy-plane that is encoded into an n_f-qubit quantum state $|\psi_f\rangle$. The encoding of this flow onto two registers is represented by

$$|\psi_f\rangle = |x,y\rangle = \sum_{i,j=0}^{2^{n_f}-1} \psi_{i,j} |i\rangle \otimes |j\rangle, \tag{5}$$

where the basis state $|i\rangle$ stores the column index corresponding to the x-register, and the basis state $|j\rangle$ stores the row index

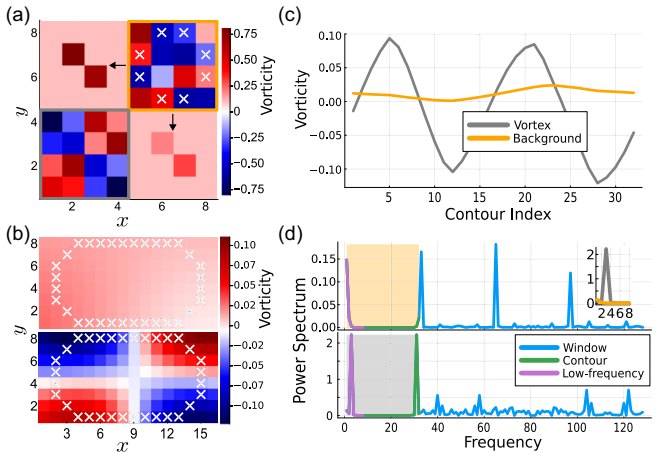

FIG. 3. Examples of performing feature extraction in QVD. (a) A two-dimensional vorticity flow in the xy-plane. A sliding window is selected, and the contour points within each window (indicated by white crosses) are extracted. Note that this example is course-grained for brevity. (b) Two seven-qubit windows are shown with five-qubit circular contours extracted. The top window is representative of a featureless background, and the bottom window is representative of a centrally located vortex. (c) The vorticity amplitudes extracted from the two contours. (d) The power spectra of the background and vortex, where frequencies are indexed by window pixel $\psi_{i,j}$. The power spectra are shown for the encoded window (QFT$|\psi_{\rm w}\rangle$ on seven qubits), contour ($|\psi_{\rm ps}\rangle$ on five qubits), and low-frequency band ($|\psi_{\rm lfps}\rangle$ on three qubits), as obtained with the quantum circuit from Fig. 2. The inset explicitly shows the magnitude of the low-frequency power spectra for the background and the vortex, illustrating a featureless spectrum for the background and a pronounced peak for the vortex. Note that the vorticity and the power spectrum are measured in arbitrary units.

corresponding to the y-register. This is repeated for each flow to create a quantum dataset $\{\boldsymbol{\rho}\} = \{|\psi_{{\rm f},q}\rangle\langle\psi_{{\rm f},q}|\}_{q=1}^{N}$ consisting of N encoded flow-field solutions. The workflow outlining how features are extracted from these flows is shown in Fig. 3. In particular, the workflow is designed to identify rotational symmetries in the states, which signal the presence of vortices.

Next, in a similar vein to the CNN, a window is selected from the flow domain. A schematic of a sliding window traversing a vorticity flow field is shown in Fig. 3(a). To achieve this, a shift gate $S(n, d)$ is applied to $|\psi_{\rm f}\rangle$ to select the desired window. This shift gate is followed by a permutation gate $P(n, l)$ which moves the elements of the window to the top register of the circuit. This results in a state $|\psi_{\rm w}\rangle$ which encodes the window in the top $n_{\rm w}$ qubits. For a detailed stepwise example of these operations, refer to the Appendix. In the QVD model, the step size between consecutive window evaluations is treated as a tunable parameter α. This helps reduce the number of circuit evaluations while ensuring that there is sufficient space between windows to detect all vortices.

Next, a circular contour is selected from each window. An example of a circular contour as extracted from two distinguishable windows is demonstrated in Fig. 3(b), with one representing the background and the other representing a vortex. Extracting the contour enables tracing the variation in the magnitude of vorticity while accounting for rotational symmetry, effectively producing a state that encodes the vorticity amplitudes. Analyzing the contour of a vorticity window allows us to isolate regions where amplitudes exhibit certain structure. Figure 3(c) shows a comparison of the vorticity amplitudes of the contours extracted from both the background window and the vortex window. To achieve this, a permutation gate is applied to $|\psi_{\rm w}\rangle$ to move the elements of the contour (in either a clockwise or anticlockwise order) to the top register of the circuit. This results in a state $|\psi_{\rm c}\rangle$ that encodes the contour in the top $n_{\rm c}$ qubits. For a detailed example of this operation, refer to the Appendix. When comparing the vorticity amplitudes, the contour from the background window has a fairly flat profile, but the contour from the vortex window has a distinguishable oscillatory profile. Since vortices are typically characterized by oscillatory vorticity profiles due to rotational symmetry, examining circular contours provides a focused representation of how vorticity varies around a central point. In the QVD model, the inverse contour radius is treated as a tunable parameter β to account for variations in vortex size and ensure optimal detection.

Given that features are hidden in oscillatoric patterns, we use QFT for the contour (state $|\psi_{\rm ps}\rangle$) and sample the corresponding power spectrum to identify signatures of vortical structures (prominent low-frequency components along the contour). Therefore, an adjustable circuit used to extract features is represented by

$$\begin{aligned}|\psi_{\rm ps}\rangle &= \hat{U}(\alpha, \beta; n_{\rm f}, n_{\rm w}, n_{\rm c})|\psi_{\rm f}\rangle \\ &= {\rm QFT}(n_{\rm c})P[n_{\rm w}, l_2(\beta)]P(n_{\rm f}, l_1)S[n_{\rm f}, d(\alpha)]|\psi_{\rm f}\rangle. \quad (6)\end{aligned}$$

Here, the shift operator selects the position of each window according to the step size α, while the permutation operator reorders the data specified by the contour with the inverse radius β.

Finally, the low-frequency region of the power spectrum is extracted. This can be achieved by performing projective measurements on the bottom part of the register (effectively filtering out high frequencies). This results in a state $|\psi_{\rm lfps}\rangle$ that encodes the low-frequency power spectrum in the top $n_{\rm lfps}$ qubits, $\mathcal{P} = |\Pi(n_{\rm lfps})|\psi_{\rm ps}\rangle|^2$. More specifically, for contour index p and flow-field index $q \in [1, N]$, this power spectrum is given by

$$\mathcal{P}_{p,q} = |\Pi(n_{\rm lfps})\hat{U}_p(\alpha, \beta)|\psi_{{\rm f},q}\rangle|^2. \quad (7)$$

If each flow in the dataset $\boldsymbol{\rho}$ is indexed by q, the power spectrum must be computed for every extracted contour p as it moves through the field. Figure 3(d) shows the power spectra of the different states extracted from the window, with a particular focus on contrasting the low-frequency signals between the background window and the vortex window. Vortex signatures are typically present in the frequency region $[0, N_{\rm lf}]$, where $N_{\rm lf}$ is 32 for our example in Fig. 3(d). The top power spectrum of the background has a weak signal with no detectable characteristics in the low-frequency band. In comparison, the bottom power spectrum of the vortex has a sharp peak in the low-frequency band with a maximum value of approximately 2.4.

The features in $|\psi_{\rm lfps}\rangle$ are analyzed to distinguish between the presence or absence of a vortex. When a vortex is present, there is a peak in the low-frequency power spectrum as

demonstrated from Fig. 3(d). A threshold value for the peak is used to classify whether a vortex is present or absent within the window. We set the threshold as a tunable parameter γ, adjusted based on available data to separate vortex signal from background fluctuations or noise.

We apply classical postprocessing to eliminate overcounting and identify unique vortex detections. Each circuit applied to each patch returns a binary value of $\mathcal{M}_{p,q}(\alpha, \beta, \gamma)$. The QVD model therefore often detects the same vortex multiple times across neighboring contour evaluations. To count unique vortices only, we average regions where multiple detections occur, such that $\mathcal{M}_q(\alpha, \beta, \gamma)$ provides 0 or 1 detection values over the set of contour values. This postprocessing can also aid in determining the central locations of the vortices.

Notably, the QVD model can be integrated into a machine learning framework by treating the three parameters as tunable variables with which to optimize vortex detections, based on known examples of vortices. This leads to the physics-informed QML approach, where inductive bias is derived from the analysis of the low-frequency power spectrum and optimized based on physically motivated variables.

C. Parallel QVD

As an important step to avoid repeated QVD application, the quantum vortex detection approach can be parallelized. This step is needed to accelerate the processing of flow-field solutions and enable coherent feature selection. Specifically, we suggest processing solutions with a linear combination of positions (shifts) $\sum_x |x\rangle\langle x| \otimes \hat{U}_x(\alpha, \beta)$, selected such that the window positions effectively cover the grid (as typical for quantum phase estimation protocols [117]). Note that the grid can be coarse-grained depending on the typical vortex size. Next, we employ the readout strategy proposed in Ref. [120], where the ancillary register stores positions in the amplitudes of states $|x\rangle$, and the relevant (low k-frequency) component is selected. For instance, this can be achieved by implementing a rank-1 projector $\Pi_k = |k\rangle\langle k|$ and acting on the shifted state as $\Pi_k|\psi_{\text{ps}}\rangle$. Finally, the QFT is applied on the ancillary register such that we read out information about the spectral (i.e., global) properties of the flow field (Fig. 4). This higher-order spectrum is referred to as the density spectrum. The associated probability distribution $p_{\text{a}}(x)$ can be sampled on the ancillary register and postprocessed to get the global properties of each example.

Let us demonstrate the utility of this approach with the following example. We consider two distinct types of flow fields, separating samples into those with vortical and nonvortical patterns [Fig. 5(a)]. Each quantum data sample is processed coherently, such that selected low-frequency components are pushed to the ancillary register, forming a state that contains information about the entire field (its spectral properties). Next, the QFT is applied on the ancillary register, aiming to highlight states with flat profiles (i.e., frequency response from all positions is similar but noisy) as compared to those with significant k-frequency components (i.e., certain areas show pronounced response while other parts of the flow have a low signal). The corresponding examples are shown in Fig. 5(b). The density spectrum of a nonvortical field is

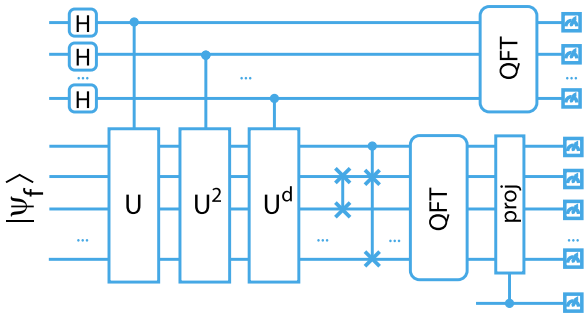

FIG. 4. Quantum circuit for parallelized feature processing in quantum vortex detection. Conditioned on the superposition state $|+\rangle^{\otimes n_a}$, operators corresponding to different shifts are applied (each corresponding to an associated ancilla state $|x\rangle$), followed by permutations and QFT. Projecting out the bottom register, the low-frequency amplitudes are pushed to the ancilla register, which is to be sampled in the frequency basis.

distinct from that of a vortical field containing any number of vortices. This higher-order spectral analysis is necessary for identifying the density of peaks in the power spectrum $\mathcal{P}(\tilde{k})$, which reflects the underlying vortex density of the flow-field solutions.

Given the variability in vorticity, as well as the number, location, and strength of vortices, it is useful to compute a representative density spectrum distribution for each flow category. These distributions capture the characteristic spectral features of each class, enabling more robust comparison and feature extraction across different datasets. To construct such distributions, the density spectrum of each field is first computed and then repeatedly sampled (via usual projective measurements). The resulting measurement results are concatenated to form the class-level distributions shown in Fig. 5(c), which are representative of the density spectra produced by fields in each class of the quantum dataset. The nonvortical distribution $p_{\text{NV}}(x)$ does not exhibit distinctive features, while the vortical distribution $p_{\text{V}}(x)$ is nonzero only at certain frequencies. This shows that the spectral signatures of nonvortical and vortical fields are sufficiently different to enable clear class separation.

Access to a representative distribution for each class can thus be leveraged for efficient classification. Empirical distributions can be sampled from their respective representative distributions $p_{\text{NV}}(x)$ and $p_{\text{V}}(x)$. These empirical distributions can be postprocessed by training a classical ML classifier that learns the separation between nonvortical and vortical regimes. Upon training, we obtain empirical test distributions by sampling the density spectrum of a new field that did not form part of the original dataset composed in Fig. 5(a). This test distribution can then be fed into the classifier to determine if its underlying field is nonvortical or vortical.

III. RESULTS

In this section, we discuss the use of supervised quantum machine learning to develop an optimized vortex detection model based on the Lamb-Oseen vortex solution of the viscous Navier-Stokes equations. For simplicity in exhibiting how the QVD model is performed, the solutions used for

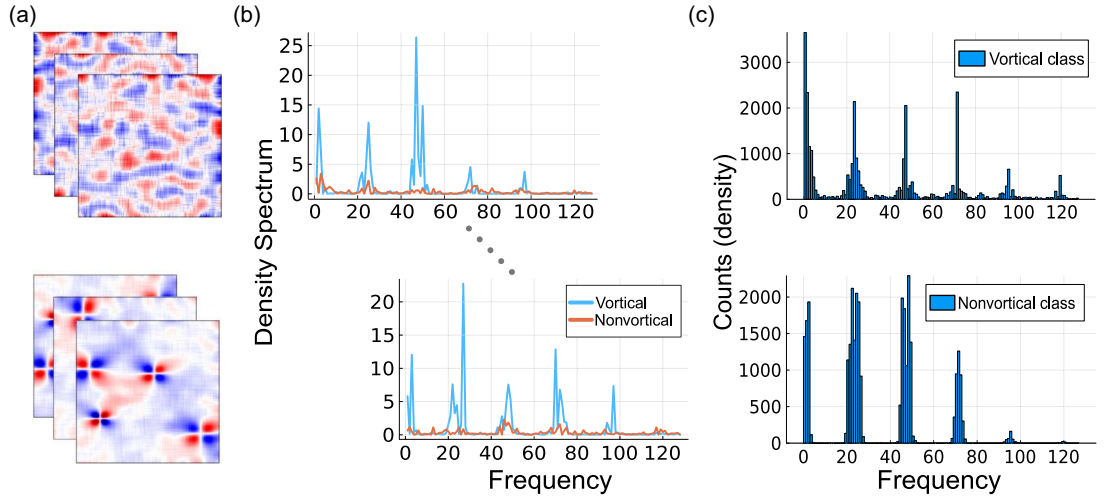

FIG. 5. Quantum data-driven feature processing in QVD. (a) A quantum dataset containing vorticity flow field solutions $|\psi_{i,j}\rangle$ categorized into nonvortical and vortical classes. (b) The density spectra for each pair of vortical and nonvortical fields, where frequencies are indexed by single power spectrum values of contours N_c processed in parallel. These spectra are obtained with the quantum circuit from Fig. 4 and truncated to seven qubits. The top vortical density spectrum corresponds to a field with four vortices and the bottom to one with seven vortices. (c) The representative density spectrum distribution of the nonvortical and vortical classes obtained from sampling the individual density spectra of all fields in the dataset. Note that the density spectrum is measured in arbitrary units.

training and testing are obtained from classical simulations using the model from Eq. (1). It is a separate challenge to prepare the input data (often with ancilla overheads that prevent testing), and here we concentrate on the readout procedure.

Vorticity fields for representative Lamb-Oseen solutions are prepared as quantum states and used as inputs to the QML model, forming the quantum dataset $\{\boldsymbol{\rho}\}$. It contains vorticity fields, each with grid dimensions of 200×200. The two-dimensional velocity fields $\boldsymbol{v} = (v_x, v_y)$ are first generated by randomly placing a set of Lamb-Oseen vortices of different strength, size, and orientation onto the grid [68], achieved by sampling the parameters Δ, $v_{\max}$, and r in Eq. (1) as well as randomly selecting the number of vortices $M \in [4, 8]$, their location, and orientation. We overlay each flow field with a random velocity field defined with a smoothed Gaussian filter to act as noise. The vorticity fields $\psi = \nabla \times \boldsymbol{v}$ are then calculated. The baseline setup assumes vortices are sufficiently separated to be individually identifiable. Nonetheless, we consider the proposed method to be adjustable such that it can be extended to handle more closely packed vortex configurations.

A. Sequential QVD

To effectively implement and evaluate the QVD model as used for supervised vortex counting, we use an optimization procedure for selected model parameters (window step size, inverse contour radius, power spectrum threshold) and evaluate the performance of trained models. Specifically, we minimize a mean squared error (MSE) loss function that encapsulates the number of correctly detected vortices. The MSE is given by

$$\mathrm{MSE}(\alpha, \beta, \gamma) = \sum_{q=1}^{N} \frac{[M_q - \mathcal{M}_q(\alpha, \beta, \gamma)]^2}{N}, \tag{8}$$

where $\mathcal{M}_q$ is the estimated number of unique detections produced by the QVD model for the flow field q and parameter set $\{\alpha, \beta, \gamma\}$. Optimization is also performed to maximize the accuracy of detections. The accuracy is defined as the proportion of exact matches between true and predicted vortex counts.

We show the results obtained using a Bayesian optimization procedure in Fig. 6. The training stage for the QVD model is demonstrated with the MSE loss history in Fig. 6(a). The optimization is summarized over five instances as applied to a sample set of $N = 60$ flow fields that is split into a 75% training set and a 25% testing set. The optimization is able to retrieve a MSE of less than 1 in fewer than 15 epochs. The physically motivated parameters directly influence the sensitivity and accuracy of the vortex detection model, as reflected in the initially high MSE that decreases significantly during early optimization stages. The generalization of the QVD model is demonstrated with the accuracy in Fig. 6(b). Here, the optimization procedure identifies the best parameters for vortex detection by maximizing accuracy on an increasing number of training samples. The performance is then measured via the accuracy on the test set, which contains a randomized selection of 30 flow fields not seen during training. This accuracy measures the success rate of vortex detection and how well the optimized parameters of the QVD model generalize to unseen data when trained on datasets of increasing size. This result demonstrates how performance differs from large to small training sets and shows that an accuracy of 80% is achievable with as few as 30 fields used for training.

In Fig. 7 we show the output of the vortex detection model as applied to an example vorticity field. This vorticity field is derived from a randomly generated Lamb-Oseen velocity field consisting of seven vortices. This result is achieved through an exhaustive grid search on all N fields, which identifies the optimal model parameters based on a ground truth MSE

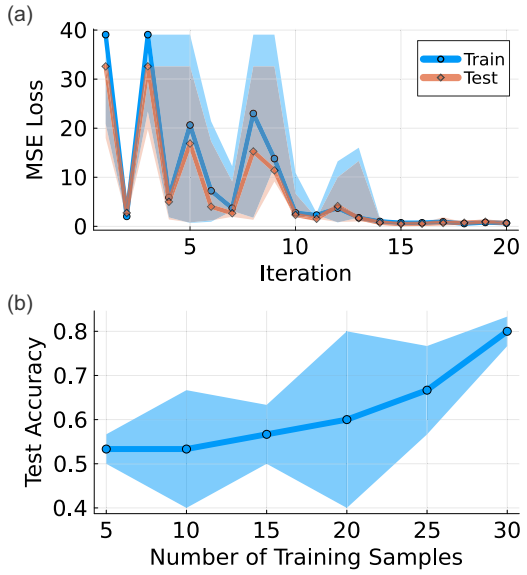

FIG. 6. Trainability and generalizability of the QVD model using Bayesian optimization. Curves and dots represent the median, and shaded areas represent the interquartile range as obtained from five instances with different seed values. (a) Training and testing of the model where we use optimization to minimize the MSE with respect to the model's parameters. The loss history is shown for a model trained on 45 fields and tested on 15 fields. A final MSE loss of 0.6 for the train set and 0.7 for the test set is reached after 20 epochs. (b) The generalizability of the model where optimization is used to maximize the accuracy with respect to the model's parameters. Accuracy is optimized on a growing training set and evaluated on a separate test set of size 30 containing only unseen fields.

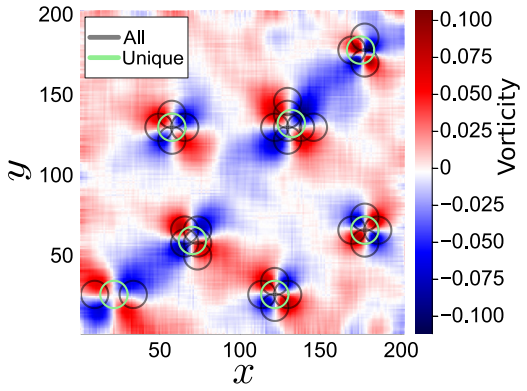

FIG. 7. Example of the output obtained from the vortex detection model as applied to a Lamb-Oseen flow. The two-dimensional vorticity flow field is shown in the xy-plane and has a dimension of 200×200. The flow contains seven Lamb-Oseen vortices of varying size, strength, and orientation positioned randomly on the grid. Black circles represent all detections made by the model, while green circles indicate the final unique detections after averaging across all detections to remove overcounting. The model uses the optimal parameter values consisting of a window step size of $\alpha = 8$, a contour with an inverse radius of $\beta = 3$, and a power spectrum threshold of $\gamma = 0.9$, resulting in a MSE of 0.25.

of 0.25. The black circles represent the contours where their low-frequency power spectrum peaks above the threshold, as used to indicate the presence of a vortex. The green circles represent the unique vortices, as obtained from averaging over regions where multiple contours overlap. This model output has accurately detected both the number of vortices in the flow field and the central positions of each vortex, demonstrating high accuracy of detection on quantum data.

B. Parallel QVD

To assess the QVD algorithm as used for classification, we use classical ML models trained on empirical (sampled) density spectra distributions. In this classification task, we aim to separate flow behavior into nonvortical or vortical classes. This can later be used to verify if an unseen density spectrum was generated from a nonvortical or vortical field, enabling predictions of system properties without tomographic procedures. Each empirical distribution is sampled with a fixed number of shots. The number of empirical distributions in each set is such that each set contains exactly 10 000 measurements; when shot size is low, the number of repetitions is high and vice versa. We consider this procedure to provide consistency in assessing classification accuracy as a function of the number of shots.

In Fig. 8 we show results obtained using a standard random forest classifier. The underlying dataset contained N fields, 30 fields in each class. The nonvortical fields contain no vortices, just random fluctuations, while vortical fields contain $M \in [1, 8]$ Lamb-Oseen vortices. Each field is processed using the parallel QVD algorithm, with optimal parameters determined by Bayesian optimization as provided in Fig. 7. We then compute the density spectrum for each field and truncate it to eight qubits. These density spectra are sampled and concatenated to generate representative distributions for each class. We generate the empirical distributions, with sample sets for different shot sizes used to train and test independent classifiers. Each sample set undergoes fivefold cross-validation, after which the F1 score and area under the curve (AUC) are calculated. The F1 score is a measure that summarizes how well a classifier identifies the correct examples of each class [121]. It combines two aspects of performance: how many of the predicted positives are correct (precision) and how many of the actual positives are found (recall). A higher F1 score indicates a better balance between these two and thus a more reliable classification performance. The AUC (area under the receiver operating characteristic curve) measures how well the classifier distinguishes between the different classes overall, regardless of the threshold used for making decisions. An AUC of 1 corresponds to perfect separation, while 0.5 indicates performance no better than random guessing. As expected, the result shows that classification accuracy improves with more measurement shots. Notably, the results indicate that a high F1 score of approximately 94% can be achieved with as few as five shots (for the given system size). Additional tests on more closely packed vortex configurations yielded comparable accuracy of approximately 89% with five shots, indicating that the QVD approach remains effective even when vortices are not well separated.

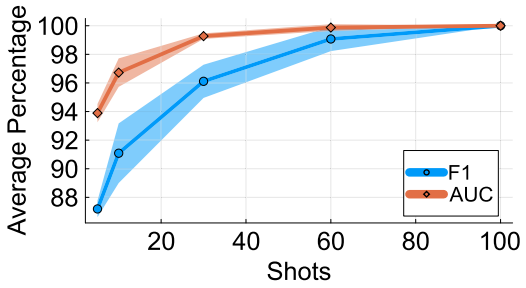

FIG. 8. Classification from compressed feature representation using spectral density. Classification is performed on empirical density spectra sampled from the representative distributions of the nonvortical and vortical classes using a random forest classifier. The representative distributions are generated from an equally balanced dataset of 60 vortical and nonvortical flow fields, with density spectra computed using parallel QVD truncated to eight qubits. Curves and dots represent the average F1 score and AUC metrics, and shaded areas represent the standard deviation obtained with fivefold cross-validation. The number of samples used to generate the empirical density spectra are given by the number of shots. The number of generated distributions used for classification is given by 10000 shots to ensure each model run contains the same number of measurements.

To assess the performance, we need to establish baselines. For this, we consider the most typical quantum machine learning approaches based on parametrized quantum circuits (also known as quantum neural networks). The QVD model demonstrates excellent performance compared to other variational QML approaches in the classification of quantum vorticity field data. A comparison in the brute-force classification of quantum data into vortical (turbulent) and nonvortical (laminar) regimes using different methods is shown in Table I. Each model is trained on a quantum dataset similar to Fig. 5(a), consisting of $N = 40$ field solutions, each encoded into quantum states using 16 qubits. The classification accuracy is evaluated on test sets across four independently generated datasets. In the QVD implementation, the representative density spectrum distribution for each class is constructed from all N flows, and the subsequent empirical distributions used in classification are sampled with 10 shots. The benchmark QML models include the quantum deep neural network (QDNN) and the QCNN [96,102]. Each neural network architecture uses a hardware efficient ansatz, trained using the Adam optimizer with a learning rate of 0.1 over 50 iterations. The comparison to a classical CNN is not appropriate, since the use of quantum data renders classical approaches inapplicable. The result of this comparison shows the importance of developing bespoke quantum algorithms and the need to design quantum CFD workflows that operate natively within quantum simulation environments.

TABLE I. Comparison of different classification methods. Scores for different approaches (QNN-based and QVD) are summarized for distinguishing turbulent and laminar flows in quantum vorticity field data. Each model is trained on 30 fields and tested on 10 fields. Accuracy is measured on the test set, with averages and standard deviations computed over four runs using different seed values. Classical neural networks are not applicable as we work with quantum datasets.

Method	Test Accuracy (%)
QDNN	63 ± 30
QCNN	63 ± 32
QVD	93 ± 0.3
CNN (classical)	not applicable

IV. CONCLUSION

We proposed an algorithm for vortex detection on quantum data, where solutions to differential equations are analyzed with circuits of specific structure. The approach, named quantum vortex detection (QVD), relies on extracting features of flow fields via sliding windows (motivated by convolutional approaches), followed by Fourier analysis along contours to identify signals inherent to structures with vorticity. We described the required sequence of operations for shifting, permuting, and transforming states, and demonstrated QVD on examples that encode Lamb-Oseen vortices. In particular, we showed that vortices can be detected with high accuracy when using a few-parameter model trained to set an optimal detection threshold. Using the coherent sample loading via quantum parallelism, we performed classification of vortical versus nonvortical solutions, greatly outperforming other quantum neural network approaches for quantum data.

As a direction for future works we consider extensions of these quantum readout tools to other coherent structures, such as eddies, which are often more chaotic and challenging to detect [122]. The modular and adaptable nature of the QVD workflow opens the possibility of extensions to detect closely spaced vortices or vortex trains through spectral feature processing, consistent with classical approaches in reduced-order modeling. Also, it would be beneficial to investigate how this detection model can work in a quantum-inspired regime with differential equation solutions stored as a tensor network [123]. Finally, an intriguing possibility for detecting topological flow features is using quantum methods for topological data analysis [124–128], if modified appropriately for the type of data in question.

ACKNOWLEDGMENTS

This work was supported by Siemens Industry Software NV. O.K. acknowledges support from the QCi3 Hub (Grant No. EP/Z53318X/1).

DATA AVAILABILITY

The data that support the findings of this article are openly available [129].

APPENDIX

To provide an explicit demonstration of the window contour extraction workflow introduced in the main text, we present a stepwise visualization for an example of quantum register manipulation (Fig. 9). Here, quantum states are visualized as heatmaps, where the top rows represent amplitudes stored in the upper registers of the quantum circuit, and the bottom rows correspond to amplitudes in the lower registers.

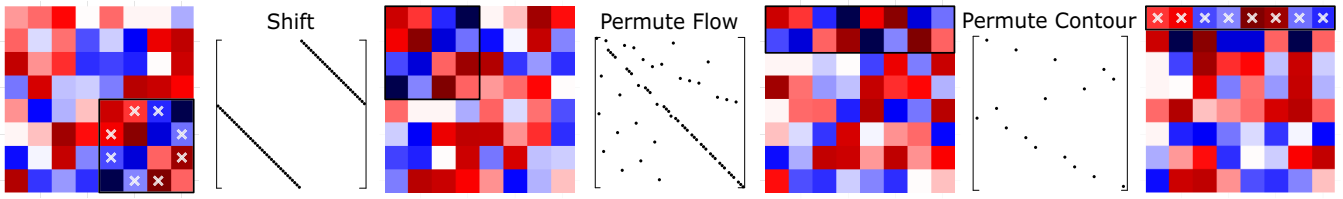

FIG. 9. Illustration of how the sequence of shift and permutation operations transforms the quantum state encoding the flow, enabling extraction of the contour for a selected window within a chosen flow field. The process shows how specific elements are repositioned and compressed as the circuit evolves until only the correctly ordered contour points remain encoded in the topmost portion of the register. The heatmaps depict quantum state representations and the sparse matrices depict unitary quantum operations. Black borders bound the important elements that belong to the selected window, whose contour is highlighted with white crosses.

The black borders are used to visualize the selected window to be rearranged. The unitary quantum operations used to transform the states are visualized using sparse matrix notation, where dots represent the locations of 1's and empty space represents 0's.

We start with a quantum state represented by 64 amplitudes, where the flow is visualized as a heatmap. The chosen flow contains a single vortex in the bottom right corner (Fig. 9, left). The objective is to select the window with the black border and extract the contour points marked by white crosses.

The first step is to select the desired window of length 16. Application of the shift gate $S(n = 6)$ promotes the chosen window from the bottom of the register to the top, effectively reindexing the circuit such that the relevant subregion populates the highest-order qubits. The structure of the required shift matrix is shown in the first sparse matrix representation. The window is now mapped to the upper half of the register. This is indicated by the relocation of the black-bordered vortex to the top (see the heatmap in the middle). We then apply the first permutation gate $P(n = 6)$ (`Permute Flow`), whose structure is shown in the second sparse matrix representation. This operation rearranges the window elements in the top half to further localize them within part of the register (top qubit lines). The second permutation gate $P(n = 4)$ (`Permute Contour`), whose structure is shown in the third sparse matrix representation, is then applied. This operation acts within the top quarter to order the contour sites themselves into the very top of the register, ensuring that the ordered contour is stored in the top three qubits. From here, the features for a clockwise vortex can be extracted.

As a result of these sequential operations, only the relevant contour points which are ordered according to the original flow geometry are encoded in the topmost register of the circuit. This isolation procedure extracts the low-dimensional state required for downstream vortex detection. The procedure can be repeated to extract contours from different windows that are located within the flow by applying modified shift and permutation operations.

[1] A. N. Kolmogorov, Proc. R. Soc. Lond. Ser. A **434**, 9 (1991).
[2] S. K. Robinson, Annu. Rev. Fluid Mech. **23**, 601 (1991).
[3] J. Jiménez, A. A. Wray, P. G. Saffman, and R. S. Rogallo, J. Fluid Mech. **255**, 65 (1993).
[4] J. O'Connor, P. Day, P. Mandal, and A. Revell, Integr. Biol. **8**, 589 (2016).
[5] J. Anderson, *Computational Fluid Dynamics: The Basics with Applications*, McGraw-Hill International Editions: Mechanical Engineering (McGraw-Hill, New York, 1995).
[6] A. K. M. F. Hussain, J. Fluid Mech. **173**, 303 (1986).
[7] J. Jeong and F. Hussain, J. Fluid Mech. **285**, 69 (1995).
[8] G. Lindner, Y. Devaux, and S. Miskovic, SoftwareX **12**, 100604 (2020).
[9] K. Shibata and T. Magara, Liv. Rev. Solar Phys. **8**, 6 (2011).
[10] P. Goldreich and S. Sridhar, Astrophys. J. **438**, 763 (1995).
[11] M.-M. Mac Low and R. S. Klessen, Rev. Mod. Phys. **76**, 125 (2004).
[12] L.-A. Couston, D. Lecoanet, B. Favier, and M. Le Bars, Phys. Rev. Res. **2**, 023143 (2020).
[13] J. C. McWilliams, J. Fluid Mech. **146**, 21 (1984).
[14] P. R. Spalart and S. R. Allmaras, Recherche Aérospatiale 5 (1994).
[15] J. Smagorinsky, Month. Weather Rev. **91**, 99 (1963).
[16] F. Sadlo, R. Peikert, and E. Parkinson, *IEEE Visualization 2004* (IEEE, Texas, USA, 2004), pp. 179–186.
[17] Q. Yang, W. Li, L. Ji, W. Shi, W. Pu, Y. Long, and X. He, J. Mater. Sci. Eng. **11**, 2209 (2023).
[18] L. J. Vermeer, J. N. Sørensen, and A. Crespo, Prog. Aerosp. Sci. **39**, 467 (2003).
[19] M. J. Lighthill, Proc. R. Soc. Lond. Ser. A **211**, 564 (1952).
[20] P. W. Bearman, Annu. Rev. Fluid Mech. **16**, 195 (1984).
[21] P. Moin and K. Mahesh, Annu. Rev. Fluid Mech. **30**, 539 (1998).
[22] Y. Kaneda, T. Ishihara, M. Yokokawa, K. Itakura, and A. Uno, Phys. Fluids **15**, L21 (2003).
[23] M. Lee and R. D. Moser, J. Fluid Mech. **774**, 395 (2015).
[24] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).
[25] A. M. Dalzell, S. McArdle, M. Berta, P. Bienias, C.-F. Chen, A. Gilyén, C. T. Hann, M. J. Kastoryano, E. T. Khabiboulline, A. Kubica, G. Salton, S. Wang, and F. G. S. L. Brandão, *Quantum Algorithms: A Survey of Applications and End-to-End Complexities* (Cambridge University Press, Cambridge, 2025).
[26] Y. Cao, A. Papageorgiou, I. Petras, J. Traub, and S. Kais, New J. Phys. **15**, 013021 (2013).

[27] A. Montanaro and S. Pallister, Phys. Rev. A **93**, 032324 (2016).
[28] T. Xin, S. Wei, J. Cui, J. Xiao, I. Arrazola, L. Lamata, X. Kong, D. Lu, E. Solano, and G. Long, Phys. Rev. A **101**, 032307 (2020).
[29] A. Martin, R. Ibarrondo, and M. Sanz, Phys. Rev. Appl. **19**, 064056 (2023).
[30] C. A. Williams, A. E. Paine, H.-Y. Wu, V. E. Elfving, and O. Kyriienko, arXiv:2306.17026.
[31] O. Kyriienko, A. E. Paine, and V. E. Elfving, Phys. Rev. Res. **6**, 033291 (2024).
[32] L. Lin and Y. Tong, Quantum **4**, 361 (2020).
[33] N. Linden, A. Montanaro, and C. Shao, Commun. Math. Phys. **395**, 601 (2022).
[34] H. Krovi, Quantum **7**, 913 (2023).
[35] P. C. S. Costa, P. Schleich, M. E. S. Morales, and D. W. Berry, npj Quantum Inf. **11**, 141 (2025).
[36] A. M. Childs, R. Kothari, and R. D. Somma, SIAM J. Comput. **46**, 1920 (2017).
[37] A. M. Childs and N. Wiebe, Quantum Inf. Comput. **12**, 901 (2012).
[38] D. W. Berry, A. M. Childs, A. Ostrander, and G. Wang, Commun. Math. Phys. **356**, 1057 (2017).
[39] D. Jennings, M. Lostaglio, S. Pallister, A. T. Sornborger, and Y. Subaşı, arXiv:2305.11352.
[40] S. Gribling, I. Kerenidis, and D. Szilágyi, ACM Trans. Quantum Comput. **5**, 1 (2024).
[41] S. Jin, N. Liu, and Y. Yu, Phys. Rev. A **108**, 032603 (2023).
[42] S. Jin and N. Liu, Quantum Sci. Technol. **9**, 035047 (2024).
[43] S. Jin and N. Liu, Proc. R. Soc. A **480**, 20230370 (2024).
[44] J. Hu, S. Jin, N. Liu, and L. Zhang, Quantum **8**, 1563 (2024).
[45] B. Liu, M. Ortiz, and F. Cirak, Comput. Methods Appl. Mech. Eng. **432**, 117403 (2024).
[46] M. Lubasch, Y. Kikuchi, L. Wright, and C. M. Keever, arXiv:2505.16895.
[47] E. Devereux and A. Datta, Phys. Rev. A **112**, 042419 (2025).
[48] Y. Subaşı, R. D. Somma, and D. Orsucci, Phys. Rev. Lett. **122**, 060504 (2019).
[49] P. C. S. Costa, D. An, Y. R. Sanders, Y. Su, R. Babbush, and D. W. Berry, PRX Quantum **3**, 040303 (2022).
[50] D. An and L. Lin, ACM Trans. Quantum Comput. **3**, 1 (2022).
[51] C. A. Williams, A. A. Gentile, V. E. Elfving, D. Berger, and O. Kyriienko, Adv. Quantum Technol. e00618 (2025).
[52] O. M. Raisuddin and S. De, arXiv:2404.07466.
[53] P. Jaksch, arXiv:2201.04513.
[54] J.-P. Liu, H. Øie Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa, and A. M. Childs, Proc. Natl. Acad. Sci. USA **118**, e2026805118 (2021).
[55] Y. Tanaka, Y. Ito, and K. Fujii, Sci. Rep. **13**, 4531 (2023).
[56] J. Ingelmann, S. S. Bharadwaj, P. Pfeffer, K. R. Sreenivasan, and J. Schumacher, Comput. Fluids **281**, 106369 (2024).
[57] H.-C. Wu, J. Wang, and X. Li, arXiv:2405.12714.
[58] J. Gonzalez-Conde, D. Lewis, S. S. Bharadwaj, and M. Sanz, Phys. Rev. Res. **7**, 023254 (2025).
[59] C. Sanavio, R. Scatamacchia, C. de Falco, and S. Succi, Phys. Fluids **36**, 057143 (2024).
[60] C. Sanavio, E. Mauri, and S. Succi, arXiv:2406.01118.
[61] A. E. Paine, V. E. Elfving, and O. Kyriienko, arXiv:2308.01827.
[62] H.-Y. Wu, A. E. Paine, E. Philip, A. A. Gentile, and O. Kyriienko, arXiv:2504.13174.
[63] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch, Phys. Rev. A **101**, 010301(R) (2020).
[64] D. Jaksch, P. Givi, A. J. Daley, and T. Rung, AIAA J. **61**, 1885 (2023).
[65] A. J. Pool, A. D. Somoza, C. McKeever, M. Lubasch, and B. Horstmann, Phys. Rev. Res. **6**, 033257 (2024).
[66] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Nature (London) **549**, 195 (2017).
[67] C. A. Williams, S. Scali, A. A. Gentile, D. Berger, and O. Kyriienko, arXiv:2411.14259.
[68] J. R. Canivete Cuissa and O. Steiner, Astron. Astrophys. **668**, A118 (2022).
[69] B. Epps, Review of vortex identification methods, in *55th AIAA Aerospace Sciences Meeting* (American Institute of Aeronautics and Astronautics, Texas, USA, 2017).
[70] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, Annu. Rev. Fluid Mech. **52**, 477 (2020).
[71] K. Duraisamy, G. Iaccarino, and H. Xiao, Annu. Rev. Fluid Mech. **51**, 357 (2019).
[72] C. M. Ströfer, J.-L. Wu, H. Xiao, and E. Paterson, Commun. Comput. Phys. **25**, 625 (2025).
[73] Y. Wang, D. Liang, Z. Yang, D. Zhao, and F. Wang, Visual Comput. **37**, 261 (2021).
[74] L. Deng, W. Bao, Y. Wang, Z. Yang, D. Zhao, F. Wang, C. Bi, and Y. Guo, Appl. Soft Comput. **115**, 108229 (2022).
[75] D. Liang, Y. Wang, Y. Liu, F. Wang, S. Li, and J. Liu, J. Visual. **22**, 65 (2018).
[76] J. Wang, L. Guo, Y. Wang, L. Deng, F. Wang, and T. Li, Int. J. Aerosp. Eng. **2020**, 8865001 (2020).
[77] H.-Y. Huang, M. Broughton, J. Cotler, S. Chen, J. Li, M. Mohseni, H. Neven, R. Babbush, R. Kueng, J. Preskill, and J. R. McClean, Science **376**, 1182 (2022).
[78] L. Schatzki, A. Arrasmith, P. J. Coles, and M. Cerezo, arXiv:2109.03400.
[79] M. C. Caro, H.-Y. Huang, M. Cerezo, K. Sharma, A. Sornborger, L. Cincio, and P. J. Coles, Nat. Commun. **13**, 4919 (2022).
[80] E. R. Anschuetz, H.-Y. Hu, J.-L. Huang, and X. Gao, PRX Quantum **4**, 020338 (2023).
[81] C. Umeano, V. E. Elfving, and O. Kyriienko, arXiv:2402.03871.
[82] C. Umeano, S. Scali, and O. Kyriienko, arXiv:2409.01496.
[83] L. Lewis, D. Gilboa, and J. R. McClean, arXiv:2503.20879.
[84] Y. Morohoshi, A. Nakayama, H. Manabe, and K. Mitarai, arXiv:2504.16370.
[85] A. Barthe, M. Y. Rad, M. Grossi, and V. Dunjko, arXiv:2506.17089.
[86] O. Kyriienko, A. E. Paine, and V. E. Elfving, Phys. Rev. A **103**, 052416 (2021).
[87] A. E. Paine, V. E. Elfving, and O. Kyriienko, Adv. Quantum Technol. **6**, 2300065 (2023).
[88] A. E. Paine, V. E. Elfving, and O. Kyriienko, Phys. Rev. A **107**, 032428 (2023).
[89] S. Markidis, Front. Appl. Math. Statist. **8**, 1036711 (2022).
[90] N. Heim, A. Ghosh, O. Kyriienko, and V. E. Elfving, arXiv:2111.06376.
[91] S. Kasture, O. Kyriienko, and V. E. Elfving, Phys. Rev. A **108**, 042406 (2023).

[92] A. Setty, R. Abdusalamov, and F. Motzoi, Mach. Learn.: Sci. Technol. **6**, 015002 (2025).
[93] B. Jaderberg, A. A. Gentile, Y. A. Berrada, E. Shishenina, and V. E. Elfving, Phys. Rev. A **109**, 042421 (2024).
[94] B. Jaderberg, A. A. Gentile, A. Ghosh, V. E. Elfving, C. Jones, D. Vodola, J. Manobianco, and H. Weiss, Phys. Rev. A **110**, 052423 (2024).
[95] J. J. M. de Lejarza, H.-Y. Wu, O. Kyriienko, G. Rodrigo, and M. Grossi, arXiv:2503.16073.
[96] I. Cong, S. Choi, and M. D. Lukin, Nat. Phys. **15**, 1273 (2019).
[97] A. Pesah, M. Cerezo, S. Wang, T. Volkoff, A. T. Sornborger, and P. J. Coles, Phys. Rev. X **11**, 041011 (2021).
[98] T. Hur, L. Kim, and D. K. Park, Quantum Mach. Intell. **4**, 3 (2022).
[99] S. Y.-C. Chen, T.-C. Wei, C. Zhang, H. Yu, and S. Yoo, Phys. Rev. Res. **4**, 013231 (2022).
[100] J. Herrmann, S. M. Llima, A. Remm, P. Zapletal, N. A. McMahon, C. Scarato, F. Swiadek, C. K. Andersen, C. Hellings, S. Krinner *et al.*, Nat. Commun. **13**, 4144 (2022).
[101] P. Zapletal, N. A. McMahon, and M. J. Hartmann, Phys. Rev. Res. **6**, 033111 (2024).
[102] C. Umeano, A. E. Paine, V. E. Elfving, and O. Kyriienko, Adv. Quantum Technol. **8**, 2400325 (2024).
[103] E. Gil-Fuster, J. Eisert, and C. Bravo-Prieto, Nat. Commun. **15**, 2277 (2024).
[104] Y. Song, J. Li, Y. Wu, S. Qin, Q. Wen, and F. Gao, Front. Phys. **12**, (2024).
[105] M. L. Goh, M. Larocca, L. Cincio, M. Cerezo, and F. Sauvage, Phys. Rev. Res. **7**, 033266 (2025).
[106] M. Cerezo, M. Larocca, D. García-Martín, N. L. Diaz, P. Braccia, E. Fontana, M. S. Rudolph, P. Bermejo, A. Ijaz, S. Thanasilp, E. R. Anschuetz, and Z. Holmes, Nat. Commun. **16**, 7907 (2025).
[107] P. Bermejo, P. Braccia, M. S. Rudolph, Z. Holmes, L. Cincio, and M. Cerezo, arXiv:2408.12739.
[108] K. O'Shea and R. Nash, arXiv:1511.08458.
[109] W. J. Devenport, M. C. Rife, S. I. Liapis, and G. J. Follin, J. Fluid Mech. **312**, 67 (1996).
[110] A. Ramakrishnan, in *Proceedings of the Conference on Clifford Algebra, its Generalization and Applications* (Matscience, Madras, 1971), pp. 87–96.
[111] C. Sünderhauf, E. Campbell, and J. Camps, Quantum **8**, 1226 (2024).
[112] D. Camps, L. Lin, R. V. Beeumen, and C. Yang, arXiv:2203.10236.
[113] L. Budinski, O. Niemimäki, R. Zamora-Zamora, and V. Lahtinen, Quantum Sci. Technol. **8**, 045031 (2023).
[114] V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes, Trans. Comp.-Aided Des. Integ. Cir. Sys. **22**, 710 (2006).
[115] S. Herbert, J. Sorci, and Y. Tang, Phys. Rev. A **110**, 012437 (2024).
[116] A. Wills and S. Strelchuk, arXiv:2305.04069.
[117] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2010).
[118] E. Bäumer, D. Sutter, and S. Woerner, arXiv:2504.20832.
[119] M. Oszmaniec, F. B. Maciejewski, and Z. Puchała, Phys. Rev. A **100**, 012351 (2019).
[120] C. Umeano, S. Scali, and O. Kyriienko, Phys. Rev. A **112**, 052422 (2025).
[121] S. Beddar-Wiesing, A. Moallemy-Oureh, M. Kempkes, and J. M. Thomas, arXiv:2507.03392.
[122] R. Lguensat, M. Sun, R. Fablet, P. Tandeo, E. Mason, and G. Chen, in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium* (IEEE, Valencia, Spain, 2018), pp. 1764–1767.
[123] N. Gourianov, M. Lubasch, S. Dolgov, Q. Y. van den Berg, H. Babaee, P. Givi, M. Kiffner, and D. Jaksch, Nat. Comput. Sci. **2**, 30 (2022).
[124] S. Lloyd, S. Garnerone, and P. Zanardi, Nat. Commun. **7**, 10138 (2016).
[125] C. Gyurik, C. Cade, and V. Dunjko, Quantum **6**, 855 (2022).
[126] D. W. Berry, Y. Su, C. Gyurik, R. King, J. Basso, A. D. T. Barba, A. Rajput, N. Wiebe, V. Dunjko, and R. Babbush, PRX Quantum **5**, 010319 (2024).
[127] S. Scali, C. Umeano, and O. Kyriienko, Phys. Rev. A **110**, 042616 (2024).
[128] S. Scali, C. Umeano, and O. Kyriienko, APL Quantum **1**, 036106 (2024).
[129] Quantum vortex detection, https://github.com/cw962/quantum_vortex_detection.