nature portfolio

# Peer Review File

# Simulating fluid vortex interactions on a superconducting quantum processor

Corresponding Author: Professor Chao Song

**This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.**

Version 0:

Reviewer comments:

Reviewer #1

(Remarks to the Author)
The vortex-based Navier- Stokes equations are being simulated on quantum systems in this research. Congrats to the authors for the excellent, innovative work and very good presentation in the paper! I strongly recommend this paper for publication due to the importance of the topic as well as innovative work. Only a few improvements are suggested for the final version on this novel paper:
- Authors should provide more details about the practical implementation of the vortex-based NSE. This form is very well known in the CFD community, but it lacks practical application.
- The core element of the quantum algorithm is the operator F. Some additional explanations on the complexity of the operator as well as the practical implementation need to be provided.
- Complexity of the overall algorithm should be provided.
- The reference [39] has a wrong author reference. The correct reference is Budinski Lj. instead of B. Ljubomir in reference [39] and L. Budinski in [42].

Reviewer #2

(Remarks to the Author)
The manuscript describes an interesting investigation into realizing an implementation on a quantum processor of of a simulated vortex-interaction. As discussed in this article, vortex interactions play an important role in many fluid dynamics problems of scientific as well as practical interest.

The method proposed is certainly interesting, and the Supplementary part provides a good amount of detail.

The re-formulation of the governing equations for vortex movement into complex numbers and its normalization is interesting, but clearly is rather convoluted. It requires either a significant level of approximation (and linearization) to facilitate implementation on a quantum processor and also a significant amount of pre- and post-processing, i.e. before and after the dynamics are modelled on the quantum processor.

A key issue is that it is not clear where the 'quantum benefit' is, i.e. relative to a classical-computer implementation, is there a theoretical computational complexity benefit for the quantum algorithm?
This important aspect needs to be clearly discussed in the main article. For example, the current method stores solutions at different time-steps while a classical method wouldn't need this. This negates (part) of a potential storage benefit because of the additional qubits requires for the multiple time levels.

With the substantial level of approximations, as well as pre- and post-processing required does the demonstrated approach offer longer-term benefit. It seems that the key contribution is a proof-of-concept showing how a specific (model) vortex problem can be realized with current quantum hardware. The 'potential to tackle longstanding challenges in fluid dynamics' (as stated in the abstract) seems to overstate the impact of this work.

A further key issue is that in the main article it should be much clearer how much the modelled problem is different from a 'full' vortex-particle based implementation of the incompressible-flow Navier-Stokes equations.

There are further important questions to address, as outlined below:

[1] by opting to use an idealized fluid where the vortices are pure potential vortices, the modelled flow is effectively a potential-flow problem (i.e. irrotational and inviscid). So quite far removed from the fluid dynamics governed by the incompressible-flow NS equations. For example, the effect of co-rotating vortices merging as occurring in many practical, turbulent flows cannot be modelled.
The implications of the use of the idealized system should be explained much more clearly in the main article.

[2] for potential-flow modeling, approaches based on hydrodynamic Schrodinger and Madelung transformation have been investigated quite extensively (as reviewed in the Introduction section of the article). So, the relative merits of the current approach as compared to the mentioned alternative approaches needs to be discussed (more) as well.

[3] As detailed in the Supplementary material, the dynamics of the discrete-vortex system is actually modeled using a DMD-type reduced-order model rather than the by discretization of the set of ODEs resulting from the re-formulated Biot-Savart type equations. It seems that this reduced modelling relies on training for a very specific problem involving a set of interacting discrete vortices.

Question: once an evolution operator is trained, can it be applied to a (somewhat) different test case than the one used for training? So, that there is an element of generalization?

This leads to the following further question: to mimic the vortex-interaction problem, it is possible to train a DMD-type problem based on a viscous simulation (e.g. using a vortex particle methods) of the vortex interaction problem (for example using POD or SVD) and then implement this DMD on a quantum processor. This avoids the need to first approximate/linearize the dynamical system before training the reduced-order model. In the revised article, this needs to be considered as well, for example in terms of how these different approaches compare in terms of generalization to different systems.

[4] Measurement and data extraction need to be discussed further. With solution at multiple-time step encoded in the quantum register, the likelihood of measuring data related to an intended time step reduced rapidly with increasing number of time steps resolved.
The 'single quantum execution' mentioned in Introduction on page 1 needs further detailing, e.g. mention that multiple realization are needed for extraction of flow data and that the number of realizations for a given accuracy will increased with $n_t$

[5] Viscous vortex particle system (page 5): required changes to learning the evolution operator, i.e. with regards using varying vortex particle strengths (circulation) needs to be explained more clearly.

Version 1:

Reviewer comments:

Reviewer #1

(Remarks to the Author)
All the comments have been appropriately addressed. No further comments.

Reviewer #2

(Remarks to the Author)
The revised manuscript represents a significant improvement over the original version, and now places the presented in a much clearer context.

**List of major changes** (marked in red in the Main Text and the Supplementary Information):

1. We expand the discussion to emphasize that the normalized wavefunction reformulation of vortex dynamics is physically grounded in conservation laws, such as Kelvin's and Helmholtz's theorems.

2. We add a thorough analysis of the algorithm's computational complexity, explicitly discussing the potential quantum benefit in circuit evolution and storage efficiency.

3. We clarify the relationship between our approach and "full" Navier–Stokes solutions, and expand the discussion on handling viscous flows.

4. We discuss how alternative quantum approaches, such as Schrödinger/Madelung-based methods, differ in flow modeling.

5. We discuss linearization and pre/post-processing as necessary compromises for current quantum hardware.

6. We incorporate a test of the model's performance on perturbed initial conditions, demonstrating its generalization to different systems.

7. We provide expanded details on the complexity, training, and hardware implementation of the core quantum evolution operator $F_k$, including a summary of experimental gate counts.

8. We clarify the data extraction process from the quantum state and discuss potential strategies to address the sampling challenge for specific time steps.

9. We moderate the language regarding the work's impact, framing it as a concrete pathway for quantum fluid simulation.

10. We add discussion and references on the established practical applications of classical Lagrangian vortex methods in computational fluid dynamics.

11. We correct author names in references and add new tables/figures to support the added analyses.

**Reply to reviewer #1** for the manuscript (ID: NCOMMS-25-39868) titled "Simulating fluid vortex interactions on a superconducting quantum processor" by Ziteng Wang *et al.*

**General comment:** The vortex-based Navier–Stokes equations are being simulated on quantum systems in this research. Congrats to the authors for the excellent, innovative work and very good presentation in the paper! I strongly recommend this paper for publication due to the importance of the topic as well as innovative work. Only a few improvements are suggested for the final version on this novel paper.

**Reply:** We sincerely appreciate the reviewer's encouraging evaluation and the constructive comments that help to improve the manuscript. All the comments have been addressed point-by-point, and the corresponding changes in the revised paper are marked in **red.**

1. Authors should provide more details about the practical implementation of the vortex-based NSE. This form is very well known in the CFD community, but it lacks practical application.

**Reply:** Thank you for this insightful comment regarding the practical application of the vortex-based Navier–Stokes equations (NES). We agree that expanding on this aspect will strengthen the manuscript.

Indeed, as a conventional vortex method, the practical application of the vortex-based NSE has traditionally been limited. However, advances in graphics processing unit (GPU) computing have enabled the community to develop the LVM into a robust and accurate methodology for practical computations. To extend the method to viscous flows, various numerical strategies for modeling viscosity have been proposed [1], including Chorin's stochastic random walk diffusion method [2], Degond and Mas-Gallic's deterministic particle strength exchange (PSE) schemes with rigorous consistency and convergence [3], and Ogami and Akamatsu's diffusion velocity formulations tailored for discrete vortex models [4]. For boundary conditions, desingularized vortex sheet formulations [5] and high-resolution cylinder benchmarks [6] established accurate wall treatments, while the combination of particle redistribution for general geometries with a modified PSE scheme for diffusion enabled nontrivial geometries without abandoning the Lagrangian backbone [7].

For practical applications, the LVM has been employed in several representative classes[8], including swimmer hydrodynamics [9–11], aircraft and rotorcraft wake prediction [12, 13], free jets [14–16], and wind energy turbines [17, 18]. Further advances in applications may be unlocked through quantum algorithms, offering a promising pathway to overcome the quadratic $O(N_p^2)$

complexity of Biot–Savart evaluations.

We provide the details in lines 19–33 of the Supplementary Information and summarize them in the Main Text in lines 65–67.

2. The core element of the quantum algorithm is the operator $F$. Some additional explanations on the complexity of the operator as well as the practical implementation need to be provided.

**Reply:** We appreciate the reviewer's comment regarding the complexity and practical implementation of the operator $\boldsymbol{F}_k$, where $k$ is the temporal qubit index.

For the complexity of the operator $\boldsymbol{F}_k$, it is heuristic and depends on empirical tuning as the system size increases, much like the widely used quantum approximate optimization algorithm (QAOA) [19, 20]. In our work, for each synthesized instance of $\boldsymbol{F}_k$, the number of CZ gates and single-qubit rotations is determined by the target synthesis accuracy and the chosen depth threshold [21], showing no significant dependence on $k$. We limit the maximum numbers of CZ gates and single-qubit rotations in our example to fewer than 15 and 25, respectively, to ensure feasibility on the quantum hardware while maintaining good agreement with the reference results. In Table R1, we summarize the gate counts used in our experiments for each synthesized instance of $\boldsymbol{F}_k$.

Table R1: Gate counts of controlled $\boldsymbol{F}_k$ circuits synthesized by CPFlow for each $k$. $U$ denotes single-qubit $U(\theta, \phi, \lambda)$ and CZ denotes the two-qubit entangling gate.

| $k$ | $2^{k-1}$ steps | $U$ gates | CZ gates | Total gates |
|---|---|---|---|---|
| 1 | 1 | 19 | 8 | 27 |
| 2 | 2 | 13 | 5 | 18 |
| 3 | 4 | 11 | 4 | 15 |
| 4 | 8 | 16 | 7 | 23 |
| 5 | 16 | 21 | 11 | 32 |
| 6 | 32 | 11 | 4 | 15 |
| Average | | 15.17 | 6.50 | 21.67 |

For the practical implementation, the Supplementary Information includes additional details on how we obtain an effective Hamiltonian and how we compile $\boldsymbol{F}_k$ into a controlled operation suitable for hardware. Specifically, we learn an effective Hamiltonian while enforcing Hermiticity by parameterizing

$$\boldsymbol{H}_{\text{eff}}(\boldsymbol{\theta}) = \boldsymbol{S}(\boldsymbol{\theta}_{\text{R}}) + \mathrm{i}\, \boldsymbol{A}(\boldsymbol{\theta}_{\text{I}}), \tag{R1}$$

where $\boldsymbol{S}(\boldsymbol{\theta}_{\mathrm{R}}) \in \mathbb{R}^{N_p \times N_p}$ is symmetric and $\boldsymbol{A}(\boldsymbol{\theta}_{\mathrm{I}}) \in \mathbb{R}^{N_p \times N_p}$ is antisymmetric, and $\boldsymbol{\theta}_{\mathrm{R}}$ and $\boldsymbol{\theta}_{\mathrm{I}}$ constitute the overall parameter vector $\boldsymbol{\theta}$. The training procedure is posed as

$$\begin{aligned} \boldsymbol{\theta}^{\star} &= \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \\ \text{with} \quad \mathcal{L}(\boldsymbol{\theta}) &= \frac{1}{N_{\text{train}}} \sum_{i=0}^{N_{\text{train}}-1} \| \, |\psi(t_i + \Delta t_{\text{train}})\rangle - U(\boldsymbol{\theta}; \Delta t_{\text{train}}) \, |\psi(t_i)\rangle \|_2^2 \, , \end{aligned} \tag{R2}$$

and $U(\boldsymbol{\theta}; \Delta t_{\text{train}}) = e^{-\mathrm{i}\, \boldsymbol{H}_{\text{eff}}(\boldsymbol{\theta})\, \Delta t_{\text{train}}}$. Once the effective Hamiltonian is trained, we compute $\boldsymbol{F}_k$ for each $k$ with $\boldsymbol{F}_k(\boldsymbol{\theta}) = e^{-\mathrm{i}\, \boldsymbol{H}_{\text{eff}}(\boldsymbol{\theta})(2^k-1)\Delta t}$, where $\Delta t$ represents the unit time step.

For circuit synthesis we use CPFlow method [21], which relaxes the gate-layout search to a continuous optimization with parametrized controlled-phase placeholders. After optimizing several random seeds, phases near 0 or $\pi$ are snapped to $I$ or CZ, and the remaining phases are realized by a constant-depth two-CZ template with single-qubit rotations, yielding controlled-$\boldsymbol{F}_k$ circuits in the device's native gate set.

In our setting, the target unitary is the controlled version of $\boldsymbol{F}_k$ with the temporal qubits as controls, and the final implementation is selected by jointly considering the unitary approximation error and the total CZ count. This procedure yields controlled-$\boldsymbol{F}_k$ circuits directly, without first compiling an uncontrolled $\boldsymbol{F}_k$ and then adding control.

We have incorporated the above discussion in the Main Text in lines 450–457 and Supplementary Information in lines 190–229.

3. Complexity of the overall algorithm should be provided.

**Reply:** We thank the reviewer for the suggestion. The algorithmic complexity can be divided into two parts, the online circuit execution involving real-time quantum operations and the offline fitting stage focusing on classical optimization and parameter tuning.

During the online circuit execution, the complexity scales as $O(\bar{g}_{\mathrm{CZ}} \log N_t)$ in two-qubit gate depth. We take two-qubit depth as the primary cost metric because entangling operations dominate runtime and error on current platforms. Here, $N_t$ the number of time steps, with $n_t = \lceil \log N_t \rceil$ control qubits. The complexity $O(\bar{g}_{\mathrm{CZ}} \log N_t)$ arises from the sequence of $n_t$ controlled-$F_k$ blocks, where the $k$-th block is compiled with $g_{\mathrm{CZ}}^{(k)}$ CZ gates. We use the average $\bar{g}_{\mathrm{CZ}} = \frac{1}{n_t} \sum_{k=1}^{n_t} g_{\mathrm{CZ}}^{(k)}$ because our preliminary observations indicate that the per-block counts exhibit no significant dependence on $k$, as noted in the previous comment. The time register is prepared by applying

$n_t$ Hadamard gates in parallel, which contributes $O(\log N_t)$ single-qubit gates but only constant depth and is not rate-limiting. Besides, the spatial register that encodes the positions of $N_p$ vortex particles is initialized at the beginning of the computation, whose cost is independent of $N_t$.

The state preparation is realized by a data loading circuit whose depth scales at most linearly with $N_p$, in line with general upper bounds for preparing quantum states from classical data [22, 23]. For relevant families of initial configurations, more structured loading schemes or hardware support such as quantum random access memory (QRAM) can reduce this dependence on $N_p$ by exploiting correlations and low dimensional parametrizations of vortex states [24–26]. Measurements are only required when estimating observables. If $M$ observables $\{O^{(\ell)}\}_{\ell=1}^{M}$ are estimated by simple sampling with a target mean-squared error $\varepsilon$ for each expectation value then the shot complexity scales as $O(M\varepsilon^{-2})$.

During the offline stage, we fit each evolution module $F$ with a heuristic procedure whose complexity is $O(N_{\text{train}} I L D \varepsilon^{-2})$. Here, $N_{\text{train}}$ state pairs constitute the training set for each $F$ block. The quantities $I$, $L$, and $D$ are empirically determined, and they denote the number of iterations, the number of trainable parameters, and the depth of each block, respectively. With the parameter shift rule, one iteration requires $O(N_{\text{train}} L)$ forward circuit evaluations to obtain the full gradient vector. Driving the single circuit estimation error to $\varepsilon$ contributes a sampling factor of $O(\varepsilon^{-2})$. Each evaluation has depth $D$, dominated by loading the training data and executing the parameterized subcircuit.

We have incorporated the above discussion in the Main Text in lines 383–393 and Supplementary Information in lines 285–326.

4. The reference [39] has a wrong author reference. The correct reference is Budinski Lj. instead of B. Ljubomir in reference [39] and L. Budinski in [42].

**Reply:** We have corrected the author name in References [39] and [42]. Both entries now correctly appear as "Budinski, Lj." in the reference list and the corresponding BibTEX records, as shown below:

[39] Lj. Budinski, Quantum algorithm for the Navier–Stokes equations by using the streamfunction–vorticity formulation and the lattice Boltzmann method, *Int. J. Quantum Inf.* **20**, 2150039 (2022).

[42] Lj. Budinski, Quantum algorithm for the advection–diffusion equation simulated with the

lattice Boltzmann method, *Quantum Inf. Process.* **20**, 57 (2021).

---

[1] G. H. Cottet and P. Koumoutsakos. *Vortex Methods: Theory and Practice*. Cambridge University Press, 2000.

[2] A. J. Chorin. Numerical study of slightly viscous flow. *J. Fluid Mech.*, 57:785–796, 1973.

[3] P. Degond and S. Mas-Gallic. The weighted particle method for convection-diffusion equations. i. the case of an isotropic viscosity. *Math. Comput.*, 53:485–507, 1989.

[4] Y. Ogami and T. Akamatsu. Viscous flow simulation using the discrete vortex model—the diffusion velocity method. *Comput. Fluids*, 19:433–441, 1991.

[5] R. Krasny. Desingularization of periodic vortex sheet roll-up. *J. Comput. Phys.*, 65:292–313, 1986.

[6] P. Koumoutsakos and A. Leonard. High-resolution simulations of the flow around an impulsively started cylinder using vortex methods. *J. Fluid Mech.*, 296:1–38, 1995.

[7] P. Ploumhans and G. Winckelmans. Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry. *J. Comput. Phys.*, 165:354–406, 2000.

[8] C. Mimeau and I. Mortazavi. A review of vortex methods and their applications: From creation to recent advances. *Fluids*, 6:68, 2021.

[9] W. M. Van Rees, M. Gazzola, and P. Koumoutsakos. Optimal shapes for anguilliform swimmers at intermediate reynolds numbers. *J. Fluid Mech.*, 722:R3, 2013.

[10] M. Gazzola, B. Hejazialhosseini, and P. Koumoutsakos. Reinforcement learning and wavelet adapted vortex methods for simulations of self-propelled swimmers. *SIAM J. Sci. Comput.*, 36:B622–B639, 2014.

[11] C. Bernier, M. Gazzola, R. Ronsse, and P. Chatelain. Simulations of propelling and energy harvesting articulated bodies via vortex particle-mesh methods. *J. Comput. Phys.*, 392:34–55, 2019.

[12] R. Cocle, G. Winckelmans, and G. Daeninck. Combining the vortex-in-cell and parallel fast multipole methods for efficient domain decomposition simulations. *J. Comput. Phys.*, 227:9091–9120, 2008.

[13] D.-G. Caprace, G. Winckelmans, and P. Chatelain. Assessment of the vortex particle-mesh method for efficient les of hovering rotors and their wakes. In *AIAA SciTech 2021*, page 0738, 2021.

[14] S. Samarbakhsh and N. Kornev. Simulation of a free circular jet using the vortex particle intensified les (v$\pi$les). *Int. J. Heat Fluid Flow*, 80:108489, 2019.

[15] G. S. Winckelmans and A. Leonard. Contributions to vortex particle methods for the computation of three-dimensional incompressible unsteady flows. *J. Comput. Phys.*, 109:247–273, 1993.

[16] E. J. Alvarez and A. Ning. Stable vortex particle method formulation for meshless large-eddy simulation. *AIAA J.*, 62:637–656, 2024.

[17] P. Chatelain, M. Duponcheel, D.-G. Caprace, Y. Marichal, and G. Winckelmans. Vortex particle-mesh simulations of vertical axis wind turbine flows: from the airfoil performance to the very far wake. *Wind Energy Sci.*, 2:317–328, 2017.

[18] P. Balty, D.-G. Caprace, J. Waucquez, M. Coquelet, and P. Chatelain. Multiphysics simulations of the dynamic and wakes of a floating vertical axis wind turbine. In *J. Phys. Conf. Ser.*, volume 1618, page 062053. IOP Publishing, 2020.

[19] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Phys. Rev. X*, 10:021067, 2020.

[20] Z. He, R. Shaydulin, D. Herman, C. Li, R. Raymond, S. H. Sureshbabu, and M. Pistoia. Parameter setting heuristics make the quantum approximate optimization algorithm suitable for the early fault-tolerant era. In *Proc. 43rd IEEE/ACM Int. Conf. Comput.-Aided Des. (ICCAD)*, pages 1–7, 2024.

[21] N. A. Nemkov, E. O. Kiktenko, I. A. Luchnikov, and A. K. Fedorov. Efficient variational synthesis of quantum circuits with coherent multi-start optimization. *Quantum*, 7:993, 2023.

[22] L. Grover and T. Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv preprint quant-ph/0208112*, 2002.

[23] M. Plesch and Č. Brukner. Quantum-state preparation with universal gate decompositions. *Phys. Rev. A*, 83:032302, 2011.

[24] V. Giovannetti, S. Lloyd, and L. Maccone. Quantum random access memory. *Phys. Rev. Lett.*, 100:160501, 2008.

[25] C. T. Hann, C. L. Zou, Y. X. Zhang, Y. W. Chu, R. J. Schoelkopf, S. M. Girvin, and L. Jiang. Hardware-efficient quantum random access memory with hybrid quantum acoustic systems. *Phys. Rev. Lett.*, 123:250501, 2019.

[26] N. Jiang, Y. F. Pu, W. Chang, C. Li, S. Zhang, and L. M. Duan. Experimental realization of 105-qubit random access quantum memory. *npj Quantum Inf.*, 5:28, 2019.

**Reply to reviewer #2** for the manuscript (ID: NCOMMS-25-39868) titled "Simulating fluid vortex interactions on a superconducting quantum processor" by Ziteng Wang *et al.*

**General comment:** The manuscript describes an interesting investigation into realizing an implementation on a quantum processor of of a simulated vortex-interaction. As discussed in this article, vortex interactions play an important role in many fluid dynamics problems of scientific as well as practical interest. The method proposed is certainly interesting, and the Supplementary part provides a good amount of detail.

**Reply:** We thank the reviewer for the accurate summary of our work. We also appreciate the reviewer for the constructive comments, which have guided us to improve the manuscript significantly. All the comments have been addressed point-by-point, and the corresponding changes in the revised paper are marked in **red.**

1. The re-formulation of the governing equations for vortex movement into complex numbers and its normalization is interesting, but clearly is rather convoluted. It requires either a significant level of approximation (and linearization) to facilitate implementation on a quantum processor and also a significant amount of pre- and post-processing, i.e. before and after the dynamics are modelled on the quantum processor.

**Reply:** We thank the reviewer for this thoughtful comment. We introduce a complex representation of vortex motion with explicit normalization to embed discrete vortex dynamics into a unitary evolution in Hilbert space, yielding a valid quantum state and an effective Hamiltonian for interacting vortices implementable on a quantum processor. The reformulation of vortex dynamics into a normalized wavefunction representation has been explored classically, such as in vortex particle method [1], vortex filament methods [2], and geometric models [3, 4]. Its extension to quantum computing is motivated by the intrinsic conservation laws that govern vorticity evolution in inviscid or high-Reynolds-number flows, including Kelvin's theorem [5], Helmholtz's theorem [6], Ertel's theorem [7], and helicity conservation [8, 9]. These conservation laws and the unitary evolution of quantum systems share underlying mathematical structures that suggest a potential connection. By embedding vortex methods into a wavefunction framework, the resulting quantum algorithm inherits a structure that is compatible with unitary evolution, thereby offering a promising pathway to designing quantum simulations that preserve the conservation laws of fluid systems.

Indeed, to implement nonlinear fluid dynamics on current quantum hardware, some level of

linearization and approximation is generally necessary to align with the operational principles of quantum processors. Classical pre-processing for mapping vortex particles from configuration space to quantum state space, as well as post-processing to extract task relevant observables through measurements, are also typically required.

In our implementation, the approximations and the linearization preserve the leading interaction structure over the horizons of interest and make the expensive interactions quantifiable, learnable, and evolvable within a linear representation. The effective model is trained on short windows and subsequently used to propagate beyond the training range to assess generalization on representative interactions, which constitutes a pragmatic compromise that enables experiments on present hardware while maintaining fidelity within the measurement resolution. Therefore, we view this linearized realization as a hardware compatible instantiation of the QVM, and note that a growing body of work on quantum algorithms for nonlinear problems indicates promising avenues for incorporating nonlinear effects in future extensions of our method [10–14].

From a broader perspective, pre- and post-processing are common components in numerical schemes based on quantum processors [15–18]. In our case, the normalization operation as pre-processing is mathematically equivalent to the classical vortex method. More importantly, the newly added numerical experiments provide preliminary evidence that normalization combined with unitary evolution may act as a form of implicit regularization and, compared with direct linearization in the original variables, can enhance robustness and generalization with respect to initial condition perturbations and noise, as further discussed in our reply to Reviewer 2, Comment 8. At the same time, we would like to emphasize that our work presents a general framework for encoding and simulating vortex dynamics on quantum computers, where both pre- and post-processing may vary depending on the specific scenario. For example, instead of extracting the full vortex state, one may want to compare the trajectories of two vortex movements, which can be realized with SWAP test technique, thereby reducing the need for complex post-processing.

In response, we have incorporated the above discussion in the Main Text in lines 67–72 and Supplementary Information in lines 88–95 and 113–116.

2. A key issue is that it is not clear where the ‘quantum benefit’ is, i.e. relative to a classical-computer implementation, is there a theoretical computational complexity benefit for the quantum algorithm? This important aspect needs to be clearly discussed in the main article. For example, the current method stores solutions at different time-steps while a classical method wouldn’t need

this. This negates (part) of a potential storage benefit because of the additional qubits requires for the multiple time levels.

**Reply:** We value the reviewer's comment on the theoretical computational complexity benefits of quantum algorithms compared to classical methods, a point of great interest within the research community. We discuss the theoretical computational complexity benefit of the QVM from two main aspects, including the complexity of circuit evolution and the storage benefit. The complexity of circuit evolution primarily depends on the intrinsic complexity of the evolution module $\boldsymbol{F}_k$ and its count $O(\log N_t)$, where $N_t$ denotes the number of discrete time steps. For different temporal qubit index $k$, the complexity of $\boldsymbol{F}_k$ shows no significant difference; as such, we set the complexity of $\boldsymbol{F}_k$ as $O(\bar{g}_{\mathrm{CZ}})$. We conclude the overall complexity of the quantum circuit execution is $O(\bar{g}_{\mathrm{CZ}} \log N_t)$, demonstrating a quantum benefit for predicting long-time flow evolution as the algorithmic complexity grows logarithmically with $N_t$. Regarding the potential storage benefit, our method processes all time steps in a quantum superposition within a single workflow, requiring $\log N_p + \log N_t$ qubits compared with the classical $N_p N_t$ units of storage for the same functionality. We now proceed to discuss in detail.

During the online circuit execution, as illustrated by the complete circuit executed on our quantum processor in Fig. R1, we employ two-qubit gate depth as the primary cost metric since entangling operations constitute the dominant source of runtime and error in current quantum platforms. The complexity scales as $O(\bar{g}_{\mathrm{CZ}} \log N_t)$ in two-qubit gate depth with $n_t = \lceil \log N_t \rceil$ control qubits, revealing a theoretical quantum benefit for simulating long-time evolution of complex vortex systems. The complexity of circuit evolution mainly arises from the sequence of $n_t$ controlled-$\boldsymbol{F}_k$ blocks. Notably, each evolution module $\boldsymbol{F}_k$ is obtained through a heuristic fitting procedure and collectively forms a linear effective Hamiltonian that steers the evolution of the quantum-encoded vortex state. The empirical complexity $O(\bar{g}_{\mathrm{CZ}})$ depends on both the vortex count $N_p$ and parameter settings, and can be controlled by adjusting optimization parameters while balancing fitting accuracy to accommodate execution on current noisy intermediate-scale quantum devices. We use the average $\bar{g}_{\mathrm{CZ}} = \frac{1}{n_t} \sum_{k=1}^{n_t} g_{\mathrm{CZ}}^{(k)}$ because our preliminary observations indicate that the per-block counts exhibit no significant dependence on $k$, as noted in the previous comment. The time register is prepared by applying $n_t$ Hadamard gates in parallel, which contributes $O(\log N_t)$ single-qubit gates but only constant depth and is not rate-limiting. Besides, the spatial register that encodes the positions of $N_p$ vortex particles is initialized at the beginning of the computation, whose cost is independent of $N_t$.

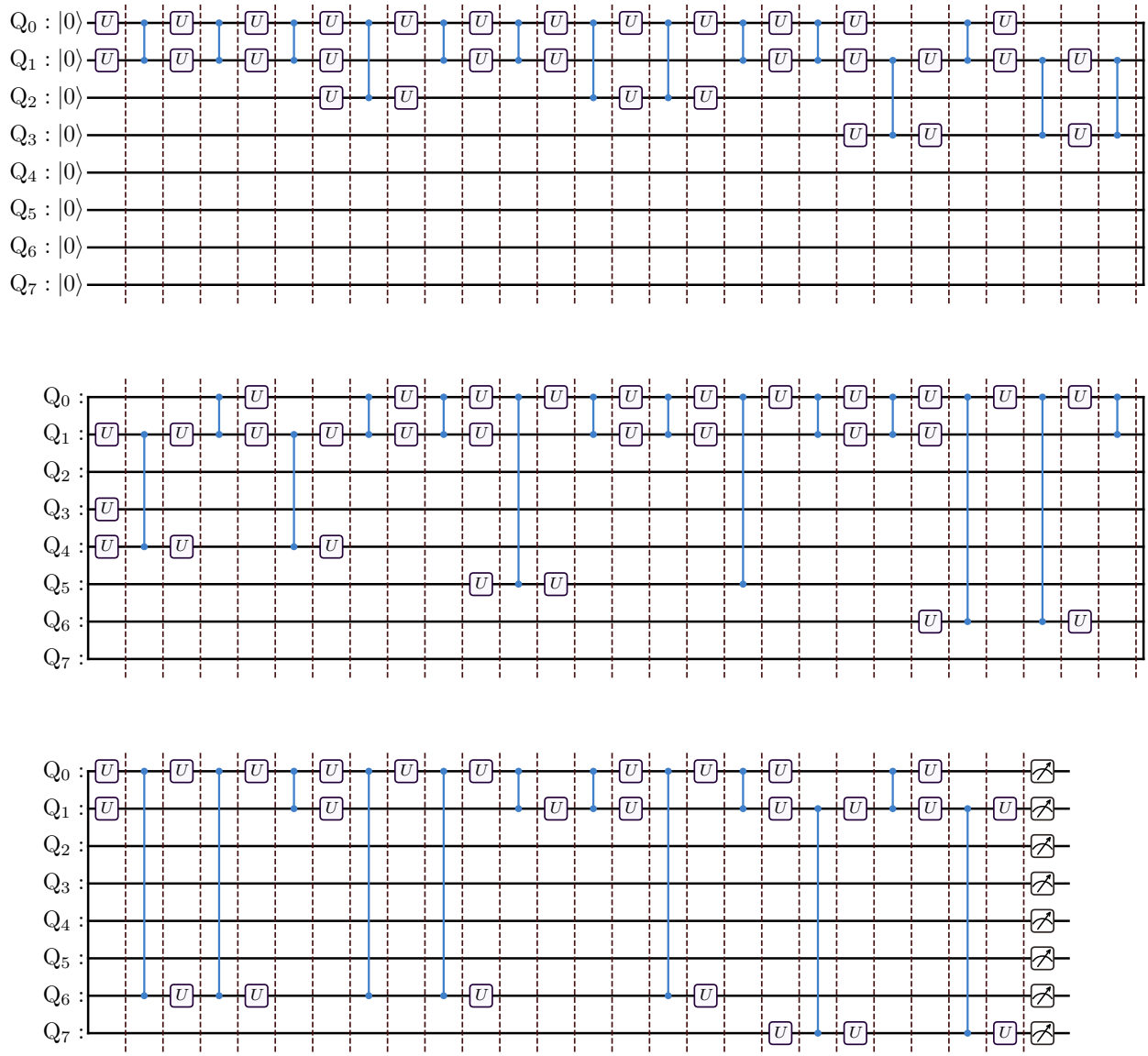


Fig. R1: Full circuit used in the hardware experiment, including initial state preparation, parallel Hadamard gates (implemented in the $\{U, \mathrm{CZ}\}$ basis as $U(\theta, \phi, \lambda)$), and the sequence of controlled $F_k$ blocks, followed by measurements. Here $U$ denotes single-qubit $U(\theta, \phi, \lambda)$, and CZ denotes two-qubit entangling gate.

The state preparation is realized by a data loading circuit whose depth scales at most linearly with $N_p$, in line with general upper bounds for preparing quantum states from classical data [19, 20]. For relevant families of initial configurations, more structured loading schemes or hardware support such as quantum random access memory can reduce this dependence on $N_p$ by exploiting correlations and low dimensional parametrizations of vortex states [21–23].

Measurements are only required when estimating observables. If $M$ observables $\{O^{(\ell)}\}_{\ell=1}^{M}$ are estimated by simple sampling with a target mean-squared error $\varepsilon$ for each expectation value, then the shot complexity scales as $O(M\varepsilon^{-2})$.

During the offline stage, we fit each evolution module $\boldsymbol{F}_k$ with a heuristic procedure whose complexity is $O(N_{\text{train}}ILD\varepsilon^{-2})$. Here, $N_{\text{train}}$ state pairs constitute the training set for each $\boldsymbol{F}_k$ block. The quantities $I$, $L$, and $D$ are empirically determined, and they denote the number of iterations, the number of trainable parameters, and the depth of each block, respectively. With the parameter shift rule, one iteration requires $O(N_{\text{train}}L)$ forward circuit evaluations to obtain the full gradient vector. Driving the single circuit estimation error to $\varepsilon$ contributes a sampling factor of $O(\varepsilon^{-2})$. Each evaluation has depth $D$, dominated by loading the training data and executing the parameterized subcircuit. Such optimization phase, once completed, allows the optimized circuit to be reused for multiple simulations under different initial conditions.

For the data storage efficiency, we break down the analysis into two specific scenarios including problems requiring post-processing across multiple time steps and tasks involving only sequential time stepping.

For problems involving post-processing across many time steps, our method offers a more memory-efficient solution. For a system with $N_p$ quantities over $N_t$ time steps, our method evaluates aggregated quantities and performs a time-domain search on a superposition over time in a single workflow, requiring $\log N_p + \log N_t$ qubits. In contrast, a classical method either stores $N_t$ frames or recomputes $N_t$ times, resulting in $N_p N_t$ units of storage for the same functionality.

For tasks involving the sequential advancement of dynamics and evaluation of each time step, classical methods typically require storage for only the current time step, with resource demand scaling linearly with the system size $N_p$. In contrast, our quantum methods leverage superposition and entanglement to efficiently represent and process spatial information. Although the logarithmic resource overhead introduced by the time register partially offsets this spatial benefit, the quantum method still maintains a benefit over the linear growth of classical methods as the problem size increases.

In the revised manuscript, we provide an explicit discussion of the quantum benefit in the Main Text in lines 383–394 and Supplementary Information in lines 285–338, incorporating the other reviewer's suggestion to include the method's complexity.

3. With the substantial level of approximations, as well as pre- and post-processing required

does the demonstrated approach offer longer-term benefit. It seems that the key contribution is a proof-of-concept showing how a specific (model) vortex problem can be realized with current quantum hardware. The 'potential to tackle longstanding challenges in fluid dynamics' (as stated in the abstract) seems to overstate the impact of this work.

**Reply:** We thank the reviewer for the responsible evaluation of our work. At the current stage of quantum applications, some level of approximations, along with pre- and post-processing, are required. These common procedures are expected to be mitigated with future advancements in hardware and algorithms. The reformulation of vortex dynamics into a wavefunction representation for quantum implementation is motivated by the intrinsic conservation laws governing vorticity evolution in fluid mechanics. This approach leverages mathematical parallels between these conservation properties and unitary quantum evolution, enabling the development of quantum algorithms that inherently preserve the physical structure of fluid systems.

Through concrete analysis and discussion of our implementation, we identify a theoretical computational complexity benefit of the QVM in two main aspects. The overall quantum circuit execution complexity demonstrates quantum benefit for long-time flow evolution prediction, while the method's ability to process all time steps in quantum superposition within a single workflow offers potential storage benefits.

We agree with the reviewer that the statement regarding "potential to tackle longstanding challenges in fluid dynamics" in the abstract may create the misleading impression that solutions to these longstanding problems are immediately within reach. Accordingly, we reword this in the Abstract to be more specific: "In this work, we establish a framework that reformulates vortex dynamics into a normalized wavefunction representation compatible with quantum system unitary evolution, combined with the designed spatiotemporal encoding scheme, providing a concrete pathway toward leveraging quantum resources in fluid systems."

4. A further key issue is that in the main article it should be much clearer how much the modelled problem is different from a 'full' vortex-particle based implementation of the incompressible-flow Navier–Stokes equations.

**Reply:** We thank the reviewer for this helpful comment. Our approach can be structured in two main stages. First, we establish that representing the vortex state as a quantum wavefunction is mathematically equivalent to conventional vortex-particle methods. Second, we design an effective linear Hamiltonian to guide the system's evolution, which provides a controlled approximation to

the incompressible Navier–Stokes equations.

Although this approximation involves some simplification, it preserves the essential physics of vortex interactions. The model is trained using short-time data and is subsequently applied to propagate the system beyond the training interval, enabling an evaluation of its generalization to representative fluid interactions. As illustrated by the two-vortex co-rotating system in the main text, the results exhibit close agreement with the direct numerical simulation of viscous flows.

We agree that an explicit clarification is necessary to distinguish our method from the 'full' vortex-particle based implementation of the incompressible-flow Navier–Stokes equations, which we have incorporated in the Main Text in lines 145–150 and in the Supplementary Information in lines 96–99 of the revised manuscript.

5. By opting to use an idealized fluid where the vortices are pure potential vortices, the modelled flow is effectively a potential-flow problem (i.e. irrotational and inviscid). So quite far removed from the fluid dynamics governed by the incompressible-flow NS equations. For example, the effect of co-rotating vortices merging as occurring in many practical, turbulent flows cannot be modelled. The implications of the use of the idealized system should be explained much more clearly in the main article.

**Reply:** We appreciate the reviewer's comment regarding the limitation of the idealized system. Our method extends the idealized framework by learning viscous diffusion from data, enabling it to model flows up to the point of vortex pre-merging dynamics; however, further study is needed to capture the entire process, including merging and subsequent topology changes, due to the fixed structure of the quantum representation. In the present work, we make a preliminary attempt to incorporate viscous effects through a special treatment within the same vortex based framework, as illustrated by the simple example of two co-rotating vortices in Fig. 4 of the Main Text. The evolution results showed good agreement with the direct numerical simulation of viscous flows, demonstrating the potential of our approach in dealing with viscous scenarios.

In our method, flows are treated with a vortex particle representation, where we implicitly handle vortex particle strengths by introducing a moving reference trajectory that follows the collective motion of the vortex ensemble and rescale the coordinates relative to this reference so that each instantaneous vortex configuration is mapped to a normalized wavefunction. This reference trajectory is updated numerically from the underlying vortex particle dynamics. Once this reference trajectory is available, the same mapping is applied in both cases and the quantum

module operates on a compressed Lagrangian time series in which viscous effects are implicitly encoded in the time dependence of the vortex positions. In order to recover the velocity field, one can subsequently apply empirical relations governing the time evolution of vortex strength.

However, how to address topological changes induced by viscosity [24], such as vortex splitting, merging, or reconnection, has not yet been investigated. In conventional vortex methods, various particle reseeding and local re-meshing techniques, including vortex element splitting, merging, and deletion, have been developed to handle vortex reconnection [25, 26]. From an implementation perspective, we could initialize partially empty vortex element data structures to accommodate increases or decreases in vortex element count, inspiring that incorporating auxiliary registers might provide a potential solution for handling vortex reconnection in future work.

In the revised manuscript, we have explicitly clarified the implications of the use of the idealized system in the Main Text in lines 419–422, and have added more details in the Supplementary information in the line 118–133.

6. For potential-flow modeling, approaches based on hydrodynamic Schrodinger and Madelung transformation have been investigated quite extensively (as reviewed in the Introduction section of the article). So, the relative merits of the current approach as compared to the mentioned alternative approaches needs to be discussed (more) as.

**Reply:** We appreciate the suggestion to further discuss the relative merits of our method compared to established alternatives, which we have thoroughly addressed in the revised manuscript.

Schrödinger / Madelung-based approaches [18, 27–29] model fluid dynamics in a quantum-like framework, particularly effective where smooth potential-flow fields are the focus and full-field observables are required, offering a comprehensive representation of the system with direct parallels to quantum mechanics.

Our method adopts a Lagrangian, point-vortex description to capture vortex–vortex interactions with fewer degrees of freedom, making it compatible with current quantum resources when the flow is governed by coherent vortices.

Specifically, our approach (i) encodes the interactive vortex topology rather than the full fluid field to focus the quantum representation on the essential degrees of freedom, creating a compact mapping to the constrained Hilbert space of near-term devices; (ii) leverages a time register to realize time-parallel state encoding and single-evolution execution, thereby avoiding repeated state re-preparation at every time step; and (iii) focuses on vortex-level observables, e.g., vortex positions

Table R2: Comparison between our quantum vortex method and full-field quantum fluid methods

| Aspect | Schrödinger/Madelung methods | Our vortex-based method |
|---|---|---|
| Representation | Full, smooth velocity field | Sparse, discrete vortex particles |
| Suitable flow regime | General incompressible flows | Vortex-dominated incompressible flows |
| Spatial encoding | Dense grid, requiring high qubit count | Compact vortex topology, scaling with vortex count |
| Temporal evolution | Typically requires sequential time-stepping | Time-parallel processing via quantum superposition |
| Observables | Full-field quantities (e.g., global velocity/pressure) | Vortex-level quantities (e.g., trajectories, interaction patterns) |

and relative configurations, rather than full-field measurements. Considering the constrained qubit numbers and gate fidelities of the current quantum hardware, our method provides a more hardware-efficient way for simulating complex vortex motions.

To clearly summarize the key distinctions, we provide a comparative analysis in Table R2.

We have added these clarifications to the Main Text in lines 368–382, and have discussed further in the Supplementary information in the line 135–146.

7. As detailed in the Supplementary material, the dynamics of the discrete-vortex system is actually modeled using a DMD-type reduced-order model rather than the by discretization of the set of ODEs resulting from the re-formulated Biot–Savart type equations. It seems that this reduced modelling relies on training for a very specific problem involving a set of interacting discrete vortices.

Question: once an evolution operator is trained, can it be applied to a (somewhat) different test case than the one used for training? So, that there is an element of generalization?

**Reply:** The circuit of our method is designed to simulate the evolution of vortex particles and can be methodologically extended to systems with a fixed number of vortex particles. To evaluate generalization of different test cases based on the same trained model, we compared time extrapolation and changes of initial conditions, as shown by the blue curve in Fig. R2, which corresponds to the QVM-based model. Results demonstrate that the solution error obtained by our method remains stable under varying initial conditions, indicating that the trained operators can be effectively reused for systems with the same number of vortices and similar initial configurations.

Specifically, we implement the following experimental setup. The initial condition is $(x_i^0, y_i^0)$

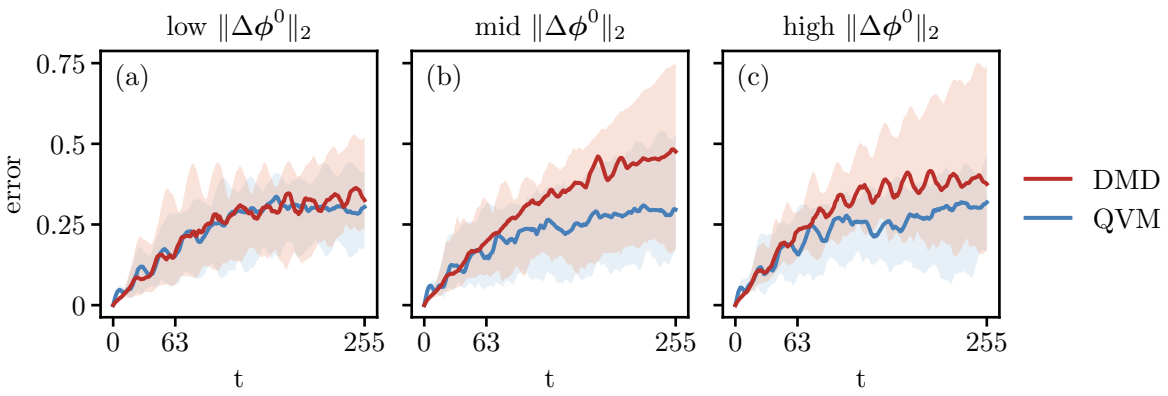


Fig. R2: Validation of generalization capability for the QVM method and comparison with the DMD model. Panels (a), (b), and (c) correspond to the low, mid, and high groups by $\|\Delta\phi^0\|_2$ with ranges $[0.024, 0.084]$, $[0.084, 0.130]$, and $[0.130, 0.240]$. Each group contains 50 cases. The horizontal axis denotes the time frame index. Both QVM and DMD are trained on the first 64 frames of the unperturbed trajectory. From each perturbed initialization we roll out 256 frames and evaluate the configuration space error at each frame. Curves show the median across cases and shaded bands indicate the 10% to 90% percentile band. Within the training window the two methods are similar.

with $x_1 = x_2 = x_3 = x_4 = 0$ and $y_1 = -y_3 = 1$ and $y_2 = -y_4 = 0.4$. We then apply symmetric random perturbations $\Delta\phi^0$ that preserve mirror symmetry about the $x$ axis and we remove rigid translation by centering the sampled offsets. For each mirror pair $(i, \bar{i})$ satisfying $x_{\bar{i}} = x_i$ and $y_{\bar{i}} = -y_i$, we draw $\Delta x_i = \Delta x_{\bar{i}}$ uniformly from $[-2s, 2s]$ and $\Delta y_i$ uniformly from $[-s, s]$, and set $\Delta y_{\bar{i}} = -\Delta y_i$. The scale $s$ equals ten percent of the minimal initial pairwise spacing and the wider range in the $x$ direction increases sample diversity. Let $\Delta\phi^0 := [\Delta x_1^0 + \mathrm{i}\Delta y_1^0, \ldots, \Delta x_4^0 + \mathrm{i}\Delta y_4^0]$. We generate 150 samples of $\Delta\phi^0$, sort them by $\|\Delta\phi^0\|_2$, and split them evenly into low, medium, and high groups, each with 50 cases, corresponding to the ranges $[0.024, 0.084]$, $[0.084, 0.130]$, and $[0.130, 0.240]$, respectively. Both models are trained on the first 64 frames of the unperturbed trajectory. Starting from each perturbed initial condition we roll out 256 frames and compute the configuration space error between the predicted and reference trajectories over time, with the error defined as $\|\phi^t - \tilde{\phi}^t\|_2$, where $\phi^t$ and $\tilde{\phi}^t$ denote the complex vectors that encode vortex positions in configuration space at time $t$, obtained from the QVM or DMD rollouts and from the numerical reference solution for each perturbed initial condition, respectively. For each group we visualize the median error together with the 10% to 90% band.

We provide details in the Supplementary Information in lines 339–366, and related discussions in the Main Text in lines 284–288.

8. This leads to the following further question: to mimic the vortex-interaction problem, it is possible to train a DMD-type problem based on a viscous simulation (e.g. using a vortex particle methods) of the vortex interaction problem (for example using POD or SVD) and then implement this DMD on a quantum processor. This avoids the need to first approximate/linearize the dynamical system before training the reduced-order model. In the revised article, this needs to be considered as well, for example in terms of how these different approaches compare in terms of generalization to different systems.

**Reply:** We appreciate the reviewer's constructive suggestion, and we refine this DMD-based approach into a concrete implementation and compare it with our method in terms of generalization and practical implementability.

In the DMD-based approach, flow snapshots are treated in the configuration space and DMD yields a stepwise linear evolution operator $\mathcal{A}$. Such an operator is generally nonunitary and, when executed on quantum hardware, needs to be block-encoded into a larger unitary, which in turn entails ancilla qubits and postselection. Simulation results indicate that, for the leap-frog case with a fixed vortex number, when the initial conditions deviate from the training distribution, the QVM operator shows slower long-horizon error accumulation and tighter uncertainty bands than the DMD surrogate; under small perturbations the two perform comparably. We hypothesize that the normalization and unitary propagation in QVM act as an implicit regularizer under distribution shift, thereby enhancing robustness, although a rigorous conclusion requires further verification.

Specifically, $\mathcal{A}$ is obtained by applying DMD to trajectories of the vortex-particle position vector $\phi_j$ extracted from viscous-flow simulations. The decomposition is truncated to retain $r$ modes, which allows flexible control of the approximation. The truncation level satisfies $r \leq \min\{N_{\text{train}}, N_p\}$, where $N_{\text{train}}$ denotes the number of training frames and $N_p$ is the number of particles.

$\mathcal{A}$ is embed into a larger unitary via the linear combination of unitaries (LCU)[30–32], and subsequently serves as an evolution module $F$ within our quantum circuit. This can be expressed as

$$\mathcal{A} = \sum_{j=0}^{J-1} \beta_j V_j, \quad \beta_j \geq 0, \; V_j \text{ unitary}, \quad \Lambda = \sum_{j=0}^{J-1} \beta_j, \tag{R3}$$

which prepares an ancilla state

$$\mathrm{PREP}\,|0^m\rangle \;=\; \sum_{j=0}^{J-1} \sqrt{\frac{\beta_j}{\Lambda}}\,|j\rangle, \tag{R4}$$

and applies the selection operator

$$\mathrm{SELECT}_{\{V_j\}} \;=\; \sum_{j=0}^{J-1} |j\rangle\langle j| \otimes V_j. \tag{R5}$$

Here $\{V_j\}_{j=0}^{J-1}$ are the unitary blocks appearing in the LCU decomposition of $\mathcal{A}$, $J$ is the number of terms in this decomposition, and $m$ is the number of ancilla qubits used to index these terms with $m \geq \lceil \log_2 J \rceil$. The composite unitary

$$U_{\mathcal{A}} \;=\; \left(\mathrm{PREP}^{\dagger} \otimes I\right) \cdot \mathrm{SELECT} \cdot \left(\mathrm{PREP} \otimes I\right) \tag{R6}$$

is a block encoding of $\mathcal{A}/\Lambda$ in the sense that

$$\left(\langle 0^m| \otimes I\right) U_{\mathcal{A}} \left(|0^m\rangle \otimes I\right) \;=\; \mathcal{A}/\Lambda, \tag{R7}$$

Thus, postselecting the ancilla in the state $|0^m\rangle$ applies $\mathcal{A}/\Lambda$ to the system.

Each evolution matrix requires multiple ancilla qubits, with the count determined by the number of LCU terms. Increasing the number of LCU terms typically increases $\Lambda$, which reduces postselection success probability, necessitating more repetitions or stronger amplification. If the spectral radius of $\mathcal{A}$ is large, which in our setting corresponds to trajectories of $\phi_j$ that move away from the origin and increase in norm, the required $\Lambda$ becomes larger, leading to a lower per-step success rate or a higher amplification cost.

To compare with the DMD-based model in terms of generalization to different systems, we study temporal extrapolation and variations in initial conditions. Under the same configuration settings as described previously, Fig. R2 shows that during the first 64 frames, which correspond exactly to the training window, the error curves of both methods remain very close. Beyond this horizon, the two remain comparable under small deviations from the training initialization, whereas for progressively larger deviations the QVM exhibits slower long-horizon error growth and tighter uncertainty bands than the DMD surrogate. We hypothesize that the normalization and unitary propagation in the

QVM act as an implicit regularizer under distribution shift, improving robustness at the cost of a small in-distribution bias, which complements DMD's strong interpolation performance near the training manifold.

From a practical perspective, DMD provides a lightweight and interpretable baseline with good accuracy. Its quantum realization, however, requires block-encoding of a nonunitary step with ancilla qubits and postselection. The QVM is hardware-native but incurs pre- and post-processing as well as data-calibration overhead. The preferable choice therefore depends on the expected degree of distribution shift and on implementation constraints. In future work, a hybrid approach combining the benefits of both methods is worth exploring to improve both performance and adaptability.

We provide details in the Supplementary Information in lines 339–366, and related discussions in the Main Text in lines 466–484.

9. Measurement and data extraction need to be discussed further. With solution at multiple-time step encoded in the quantum register, the likelihood of measuring data related to an intended time step reduced rapidly with increasing number of time steps resolved. The 'single quantum execution' mentioned in Introduction on page 1 needs further detailing, e.g. mention that multiple realization are needed for extraction of flow data and that the number of realizations for a given accuracy will increased with $n_t$.

**Reply:** We thank the reviewer for raising this point. As suggested by the reviewer, we have clarified in the introduction that the phrase "single quantum execution" refers to one circuit structure that prepares a quantum state encoding multiple time steps in superposition, and mentioned that data extraction still requires repeated executions of the quantum circuit to collect samples, and the number of realizations for a given accuracy increase with $n_t$. We note that when particular time slices are required, amplitude amplification can target the subspace associated with the desired index. This boosts the probability of success from $p$ to a constant level with $O(1/\sqrt{p})$ iterations and reduces the number of shots compared with naive sampling.

Furthermore, we would like to note that the above-mentioned data extraction may not be necessary in specific scenarios. For example, a superposition state that carries multiple time steps can be processed coherently by applying specific oracles that read the time index register and apply operators that compute flow relevant quantities, correlate information across different time steps, or flag events of interest without collapsing the state. Besides, instead of extracting the full vortex

state, one may want to compare the trajectories of two vortex movements, which can be realized with SWAP test technique, thereby reducing the need for complex post-processing.

We have made clarifications in the Main Text in lines 85–89, and have add the discussion of measurement and data extraction in the Supplementary Information in lines 311–319.

10. Viscous vortex particle system (page 5): required changes to learning the evolution operator, i.e. with regards using varying vortex particle strengths (circulation) needs to be explained more clearly.

**Reply:** As outlined in Comment 5, our method implicitly incorporates vortex particle strengths through a dynamically updated reference frame that maps vortex configurations to normalized wavefunctions, encodes viscous effects in positional evolution, and enables velocity field recovery via empirical vortex strength relations.

Specifically, the core idea of the normalization transformation is to choose a suitable reference point whose motion represents the collective movement of the vortex particle system, while the absolute motion of each vortex particle is thus redefined as motion relative to this reference point, which itself is independent of the presence of viscosity. The difference between these two treatments lies in whether the trajectory of the reference point $C(t)$ can be determined analytically. To select a single point from the admissible circle and obtain a continuous and smooth reference trajectory, an additional selection rule is required. In general, for a given time $t$, the reference point can be located in the circle centered at $\bar{\phi}(t)$ with radius

$$\sqrt{\frac{\lambda^{-2} - S\big(\bar{\phi}(t)\big)}{N_p}}. \tag{R8}$$

where $S(C) = S(\bar{\phi}) + N_p \left|C - \bar{\phi}\right|^2, \quad \bar{\phi} = \frac{1}{N_p}\sum_j \phi_j.$

For viscous flows, we set

$$C(t) = \bar{\phi}(t) + \sqrt{\frac{\lambda^{-2} - S\big(\bar{\phi}(t)\big)}{N_p}}\,\frac{\bar{\phi}(t) - \bar{\phi}(t-\Delta t)}{\left|\bar{\phi}(t) - \bar{\phi}(t-\Delta t)\right|}, \tag{R9}$$

which determine the time evolution of vortex particles, and their strengths can be recovered from the particle trajectories.

We have expanded the Supplementary Information in lines 58-74 to clarify how viscous systems

are mapped to a quantum formulation, and reflected in the Main Text in lines 316-321.

---


[1] S. Xiong, Y. Tong, X. He, S. Yang, C. Yang, and B. Zhu. Nonseparable symplectic neural networks. In *International Conference on Learning Representations*, 2021.

[2] S. Weißmann, U. Pinkall, and P. Schröder. Smoke rings from smoke. *ACM Trans. Graph.*, 33:140, 2014.

[3] A. Chern, F. Knöppel, U. Pinkall, P. Schröder, and S. Weißmann. Schrödinger's smoke. *ACM Trans. Graph.*, 35:77, 2016.

[4] A. Chern, F. Knöppel, U. Pinkall, and P. Schröder. Inside fluids: Clebsch maps for visualization and processing. *ACM Trans. Graph.*, 36:142, 2017.

[5] W. Thompson (Lord Kelvin). On vortex motion. *Trans. R. Soc. Edinburgh*, 25:217–260, 1869.

[6] H. Helmholtz. Über integrale der hydrodynamischen gleichungen welche den wirbelbewegungen entsprechen. *J. Reine Angew. Math.*, 55:25–55, 1858.

[7] H. Ertel. Ein neuer hydrodynamischer wirbelsatz. *Meteorol. Z.*, 59:271–281, 1942.

[8] J. J. Moreau. Constantes d'un îlot tourbillonnaire en fluide parfait barotrope. *C. R. Acad. Sci. Paris*, 252:2810–2812, 1961.

[9] H. K. Moffatt. The degree of knottedness of tangled vortex lines. *J. Fluid Mech.*, 35:117–129, 1969.

[10] J.-P. Liu, H. Ø. Kolden, H. K. Krovi, N. F. Loureiro, K. Trivisa, and A. M. Childs. Efficient quantum algorithm for dissipative nonlinear differential equations. *Proc. Natl. Acad. Sci. U.S.A.*, 118:e2026805118, 2021.

[11] H. Krovi. Improved quantum algorithms for linear and nonlinear differential equations. *Quantum*, 7:913, 2023.

[12] H.-C. Wu, J. Wang, and X. Li. Quantum algorithms for nonlinear dynamics: Revisiting carleman linearization with no dissipative conditions. *SIAM J. Sci. Comput.*, 47:A943–A970, 2025.

[13] P. C. S. Costa, P. Schleich, M. E. S. Morales, and D. W. Berry. Further improving quantum algorithms for nonlinear differential equations via higher-order methods and rescaling. *npj Quantum Inf.*, 11:141, 2025.

[14] F. Tennie, S. Laizet, S. Lloyd, and L. Magri. Quantum computing for nonlinear differential equations and turbulence. *Nat. Rev. Phys.*, 7:220–230, 2025.

[15] M. Rath and H. Date. Quantum data encoding: A comparative analysis of classical-to-quantum

mapping techniques and their impact on machine learning accuracy. *EPJ Quantum Technol.*, 11:72, 2024.

[16] R. Au-Yeung, B. Camino, O. Rathore, and V. Kendon. Quantum algorithms for scientific computing. *Rep. Prog. Phys.*, 87:116001, 2024.

[17] A. M. Childs, J.-P. Liu, and A. Ostrander. High-precision quantum algorithms for partial differential equations. *Quantum*, 5:574, 2021.

[18] S. Succi, W. Itani, K. Sreenivasan, and R. Steijl. Quantum computing for fluids: Where do we stand? *Europhys. Lett.*, 144:10001, 2023.

[19] L. Grover and T. Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv preprint quant-ph/0208112*, 2002.

[20] M. Plesch and Č. Brukner. Quantum-state preparation with universal gate decompositions. *Phys. Rev. A*, 83:032302, 2011.

[21] V. Giovannetti, S. Lloyd, and L. Maccone. Quantum random access memory. *Phys. Rev. Lett.*, 100:160501, 2008.

[22] C. T. Hann, C. L. Zou, Y. X. Zhang, Y. W. Chu, R. J. Schoelkopf, S. M. Girvin, and L. Jiang. Hardware-efficient quantum random access memory with hybrid quantum acoustic systems. *Phys. Rev. Lett.*, 123:250501, 2019.

[23] N. Jiang, Y. F. Pu, W. Chang, C. Li, S. Zhang, and L. M. Duan. Experimental realization of 105-qubit random access quantum memory. *npj Quantum Inf.*, 5:28, 2019.

[24] S. Kida and M. Takaoka. Vortex reconnection. *Annu. Rev. Fluid Mech.*, 26:169–189, 1994.

[25] A. J. Chorin. Hairpin removal in vortex interactions. *J. Comput. Phys.*, 91:1–21, 1990.

[26] S. Xiong, R. Tao, Y. Zhang, F. Feng, and B. Zhu. Incompressible flow simulation on vortex segment clouds. *ACM Trans. Graph.*, 40:98, 2021.

[27] L. Salasnich, S. Succi, and A. Tiribocchi. Quantum wave representation of dissipative fluids. *Int. J. Mod. Phys. C*, 35:2450100, 2024.

[28] Z. Meng and Y. Yang. Quantum computing of fluid dynamics using the hydrodynamic schrödinger equation. *Phys. Rev. Res.*, 5:033182, 2023.

[29] Z. Meng and Y. Yang. Quantum spin representation for the navier-stokes equation. *Phys. Rev. Res.*, 6:043130, 2024.

[30] D. W. Berry, A. M. Childs, and R. Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *2015 IEEE 56th annual symposium on foundations of computer science*, pages

792–809, 2015.

[31] G. H. Low and I. L. Chuang. Hamiltonian simulation by qubitization. *Quantum*, 3:163, 2019.

[32] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proc. STOC 2019*, pages 193–204, 2019.