Author/s:
Heredge, J;West, M;Hollenberg, L;Sevior, M

Title:
Nonunitary quantum machine learning

Date:
2025-04

Citation:
Heredge, J., West, M., Hollenberg, L. & Sevior, M. (2025). Nonunitary quantum machine learning. Physical Review Applied, 23 (4), https://doi.org/10.1103/PhysRevApplied.23.044046.

PHYSICAL REVIEW APPLIED **23,** 044046 (2025)

Nonunitary quantum machine learning

Jamie Heredge,* Maxwell West, Lloyd Hollenberg, and Martin Sevior
School of Physics, University of Melbourne, Parkville, Victoria 3010, Australia

(Received 2 July 2024; revised 15 February 2025; accepted 19 February 2025; published 21 April 2025)

We introduce several probabilistic quantum algorithms that overcome the normal unitary restrictions in quantum machine learning by leveraging the linear combination of unitaries (LCU) method. We cover three distinct topics, beginning with quantum native implementations of residual networks (ResNets). We demonstrate that while residual connections between layers of a variational *Ansatz* can prevent barren plateaus in models, this approach is accompanied by a trade-off in success probability. Second, we implement a quantum analogue of average-pooling layers from convolutional networks using single-qubit-controlled basic arithmetic operators and show that the LCU success probability remains stable for the Modified National Institute of Standards and Technology (MNIST) database. This method can be further generalized to convolutional filters, while using exponentially fewer controlled unitaries than previous approaches. Finally, we propose a general framework for applying a linear combination of irreducible-subspace projections on quantum encoded data for any finite group. This enables a quantum state to remain within an exponentially large space, while selectively amplifying specific subspaces relative to others, alleviating simulability concerns that arise when fully projecting to a polynomially sized subspace. We demonstrate improved classification performance for partially amplified permutation-invariant encoded point-cloud data when compared to noninvariant or fully permutation-invariant encodings. We also demonstrate a novel rotationally invariant encoding for point-cloud data via Schur-Weyl duality. These quantum computing frameworks are all constructed using the LCU method, suggesting that further novel quantum machine-learning (QML) algorithms could be created by utilizing the LCU technique.

DOI: 10.1103/PhysRevApplied.23.044046

I. INTRODUCTION

Quantum computing is an emerging technology with the potential to solve certain problems far more efficiently than classical techniques [1–3]. The search for useful applications of quantum computers within the domain of machine learning is of particular importance given the significant impact that classical machine learning has had on many fields and industries in recent years [4–6]. Various quantum machine-learning (QML) algorithms have been proposed, with some of the most common examples attempting to recreate a quantum analogue of the classical neural network [7,8]. However, the inherently unitary operations of quantum algorithms impose significant limitations, as they restrict the implementation of nonunitary operations that are essential in many classical machine-learning models. To address this issue, we explore the linear combination of unitaries (LCU) method [9,10], a technique that employs ancilla qubits to probabilistically implement a linear combination of unitary operations, which therefore permits the implementation of nonunitary operations in quantum circuits. This unlocks the possibility of implementing nonunitary operations within QML applications. We utilize this to present several novel applications of the LCU method with various advantages in the field of QML.

In Sec. II, we adapt the classical residual-learning framework to quantum variational circuits, facilitating a quantum native implementation of residual networks (ResNets) [11]. By partially skipping layers of the network, classical residual networks are able to avoid the vanishing-gradient problem by allowing the gradients to flow through shallower sections of the network [11,12]. We show that a quantum ResNet may similarly provide a method of avoiding barren plateaus in variational quantum circuit (VQC) models, by maintaining shallow-depth contributions in the final loss function. Furthermore, we show that by including terms that parametrize the strength of the residual connections, it is possible to increase the lower bound of the probability of success of the LCU procedure, helping to alleviate one of the principal issues of the LCU method.

---

[*] Contact author: jamie.heredge@student.unimelb.edu.au, jamieheredge@hotmail.com

In Sec. III, we implement a quantum analogue of average-pooling operations from convolutional neural networks (CNNs) [13–15] by using the LCU method. This provides an efficient implementation of average pooling for amplitude-encoded image data which we demonstrate for any size pooling window. This may further be generalized to convolutional filters, leading to an exponential improvement in the number of controlled unitaries required compared to previous techniques [16,17]. We demonstrate that the LCU probability of success will equal 1 in the case in which all pixels are of the same color and will decrease when pixels in the local pooling window of dimension $D \times D$ become more diverse. For real-world images, we provide the intuition that the probability remains relatively stable since most pixels are similar locally, except at the edges of subjects in the image. This intuition is supported by empirical evidence on $N \times N$ pixel images in the Modified National Institute of Standards and Technology (MNIST) [18] database, which shows that the probability decreases but levels off to a finite value as D increases and shows no discernible trend when increasing the image size N.

In Sec. IV, we present a method for projecting quantum encoded data to any combination of irreducible-representation subspaces of a finite group, presenting a general framework for implementing full or partial symmetries in the encoding step of QML models. These techniques are intended to reduce the effective dimension of quantum encoded data in an effort to improve generalization performance, which has been reported to decline as the number of qubits and hence the dimension of the encoded quantum states increases [19]. We show that this technique can recreate previous work on permutation-invariant encodings for point-cloud data [20] as a special case. Furthermore, we implement a novel rotationally invariant encoding for point-cloud data using the new technique by leveraging Schur-Weyl duality. This results in an encoded quantum state for point-cloud data that is invariant if classical input point-cloud data are rotated in three-dimensional (3D) space, hence strongly enforcing rotation invariance on any model that subsequently uses this rotationally invariant input state. We further show that any combination of projections can be implemented at once, allowing certain symmetric subspaces of the data to be amplified or contracted to give increased flexibility over the amount of symmetry in the encoding. We demonstrate that intermediate levels of permutation symmetry for point-cloud encoded data leads to an improved classification performance when compared to a nonsymmetric or a fully permutation-symmetric encoding.

These implementations illustrate the ability of the LCU method to benefit the field of QML and a summary of these contributions is provided in Table I. This work covers three different algorithmic frameworks that all utilize the same LCU method in their construction, which are located in self-contained sections that may be read in any order:

(1) *Quantum native ResNet* is detailed in Sec. II.

(2) *Quantum native average pooling* is detailed in Sec. III.

(3) *Irreducible-representation projections* are detailed in Sec. IV.

The remainder of this introductory section will introduce these three frameworks, followed by a summary of the LCU method used in their constructions.

A. Quantum native ResNets

A common issue in the training of variational quantum circuit (VQC) models is the issue of barren plateaus leading to vanishing gradients. Classical ResNets have profoundly impacted deep learning by enabling the training of extremely deep neural networks through the introduction of skip connections that mitigate the vanishing-gradient problem [11,12]. This suggests that the implementation of residual networks within a quantum variational model could provide a significant advancement in QML if they could similarly be utilized to avoid the problem of barren plateaus inherent in many VQC models. This line of reasoning has been suggested in a recent review of barren plateaus [21], with the caveat that the no-cloning theorem may make residual and skipped connections difficult to achieve in quantum circuits. In this work, we shall show a probabilistic implementation of a quantum native ResNet that does not require cloning states and could show promise against barren plateaus in VQC models.

There has been active interest in building quantum ResNet inspired algorithms in the literature, with a primary focus being on quantum-classical hybrid models that use the powerful modern architecture of classical ResNet models, while including QML subroutines to produce a hybrid algorithm [22–24]. There have also been proposals to implement quantum residual connections by utilizing several quantum neural network models in series with residual connections between them [25]. A native VQC implementation of a quantum ResNet algorithm has previously been introduced in Refs. [26,27], in which skipped connections are possible in a VQC model if the implemented variational layers are restricted to be of the form of a unitary circuit followed by a product Pauli encoding, followed by the unitary-circuit conjugate. This implements the quantum ResNet natively on a quantum device; however, as the authors note, this does not cover any general unitary variational layer $W_l(\theta_l)$, since implementing residual connections for a general $W_l(\theta_l)$ would not be a unitary process overall and has therefore not been considered in this approach. A common theme explored in these previous works has been utilizing the effectiveness of classical

TABLE I. A summary of the algorithms introduced in this work using the LCU framework. G is a finite group. $|\psi_r\rangle$ is the portion of a state $|\psi\rangle$ that occupies the subspace of the irreducible representation r, with $|\psi_{\text{rot}}\rangle$ meaning the portion of $|\psi\rangle$ that is in the rotationally invariant subspace. Note that $\langle\psi_r|\psi_r\rangle$ is not necessarily equal to 1, as the condition is $\langle\psi|\psi\rangle = \sum_r\langle\psi_r|\psi_r\rangle = 1$. The parameters a_r can be chosen by the user subject to normalization constraints. In average pooling, we use a pooling window of dimension $D \times D$ pixels. L is the number of layers in the quantum ResNet. $\beta_l \in [0, 1]$, $\beta_l \in \mathbb{R}$ parametrize the strength of the skipped connections in the quantum ResNet. Throughout this work, $\log \equiv \log_2$ is the logarithmic function for base 2.

Algorithm	Result	Ancilla qubits	Success probability	Section
Quantum ResNet residual layers	Residual connections can avoid barren plateaus	$\mathcal{O}(L)$	$[\prod_l (1 - 4\beta_l(1 - \beta_l)), 1]$	Sec. II
Average pooling	Exponentially fewer controlled unitaries in averaging or convolutional step than previous techniques [16,17]	$\mathcal{O}(\log(D))$	Shown for MNIST in Figs. 14 and 15	Sec. III
Irreducible-subspace projections	Can freely enforce and parametrize symmetry of a quantum encoded state	$\mathcal{O}(\log(\|G\|))$	$\sum_r \|a_r\|^2\langle\psi_r\|\psi_r\rangle$	Sec. IV
Rotationally invariant encoding	Encoding respects rotational invariance of point-cloud data	$\mathcal{O}(\log(\|G\|))$	$\langle\psi_{\text{rot}}\|\psi_{\text{rot}}\rangle$	Sec. IV C 2

ResNet models in tackling the vanishing-gradients problem of deep neural networks and applying this reasoning in a quantum setting to the problem of barren plateaus.

Applying residual connections in a quantum setting corresponds to operating on a state $|\psi_{l-1}\rangle$ to produce

$$|\psi_l\rangle = W_l(\theta_l)|\psi_{l-1}\rangle + |\psi_{l-1}\rangle, \tag{1}$$

whereby a portion of the previous state $|\psi_{l-1}\rangle$ is able to skip the variational operator $W_l(\theta_l)$ in that layer. This allows the overall model cost function to retain terms that have only passed through one variational layer, corresponding to very shallow circuits, while still providing terms that have been passed through all layers and hence potentially very deep circuits. The operator that would result in this would be of the form

$$A_l = W_l(\theta_l) + I, \tag{2}$$

which in general is not a unitary operator. The key challenge therefore in implementing a quantum native ResNet analogue is that it would require implementation of nonunitary operators, something that becomes possible with the LCU framework. In this work, we translate the ResNet architecture to a quantum setting by applying residual connection with the LCU framework to facilitate the flow of quantum information across deeper or more complex quantum circuits, while allowing the strength of the residual connection to be chosen freely. This introduces a potential new class of VQC ResNet models that could provide possible protection against barren plateaus in complex VQC models.

The use of ancilla qubits to implement ResNet-like architectures in QML models has previously been explored [28], revealing that residual connections within the data-encoding segment of a circuit can expand the frequency spectrum of the resulting model, leading to more expressive encodings. In contrast, our work introduces a framework for quantum ResNets exclusively within the variational portion of a VQC model. We provide a detailed proof of the probability of success of the LCU procedure, demonstrating that the lower bound of this probability can be adjusted by varying the residual-connection strength. Additionally, we illustrate that applying quantum residual layers to a model can mitigate the occurrence of barren plateaus in circuits that would otherwise contain them. We also demonstrate that quantum ResNets can be viewed as equivalent to ensembles of unitary VQC models with additional nonunitary terms. While we show that quantum ResNets can avoid barren plateaus, we also discuss how they may likely be classically simulatable in many cases, at least in the case in which the connection between absence of barren plateaus and classical simulatability [29] is valid for the constituent components. We propose that the solution to this quantum ResNet simulatability issue may lie in the nonunitary terms and suggest a characterization of these terms as a topic for further research.

B. Average-pooling layers

Classical CNNs are of significant importance to machine learning, mainly due to their structured manner of handling image and video data [13–15,30]. Inspiration from these models has led to the development of quantum analogues. Quantum CNNs have been previously proposed [31,32] to classify quantum states with certain symmetry-protected topological phases and to classify image data sets [33,34]. In these models, variational circuits are used during convolutional layers, followed by further variational circuits and measurements in the pooling layer to perform dimensionality reduction such that the operations are performed in a manner that respects certain symmetries of the data. Significant benefits of these models, such as avoiding barren plateaus [35], have been identified. Quantum-classical hybrid techniques have also been developed that utilize classical CNN architectures alongside quantum models [36].

The implementation that we present here differs in that we focus entirely on implementing a subroutine of classical CNN models, the average-pooling layer, for amplitude-encoded image data. We consider the subroutine in which a pooling window of size $D \times D$ passes over the image and outputs the average of all pixels found within the pooling window. We show that this can be implemented natively on a quantum circuit by utilizing the LCU method, which could lead to an improvement over performing the subroutine classically, as quantum parallelism allows the averaging operation to apply to all pixel simultaneously. This demonstrates the possible utility of the LCU method, while providing a potentially advantageous subroutine for future quantum CNN models.

Previous work has investigated the LCU technique for creating convolutional-layer filters [16] based on spatial filtering [37], which can recover average pooling as a special case. This work has not generalized to any size D, restricting instead to $D = 3$, but has stated that the method would require D^2 multicontrolled operators in general. In contrast, we show a valid proof for any D and an efficient circuit implementation that requires only $\mathcal{O}(\log(D))$ single-qubit controlled unitaries, leading to an exponential improvement in D over previous work [16]. We also show that our construction can be generalized by adjusting the ancilla-qubit state initialization in order to implement a general convolutional-layer filter, recovering the main result of Ref. [16] while maintaining an exponential improvement in D.

C. Irreducible-representation-subspace symmetry projections

A known issue in QML is that as the number of qubits increases, there is a decrease in the generalization performance of algorithms [19]. A common empirical explanation of this is that the exponentially large Hilbert space leads to an overly expressive feature encoding where overtraining on the data becomes commonplace. Without an accompanied exponential increase in the training data, this leads to an overall reduction in the performance in the validation data set. A possible solution to the issue of overtraining is to reduce the expressibility of the encoding. Examples of techniques that have attempted this range from projecting kernels to a lower-dimensional space [38] to approaches that are capable of encoding inductive biases directly into quantum states [39]. Furthermore, geometric QML techniques have been used to study methods for creating variational circuits that are equivariant with respect to data symmetries [40–44], in similar attempts to reduce the expressibility of QML models. We instead focus on implementing symmetries directly into the quantum encoded data, which is a stricter implementation of symmetry, meaning that our procedure is agnostic to the trainable classification procedure.

In previous work, a quantum encoding has been proposed using permutation symmetry, which has led to a reduction in the dimensionality of the encoding and improved classification performance [20]. This has been discussed in the context of point-cloud data (unordered collection of points in three dimensions that collectively represent an image), where each point cloud X consists of n points and each point $\mathbf{p}_i$ is a 3D vector. As points do not have intrinsic ordering, the ordering of the points as they are input into a classification algorithm should ideally have no effect on the outcome. Therefore, the point-cloud data naturally has point-ordering permutation invariance. In general, a machine-learning classifier f could return a different result depending on the order of the points, i.e., $f(\mathbf{p}_1, \mathbf{p}_2) \neq f(\mathbf{p}_2, \mathbf{p}_1)$, unless it has been specifically constructed to respect the permutation symmetry. The permutation-invariant encoding [20] acts by creating an equal quantum superposition of all permutations of the data. In the two-state case, given two points, $\mathbf{p}_1$ and $\mathbf{p}_2$, encoded into the quantum state $|\mathbf{p}_1\rangle \otimes |\mathbf{p}_2\rangle$, this would correspond to preparing the permutation-invariant state

$$|X_s\rangle = \frac{1}{\sqrt{2}}(|\mathbf{p}_1\rangle|\mathbf{p}_2\rangle + |\mathbf{p}_2\rangle|\mathbf{p}_1\rangle). \tag{3}$$

However, it has been shown that the permutation-invariant state preparation procedure

$$|\mathbf{p}_1\rangle|\mathbf{p}_2\rangle \Rightarrow \frac{1}{\sqrt{2}}(|\mathbf{p}_1\rangle|\mathbf{p}_2\rangle + |\mathbf{p}_2\rangle|\mathbf{p}_1\rangle) \tag{4}$$

cannot be implemented via a unitary operation [45]. However, this process can be implemented in a probabilistic manner using ancilla qubits [46]. This has been shown to lead to improved classification performance for point-cloud image classification [20].

In this work, we demonstrate a generalization of this technique that allows projections to any irreducible-representation subspace of a finite group. Unlike previous works [20], this means that we are no longer restricted to the symmetric subspace or the permutation group S_n. Furthermore, our technique allows for linear combinations of projections to any irreducible-representation subspaces.

Utilizing this framework, we demonstrate a rotationally invariant encoding for point-cloud data as an example use case. This encoding produces the same quantum encoded state each time, even if the data-input point cloud is rotated by any amount in three dimensions. We show that this is achieved both theoretically and numerically. This is a highly desirable property of the model, as point-cloud data naturally have rotational symmetry. Especially in applications such as computer vision for autonomous vehicles, it is of the utmost importance that subjects in the image, such as pedestrians, are correctly identified regardless of the angle from which they are being viewed. Previous work has focused on implementing rotational and permutation symmetry in models for point-cloud data [47], which did so by implementing an equivariant variational model. We highlight that our work does not use equivariant variational models but projects quantum input states to a rotationally invariant subspace, hence strictly enforcing rotational invariance into any model for which this input encoded state is passed, as we effectively delete all information of the state that is not rotationally invariant.

The permutation and rotationally invariant encodings mentioned previously succeed in reducing the dimension of the encoding but we note that this reduction may indeed be too drastic, which could lead to classically simulatable approaches or could simply delete too much information about the input state, hindering the model performance. We therefore show how our new framework allows for linear combinations of projections that can be utilized to introduce parametrized symmetry subspace amplification. In this setting, the dimension of the quantum state can remain exponentially large, while subspaces associated with certain symmetries can have their relative weightings adjusted. We focus on the weighting of the permutation-symmetric subspace relative to all other subspaces, which can be continuously adjusted using a hyperparameter α. We show that by implementing an intermediate amount of permutation symmetry for point-cloud data classification, it is possible to gain higher accuracy scores than using either noninvariant encodings or the fully permutation-invariant encodings suggested in previous work [20]. A visualization of these applications is shown in Fig. 1.

D. Linear combination of unitaries method

All results in this paper are specific cases of the LCU method described in this section. Let us define the general framework of how the LCU method works in a quantum circuit as detailed in Refs. [9,10]. An insightful tutorial on the LCU method can also be found in Ref. [48]. The LCU procedure allows the implementation of any operator A that is itself a linear combination of N unitary operators $U_j \in SU(2^n)$. The operator A acts on a target state that is contained in an n-qubit target register. The target state is denoted $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$. We can define the operator A acting on $|\psi\rangle$ as

$$A|\psi\rangle = \frac{1}{\Omega'}\sum_{j=1}^{N}\alpha_j U_j|\psi\rangle, \tag{5}$$

where for simplicity $\alpha_j \geq 0, \alpha_j \in \mathbb{R}$ and any negative sign or complex phase can be absorbed into the unitary $U_j \in SU(2^n)$ and Ω' is a normalization constant for the final target state.

In order to implement this, we need to define an ancilla preparation operator P_{PREP} that prepares the k ancilla qubits, which are initially in a basis state $|b_1\rangle$, into the following state:

$$P_{\text{PREP}}|b_1\rangle = \frac{1}{\sqrt{\Omega}}\sum_{j=1}^{N}\sqrt{\alpha_j}|b_j\rangle, \tag{6}$$

where $\Omega = \sum_{j=1}^{N}\alpha_j$ is a normalization constant for the ancilla state and $\{|b_j\rangle\} \in (\mathbb{C}^2)^{\otimes k}$ are basis states of the 2^k-dimensional Hilbert space for the k-qubit ancilla register denoted by $\mathcal{H} = (\mathbb{C}^2)^{\otimes k}$, which can be taken to be the computational basis states. The operator itself can be explicitly written as

$$P_{\text{PREP}} = \frac{1}{\sqrt{\Omega}}\sum_{j=1}^{N}\sqrt{\alpha_j}|b_j\rangle\langle b_1| + \sum_{i=2}^{2^k}\sum_{j=1}^{2^k}u_{i,j}|b_j\rangle\langle b_i|, \tag{7}$$

where any terms $(\ldots)\langle b_i|$ for $i \geq 2$ will not be used and hence can be ignored.

After the ancilla qubits are prepared, we then apply a selection operator S_{SELECT}. The selection operator applies the unitary operation $U_j \in SU(2^n)$ to the target register state $|\psi\rangle$ on the condition that the ancilla qubit is in the state $|b_j\rangle$, which can be defined as

$$S_{\text{SELECT}}|b_j\rangle|\psi\rangle = |b_j\rangle U_j|\psi\rangle. \tag{8}$$

If we now combine the preparation and selection operators, we have

$$S_{\text{SELECT}}P_{\text{PREP}}|b_1\rangle|\psi\rangle = \frac{1}{\sqrt{\Omega}}\sum_{j=1}^{N}\sqrt{\alpha_j}|b_j\rangle U_j|\psi\rangle. \tag{9}$$

The final step consists of applying $P^{\dagger}_{\text{PREP}}$, which is defined as

$$P^{\dagger}_{\text{PREP}} = \frac{1}{\sqrt{\Omega}}\sum_{j=1}^{N}\sqrt{\alpha_j}|b_1\rangle\langle b_j| + \sum_{i=2}^{2^k}\sum_{j=1}^{2^k}u^{*}_{i,j}|b_i\rangle\langle b_j|. \tag{10}$$

Applying this to the circuit results in

$$\begin{aligned} &P^{\dagger}_{\text{PREP}}S_{\text{SELECT}}P_{\text{PREP}}|0\rangle|\psi\rangle \\ &\quad = \frac{1}{\Omega}\sum_{j=1}^{N}\alpha_j|b_1\rangle U_j|\psi\rangle + \sum_{i=2}^{2^k}(\ldots)|b_i\rangle|\psi\rangle, \end{aligned} \tag{11}$$

where $\sum_{i=2}^{2^k}(\ldots)|b_i\rangle|\psi\rangle$ collects terms that will be discarded and can therefore be ignored. We now need to

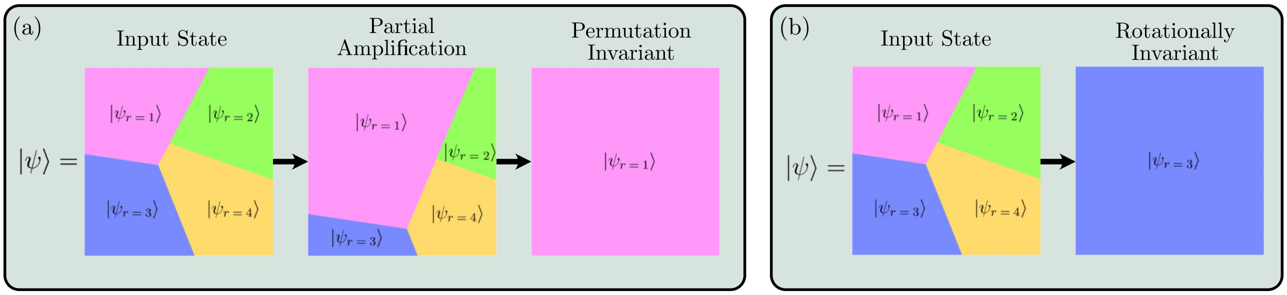

FIG. 1. A visual representation of the irreducible-subspace projection of a quantum state. The input state $|\psi\rangle$ has components in each irreducible-representation subspace denoted by $|\psi_r\rangle$. (a) The component of $|\psi\rangle$ in the subspaces can be partially amplified to any desired ratio between the subspace components. Controlling the amplification of the permutation-symmetric subspace $|\psi_{r=1}\rangle$ for point-cloud data is the focus of Sec. IV D. The state can also be fully projected to $|\psi_{r=1}\rangle$, which for the permutation group S_n would correspond to a fully permutation-invariant state. Permutation-invariant encodings have been the subject of previous work [20] and are reviewed within our new framework in Sec. IV C 1. (b) The input state $|\psi\rangle$ is fully projected to $|\psi_{r=3}\rangle$ within the $r = 3$ irreducible subspace. In Sec. IV C 2, we show that this corresponds to creating rotationally invariant encodings of point-cloud data.

measure the ancilla register and discard any results when the ancilla is not measured in the $|b_1\rangle$ state. The probability of measuring the $|b_1\rangle$ state will equal the probability of success of the LCU method π_S, which can be written as

$$\pi_S = \left| \frac{1}{\Omega} \sum_{j=1}^{N} \alpha_j U_j |\psi\rangle \right|^2 . \tag{12}$$

Discarding any results in which the ancilla is not measured in the $|b_1\rangle$ state, we see that the remaining state will be projected to

$$\langle 0|P^{\dagger}_{\text{PREP}}|S_{\text{SELECT}}|P_{\text{PREP}}|0\rangle|\psi\rangle = \frac{1}{\Omega'} \sum_{j=1}^{N} \alpha_j U_j |\psi\rangle, \tag{13}$$

where $\Omega' = \sqrt{\pi_S}\Omega$ is the normalization constant for the final state, which can then be written as

$$\frac{1}{\Omega'} \sum_{j=1}^{N} \alpha_j U_j |\psi\rangle = A|\psi\rangle. \tag{14}$$

Hence, the operator A, which is an LCU and hence may itself be nonunitary, has been applied to the state $|\psi\rangle$ [49].

The goal of this work is to demonstrate how the LCU method described above can be used in QML tasks to achieve desirable traits in the model architecture that are not possible in a strict unitary setting. The main results of this work rely on specifying preparation and selection operators, showing that they can be implemented on a quantum device, and then repeating the LCU framework detailed here to prove that they result in the desired nonunitary operation. An example LCU circuit for the implementation of $A|\psi\rangle = 1/\Omega'(\alpha_1 U_1 + \alpha_1 U_2 + \alpha_1 U_3 + \alpha_1 U_4)|\psi\rangle$ is shown in Fig. 2.

II. QUANTUM NATIVE ResNet

A. Variational quantum circuit model preliminaries

For a vector of classical input data $\mathbf{x} \in \mathbb{R}^d$, a standard quantum variational circuit model consists of an n-qubit encoding circuit $V(\mathbf{x}) \in SU(2^n)$ that encodes the classical data into a quantum state $|\psi_0\rangle \equiv |\phi(\mathbf{x})\rangle = V(\mathbf{x})|0\rangle^{\otimes n}$. This encoded state can also be represented as a density matrix, defined by

$$\rho(\mathbf{x}) = V(\mathbf{x})|0\rangle^{\otimes n}\langle 0|^{\otimes n} V(\mathbf{x})^{\dagger}. \tag{15}$$

Alternatively, $|\psi_0\rangle$ and $\rho = |\psi_0\rangle\langle\psi_0|$ could be quantum data in which no encoding process needs to be considered.

The input state $|\psi_0\rangle$ is then passed through L layers of variational quantum circuits $W_l(\theta_l) \in SU(2^n)$, where $\theta_l = (\theta_{l1}, \theta_{l2}, \ldots \theta_{lD_l})$ is a vector of variational parameters that

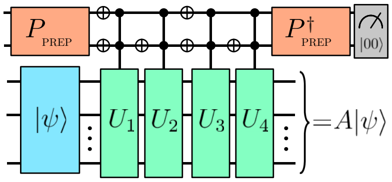

FIG. 2. A visual representation of the LCU procedure to implement $A|\psi\rangle = 1/\Omega'(\alpha_1 U_1 + \alpha_1 U_2 + \alpha_1 U_3 + \alpha_1 U_4)|\psi\rangle$. The upper two qubits are the ancilla qubits initially prepared by the P_{PREP} operator to be in the state $P_{\text{PREP}}|b_1\rangle = 1/\sqrt{\Omega}\sum_{j=1}^{4}\sqrt{\alpha_j}|b_j\rangle$. Note that in this case $|b_1\rangle \equiv |00\rangle$ and will implement U_1. $|b_2\rangle \equiv |01\rangle$ and will implement U_2. $|b_3\rangle \equiv |10\rangle$ and will implement U_3. $|b_4\rangle \equiv |11\rangle$ and will implement U_4. The inverse preparation $P^{\dagger}_{\text{PREP}}$ is then applied and the ancillas are measured. If the ancillas are measured to be in the state $|b_1\rangle \equiv |00\rangle$ then A is successfully implemented on the target state $|\psi\rangle$. This example demonstrates how multicontrol unitary gates that are controlled by the ancilla qubits and applied to the target register containing $|\psi\rangle$ can be implemented in practice.

can be adjusted as the model is trained. We can therefore represent the overall variational circuit as

$$W(\theta) = \prod_{l=1}^{L} W_l(\theta_l), \tag{16}$$

where θ is a vector containing all variational parameters for all layers such that $\theta = \{\theta_l\}, l \in [1, L]$. Finally, measurement is made of some Hermitian observable O. For simplicity, we will consider a loss function for the model, defined as

$$\mathcal{L}_\theta(\rho, O) \equiv \langle O \rangle_{\rho,\theta} = \mathrm{Tr}(W(\theta)\rho W(\theta)^\dagger O), \tag{17}$$

where any insights from this loss function with regard to barren plateaus can often be extended to other, more general, loss functions. Training a model consists of variationally adjusting the parameters θ until the cost function matches some known data labels y to within some acceptable error.

B. Quantum ResNet implementation

The key concept of the ResNet [11] is the introduction of residual blocks. Instead of learning a mapping $H(a)$ on some data, the model instead learns some residual function $W(a)$ such that

$$H(a) = a + W(a), \tag{18}$$

where a is the input to the block, $W(a)$ is the learned residual mapping, and $H(a)$ is the output of the block. This means that the output a_l of each layer l can be defined by

$$a_l = a_{l-1} + W_l(a_{l-1}, \theta_l), \tag{19}$$

where a_0 is the initial data input. In order to create a quantum version of this framework, we consider the quantum ResNet [26,27] that can be defined as

$$|\psi_l\rangle = |\psi_{l-1}\rangle + W_l(\theta_l)|\psi_{l-1}\rangle, \tag{20}$$

where $W_l(\theta_l)$ is the variational unitary for layer l, $|\psi_l\rangle$ is the output of the layer, $|\psi_{l-1}\rangle$ is the input of the layer, and $|\psi_0\rangle$ corresponds to the initial input state. This initial data-input state could be quantum data $|\psi_0\rangle$ or it could be a quantum state that encodes classical data $|\psi_0\rangle = |\phi(\mathbf{x})\rangle$, where $\mathbf{x} \in \mathbb{R}^d$ is a d-dimensional classical input data vector encoded into a quantum state through a data-encoding circuit $V(\mathbf{x})$ such that $|\phi(\mathbf{x})\rangle = V(\mathbf{x})|0\rangle^{\otimes n}$. Implementation of quantum ResNet of this form involves applying an operator

$$A_{\mathrm{RES}} = (I + W_l(\theta_l)) \tag{21}$$

to the state $|\psi_{l-1}\rangle$. This presents a problem, as the operator is not necessarily unitary in general for all $W_l(\theta_l) \in SU(2^n)$. Therefore, we proceed with implementing it via the LCU method.

To maintain the most general case, we shall consider adding some additional control into the magnitude of the skipped connections parametrized for each layer by some amount $\beta_l \in \mathbb{R}$. Therefore, the definition of a quantum residual connection that we shall use is

$$|\psi_l\rangle = \frac{1}{\Omega'}\left((1-\beta_l)|\psi_{l-1}\rangle + \beta_l W_l(\theta_l)|\psi_{l-1}\rangle\right), \tag{22}$$

where Ω' is a normalization constant.

This can be encoded with the LCU framework in a quantum circuit by applying the $P^{(l)}_{\mathrm{PREP}}$ single-qubit operator to the lth ancilla qubit denoted by $|0\rangle^{(l)}$, defined by

$$P^{(l)}_{\mathrm{PREP}} = (\sqrt{1-\beta_l}|0\rangle^{(l)} + \sqrt{\beta_l}|1\rangle^{(l)})\langle 0|^{(l)} + (\ldots)\langle 1|^{(l)}, \tag{23}$$

where terms involving $(\ldots)\langle 1|^{(l)}$ do not have an effect on the circuit and can be ignored. Hence we can write the action of preparing the lth ancilla qubit as

$$P^{(l)}_{\mathrm{PREP}}|0\rangle^{(l)} = (\sqrt{1-\beta_l}|0\rangle^{(l)} + \sqrt{\beta_l}|1\rangle^{(l)}), \tag{24}$$

such that $P_{\mathrm{PREP}}|0\rangle^{\otimes L} = \bigotimes_{l=1}^{L} P^{(l)}_{\mathrm{PREP}}|0\rangle^{(l)}$ and hence

$$P_{\mathrm{PREP}}|0\rangle^{\otimes L} = \bigotimes_{l=1}^{L}(\sqrt{1-\beta_l}|0\rangle^{(l)} + \sqrt{\beta_l}|1\rangle^{(l)}), \tag{25}$$

where for simplicity $\beta_l \in \mathbb{R}$. The selection operator is defined to be controlled by the lth ancilla qubit as

$$S^{(l)}_{\mathrm{SELECT}}|0\rangle^{(l)}|\psi\rangle = |0\rangle^{(l)}|\psi\rangle, \tag{26}$$

$$S^{(l)}_{\mathrm{SELECT}}|1\rangle^{(l)}|\psi\rangle = |1\rangle^{(l)}W_l(\theta_l)|\psi\rangle. \tag{27}$$

We can then define the overall selection operator by specifying the ordering

$$S_{\mathrm{SELECT}} = \prod_{l=1}^{L} S^{(L-l+1)}_{\mathrm{SELECT}} = S^{(L)}_{\mathrm{SELECT}} \cdots S^{(1)}_{\mathrm{SELECT}}, \tag{28}$$

such that $S^{(1)}_{\mathrm{SELECT}}$ is applied first.

Theorem 1 (Quantum ResNet). It is possible to probabilistically implement a quantum ResNet architecture on a quantum circuit where each layer of the variational circuit permits a residual connection, defined as

$$|\psi_l\rangle = \frac{1}{\Omega'}\left((1-\beta_l)|\psi_{l-1}\rangle + \beta_l W_l(\theta_l)|\psi_{l-1}\rangle\right),$$

where $W_l(\theta_l)$ is the lth layer of the variational circuit, the constants $\beta_l \in \mathbb{R}, \beta_l \in [0, 1]$ can be freely chosen, the input to the layer is the state $|\psi_{l-1}\rangle$, and the output of the layer is the state $|\psi_l\rangle$. The term Ω' is a normalization constant. The algorithm can be applied for all layers $l \in [1, L]$ simultaneously with a fixed probability of success

$$\pi_S = \prod_{l=1}^{L} \left(1 - 2\beta_l(1-\beta_l)\left(1 - \mathrm{Re}(\langle\psi_{l-1}|W_l(\theta_l)|\psi_{l-1}\rangle)\right)\right),$$

where $|\psi_0\rangle$ is an initial input state given to the algorithm, which can be quantum data or a quantum state encoded by classical data via an encoding circuit $V(\mathbf{x})$ such that $|\psi_0\rangle = |\phi(\mathbf{x})\rangle = V(\mathbf{x})|0\rangle^{\otimes n}$.

Proof. We use the preparation and selection operators defined in Eqs. (25) and (28), respectively, with the LCU framework as outlined in Sec. I D.

The P_{PREP} operator corresponds to performing single-qubit initialization on L qubits and is clearly implementable on a quantum device. Likewise, S_{SELECT} corresponds to a controlled $W_l(\theta_l)$ gate, where $W_l(\theta_l)$ is defined to be unitary, which is controlled by the lth qubit. As $W_l(\theta_l)$ is unitary, the S_{SELECT} operation can also be implemented on a quantum device.

We consider applying the LCU procedure to the first ancilla qubit only, starting with the first-qubit preparation operator:

$$P^{(1)}_{\text{PREP}}|0\rangle^{\otimes L}|\psi_0\rangle = (\sqrt{1-\beta_1}|0\rangle^{(1)} + \sqrt{\beta_1}|1\rangle^{(1)}) \times \bigotimes_{l=2}^{L}|0\rangle^{(l)}|\psi_0\rangle. \tag{29}$$

Subsequently, the selection operator that is controlled by the first qubit is applied:

$$\begin{aligned} S^{(1)}_{\text{SELECT}}P^{(1)}_{\text{PREP}}|0\rangle^{\otimes L}|\psi_0\rangle &= S^{(1)}_{\text{SELECT}}(\sqrt{1-\beta_1}|0\rangle^{(1)} + \sqrt{\beta_1}|1\rangle^{(1)})\bigotimes_{l=2}^{L}|0\rangle^{(l)}|\psi_0\rangle \\ &= \sqrt{1-\beta_1}|0\rangle^{(1)}\bigotimes_{l=2}^{L}|0\rangle^{(l)}|\psi_0\rangle + \sqrt{\beta_1}|1\rangle^{(1)} \\ &\quad \times \bigotimes_{l=2}^{L}|0\rangle^{(l)}W_1(\theta_1)|\psi_0\rangle. \end{aligned} \tag{30}$$

Consider the conjugate preparation operator,

$$(P^{(1)}_{\text{PREP}})^{\dagger} = |0\rangle^{(1)}(\sqrt{1-\beta_1}\langle 0|^{(1)} + \sqrt{\beta_1}\langle 1|^{(1)}) + |1\rangle^{(1)}(\ldots), \tag{31}$$

where $|1\rangle^{(1)}(\ldots)$ collects terms that will not be used, as we will require the ancilla to be in the $|0\rangle^{(1)}$ state. Applying this conjugate operator, we can write the state as

$$\begin{aligned} &(P^{(1)}_{\text{PREP}})^{\dagger}S^{(1)}_{\text{SELECT}}P^{(1)}_{\text{PREP}}|0\rangle^{\otimes L}|\psi_0\rangle \\ &\quad = |0\rangle^{(1)}\left((1-\beta_1)\bigotimes_{l=2}^{L}|0\rangle^{(l)}|\psi_0\rangle \right. \\ &\qquad \left. + \beta_1\bigotimes_{l=2}^{L}|0\rangle^{(l)}W_1(\theta_1)|\psi_0\rangle\right) + |1\rangle^{(1)}(\ldots), \end{aligned} \tag{32}$$

where $|1\rangle^{(1)}(\ldots)$ collects terms that will be discarded and can therefore be ignored. The first ancilla qubit is now measured and the result is discarded unless it is found to be in the $|0\rangle^{(1)}$ state. The probability of measuring the ancilla in the $|0\rangle^{(1)}$ state will be equal to

$$\begin{aligned} \pi_1 &= |(1-\beta_1)|\psi_0\rangle + \beta_1 W_1(\theta_1)|\psi_0\rangle)|^2 \\ &= 1 - 2\beta_1(1-\beta_1)\left(1 - \mathrm{Re}(\langle\psi_0|W_1(\theta_1)|\psi_0\rangle)\right), \end{aligned} \tag{33}$$

which will depend on the strength of the residual connection parametrized by β_1 as well as the real component of $\langle\psi_0|W_1(\theta_1)|\psi_0\rangle$, which will depend on the quantum encoded state $|\psi_0\rangle$ and the variational operator used $W_1(\theta_1)$.

If we succeed in finding the first ancilla in the $|0\rangle^{(1)}$ state, then the target state will now be

$$\begin{aligned} &\langle 0|^{(1)}(P^{(1)}_{\text{PREP}})^{\dagger}S^{(1)}_{\text{SELECT}}P^{(1)}_{\text{PREP}}|0\rangle^{\otimes L}|\psi_0\rangle \\ &= \frac{1}{\sqrt{\pi_1}}\left((1-\beta_1)\bigotimes_{l=2}^{L}|0\rangle^{(l)}|\psi_0\rangle + \beta_1\bigotimes_{l=2}^{L}|0\rangle^{(l)}W_1(\theta_1)|\psi_0\rangle\right) \end{aligned} \tag{34}$$

and hence ignoring the extra ancilla qubits, we have

$$|\psi_1\rangle = \frac{1}{\Omega'_1}\left((1-\beta_1)|\psi_0\rangle + \beta_1 W_1(\theta_1)|\psi_0\rangle\right), \tag{35}$$

where $\Omega'_1 = \sqrt{\pi_1}$ is the required normalization constant. Hence, for layer $l = 1$, we have shown that the quantum residual connection defined in Eq. (22) is implemented for the original input state $|\psi_0\rangle$.

Now assume that $|\psi_{p-1}\rangle$ is correctly implemented by the procedure and consider the subsequent implementation of $|\psi_p\rangle$. We prepare the pth ancilla using

$$\begin{aligned} &P^{(p)}_{\text{PREP}}|0\rangle^{\otimes L-p}|\psi_{p-1}\rangle \\ &\quad = \left(\sqrt{1-\beta_p}|0\rangle^{(p)} + \sqrt{\beta_p}|1\rangle^{(p)}\right)\bigotimes_{l=p+1}^{L}|0\rangle^{(l)}|\psi_{p-1}\rangle. \end{aligned} \tag{36}$$

Through the same procedure as previously, we see that

$$\begin{aligned}&S^{(p)}_{\text{SELECT}}P^{(p)}_{\text{PREP}}|0\rangle^{\otimes L-p}|\psi_{p-1}\rangle\\&=\sqrt{1-\beta_p}|0\rangle^{(p)}\bigotimes_{l=p+1}^{L}|0\rangle^{(l)}|\psi_{p-1}\rangle\\&\quad+\sqrt{\beta_p}|1\rangle^{(p)}\bigotimes_{l=p+1}^{L}|0\rangle^{(l)}W_p(\theta_p)|\psi_{p-1}\rangle.\end{aligned}\tag{37}$$

Applying the conjugate preparation operator for the pth qubit, which may be written as

$$\begin{aligned}(P^{(p)}_{\text{PREP}})^{\dagger}=|0\rangle^{(p)}(\sqrt{1-\beta_p}\langle 0|^{(p)}+\sqrt{\beta_p}\langle 1|^{(p)})\\+|1\rangle^{(p)}(\ldots),\end{aligned}\tag{38}$$

to the ancillas, we see that

$$\begin{aligned}&(P^{(p)}_{\text{PREP}})^{\dagger}S^{(p)}_{\text{SELECT}}P^{(p)}_{\text{PREP}}|0\rangle^{\otimes L-p}|\psi_0\rangle\\&=(1-\beta_p)|0\rangle^{(p)}\bigotimes_{l=p+1}^{L}|0\rangle^{(l)}|\psi_{p-1}\rangle\\&\quad+\beta_p|0\rangle^{(p)}\bigotimes_{l=p+1}^{L}|0\rangle^{(l)}W_p(\theta_p)|\psi_{p-1}\rangle+|1\rangle^{(p)}(\ldots),\end{aligned}\tag{39}$$

where $|1\rangle^{(p)}(\ldots)$ collects terms that will end up being discarded and hence can be ignored. Measuring the ancilla qubit, we can see that the probability of measuring the pth qubit in the $|0\rangle^{(p)}$ state is given by

$$\begin{aligned}\pi_p&=\left|((1-\beta_p)|\psi_{p-1}\rangle+\beta_p W_p(\theta_p)|\psi_{p-1}\rangle)\right|^2\\&=1-2\beta_p(1-\beta_p)\left(1-\text{Re}(\langle\psi_{p-1}|W_p(\theta_p)|\psi_{p-1}\rangle)\right).\end{aligned}\tag{40}$$

Ensuring that the process is only continued if the ancilla is in the $|0\rangle^{(p)}$ state, we find

$$\begin{aligned}&\langle 0|^{(p)}(P^{(p)}_{\text{PREP}})^{\dagger}S^{(p)}_{\text{SELECT}}P^{(p)}_{\text{PREP}}|0\rangle^{\otimes L-p}|\psi_0\rangle\\&=\frac{1}{\sqrt{\pi_p}}\left((1-\beta_p)\bigotimes_{l=p+1}^{L}|0\rangle^{(l)}|\psi_{p-1}\rangle\right.\\&\quad\left.+\beta_p\bigotimes_{l=p+1}^{L}|0\rangle^{(l)}W_p(\theta_p)|\psi_{p-1}\rangle\right).\end{aligned}\tag{41}$$

Ignoring the unused ancillas, we can therefore write the output of layer $l=p$ as

$$|\psi_p\rangle=\frac{1}{\Omega'_p}\left((1-\beta_p)|\psi_{p-1}\rangle+\beta_p W_p(\theta_p)|\psi_{p-1}\rangle\right),\tag{42}$$

where $\Omega'_p=\sqrt{\pi_p}$ is a normalization constant. Hence, assuming that the state $|\psi_{p-1}\rangle$ is correctly implemented by the procedure, we have shown that the output state $|\psi_p\rangle$ will be correctly implemented according to the quantum residual connection defined in Eq. (22). As we have shown that $|\psi_1\rangle$ can be implemented correctly from the data-input state $|\psi_0\rangle$, then by inductive reasoning, $|\psi_l\rangle$ is correctly implemented for all $l\in[L]$.

The overall probability of success π_S will rely on all L qubits being successfully measured in the $|0\rangle^{(l)}$ state such that $\pi_S=\prod_{l=1}^{L}\pi_l$ and hence

$$\pi_S=\prod_{l=1}^{L}\left(1-2\beta_l(1-\beta_l)\left(1-\text{Re}(\langle\psi_{l-1}|W_l(\theta_l)|\psi_{l-1}\rangle)\right)\right),\tag{43}$$

as required. ■

We therefore show that the most general form of ResNet, in which connections can be skipped at every layer in a variational quantum circuit, is probabilistically implementable. An example circuit architecture that demonstrates the process in the first two layers is shown in Fig. 3. This implementation requires the number of ancilla qubits to equal L. For a more qubit-efficient but less general version, see Appendix G, in which the number of ancilla used scales as $\mathcal{O}(\log(L))$.

C. Probabilistic scaling

During the proof in the previous section, we found in Eq. (40) that the probability π_l of measuring the lth qubit in the $|0\rangle^{(l)}$ state, which is required for implementing the LCU method, is

$$\pi_l=1-2\beta_l(1-\beta_l)\left(1-\text{Re}(\langle\psi_{l-1}|W_l(\theta_l)|\psi_{l-1}\rangle)\right).\tag{44}$$

This depends on the strength of the residual connection β_l, the quantum state of the previous step $|\psi_{l-1}\rangle$, and the variational circuit $W_l(\theta_l)$.

Note that, in general, $\text{Re}(\langle\psi_{l-1}|W_l(\theta_l)|\psi_{l-1}\rangle)$ lies in the range $[-1,1]$. Therefore, the probability of success lies in the range $[1-4\beta_l(1-\beta_l),1]$. The worst-case scenario therefore occurs when $\beta_l=\frac{1}{2}$, as in this case the probability of success is within the range $[0,1]$ and therefore there is a chance that the algorithm cannot be implemented. However, as β_l is varied to be closer to 0 or 1, this gives an increasing lower bound of $1-4\beta_l(1-\beta_l)$ for the probability of success. (If $\beta_l=1$, the probability of success is equal to 1, but this would trivially mean that the skip connection is simply not performed. Likewise, if $\beta_l=0$, then the success probability is equal to 1 but corresponds to not implementing the variational circuit $W_l(\theta_l)$ at all.) This means that the lower bound of the probability of success varies with the strength of the residual connection for the layer, as shown in Fig. 4.

The term $\mathrm{Re}(\langle\psi_{l-1}|W_l(\theta_l)|\psi_{l-1}\rangle)$ determines whether we will be close to the lower-bound probability $1-4\beta_l(1-\beta_l)$ or the upper bound of 1. We see that if $W_l(\theta_l)=\mathbb{I}$, then we reach the upper bound and if $W_l(\theta_l)=-\mathbb{I}$, we reach the lower bound. If restrictions were placed on the form that $W_l(\theta_l)$ can take, there would be potential to increase the lower-bound probability even further.

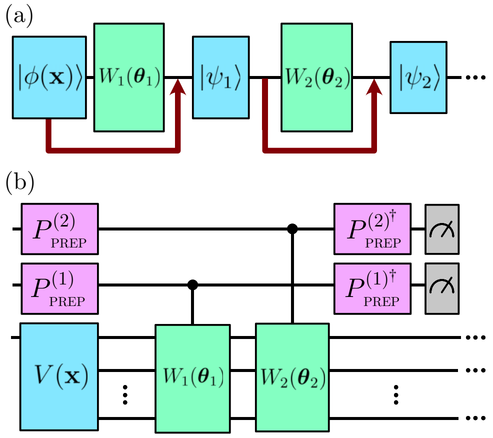

FIG. 3. (a) A conceptual illustration of classical ResNet, in which residual information can be skipped over each layer. (b) A quantum ResNet implementation using LCU methods, showing the first two layers only. In this case, a quantum analogue of ResNet can be probabilistically implemented using ancilla qubits. The $P^{(1)}_{\mathrm{PREP}}$ prepares the first ancilla qubit into the state $(\sqrt{1-\beta_1}|0\rangle^{(1)}+\sqrt{\beta_1}|1\rangle^{(1)})$; when this ancilla is measured at the end in the $|0\rangle^{(1)}$ state, it corresponds to applying the operator $(1-\beta_1)\mathbb{I}+\beta_1 W_1(\theta_1)$ to the initial input state $|\phi(\mathbf{x})\rangle = V(\mathbf{x})|0\rangle^{\otimes n}$. The process is repeated for future layers, allowing a portion of the quantum state to skip each layer and be added to the output of the layer.

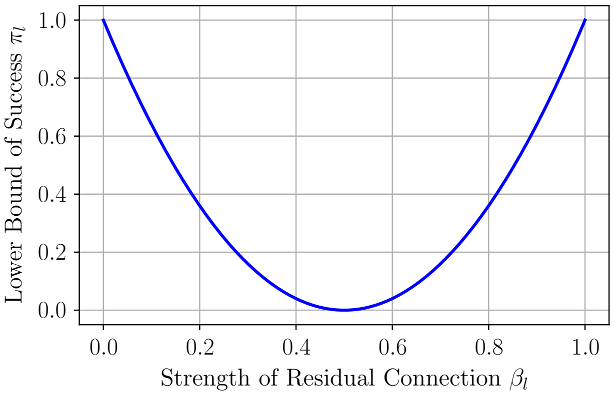

FIG. 4. The lower bound of the success probability π_l for layer l in the LCU quantum native ResNet, as the strength of residual-connection parameter β_l varies. The lower bound is equal to $1-4\beta_l(1-\beta_l)$.

As shown previously, the overall probability of success π_S will rely on all L qubits being successfully measured in the $|0\rangle$ state such that $\pi_S=\prod_{l=1}^{L}\pi_l$ and hence

$$\pi_S=\prod_{l=1}^{L}\left(1-2\beta_l(1-\beta_l)\left(1-\mathrm{Re}(\langle\psi_{l-1}|W_l(\theta_l)|\psi_{l-1}\rangle)\right)\right). \tag{45}$$

Therefore, the lower bound of the overall probability of success is given by $\prod_l^L(1-4\beta_l(1-\beta_l))$, which can be adjusted to be between 0 and 1 by varying the strength of the residual layers through the variables β_l. Although we highlight that this cannot be arbitrarily adjusted, as if β_l reaches 0 or 1, this would trivially correspond to no residual connection at all and, more generally, the value of β_l may be decided by alternative factors for a particular model or data set.

If, for a given architecture, the probability of success for any layer is bounded in the range $[\pi_{\mathrm{low}},\pi_{\mathrm{high}}]$, then the probability of overall success must be bounded in the range $[\pi^L_{\mathrm{low}},\pi^L_{\mathrm{high}}]$. In the case of $\pi_{\mathrm{low}},\pi_{\mathrm{high}}<1$, this means that the probability will decay exponentially with L, which is a key drawback of this method. However, if the number of layers used is chosen to scale logarithmically with the number of qubits $L=\mathcal{O}(\log(n))$, then the algorithm can still run in time $\mathcal{O}(\mathrm{poly}(n))$ in general, where n is the number of qubits for the target state register that initially contains $|\psi_0\rangle$. Note that when $L=1$, we have a purely unitary model. Therefore, for $L>1$, we will have access to models that are at least as expressive as a standard unitary VQC model, with the potential to be even more expressive. Hence, even setting $L=\mathcal{O}(1), L>1$ would still allow a greater variation in expressivity of the models compared to the standard unitary case. Furthermore, as discussed previously, the lower bounds can be adjusted by varying the strength of the residual connections for a given architecture.

D. Avoiding barren plateaus with residual connections

Several works have been undertaken to characterize the conditions under which VQC models exhibit barren plateaus [21]. A model is said to contain barren plateaus if $\forall\theta_\mu\in\theta$ the following condition holds [50]:

$$\mathrm{Var}_\theta\left(\frac{\partial\langle O\rangle_{\rho,\theta}}{\partial\theta_\mu}\right)\le F(n),\quad F(n)=\mathcal{O}\left(\frac{1}{b^n}\right),\quad b>1, \tag{46}$$

where Var_θ is the variance calculated over a uniform distribution of parameters θ. If the model exhibits exponential gradient concentration, then this makes training models difficult. The structure of the variational *Ansatz* plays an important role in determining this value, as for a wide range of cases the loss-function gradient variance will decay exponentially if the dynamical Lie algebra (DLA) of the variational circuit generators is exponential in dimension [51,52]. Note that an important assumption in these characterizations is usually that the circuits are sufficiently deep to form approximate designs, meaning that the depth of the circuit is also an important consideration regarding the presence of barren plateaus [52]. Previous work has shown that using an ensemble of several shallow-depth models can avoid loss-function concentration [53]. We show that the quantum ResNet provides a method of creating exponentially large ensembles of unitary VQC models, in which shallower-depth components can help avoid barren plateaus, while also providing additional nonunitary contributions to the variance. However, if these shallower layers can be efficiently simulated classically and the nonunitary terms exponentially vanish, then the entire model may be at risk of being classically simulated. This finding suggests the importance of further exploring the nonunitary components of quantum ResNet models.

1. Avoiding barren plateaus

By introducing residual connections between layers of the *Ansatz*, it is possible to effectively mitigate barren plateaus in the model. This is similar to the classical role of ResNet in which vanishing gradients are mitigated in deep networks by allowing the gradients to skip over portions of the overall network [11,12]. For example, take the case in which a variational *Ansatz* can be separated into two separate sections, $W_1(\theta_1)$ followed by $W_2(\theta_2)$. Furthermore, assume that when combined in series, the overall operator $W_2(\theta_2)W_1(\theta_1)$ exhibits barren plateaus. Consider a residual connection introduced between the input and the output of the first layer $W_1(\theta_1)$ with residual-connection strength β, and no residual connection over the second layer $W_2(\theta_2)$. Starting with an initial state $|\psi_0\rangle$, we see that the states produced in this circuit are given by

$$|\psi_1\rangle = \frac{1}{\Omega'}((1-\beta)|\psi_0\rangle + \beta W_1(\theta_1)|\psi_0\rangle), \tag{47}$$

$$|\psi_2\rangle = W_2(\theta_2)|\psi_1\rangle, \tag{48}$$

where Ω' is a normalization constant. It follows that the final state can be written as

$$|\psi_2\rangle = \frac{1}{\Omega'}((1-\beta)W_2(\theta_2)|\psi_0\rangle + \beta W_2(\theta_2)W_1(\theta_1)|\psi_0\rangle). \tag{49}$$

This corresponds to applying the operator

$$A(\theta)_{\text{RES}} = ((1-\beta)W_2(\theta_2) + \beta W_2(\theta_2)W_1(\theta_1)) \tag{50}$$

and normalizing the resultant state by a factor $1/\Omega'$. We now consider some quantum encoded density matrix state ρ, where $\rho \equiv |\psi_0\rangle\langle\psi_0|$, which is evolved by this variational operator $A(\theta)_{\text{RES}}$. The resulting loss function for the model with respect to a measurement operator O is defined by

$$\langle O\rangle_{\rho,\theta} = \text{Tr}(A(\theta)_{\text{RES}}\rho A(\theta)^{\dagger}_{\text{RES}}O). \tag{51}$$

While $A(\theta)_{\text{RES}}$ is not in general unitary, we can decompose its corresponding loss function into unitary evolution-type terms and nonunitary evolution terms by considering the following expansion:

$$\begin{aligned}\langle O\rangle_{\rho,\theta} = \frac{1}{\Omega'^2}\big(&(1-\beta)^2\text{Tr}(W_2(\theta_2)\rho W_2(\theta_2)^{\dagger}O)\\ &+\beta^2\text{Tr}(W_2(\theta_2)W_1(\theta_1)\rho W_1(\theta_1)^{\dagger}W_2(\theta_2)^{\dagger}O)\\ &+\beta(1-\beta)\text{Tr}(W_2(\theta_2)\rho W_1(\theta_1)^{\dagger}W_2(\theta_2)^{\dagger}O)\\ &+\beta(1-\beta)\text{Tr}(W_2(\theta_2)W_1(\theta_1)\rho W_2(\theta_2)^{\dagger}O)\big),\end{aligned} \tag{52}$$

where we can collect the nonunitary terms together by representing the expression as

$$\begin{aligned}\langle O\rangle_{\rho,\theta} = \frac{1}{\Omega'^2}\big(&(1-\beta)^2\text{Tr}(W_2(\theta_2)\rho W_2(\theta_2)^{\dagger}O)\\ &+\beta^2\text{Tr}(W_2(\theta_2)W_1(\theta_1)\rho W_1(\theta_1)^{\dagger}W_2(\theta_2)^{\dagger}O)\\ &+\beta(1-\beta)\text{Tr}(W_2(\theta_2)\tilde{\rho}_{\theta_1}W_2(\theta_2)^{\dagger}O),\end{aligned} \tag{53}$$

where $\tilde{\rho}_{\theta_1} = \rho W_1(\theta_1)^{\dagger} + W_1(\theta_1)\rho$. It is now possible to identify three distinct terms in the loss function.

Deep—$L_{\text{BP}} = \text{Tr}(W_2(\theta_2)W_1(\theta_1)\rho W_1(\theta_1)^{\dagger}W_2(\theta_2)^{\dagger}O)$. This component is identical to the unitary VQC loss function if no residual component had been implemented. It corresponds to a deep circuit consisting of $W_1(\theta_1)$ followed by $W_2(\theta_2)$ in series, which we will assume exhibits barren plateaus.

Shallow—$L_{\text{No-BP}} = \text{Tr}(W_2(\theta_2)\rho W_2(\theta_2)^{\dagger}O)$. This component is identical to the unitary VQC loss function if the variational circuit consisted of $W_2(\theta_2)$ only. We will assume that this model does not exhibit barren plateaus, as has been shown to be the case for many variational circuits [42,43,54,55], but may be vulnerable to being classically simulated [29].

Nonunitary—$L_{\tilde{\rho}} = \text{Tr}(W_2(\theta_2)\tilde{\rho}_{\theta_1}W_2(\theta_2)^{\dagger}O)$. This component contains the nonunitary element of the loss function. It appears similar to the $L_{\text{No-BP}}$ term, with the key difference that instead of ρ being unitary evolved, we have $\tilde{\rho}_{\theta_1} = \rho W_1(\theta_1)^{\dagger} + W_1(\theta_1)\rho$. While $\tilde{\rho}_{\theta_1}$ is Hermitian, it does not necessarily have a trace equal to 1; nor is it positive semidefinite in general.

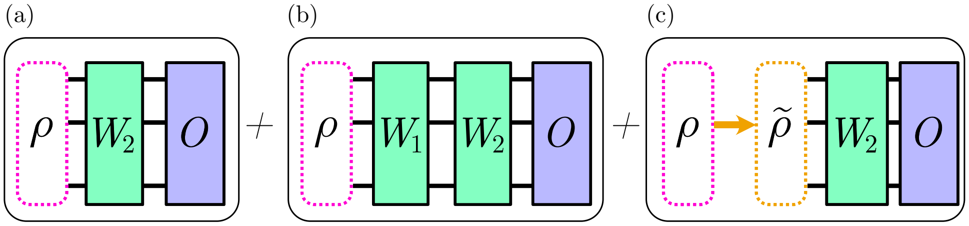

FIG. 5. Visualization of the single residual connection $L = 1$ quantum ResNet loss function $\langle O\rangle_{\rho,\theta}$ when decomposed into the individual terms corresponding to (a) the shallow component $L_{\text{No-BP}}$, (b) the deep component L_{BP}, and (c) the nonunitary component $L_{\tilde{\rho}}$. In (c), the arrow corresponds to the transformation $\rho \rightarrow \tilde{\rho} = \rho W_1^\dagger + W_1\rho$.

Setting $\beta = \frac{1}{2}$ and absorbing this factor along with the normalization constant into the loss definition, we can write

$$\langle O\rangle_{\rho,\theta} = L_{\text{No-BP}} + L_{\text{BP}} + L_{\tilde{\rho}}. \tag{54}$$

A visual interpretation of this resulting quantum ResNet model is shown in Fig. 5. We can therefore find the gradient of this loss function with respect to parameters $\theta_\mu \in \theta$ as

$$\frac{\partial\langle O\rangle_{\rho,\theta}}{\partial\theta_\mu} = \frac{\partial L_{\text{No-BP}}}{\partial\theta_\mu} + \frac{\partial L_{\text{BP}}}{\partial\theta_\mu} + \frac{\partial L_{\tilde{\rho}}}{\partial\theta_\mu}. \tag{55}$$

The resulting gradient-variance term can then be written as

$$\begin{aligned}
&\text{Var}_\theta\left(\frac{\partial\langle O\rangle_{\rho,\theta}}{\partial\theta_\mu}\right)\\
&\quad= \text{Var}_\theta\left(\frac{\partial L_{\text{No-BP}}}{\partial\theta_\mu}\right)\\
&\quad+ \text{Var}_\theta\left(\frac{\partial L_{\text{BP}}}{\partial\theta_\mu}\right) + \text{Var}_\theta\left(\frac{\partial L_{\tilde{\rho}}}{\partial\theta_\mu}\right)\\
&\quad+ \text{Cov}_\theta\left(\frac{\partial L_{\text{No-BP}}}{\partial\theta_\mu}, \frac{\partial L_{\text{BP}}}{\partial\theta_\mu}\right) + \text{Cov}_\theta\left(\frac{\partial L_{\text{No-BP}}}{\partial\theta_\mu}, \frac{\partial L_{\tilde{\rho}}}{\partial\theta_\mu}\right)\\
&\quad+ \text{Cov}_\theta\left(\frac{\partial L_{\text{BP}}}{\partial\theta_\mu}, \frac{\partial L_{\tilde{\rho}}}{\partial\theta_\mu}\right).
\end{aligned} \tag{56}$$

By construction, we know that $L_{\text{No-BP}}$ does not lead to barren plateaus and we can assume that $\text{Var}_\theta(\partial L_{\text{No-BP}}/\partial\theta_\mu) \sim 1/\text{poly}(n)$ does not decay exponentially. Hence the overall quantum ResNet model has gradient variance $\text{Var}_\theta\left(\partial\langle O\rangle_{\rho,\theta}/\partial\theta_\mu\right) \sim 1/\text{poly}(n)$ and therefore does not exhibit barren plateaus. The shallow-depth component has mitigated barren plateaus in the overall model. This is demonstrated experimentally in Fig. 6, which shows that the gradient variance $\text{Var}_\theta\left(\partial\langle O\rangle_{\rho,\theta}/\partial\theta_\mu\right)$ for the overall quantum ResNet model decays subexponentially.

2. Classical simulation

It has recently been reported that a broad class of VQC models that avoid barren plateaus are also classically simulatable [29], limiting their potential for quantum advantage. While quantum residual connections offer a means to circumvent barren plateaus, they may still lead to scenarios in which the model is well approximated by the shallow component of the loss function $L_{\text{No-BP}}$, which

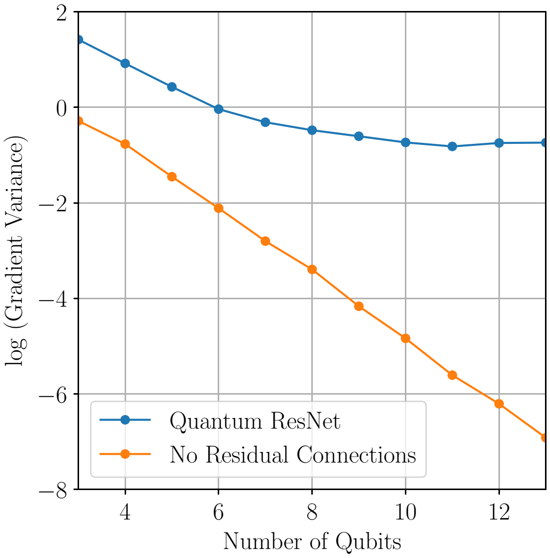

FIG. 6. The logarithm of the gradient variance as the number of qubits is increased. (Quantum ResNet) The model in which one residual layer is implemented between the input and output of the W_1 portion of the BP *Ansatz*, which consists of W_2W_1. (No Residual Connections) The model with no residual connections is an *Ansatz* that is known to exhibit barren plateaus and corresponds to the operator W_2W_1. In this example, W_1 has been generated by $[XY, YX, YZ]$, leading to an exponential DLA dimension [54], while W_2 has been generated by $[XY]$, leading to a polynomial DLA dimension [54]. Both W_1 and W_2 consist of 25 repeated sublayers of parametrized gates. Each sublayer consists of parametrized two-qubit gates that act on all adjacent qubits of the form $\prod_{k=1}((\bigotimes_{j=\text{odd}} e^{i\theta_j H^{(k)}_{j,j+1}})(\bigotimes_{j'=\text{even}} e^{i\theta_{j'} H^{(k)}_{j',j'+1}}))$, where $H^{(k)}_{j,j+1}$ is the kth element in the specified set of generators that acts on the qubits indexed by j and $j+1$. The variational parameters $\theta \in \theta$ have been sampled over the range $\theta \in [0, 2\pi]$ and 1000 samples have been taken in total. The initial state $\rho_0 = |0\rangle^{\otimes n}\langle 0|^{\otimes n}$ and the final observable is $O = Y \otimes Y \otimes \mathbb{I}^{n-2}$.

may be classically simulatable [29]. By construction, we expect the L_{BP} to exponentially vanish and if the same happens to the $L_{\tilde{\rho}}$ term, then this implies that the entire quantum ResNet model could be approximated efficiently on a classical computer within a given error. This raises the question: Are quantum ResNet models that avoid barren plateaus classically simulatable in general? If we take the assumption that L_{BP} exponentially vanishes and $L_{\text{No-BP}}$ is classically simulatable, this means that the answer depends on the characteristics of the nonunitary term $L_{\tilde{\rho}}$. If the $L_{\tilde{\rho}}$ term does not decay exponentially and cannot be classically simulated, then this condition could lead to a situation in which a model with at least one residual connection could be difficult to simulate and simultaneously avoid barren plateaus. This nonunitary term has not yet been characterized in general and therefore highlights a possible search space for future research; however, we note that this may not necessarily be a common situation. We show in Fig. 7 that for the particular quantum ResNet architecture that we consider in our numerics, $L_{\tilde{\rho}}$ exponentially decays with an increasing number of qubits, which suggests that this specific architecture may be vulnerable to classical simulation. However, we only consider a limited example and note that there is a large space that remains unknown and is open to further work. Generalized quantum operations of the form $\sum_i \alpha_i U_i$, which are implemented by the LCU procedure, have been shown to form a convex set and the extreme points of this set are the unitary operations [56]. This means that the space of operators that can be considered is much larger than that which has previously been studied in the literature, and a full characterization of this general space is beyond the scope of this current work.

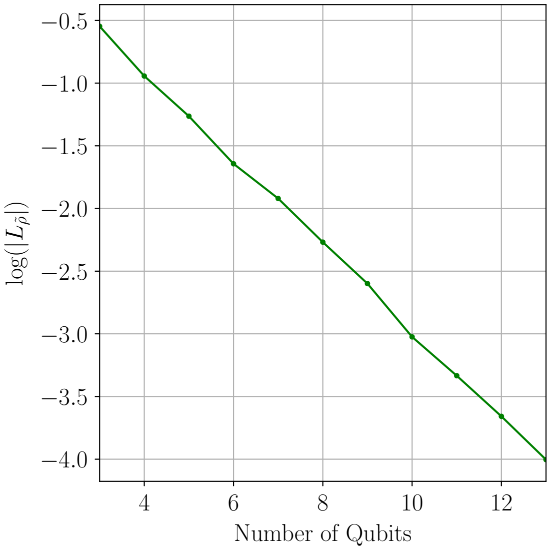

FIG. 7. The logarithm of the average magnitude of the nonunitary term $L_{\tilde{\rho}}$. The average has been performed over 1000 initializations. The data are from the same experiment as specified in Fig. 6. In this particular setting, the nonunitary contribution vanishes exponentially, suggesting that it may be possible to approximate the model by $L_{\text{BP}} + L_{\text{No-BP}}$. If the $L_{\text{No-BP}}$ loss also vanishes exponentially, then the entire quantum ResNet could be approximated by L_{BP}.

3. Quantum ResNet as an ensemble

In the following subsection, we use the term “layer” to refer to quantum ResNet layers, such that L corresponds to the number of residual connections and hence the number of controlled unitaries in the model. Each individual controlled unitary W_l is a variational *Ansatz*, which in many cases is composed of repeated layers of a certain gate architecture; to avoid confusion, we will refer to the repeated layers within these unitary W_l terms as sublayers. We will now denote the unitary in the lth residual network layer as $W_l^{(\lambda)}$, where λ specifies the number of repeated sublayers in the *Ansatz* corresponding to that particular unitary.

Previous work on the characterization of barren plateaus makes assumptions that the circuits involved are sufficiently deep [51,52]. As the quantum ResNet allows layer skipping, this means that there may always be shallow-depth circuit components in the final loss function, which will not satisfy this assumption. We now consider a deep circuit composed of many repeated sublayers of the same structure of variational gates. We show that it is possible with L layers of quantum ResNet to effectively implement an ensemble of 2^L different *Ansätze* in which the number of sublayers of the individual *Ansatz* terms in the ensemble ranges from 1 to 2^L.

The uniform-ensemble quantum ResNet architecture is created using L quantum residual layers by ensuring that the number of repeated sublayers λ in the lth unitary $W_l^{(\lambda)}$ is given by $\lambda = 2^{l-1}$ for $1 \leq l \leq L$. To prevent a constant term in the loss function arising from the identity, we also initially apply a unitary operator consisting of a single sublayer $\lambda = 1$, without using a residual connection, which we denote as $W_0^{(1)}$. This corresponds to applying the operator

$$(\mathbb{I} + W_L^{(2^{L-1})}) \dots (\mathbb{I} + W_3^{(4)})(\mathbb{I} + W_2^{(2)})(\mathbb{I} + W_1^{(1)})W_0^{(1)} \tag{57}$$

and normalizing the resultant state by a factor $1/\Omega'$. This term expands out to give an operation consisting of a linear combination of 2^L terms. When unitaries multiply to create these terms, the number of sublayers add together. It is therefore clear that this procedure results in an operator

$$\sum_{\lambda'=1}^{2^L} V^{(\lambda')}, \tag{58}$$

where $V^{(\lambda')}$ is a unitary corresponding to

$$V^{(\lambda')} = \left(\prod_{k \in S_{\lambda'}} W_k^{(2^{k-1})} \right) W_0^{(1)}, \tag{59}$$

and where $S_{\lambda'} \subseteq [L, 1]$ is the set that indexes the $W_k^{(2^{k-1})}$ terms that multiply together to give a total number of sublayers $\lambda' = 1 + \sum_{k \in S_{\lambda'}} 2^{k-1}$. The state is also normalized by a factor $1/\Omega''$ after the operation is applied. In total, we have 2^L terms where the number of sublayers in each term is in the range $\lambda' \in [1, 2^L]$. This will lead to a loss function with 2^L unitary VQC model loss functions, covering sublayer depths from 1 to 2^L, in addition to nonunitary cross terms.

Many investigations into barren plateaus specifically require circuits to be of a certain depth, e.g., requiring sufficient depth to form an approximate design over a Lie group [52]. While bounds have been found for this depth for certain circuits, the quantum ResNet would allow an agnostic approach in which many different layers can be trialed in the same model run and in which shallow-circuit components could guarantee that the overall model does not exhibit barren plateaus. We see that for L quantum ResNet layers, one can implement 2^L different unitary terms in an ensemble, in addition to further nonunitary cross terms.

Loss-function concentration implies loss-function gradient concentration and hence this metric can be similarly used to discuss the presence of barren plateaus [57]. An ensemble sum of all terms with differing numbers of sublayers may be able to avoid barren plateaus due to the presence of shallower-depth components within the average ensemble, as shown in Fig. 8. This reaffirms previous results showing that quantum ensemble methods can avoid barren plateaus [53]. However, the quantum ResNet will also provide additional sources of variance from the nonunitary terms in the loss function, although a detailed characterization of these terms is beyond the scope of this work.

Note that as the number of quantum ResNet layers L increases, the number of terms grows exponentially, meaning that any individual term will have a weighting that decays exponentially due to the normalization of the state after the operation is applied. Therefore, if L is increased, one would need to ensure that a sufficient amount of nonexponentially concentrating terms survive to still avoid barren plateaus. By initializing the ancilla qubits, one can vary the weightings of particular terms in the ensemble. This flexibility could allow the creation of models that can be pushed close to the boundary of barren plateaus but still remain in the trainable region. Similarly to the arguments made for the previous model, if the nonunitary terms vanish, then we would expect the loss function to be on average well approximated by the shallow nonvanishing terms, which do not exhibit barren plateaus individually (excluding rare potential configurations in which deep terms still provide nonzero contributions). If these shallow terms turn out to be classically simulatable [29], then this means that the overall quantum ResNet will be classically simulatable to within some given error. Efforts to avoid this situation should focus on finding settings in which the nonunitary terms do not exponentially vanish; it remains an open question whether this is possible.

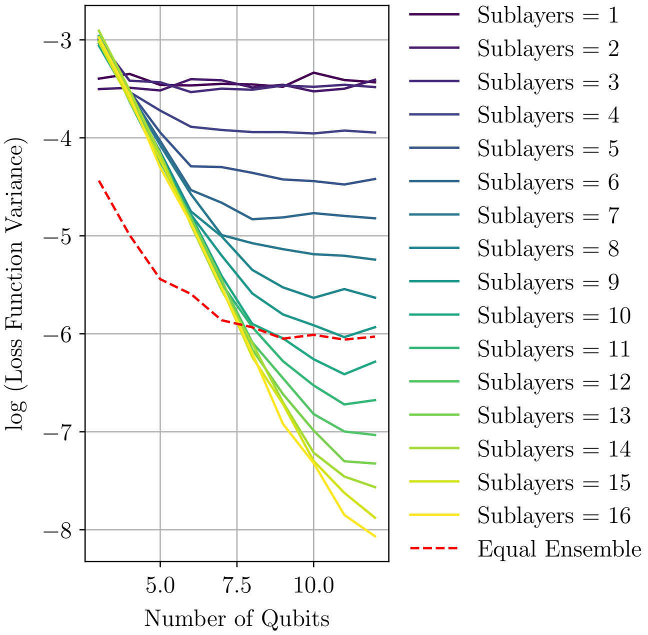

FIG. 8. The logarithm of the variance of the loss function for differing numbers of sublayers of the same *Ansatz* architecture as the number of qubits increases. The variance has been calculated over 500 random initializations of the variational parameters. The *Ansatz* has been created using the generators $[XY, YX, YZ]$ as specified in Fig. 6. The dashed red line corresponds to the variance of a normalized average of all the loss functions.

Previously, we have shown in Eq. (40) that the strength of the residual connection determines a lower bound of the probability of success of a given layer. We have also noted that the probability of overall success decays exponentially in the number of residual connections L. In Fig. 9, we show the impact of these effects on the expected number of repeated attempts required to achieve a successful LCU procedure. While the success probability decays exponentially in L, it allows the construction of ensembles of size 2^L, which therefore leads to a linear relationship between expected attempts and ensemble size.

We also note that in the case in which a given residual layer provides an exponentially vanishing contribution to the overall loss, from both its unitary and nonunitary contributions, then this could potentially be used to improve the probability of success scaling of the algorithm by allowing failed implementations of that particular layer to

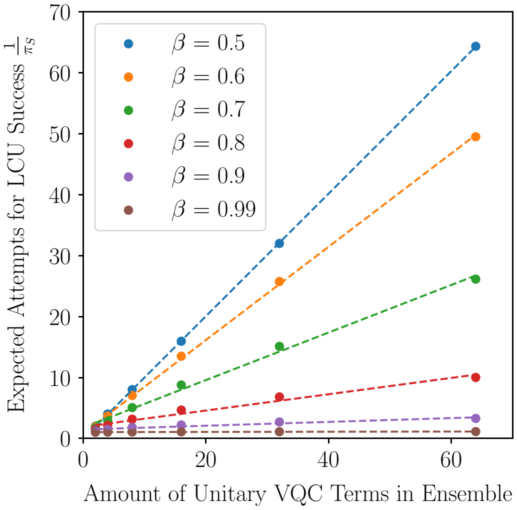

FIG. 9. The expected repeated attempts to implement one successful LCU procedure (equal to the reciprocal of the overall success probability $1/\pi_S$) plotted against the total number of unitary VQC terms in the ensemble for various fixed residual-connection strengths β. The ensemble will contain 2^L terms, where each term corresponds to a unitary VQC model loss function and where the number of sublayers ranges in increments of 1 from $\lambda = 1$ to $\lambda = 2^L$. In these numerics, we start with an initial state $\rho = |0\rangle\langle 0|$. At each residual-connection layer l, we calculate the probability of success of that layer π_l and then evolve the state by the operator $(I + W_l^{(\lambda)})$, where $W_l^{(\lambda)}$ consists of $\lambda = 2^l$ sublayers. In this case, we have chosen each individual sublayer to be a Haar-random unitary. The β terms correspond to fixing $\beta_l = \beta, \forall l$ such that every residual connection has the same strength parameter. Due to the symmetry around $\beta = 0.5$, we have restricted to plotting $0.5 \leq \beta < 1$ for readability.

be accepted as successes. In a scenario in which multiple *Ansätze* are being tested simultaneously, in which one may not know which contain or do not contain barren plateaus, it would be possible to ignore any failures for *Ansätze* that do not contribute meaningfully to the loss. This means that the probability of successful implementation would decay exponentially with the number of useful quantum ResNet layers implemented, L', that provide nonvanishing contributions to the loss, rather than the total number of layers, L. The implication of this, however, is that one could simply discard these unused terms completely from the model and restrict to L'. In this case, an ensemble of $2^{L'}$ models could then be manually constructed, which may be able to approximate the quantum ResNet results. Therefore, any benefit from future research in this direction would be focused on the ability to search through a wide range of *Ansätze* at the same time and find those that provide meaningful contributions, rather than any quantum advantage inherent in the model performance itself, although in the worst-case settings the cost related to the probabilistic nature of the LCU procedure may render the quantum ResNet infeasible.

Although we show that quantum ResNet frameworks can avoid barren plateaus, it is possible that many common architectures may approach a solution that is vulnerable to classical simulation. This would occur when the quantum ResNet is well approximated by an ensemble of classically simulatable shallow-depth terms. Perhaps this situation should be expected, as it has previously been observed that classical deep ResNets can behave as ensembles of shallow networks, where a previous study has reported that the gradient of a 110 layer ResNet was dominated by contributions from paths of depth between 10 and 34 layers [58]. We again highlight that the key to further advancements in the quantum ResNet implementation will likely rest on constructing models in which the nonunitary terms do not exponentially vanish and remain hard to simulate classically. This provides an expanded search space compared to strictly unitary models, which could be subject to a more in depth characterization in future work. Overall, the question of whether this ensemble-creating property of the quantum ResNet can be used effectively therefore remains a question for further research, although the reported success of VQC ensemble models [53] may motivate further investigation into the efficient creation of ensembles within a single quantum device.

III. AVERAGE-POOLING LAYERS

A. Quantum average-pooling implementation

In a CNN, it is common to find a pooling layer. This layer acts to reduce the dimension of the data by considering a tile of fixed size that passes over the data and performs some operation, such as averaging, on all the data points in the tile.

A classical average-pooling layer consists of a pooling window of size $D \times D$ pixels that passes over an image consisting of $N \times N$ pixels, with a certain stride K_{STRIDE} [13–15]. We shall focus on the case in which the pooling window moves across the image one pixel at a time, corresponding to a stride value $K_{\text{STRIDE}} = 1$. For each position of the pooling window, all pixels within the pooling window are averaged together and this average is output as the value of a pixel in a new image. In classical techniques depending on the size of K_{STRIDE} and D, the number of output pixels will be some number less than N^2 and hence the pooling corresponds to reducing the dimension of the image. In the quantum technique, there is no penalty for calculating all N^2 terms, as the operations are done in parallel on the quantum state; hence we shall consider this case and leave the dimensionality reduction as a subroutine that can be implemented after the averaging.

We consider an image of $N \times N$ pixels where each pixel is indexed by i and j and the pixel color value corresponds to $v_{i,j}$. If we focus on the average-pooling window that is $D \times D$ in dimension, then classically we wish to redefine each pixel label by

$$v'_{i,j} = \frac{1}{D^2} \sum_{\Delta x=0}^{D-1} \sum_{\Delta y=0}^{D-1} T_{\Delta x,\Delta y}(v_{i,j}), \tag{60}$$

where $T_{\Delta x,\Delta y}$ is a pixel translation action that acts on the i and j indices of $v_{i,j}$ to retrieve the values of other pixels through

$$T_{\Delta x,\Delta y}(v_{i,j}) = v_{i+\Delta x,j+\Delta y}, \tag{61}$$

so that they can be added to the average. The pixels accessed by $T_{\Delta x,\Delta y}$ define the pooling tile. For occasions on which the pooling window covers pixels outside the image, it is common to include some padding of the image [59], e.g., any pixels outside the $N \times N$ region can be padded by zeros, giving a padded image of dimension $(N+2D) \times (N+2D)$.

There have been many investigations focused on finding quantum parallels to CNN models, whereby measurements are taken of certain qubits such that the dimensionality of the data is reduced. For example, in some setups the pooling layer corresponds to a variational *Ansatz* followed by a measurement of a qubit [33]. In Ref. [31], an effective measurement has been chosen corresponding to the symmetry and details of their specific problem concerning quantum phase estimation. Although these popularized methods achieve the goal of pooling through reducing the dimensionality of the quantum state, they have not focused on giving a replication of an averaging pooling layer from classical CNN models for image data. In this section, we shall therefore demonstrate how LCU methods can be used to perform average pooling on quantum encoded image data. We shall utilize amplitude encoding with distinct coordinates x and y for the image, although any encoding for the data could be used, as long as the correct transformations are implemented with the LCU method.

Take an $N \times N$ image sample $(x_i, y_j, v_{i,j})$, where v represents the pixel color value and x_i and y_j are the index coordinates of the pixel, with $i,j \in [1,N]$. We shall utilize real-amplitude encoding by defining a quantum state with two registers as

$$|\psi\rangle = \sum_{x_i,y_j} \frac{v_{i,j}}{\Omega} |x_i\rangle |y_j\rangle, \tag{62}$$

where $\Omega^2 = \sum v_{i,j}^2$ is a normalization constant and $v_{i,j} \in \mathbb{R}$. Once we have an image encoded into a quantum state in this manner, we can consider the type of operation required by average pooling. In a simple case, we can consider only the x register for $D=2$. We will need to apply an operation that takes $|x\rangle \to 1/\Omega'(|x\rangle + |x \ominus 1\rangle)$ for some normalization constant Ω', to perform an averaging over pixel amplitudes. This will result in the state $|x\rangle$ having its amplitude transformed to the sum of the amplitudes of the $|x\rangle$ and $|x \oplus 1\rangle$ states before being normalized (and therefore averaged). This operation is implemented by $A_{\text{POOL}} = \frac{1}{2}(I + \text{SUBTRACT}_1)$, where $\text{SUBTRACT}_1|x\rangle = |x \ominus 1\rangle$ and $\text{ADD}_1|x\rangle = |x \oplus 1\rangle$. These operators are implemented by decrement and increment gates. We can see, by taking

$$A_{\text{POOL}}^\dagger A_{\text{POOL}} = \frac{1}{4}(2\mathbb{I} + \text{ADD}_1 + \text{SUBTRACT}_1) \neq \mathbb{I}, \tag{63}$$

that it is not a unitary operation. Hence, to implement average pooling on a quantum device, we utilize LCU methods.

Theorem 2 (Quantum Average Pooling). It is possible to probabilistically implement an averaging pooling layer on an image that has been amplitude encoded into a quantum state as specified in Eq. (62), such that the average-pooling operation A_{POOL} has the effect

$$A_{\text{POOL}} \sum_{x_i,y_j} \frac{v_{i,j}}{\Omega} |x_i\rangle |y_j\rangle = \sum_{x_i,y_j} \frac{v'_{i,j}}{\Omega'} |x_i\rangle |y_j\rangle,$$

where

$$v'_{i,j} = \frac{1}{D^2} \sum_{\Delta x=0}^{D-1} \sum_{\Delta y=0}^{D-1} v_{i+\Delta x,j+\Delta y},$$

by using the LCU framework from Sec. I D. The terms Ω and Ω' are the normalization constants for the initial and final states, respectively.

Proof. For simplicity, we consider the ancilla qubits to be composed of two registers. We initialize these registers to the following:

$$P_{\text{PREP}} |0\rangle^{\otimes k} |0\rangle^{\otimes k} = \frac{1}{D} \left(\sum_{\Delta x=0}^{D-1} |\Delta x\rangle \right) \left(\sum_{\Delta y=0}^{D-1} |\Delta y\rangle \right). \tag{64}$$

In this setting, $|\Delta x\rangle$ and $|\Delta y\rangle$ correspond to the computational basis states of a Hilbert space of k qubits $\mathcal{H} = (\mathbb{C}^2)^{\otimes k}$, which are labeled by Δx and Δy. In this basis, $|0\rangle \equiv |0\rangle^{\otimes k}$, $|1\rangle \equiv |0\rangle^{\otimes k-1}|1\rangle$ and $|2\rangle \equiv |0\rangle^{\otimes k-2}|1\rangle|0\rangle$, continuing for all values until $|2^k - 1\rangle \equiv |1\rangle^{\otimes k}$. The preparation operator therefore takes the form

$$P_{\text{PREP}} = \frac{1}{D} \left(\sum_{\Delta x=0}^{D-1} |\Delta x\rangle \right) \left(\sum_{\Delta y=0}^{D-1} |\Delta y\rangle \right) \langle 0|^{\otimes k} \langle 0|^{\otimes k} + \sum_{i=0}^{2^k-1} \sum_{j=1}^{2^k-1} (\ldots) \langle i| \langle j| + \sum_{i=1}^{2^k-1} (\ldots) \langle i| \langle 0|, \tag{65}$$

where $(\ldots)$ collects terms with $\langle i|\langle j|$ such that $i+j \geq 1$ and will therefore not be used in the algorithm and can

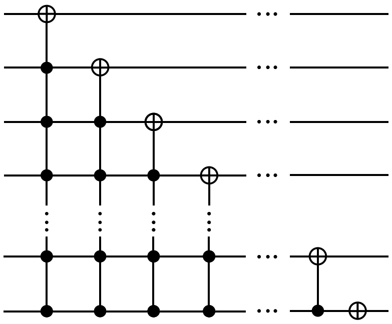

FIG. 10. An example circuit architecture to perform binary addition increasing all binary values by one. This takes $\text{ADD}_1|x\rangle = |x \oplus 1\rangle$ [60].

be ignored. This can be clearly implemented by a unitary operation. It can also be shown that $T_{\Delta x,\Delta y}$ can be realized with a unitary operator. It is possible to implement binary addition on a quantum circuit [60], as demonstrated in Fig. 10, which shows $\text{ADD}_1|x\rangle = |x \oplus 1\rangle$. In general, we shall define a k adding operator as $\text{ADD}_k|x\rangle = |x \oplus k\rangle$, which may be formed by applying ADD_1 in series k times or through a more optimized gate specific for that value of k. The inverse of this operator, found by taking the Hermitian conjugate, will subsequently be the subtraction operator $\text{SUBTRACT}_k|x\rangle = |x \ominus k\rangle$.

In order for $T_{\Delta x,\Delta y}(v_{i,j})$ to recover the correct amplitude, we can utilize the ADD_k and SUBTRACT_k operators. We can therefore define $\hat{T}_{\Delta x,\Delta y}$ through the action of these operators as

$$\hat{T}_{\Delta x,\Delta y}|x_i\rangle|y_j\rangle = |x_i \ominus \Delta x\rangle|y_j \ominus \Delta y\rangle; \quad \Delta x, \Delta y > 0, \tag{66}$$

$$\hat{T}_{\Delta x,\Delta y}|x_i\rangle|y_j\rangle = |x_i \oplus |\Delta x|\rangle|y_j \oplus |\Delta y|\rangle; \quad \Delta x, \Delta y \le 0. \tag{67}$$

Now consider the action of $\hat{T}_{\Delta x,\Delta y}$ on the amplitude-encoded image as

$$\begin{aligned}\hat{T}_{\Delta x,\Delta y}|\psi\rangle &= \hat{T}_{\Delta x,\Delta y}\left(\sum_{x_i,y_j}\frac{v_{i,j}}{\Omega}|x_i\rangle|y_j\rangle\right)\\ &= \sum_{x_i,y_j}\frac{v_{i,j}}{\Omega}|x_i \ominus \Delta x\rangle|y_j \ominus \Delta y\rangle\\ &= \sum_{x_i,y_j}\frac{v_{i,j}}{\Omega}|x_{i-\Delta x}\rangle|y_{j-\Delta y}\rangle.\end{aligned} \tag{68}$$

Relabeling the indices in the sum as $(i \to i + \Delta x)$ and $(j \to j + \Delta y)$, this can be written as

$$\hat{T}_{\Delta x,\Delta y}|\psi\rangle = \sum_{x_i,y_j}\frac{v_{i+\Delta x,j+\Delta y}}{\Omega}|x_i\rangle|y_j\rangle. \tag{69}$$

We can now define the selection operator as

$$\begin{aligned}S_{\text{SELECT}}|\Delta x\rangle|\Delta y\rangle|\psi\rangle &= |\Delta x\rangle|\Delta y\rangle\hat{T}_{\Delta x,\Delta y}|\psi\rangle\\ &= |\Delta x\rangle|\Delta y\rangle\sum_{x_i,y_j}\frac{v_{i+\Delta x,j+\Delta y}}{\Omega}|x_i\rangle|y_j\rangle,\end{aligned} \tag{70}$$

which corresponds to implementing controlled $\hat{T}_{\Delta x,\Delta y}$ operators, which are, respectively, applied when the ancilla is in the state $|\Delta x\rangle|\Delta y\rangle$. The selection operator is therefore unitary and can be applied on a quantum circuit.

Following the LCU procedure by first preparing the ancilla qubits gives

$$P_{\text{PREP}}|0\rangle^{\otimes k}|0\rangle^{\otimes k}|\psi\rangle = \frac{1}{D}\left(\sum_{\Delta x=0}^{D-1}|\Delta x\rangle\right)\left(\sum_{\Delta y=0}^{D-1}|\Delta y\rangle\right)|\psi\rangle, \tag{71}$$

while applying the selection operator results in

$$\begin{aligned}&S_{\text{SELECT}}P_{\text{PREP}}|0\rangle^{\otimes k}|0\rangle^{\otimes k}|\psi\rangle\\ &\quad = \frac{1}{D}\sum_{\Delta x=0}^{D-1}\sum_{\Delta y=0}^{D-1}|\Delta x\rangle|\Delta y\rangle\hat{T}_{\Delta x,\Delta y}|\psi\rangle.\end{aligned} \tag{72}$$

Applying the inverse preparation operator, defined as

$$\begin{aligned}P^{\dagger}_{\text{PREP}} &= \frac{1}{D}|0\rangle^{\otimes k}|0\rangle^{\otimes k}\left(\sum_{\Delta x=0}^{D-1}\langle\Delta x|\right)\left(\sum_{\Delta y=0}^{D-1}\langle\Delta y|\right)\\ &\quad + \sum_{i=0}^{2^k-1}\sum_{j=1}^{2^k-1}|i\rangle|j\rangle(\ldots) + \sum_{i=1}^{2^k-1}|i\rangle|0\rangle(\ldots),\end{aligned} \tag{73}$$

we can therefore see the state can be written as

$$\begin{aligned}&P^{\dagger}_{\text{PREP}}S_{\text{SELECT}}P_{\text{PREP}}|0\rangle^{\otimes k}|0\rangle^{\otimes k}|\psi\rangle\\ &\quad = \frac{1}{D^2}|0\rangle^{\otimes k}|0\rangle^{\otimes k}\sum_{\Delta x=0}^{D-1}\sum_{\Delta y=0}^{D-1}\hat{T}_{\Delta x,\Delta y}|\psi\rangle\\ &\qquad + \sum_{i=0}^{2^k-1}\sum_{j=1}^{2^k-1}|i\rangle|j\rangle(\ldots) + \sum_{i=1}^{2^k-1}|i\rangle|0\rangle(\ldots).\end{aligned} \tag{74}$$

We now wish to measure the ancilla qubits, discarding any states when $|0\rangle^{\otimes k}|0\rangle^{\otimes k}$ is not measured; hence we can ignore the $(\ldots)$ terms. The probability π_S of measuring the $|0\rangle^{\otimes k}|0\rangle^{\otimes k}$ state will therefore be

$$\pi_S = \left|\frac{1}{D^2}\sum_{\Delta x=0}^{D-1}\sum_{\Delta y=0}^{D-1}\hat{T}_{\Delta x,\Delta y}|\psi\rangle\right|^2. \tag{75}$$

After selecting only the cases in which the ancillas are measured in the $|0\rangle^{\otimes k}|0\rangle^{\otimes k}$ state, the final state will be projected to

$$\langle 0|^{\otimes k}\langle 0|^{\otimes k}P^{\dagger}_{\mathrm{PREP}}S_{\mathrm{SELECT}}P_{\mathrm{PREP}}|0\rangle^{\otimes k}|0\rangle^{\otimes k}|\psi\rangle = \frac{1}{\sqrt{\pi_S}D^2}\sum_{\Delta x=0}^{D-1}\sum_{\Delta y=0}^{D-1}\hat{T}_{\Delta x,\Delta y}|\psi\rangle, \quad (76)$$

where the $\sqrt{\pi_S}$ term is required such that the state is normalized. Expanding out the operation of $\hat{T}_{\Delta x,\Delta y}$, it is possible to see that

$$\frac{1}{\sqrt{\pi_S}D^2}\sum_{\Delta x=0}^{D-1}\sum_{\Delta y=0}^{D-1}\hat{T}_{\Delta x,\Delta y}|\psi\rangle = \frac{1}{\sqrt{\pi_S}D^2}\sum_{x_i,y_j}\sum_{\Delta x=0}^{D-1}\sum_{\Delta y=0}^{D-1}\frac{v_{i+\Delta x,j+\Delta y}}{\Omega}|x_i\rangle|y_j\rangle = \sum_{x_i,y_j}\frac{v'_{i,j}}{\Omega'}|x_i\rangle|y_j\rangle. \quad (77)$$

The term $\Omega' = \sqrt{\pi_S}\Omega$ is the overall normalization factor of the final state. Hence, the average-pooling operation

$$v'_{i,j} = \frac{1}{D^2}\sum_{\Delta x=0}^{D-1}\sum_{\Delta y=0}^{D-1}v_{i+\Delta x,j+\Delta y} \quad (78)$$

has been implemented on a quantum circuit. ■

Through this framework, the average-pooling operation can be implemented as is visualized for a 2 × 2 pooling window in Fig. 11.

In terms of the image boundaries, it would be possible to apply null padding to the image by introducing extra qubits in the $|0\rangle$ states. Without any padding, the quantum pooling window will treat the image as periodic and could include pixels from opposite sides of the image when located close to the image edge. It is worth mentioning that classical CNNs perform this averaging as a means of dimensionality reduction, as the total number of averages taken is usually less than the total number of pixels in the image. In the quantum case, however, it takes no additional time to average every single pixel in the image and therefore we have presented this as the most general case. Subsequently, to truly perform the quantum analogue of pooling, certain values would be discarded to reduce the dimensionality of the problem. The exact method will depend on exactly what kind of dimensionality reduction is desired, although we discuss how this could be performed in Appendix D.

B. Algorithm scaling

The advantage of average-pooling layers is well documented in classical neural networks [61–63]. The method suggested in our work would allow this feature to be implemented on quantum encoded images in QML models. The utility of this in the context of a quantum advantage will therefore be dependent on whether other quantum operations within the model can achieve some advantage over classical models. For example, recent empirical studies for amplitude-encoded image data suggest that quantum models may in some circumstances display an improved robustness against adversarial attacks compared to classical methods [64,65]. Indeed, theoretical guarantees on quantum robustness have recently been reported [66]. While this is not the focus of our work, the quantum average-pooling layers that we introduce may be of use in further studies regarding quantum encoded image data. We will therefore focus instead on examining the circuit complexity and probabilistic scaling of the method.

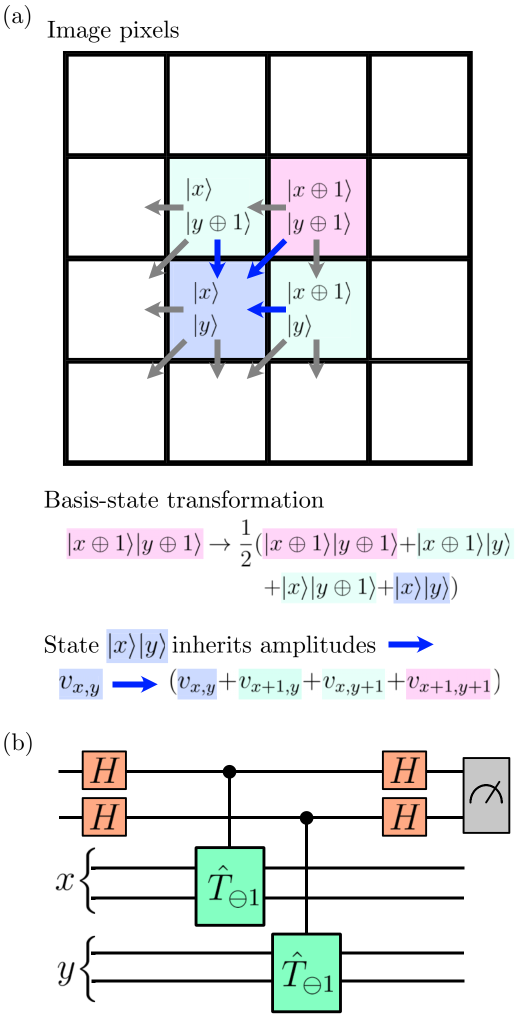

FIG. 11. (a) An example of a 2 × 2 pooling window on a 4 × 4 pixel image, indicating the desired transformation to perform average pooling. (b) The circuit that implements the average pooling in this example through application of controlled $\hat{T}_{\ominus 1}$ operators to both registers.

1. Circuit scaling

In order to implement an average-pooling layer with a pooling window size of $D \times D$, one needs to implement a linear combination of D^2 unitaries via the LCU method, whereby these unitaries correspond to selecting all pixels within the average-pooling window. This means that in a given x/y register, one needs to create a linear combination of D unitaries. A general technique for doing this using only $\log(D)$ ancillas for any LCU circuit would be to use D multicontrolled operators, whereby each ancilla basis state activates exactly one of the multicontrolled operators and no others; i.e., $S_{\text{SELECT}}|i\rangle|\psi\rangle = |i\rangle\hat{T}_i|\psi\rangle$. Previous related work on implementing general filter masks for quantum convolutional layers have used this approach, in which they state that the number of multicontrolled unitaries that must be implemented is equal to the size of the filter mask D^2 [16,17]. However, we show that by focusing on average pooling and considering that subtraction operators form a closed set, a more efficient circuit implementation is possible that uses $\mathcal{O}(\log(D))$ single-qubit controlled operators.

Without loss of generality, consider only the x register. We need to ensure that this has an equal superposition of $\hat{T}_{\Delta x}$ terms such that Δx runs from 0 to $D-1$. This can be achieved using $\mathcal{O}(\log(D))$ qubits and single-qubit controlled applications of $\hat{T}_k$ operators. In this case, $\hat{T}_k$ is implemented by the kth ancilla qubit and corresponds to the $\hat{T}_k = \text{SUBTRACT}_{2^{k-1}}$ operation. This operation subtracts 2^{k-1} from the binary register and can be formed by the SUBTRACT_1 operation applied to the $n-k+1$ most significant qubits and the identity operator applied to the $k-1$ least significant qubits. If $k \in [1, L]$, where L is the total number of ancilla qubits and controlled operations, then we have a set of operators $T = [\hat{T}_1, \hat{T}_2, \hat{T}_4, \ldots, \hat{T}_{2^{L-1}}]$, where the kth element is applied if the kth ancilla is in the 1 state. As we have an equal superposition of all 2^L basis states in the ancillas, this means that we will find an equal superposition of 2^L terms that consist of all possible combinations of the operators in T. The ancilla basis states can be thought of as selecting subsets of T, with the resultant operation found by multiplying all operators together in that subset such that the indices decrease from left to right. As we have chosen the scaling of these elements to be $\hat{T}_k = \text{SUBTRACT}_{2^{k-1}}$, we can see that considering all possible combinations of these terms will result in an equal sum of $\hat{T}_{\Delta x}$ operators, where Δx runs from 0 to $2^L - 1$.

We can then set $D = 2^L$ to see that the number of ancillas and controlled operators must scale as $L = \mathcal{O}(\log(D))$. Therefore, the procedure requires $\mathcal{O}(\log(D))$ unitaries that correspond to subtraction operators, which act on the separate x and y target registers and are controlled by $\mathcal{O}(\log(D))$ ancilla registers. An example circuit that involves four ancilla qubits and four controlled unitaries is shown in Fig. 12, in which 16 total combinations of operators are implemented. Note that if D is less than 2^L, then the final layer could be adjusted to be $\hat{T}_{D-2^{L-1}}$, but we may require multicontrol qubits in order to prevent degeneracy and maintain and an equally weighted combination of operators (see Appendix F).

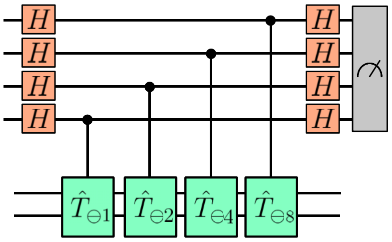

FIG. 12. An example of an efficient LCU circuit that implements a linear combination of 16 operators in the range $[\mathbb{I}, \hat{T}_1, \hat{T}_2, \ldots, \hat{T}_{15}]$. Each operator is controlled by a single qubit and corresponds to a subtraction operator, where the amount subtracted is doubled for each successive qubit. This approach is more efficient than the most general possible implementation and is possible due to the fact that subtraction operators applied in sequence form a closed set.

To perform the averaging classically for a stride size K would require finding the average of D^2 values for N^2 pixels, giving a scaling of $\mathcal{O}(D^2N^2)$ in general. Therefore classical averaging requires $\mathcal{O}(D^2N^2)$ arithmetic operations, while the quantum average pooling that we suggest requires $\mathcal{O}(\log(D))$ controlled-subtraction operations. The exact scaling of the controlled-subtraction operations depends on whether ancilla qubits can be utilized; however, we show that this uses at worst $\mathcal{O}((\log(N))^2)$ basic operations (see Appendix E), giving a total of $\mathcal{O}(\log(D)(\log(N))^2)$ basic operations in the overall LCU method while using $\mathcal{O}(\log(N))$ target qubits for the image encodings and $\mathcal{O}(\log(D))$ ancilla qubits.

Previous work has built an LCU-based quantum convolutional layer using a 3×3 filter of the form

$$\begin{pmatrix} \omega_{00} & \omega_{01} & \omega_{02} \\ \omega_{10} & \omega_{11} & \omega_{12} \\ \omega_{20} & \omega_{21} & \omega_{22} \end{pmatrix} \tag{79}$$

such that the filter passes over the image and calculates $\sum \omega_{\Delta x \Delta y} v_{i+\Delta x, j+\Delta y}$ for the pixel value in the window [16]. The authors show that average pooling can be obtained as a special case in which $\omega_{ij} = 1; \forall i,j$. We note that this work has not explicitly shown a proof for a general $D \times D$ window, instead considering only $D = 3$. The authors have used multicontrolled operators in which each ancilla-qubit state only implements exactly one unitary, meaning that their technique requires D^2 multicontrolled operators. They report their unitary operations as consisting of $\mathcal{O}((\log(N))^6)$ basic operators, which would correspond to an overall scaling of $\mathcal{O}(D^2(\log(N))^6)$. Hence, the circuit implementation that we present, which scales as

$\mathcal{O}(\log(D)(\log(N))^2)$, corresponds to a polynomial advantage in N and an exponential advantage in D over this previous technique when considering average pooling specifically.

Furthermore, although we started with the more specific case of average pooling, our entire algorithm can be generalized to recreate the quantum convolutional filters as presented in Ref. [16]. This is achieved by introducing amplitudes into the ancilla qubits instead of using an equal superposition. We see this by considering a new preparation operator in which all ancillas (for both x and y registers) are entangled such that

$$P_{\text{PREP}}|0\rangle^{\otimes k}|0\rangle^{\otimes k}|\psi\rangle = \frac{1}{D}\left(\sum_{\Delta x=0}^{D-1}\sum_{\Delta y=0}^{D-1}\sqrt{\omega_{\Delta x,\Delta y}}|\Delta x\rangle|\Delta y\rangle\right)|\psi\rangle, \tag{80}$$

where for simplicity we say that $\sqrt{\omega_{\Delta x,\Delta y}} \in \mathbb{R}$ is a real amplitude of the ancilla basis state $|\Delta x\rangle|\Delta y\rangle$. This will mean that the terms in the linear combination of $\hat{T}_{\Delta x,\Delta y}$ operators will have an associated weighting equal to $\omega_{\Delta x,\Delta y}$. Therefore, the transformation on the pixel-value amplitudes in the image will be

$$v'_{i,j} = \frac{1}{D^2}\sum_{\Delta x=0}^{D-1}\sum_{\Delta y=0}^{D-1}\omega_{\Delta x,\Delta y}v_{i+\Delta x,j+\Delta y}. \tag{81}$$

We can therefore identify that this performs a general quantum convolutional filter as originally described in Ref. [16], while utilizing exponentially fewer unitary operators in terms of D. A comparison of this reduction in circuit complexity is shown in Fig. 13 when applying a 4×4 convolutional filter, defined as

$$\begin{pmatrix} \omega_{00} & \omega_{01} & \omega_{02} & \omega_{03} \\ \omega_{10} & \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{20} & \omega_{21} & \omega_{22} & \omega_{23} \\ \omega_{30} & \omega_{31} & \omega_{32} & \omega_{33} \end{pmatrix} \tag{82}$$

This general-convolution case reduces to the previously discussed averaging case when the ancilla qubits are in an equal superposition. We highlight that in this section we do not discuss the actual pooling step itself and hence any improvement is confined to the averaging or convolutional subroutine of the circuit. Further details on the practical implementation of the pooling step and the quantum CNN framework can be found in Ref. [16].

We also note that our implementation is easily generalized to data types with dimension greater than two by adding additional registers for extra dimensions. For Q-dimensional data, we require $Q\log(N)$ qubits to encode the image into Q different registers. In this case, the window has a total Q-dimensional volume of D^Q and hence using the implementation from previous work [16] would require D^Q unitaries. In contrast, our technique requires $Q\log(D)$ operators in total, with $Q\log(D)$ ancilla qubits, when considering all registers and would therefore use exponentially fewer unitaries in both data dimension Q and window size D.

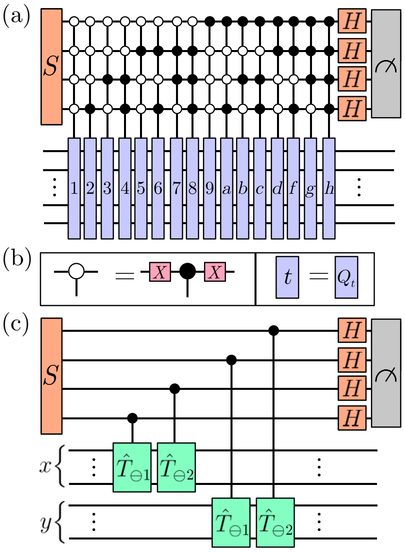

FIG. 13. (a) The circuit used to implement quantum convolutional filters in previous work [16]. For an equal comparison, we show the generalization of the authors' method to allow the implementation of a 4×4 size convolutional window, using 16 multicontrolled unitaries. (b) White circles correspond to controls that activate if the control qubit is in the $|0\rangle$ state. The t-labeled controlled gates apply the unitary operator Q_t, which when combined using the LCU technique reconstruct the desired convolutional filter, as outlined in Ref. [16]. A complete definition of the Q_t unitaries within the authors' framework is extensive and is omitted here. For both circuits, S is the unitary used to initialize the ancilla-qubit amplitudes corresponding to the desired filter, which is then achieved when the ancillas are all measured in the $|0000\rangle$ state at the end of the circuit. (c) The circuit implementation of the same 4×4 convolutional filter using our framework as described in the text. Only a single qubit controls each $\hat{T}_{\ominus t}$, which acts to subtract t in binary on the x or y register. This arithmetic interpretation simplifies the circuit considerably as we can see that each ancilla basis state applies one of 16 different transformations $\hat{T}_{\Delta x,\Delta y}$ on the pixel registers, which provides all the transformations required to apply a 4×4 size convolutional window, using only four controlled unitary gates in total, rather than having to implement 16 terms individually as multicontrolled unitary gates. For example, $|1111\rangle$ will be responsible for $\Delta x = 3, \Delta y = 3$ and hence the ancilla amplitude α_{1111} should control the relative weight of the top-right pixel of the convolution filter [ω_{03} in Eq. (82)], while α_{0000} corresponds to the identity and would control the bottom-left pixel weighting [ω_{30} in Eq. (82)].

2. Probabilistic scaling

When considering the fact that this is a probabilistic algorithm with a chance of failure, the situation becomes slightly more complicated, as the probability will depend on the image itself. As shown previously, the probability of success is equal to

$$\pi_S = \left| \frac{1}{D^2} \sum_{\Delta x=0}^{D-1} \sum_{\Delta y=0}^{D-1} \hat{T}_{\Delta x,\Delta y} |\psi\rangle \right|^2 .$$

If we have an image in which every pixel has the exact same color value, we can effectively set $\hat{T}_{\Delta x,\Delta y} = \mathbb{I}$ and therefore $\pi_S = 1$. We expect real-world images to have pixels that are close in color value to other local pixels nearby, on average, with the exceptions occurring at the edges of subjects within the image. We therefore expect the probability to remain relatively stable overall, although one could construct adversarial example images that result in low probabilities. To gain a practical understanding of this probability scaling, we have considered the MNIST fashion data set [18]; we empirically show the scaling of the probability of success with respect to D and N in Figs. 14 and 15, respectively. These results indicate that the probability of success decreases but levels off after a certain point as D increases and that there is no discernible trend when increasing N. These results align with our intuition for real-world images. The general structure and content of real-world images (and their local pixel similarities) remain consistent as N increases; therefore, there should be no discernible decrease in π_S from increasing N. On the other hand, increasing D means considering a larger neighborhood of pixels. Initially, as D increases, the probability of encountering edges within this neighborhood also increases, causing π_S to decrease. However, beyond a certain point, further increasing D adds less new information, because most of the image regions are already accounted for and edges are covered, so π_S levels off. The exact probability will be very image dependent; there may also exist techniques to prepare images such that the probabilities are improved, which we leave as an open question for further research. In cases in which the probability of success remains high, it corresponds to pixels being similar to each other locally; this in turn means that the average-pooling action on those qubits locally will be well approximated by the identity operation. It is important to note, therefore, that these conditions could be more prone to classical simulation techniques, and further work should be carried out on this possibility before any concrete advantages are claimed.

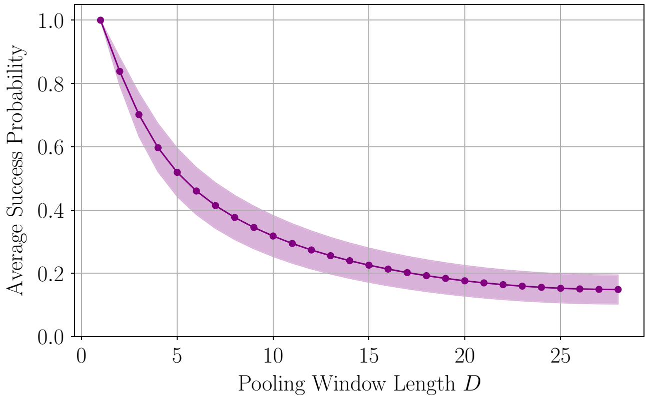

FIG. 14. The average success probability of the LCU procedure for implementing an average-pooling layer as the pooling window dimension D is increased, while the image size is set to $N = 28$. The average is taken over 100 image samples from the MNIST database [18], with the shaded area indicating the standard deviation of the samples.

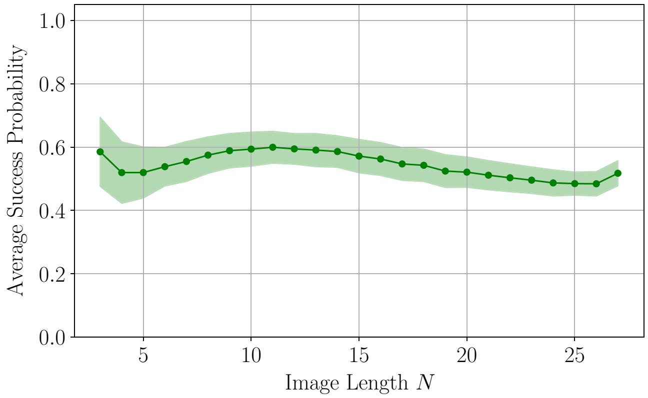

FIG. 15. The average success probability of the LCU procedure for implementing an average-pooling layer as the image size N is increased, while the pooling window size is kept constant at $D = 3$. The average is taken over 100 image samples from the MNIST database [18], with the shaded area indicating the standard deviation of the samples.

IV. IRREDUCIBLE-SUBSPACE PROJECTIONS

In order to discuss the irreducible-subspace-projections circuit, we first give a brief overview of representation-theory definitions.

A. Representation-theory preliminaries

Following standard texts on group and representation theory [67,68], we provide a brief summary and introduction to the underlying mathematical framework utilized in this section.

The definition of a representation of a group G can be given as follows.

Definition 1 (Representation). A representation of a group G refers to a pair (U, V), where V is a vector space and U is a group homomorphism $U : G \to GL(V)$, where $GL(V)$ is the general linear group of invertible matrices that acts on the vector space V of the representation. The homomorphism maps group elements $g \in G$ to matrices $U_g \in GL(V)$.

Often, the homomorphism U, the vector space V, or the image subgroup $U_G \subseteq GL(V)$ can be referred to as the "representation," depending on the context.

Definition 2 (Subrepresentation). For a given representation (U, V), a subrepresentation is a vector subspace $W \subseteq V$ for which any vector $w \in W$ remains within W when acted on by any representation of the group, i.e., $U_g w \in W, \forall g \in G$.

Definition 3 (Irreducible Representation). If a representation does not contain any nontrivial subrepresentations (the trivial spaces being the empty subspace $\{0\}$ and the entire space V), then it is said to be an irreducible representation.

If a representation does contains a nontrivial subrepresentation, then it is called a reducible representation. Representations can often be decomposed into a direct sum of their irreducible representations, in which case they are called completely reducible. Maschke's theorem [69] states that any finite-dimensional representation of a finite group G over a field F will be completely reducible (so long as the field characteristic does not divide the order of the group). It is also the case that every finite-dimensional representation of compact Lie groups is completely reducible [70]. If a representation (U, V) is completely reducible, then there exists a basis in which we can write U_g as

$$U_g \to \bigoplus_{r=1}^{R} \bigoplus_{j=1}^{m_r} u_g^{(r)} = \bigoplus_{r=1}^{R} u_g^{(r)} \otimes \boldsymbol{I}_{m_r}, \tag{83}$$

where $(u^{(r)}, V_r)$ are the irreducible representations indexed by $r \in [1, R]$, where R is the total number of irreducible representations. In this work, we will often refer to r as the irreducible representation for simplicity, although strictly it is the label that indexes the irreducible representations $(u^{(r)}, V_r)$. The quantity m_r is called the multiplicity of the irreducible representation. We can also define the degree of a representation as $n_r \equiv \dim(V_r)$. In this basis, U_g is in a block-diagonal form. This change of basis also decomposes the representation vector space V as

$$V \to \bigoplus_{r=1}^{R} V_r \otimes \mathbb{C}^{m_r}. \tag{84}$$

The conjugacy class of a group and the character of a representation for a given group element can be defined as follows.

Definition 4 (Conjugacy Class). For a group G, two given elements $g, h \in G$ are said to be conjugate if $\exists x \in G$ such that

$$g = xhx^{-1}$$

and hence the conjugacy class can be defined as

$$C_{(g)} = \{xgx^{-1} \mid x \in G\},$$

corresponding to the set of all elements that are conjugate to g.

Definition 5 (Character). For a representation (U, V) of the group G, the character can be defined as

$$\chi_U(g) = \mathrm{Tr}\left(U_g\right),$$

which is the trace of the representation of g on V. For the irreducible representations $(u^{(r)}, V_r)$ indexed by r, we shall define $\chi_{u^{(r)}}(g) \equiv \chi_r(g)$ for simplicity.

In particular, as the trace permits cyclic permutation, this means that $\chi_r(xgx^{-1}) = \chi_r(g)$ and therefore $\chi_r(g)$ is constant on the conjugacy classes of G; i.e., if $g_1, g_2 \in C_{(g_1)}$, then $\chi_r(g_1) = \chi_r(g_2)$. It is also worth noting that as the identity element corresponds to an identity matrix, taking the trace of an identity matrix will result in the fact that $\chi_r(I) = \dim(V_r)$. Hence, the degree of an irreducible representation can be found by $n_r = \dim(V_r) = \chi_r(I)$.

We will now introduce a key result of this work regarding projecting quantum encoded states to any linear combination of irreducible subspaces of a given finite group.

B. Subspace projection circuit

We start by observing a result from the representation theory of finite groups, which states the following [71].

Theorem 3 (Irreducible-Subspace Projection). Let $U : G \to GL(V)$ be a representation of G. The canonical decomposition into an irreducible representation is given by $V = V_1 \oplus V_2 \oplus \ldots \oplus V_R$, where the irreducible representations have characters $\chi_1, \ldots, \chi_R$ and degrees $n_1, \ldots, n_R$. Then, the projection $\hat{P}_r$ of V onto the space V_r is given by

$$\hat{P}_r = \frac{n_r}{|G|} \sum_{g_i \in G} \chi_r(g_i)^* U_{g_i}, \tag{85}$$

where $\chi_r(g_i)$ is the character of group element $g_i \in G$ for irreducible representation r and U_{g_i} is the matrix representation of group element $g_i \in G$ acting on the space V.

This is a known result in representation theory; for more details, see reference texts such as Refs. [67,71]. In this section, we explore a method for implementing this projection on a quantum circuit using the LCU technique. Specifically, we demonstrate the practicality of realizing the projection described in Theorem 3 on a quantum device. In the most comprehensive scenario, we establish that a quantum circuit can execute combinations of such $\hat{P}_r$ projections at the same time via the LCU method.

Theorem 4 (Quantum Generic Irreducible-Subspace Projection). Let $|\psi\rangle \in V = (\mathbb{C}^2)^{\otimes n}$ be a quantum state encoded on n qubits. Let $U : G \to SU(2^n)$ be a unitary representation of a finite group G, where the representatives for group elements $g_i \in G$ are denoted $U_{g_i} \in SU(2^n)$ and are unitary operators that act on $|\psi\rangle$ and can be implemented in a quantum device. Utilizing the LCU framework on a quantum circuit, one can probabilistically apply a linear combination of irreducible-representation projections

$$\sum_{r=1}^{R} a_r \hat{P}_r |\psi\rangle = \frac{1}{\Omega'} \sum_{r=1}^{R} a_r \left(\frac{n_r}{|G|} \sum_{g_i \in G} \chi_r(g_i)^* U_{g_i} |\psi\rangle \right), \tag{86}$$

such that $\hat{P}_r$ projects $|\psi\rangle \in V$ to the subspace V_r corresponding to the irreducible-representation subspace indexed by r with degree n_r. The constants a_r can be freely chosen, Ω' is the normalization constant for the final state after projection, and R is the total number of irreducible representations.

Proof. As we intend to implement a linear combination of projections, we shall consider a slightly more general form of the LCU method. We first perform a preinitialization step on the k ancilla qubits, initially in the basis state $|b_1\rangle$, using an operator $\hat{\gamma}$ to prepare the state $|c\rangle = \hat{\gamma}|b_1\rangle$ defined as

$$|c\rangle = \frac{1}{\Omega} \sum_{r=1}^{R} a_r n_r |b_r\rangle, \tag{87}$$

where a_r can be chosen to adjust the relative amounts of a given representation, n_r is the degree of the representation, and $\Omega = \sum_{r=1}^{R} |a_r n_r|^2$ is a normalization constant for the quantum state. The states $\{|b_j\rangle\}_{j \in [1,2^k]} \in (\mathbb{C}^2)^{\otimes k}$ are basis states of the 2^k-dimensional Hilbert space for the k-qubit ancilla register denoted by $\mathcal{H} = (\mathbb{C}^2)^{\otimes k}$. They can be taken to be computational basis states. We highlight that after this preinitialization, the ancilla register is in a combination up to R basis states, which are indexed by the irreducible representations r. The inclusion of the n_r term here ensures that the representation weightings are correct later on. At this point, every state $|b_r\rangle$ corresponds to a different irreducible representation of the group G. We define $r = 1$ to correspond to the trivial representation.

The next step corresponds to applying an operator that takes each representation labeled state $|b_r\rangle$ into a sum of basis states $|b_i\rangle$, $i \in [1, |G|]$, which are now labeled by the elements of the group $g_i \in G$, where each basis state $|b_i\rangle$ inherits a weighting according to the character $\chi_r(g_i)^*$ corresponding to the representation r and the relevant group element g_i. To help improve readability, we define $|g_i\rangle \equiv |b_i\rangle$. These correspond to the same basis states in the ancilla register but they emphasize the fact that they are labeled by the group elements g_i.

As we consider a combination of multiple projections at the same time, then, in order to maintain generality of our framework, instead of requiring $P_{\text{PREP},r}$ to be a different operator for each individual irreducible representation (indexed by r), as would be the standard case in LCU methods, we will consider a generalized preparation operator corresponding to the unitary $\hat{\chi}$, which has the effect of applying the correct character for every irreducible representation r and every group element $g_i \in G$ such that

$$\hat{\chi}|b_r\rangle = \frac{1}{\sqrt{|G|}} \sum_{g_i \in G} \chi_r(g_i)^* |g_i\rangle, \tag{88}$$

where in the most general form,

$$\hat{\chi} = \frac{1}{\sqrt{|G|}} \sum_{r=1}^{R} \sum_{g_i \in G} \chi_r(g_i)^* |g_i\rangle\langle b_r| + \sum_{r=R+1}^{2^k} (\ldots)\langle b_r|, \tag{89}$$

where the terms grouped together as $\sum_{r=R+1}^{2^k} (\ldots)\langle b_r|$ will not be used in the algorithm and can hence be ignored. Implementing this corresponds to constructing a matrix that contains the character of every group element for every irreducible representation. This matrix will have the form

$$[\hat{\chi}]_{i,j} = \begin{cases} \frac{1}{\sqrt{|G|}} \chi_j(g_i), & \text{if } i \in [1, |G|], j \in [1, R], \\ 0, & \text{if } i \in (|G|, 2^k], j \in [1, R], \\ \frac{1}{\sqrt{|G|}} u_{i,j}, & \text{if } i \in [1, 2^k], j \in (R, 2^k], \end{cases} \tag{90}$$

where the $u_{i,j}$ will never affect the algorithm but are required as the matrices involved must be of size 2^k. The first R columns correspond to the vectors that contain the characters for each element in a given representation. Considering the character orthogonality theorem [72], which states that

$$\frac{1}{|G|} \sum_{g_i \in G} \chi_j(g_i) \chi_k(g_i)^* = \delta_{jk}, \tag{91}$$

this means that the character vectors of irreducible representations form an orthonormal basis. The $u_{i,r}$ terms can be chosen utilizing the Gram-Schmidt procedure or otherwise to ensure that all column vectors of $\hat{\chi}$ form an orthonormal basis. A matrix the column vectors of which form an orthonormal basis is unitary. Hence, our generalized preparation step $P_{\text{PREP}} = \hat{\chi}$ can be implemented on a quantum circuit.

The S_{SELECT} operator can be implemented by applying the unitary quantum gate representation $U_g \in SU(2^n)$ to the n-qubit target state $|\psi\rangle \in V = (\mathbb{C}^2)^{\otimes n}$ for each corresponding group element $|g_i\rangle$ in the ancilla register

$$S_{\text{SELECT}}|g_i\rangle|\psi\rangle = |g_i\rangle U_{g_i}|\psi\rangle. \tag{92}$$

As long as G is a compact group, there will be unitary irreducible representations U_{g_i} via the Peter-Weyl theorem [73]. Hence U_{g_i} can be implemented in a quantum circuit for compact groups G. Using controlled gates U_{g_i}, in which the operation U_{g_i} is applied only when the ancilla qubit is in the state $|g_i\rangle$, then the operator S_{SELECT} is successfully implemented.

We have shown that the preparation and selection operators can be implemented on a quantum circuit. Now all that remains is to combine the operators together to view the full action of the LCU procedure. Preinitializing the ancillas

$$\hat{\gamma}|b_1\rangle|\psi\rangle = \frac{1}{\Omega}\sum_{r=1}^{R} a_r n_r |b_r\rangle|\psi\rangle, \tag{93}$$

followed by applying preparation operator $\hat{\chi}$ on the ancillas, gives

$$\hat{\chi}\hat{\gamma}|b_1\rangle|\psi\rangle = \frac{1}{\Omega}\sum_{r=1}^{R} a_r n_r \frac{1}{\sqrt{|G|}}\sum_{g_i\in G}\chi_r(g_i)^*|g_i\rangle|\psi\rangle. \tag{94}$$

The selection operator applied to the circuit can then be written

$$\begin{aligned} &S_{\text{SELECT}}\hat{\chi}\hat{\gamma}|b_1\rangle|\psi\rangle \\ &\quad = \frac{1}{\Omega}\sum_{r=1}^{R} a_r n_r \frac{1}{\sqrt{|G|}}\sum_{g_i\in G}\chi_r(g_i)^*|g_i\rangle U_{g_i}|\psi\rangle. \end{aligned} \tag{95}$$

The conjugate $\hat{\chi}$ term can be written as

$$\hat{\chi}^\dagger = \frac{1}{\sqrt{|G|}}\sum_{r=1}^{R}\sum_{g_i\in G}\chi_r(g_i)|b_r\rangle\langle g_i| + \sum_{r=R+1}^{2^k}|b_r\rangle(\ldots), \tag{96}$$

where $\sum_{r=R+1}^{2^k}|b_r\rangle(\ldots)$ represents terms that will not end up contributing and hence can be ignored. Applying the conjugate $\hat{\chi}^\dagger$ term and separating out terms that will not be used further, we find that

$$\begin{aligned} &\hat{\chi}^\dagger S_{\text{SELECT}}\hat{\chi}\hat{\gamma}|b_1\rangle|\psi\rangle \\ &= \frac{1}{\Omega}\sum_{r=1}^{R} a_r \frac{n_r}{|G|}\sum_{g_i\in G}|b_1\rangle\chi_1(g_i)\chi_r(g_i)^* U_{g_i}|\psi\rangle + \sum_{r=2}^{2^k}|b_r\rangle(\ldots), \end{aligned} \tag{97}$$

where we can ignore all terms $\sum_{r=R+1}^{2^k}|b_r\rangle(\ldots)$, as they will be discarded if ever measured. Note that since $r = 1$ is defined as the trivial representation, it follows that $\chi_1(g_i) = 1, \forall g_i \in G$. We now measure the ancilla qubits and find that they will be found in the state $|b_1\rangle$ with a probability π_S given by

$$\pi_S = \left|\frac{1}{\Omega}\sum_{r=1}^{R} a_r \frac{n_r}{|G|}\sum_{g_i\in G}\chi_r(g_i)^* U_{g_i}|\psi\rangle\right|^2. \tag{98}$$

By only retaining states in which the ancilla was measured in the $|b_1\rangle$ state, we will have prepared the state

$$\begin{aligned} &\langle b_1|\hat{\chi}^\dagger S_{\text{SELECT}}\hat{\chi}\hat{\gamma}|b_1\rangle|\psi\rangle \\ &\quad = \frac{1}{\Omega'}\sum_{r=1}^{R} a_r \frac{n_r}{|G|}\sum_{g_i\in G}\chi_r(g_i)^* U_{g_i}|\psi\rangle, \end{aligned} \tag{99}$$

where $\Omega' = \sqrt{\pi_S}\,\Omega$ is the normalization constant of the final state, as required. ■

A simple corollary follows, regarding the special case in which during the preinitialization stage $a_{r'=r} = 1$ and $a_{r'\neq r} = 0$, such that only one irreducible representation is selected.

Corollary IV.1 (Quantum Individual Subspace Projection). The projection given in Theorem 3 can be probabilistically implemented on an n-qubit quantum state $|\psi\rangle \in V = (\mathbb{C}^2)^{\otimes n}$ by utilizing an LCU in a quantum circuit where

$$\hat{P}_r|\psi\rangle = \frac{n_r}{\Omega'|G|}\sum_{g_i\in G}\chi_r(g_i)^* U_{g_i}|\psi\rangle, \tag{100}$$

such that $\hat{P}_r$ projects $|\psi\rangle$ from the space V to an irreducible-representation subspace V_r.

This shows that the LCU method can be adapted through careful selection of the preparation and selection operators to reproduce the projection onto any combination of irreducible representations. An overall schematic of the

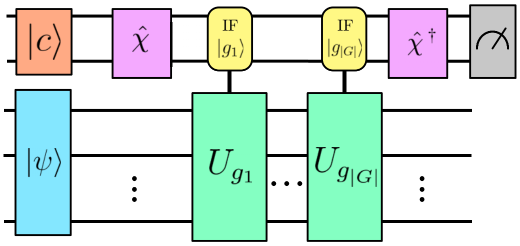

FIG. 16. The general irreducible-representation projection circuit. The upper register consisting of k qubits is prepared into a superposition state $|c\rangle$, where the basis states $|b_r\rangle$ are labeled by the irreducible representations r. The operator $\hat{\chi}$ then takes each irreducible-representation labeled basis state to its corresponding group-element labeled basis state multiplied by the correct character. Each group-element labeled basis state $|g_i\rangle \equiv |b_i\rangle$, where $g_i \in G$, then controls the application of its corresponding representation on the quantum states in the lower register, represented by $U_{g_i} \in SU(2^n)$. This is achieved by using multiqubit control gates, such that if the ancilla is in the state $|g_i\rangle$, the operator U_{g_i} is applied and nothing else. This is demonstrated in the graphic using yellow boxes for these conditional control gates, which are controlled by specific multiqubit basis states $|g_i\rangle$. These can be constructed as multiqubit control gates along with the correct implementation of X gates to ensure that the controlled action U_{g_i} is only applied when the ancilla register is in the correct basis state $|g_i\rangle$. We apply $\hat{\chi}^\dagger$ on the upper register and then measure and postselect for the $|b_1\rangle \equiv |0\rangle^{\otimes k}$ state. We demonstrate that this will create the appropriate combinations of projections onto the desired subspaces, with the proportion of each projector determined by the initialization of the state $|c\rangle$.

representation projection circuit is shown in Fig. 16. Note that after the projection, the state may no longer be a pure state.

1. Circuit scaling

In order for k ancilla qubits to have a basis state for each group-element representation of G, it would require $2^k \geq |G|$. Hence, the number of qubits required scales as $k \approx \mathcal{O}(\log(|G|))$. However, the character-implementation unitary $\hat{\chi}$, which is used in the ancilla preparation stage, has a dimension that scales with $\mathcal{O}(|G|)$, which may be prohibitive in certain cases. For example, if G is taken to be the permutation group $G = S_n$, where the representation operators $U_{\sigma_i}, \sigma_i \in S_n$ correspond to permuting n qubits in the target state, then we require $k \approx \mathcal{O}(\log(n!)) \approx \mathcal{O}(n\log(n))$ ancilla qubits; however, we are required to initialize a unitary of size $\mathcal{O}(n!)$, which may be difficult. A proposed solution outlined in Appendix A is to utilize a smaller version of $\tilde{\chi}$ corresponding to the character table, where each ancilla-qubit state represents a conjugacy class rather than individual group elements. In this case, the size of the unitary required could be significantly reduced, as it would scale only with the number of conjugacy classes, denoted by $|\mathcal{C}|$. Hence, this alternative implementation would require ancilla qubits of order $\mathcal{O}(\log(|\mathcal{C}|))$ to encode the character-table state, but then it would also require $|\mathcal{C}|$ extra registers, one per conjugacy class, with sufficient qubits to produce a superposition of states for every element in the conjugacy class. This means more qubits are required in total but the unitary is easier to implement. For example, the number of conjugacy classes in S_n, which is approximately approached in the asymptotic limit, is

$$|\mathcal{C}_n| \approx \frac{1}{4n\sqrt{3}} \exp\left(\pi\sqrt{\frac{2n}{3}}\right); \quad n \to \infty, \tag{101}$$

which is superpolynomial but subexponential [67]. This would be a significant improvement in scaling compared to $\mathcal{O}(n!)$. This allows a trade-off between total qubits and the ease of unitary preparation. Note that in this section, we have introduced a very general framework and improvements in circuit efficiency to implement $\hat{\chi}$ may well be possible for specific groups, representations, and data types.

2. Probabilistic scaling

The probability of success π_S is shown in Eq. (98) and is equal to the probability of measuring the ancillas in the $|b_1\rangle$ state. This can be simplified by utilizing the result of Theorem 4 to observe that

$$\pi_S = \left|\frac{1}{\Omega}\sum_{r=1}^{R} a_r \hat{P}_r |\psi\rangle\right|^2 . \tag{102}$$

Any quantum state can be written as a decomposition in terms of its components on the irreducible subspaces

$$|\psi\rangle = \bigoplus_{r=1}^{R}\bigoplus_{j=1}^{m_r} |\psi_r\rangle_j , \tag{103}$$

where m_r is the multiplicity of representation r. We define $|\psi_r\rangle \equiv \bigoplus_{j=1}^{m_r} |\psi_r\rangle_j$ and see that it denotes the component of $|\psi\rangle$ that occupies the subspaces of the irreducible representation r. After $|\psi\rangle$ is projected by $\hat{P}_r$ onto the irreducible subspace of r, it will equal $|\psi_r\rangle$. Noting that $|\psi_r\rangle$ will be orthogonal for different r values, we can therefore write

$$\pi_S = \frac{1}{\Omega^2}\sum_{r=1}^{R} |a_r|^2 \langle\psi_r|\psi_r\rangle. \tag{104}$$

Therefore, the probability of success depends on the extent to which the initial state $|\psi\rangle$ occupies the relevant irreducible subspaces in the projection. The normalization condition in the general state means that $\langle\psi|\psi\rangle =$

$\sum_r \langle\psi_r|\psi_r\rangle = 1$. In the case in which we project fully to one specific space $a_r = 1, a_{r'\neq r} = 0$, then

$$\pi_S = \langle\psi_r|\psi_r\rangle, \tag{105}$$

which will depend on how much of the state $|\psi\rangle$ lies in the subspace of irreducible representation r. If $|\psi\rangle$ lies fully within the space, then $\pi_S = \langle\psi_r|\psi_r\rangle = 1$. Conversely, if $|\psi\rangle$ has no components in the subspace for irreducible representation r, then $\pi_S = \langle\psi_r|\psi_r\rangle = 0$ and the algorithm is impossible to run. An advantage of the additional control introduced by the a_r parameters is that we can freely choose the weightings to potentially improve the probability of success. In general, the probability of success will depend greatly on data encoding and choice of representations.

C. Symmetry-invariant encodings for point-cloud data

In this subsection, we discuss projections to a single irreducible subspace in order to highlight how these subspaces can correspond to symmetries of the input data. However, it should be noted that these projections could take the quantum state to a polynomially sized space that may be classically simulatable, as has previously been shown to be the case with equivariant variational models [74,75]. The following subsection will address this issue by instead considering all subspaces, and hence maintaining an exponentially large space overall, but allowing certain subspaces to be amplified relative to the others. Investigating single subspace projections remains useful in this context, as it allows the identification of which irreducible-representation subspaces correspond to specific symmetries of the underlying data. We therefore proceed to highlight how certain irreducible-representation subspaces correspond to symmetries of the underlying data when it has been encoded into a quantum state. In particular, we focus on point-cloud data due to its inherent permutation and rotation symmetry.

A point cloud is a collection of 3D vectors (the points) that when viewed as a collective, represent an image. Amongst other applications, they are often associated with computer-vision algorithms, as a primary method for performing 3D imaging is the use of the Light Detection and Ranging (LiDAR) system, which produces point clouds as its data output [76]. We can consider a point cloud P as a set of 3D vectors $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$ that overall forms an image. We can consider a form of quantum encoding in which each individual point $\mathbf{p}_i$ is encoded into a quantum state $|\mathbf{p}_i\rangle$, leading to a separable quantum state for the overall point cloud as

$$|P\rangle = |\mathbf{p}_1\rangle \otimes |\mathbf{p}_2\rangle \otimes \cdots \otimes |\mathbf{p}_n\rangle. \tag{106}$$

The states $|\mathbf{p}_i\rangle$ could be single qubits, in which case the total number of qubits in this target register would be n. However, in general, each $|\mathbf{p}_i\rangle$ could be a state encoded on t qubits, in which case we can consider each $|\mathbf{p}_i\rangle$ as a 2^t-dimensional qudit and n would denote the number of qudits in the system that encodes $|P\rangle$.

1. *Permutation invariance*

The ordering of the points within the set P does not affect the overall image; however, in general, a machine-learning algorithm may output different results depending on the ordering of the points in the data-input array. A solution to prevent this by utilizing permutation-symmetric encodings for point clouds has been suggested in previous work [20], where a quantum superposition of all possible permutations has been used, leading to a permutation-invariant quantum encoding such that

$$|P\rangle_{\text{perm}} = \frac{1}{\Omega'} \sum_{\sigma\in S_n} |\mathbf{p}_{\sigma^{-1}(1)}\rangle \otimes |\mathbf{p}_{\sigma^{-1}(2)}\rangle \otimes \cdots \otimes |\mathbf{p}_{\sigma^{-1}(n)}\rangle. \tag{107}$$

This previous work can be viewed as a special case of the LCU method described previously, in which the group is S_n and the state $|P\rangle$ is projected to the symmetric irreducible-representation subspace (in which the character for all group elements is equal to 1). The representations of the group elements $\sigma \in S_n$ that act on $|P\rangle$ are denoted by U_σ and correspond to SWAP gates that permute the constituent states $|\mathbf{p}_i\rangle$ accordingly. This projection would correspond to applying the projector

$$\sum_{\sigma\in S_n} U_\sigma |P\rangle = \frac{1}{\Omega'} \sum_{\sigma\in S_n} |\mathbf{p}_{\sigma^{-1}(1)}\rangle \otimes \cdots \otimes |\mathbf{p}_{\sigma^{-1}(n)}\rangle, \tag{108}$$

providing the result proposed in Eq. (107) and recreating the work of Ref. [20] as a special case of a more general framework.

2. *Rotational invariance*

The novelty of the projection technique that we propose in the LCU framework is that it allows expansion to many other possible symmetries beyond permutation invariance. Here, we shall outline another example, rotationally invariant encodings of point clouds that contain four points utilizing an $SU(2)^{\otimes 4}$ invariant encoding. This is motivated by the fact that $SU(2)$ is the double cover of $SO(3)$, which is the group that corresponds to rotations in three dimensions, which means that each element of $SO(3)$ corresponds to exactly two elements in $SU(2)$.

In this framework, we consider point clouds in which the points $\mathbf{p}_i$ are represented in spherical coordinates $\mathbf{p}_i = (r_i, \theta_i, \phi_i)$. We shall ignore the radial component r_i for the sake of simplicity, although this could be included in any practical implementation in an appropriate manner.

We propose an encoding in which for each point $\mathbf{p}_i = (r_i, \theta_i, \phi_i)$ is encoded into the following quantum state,

$$|\mathbf{p}_i\rangle = \cos\left(\frac{\theta_i}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta_i}{2}\right)|1\rangle, \tag{109}$$

such that the θ_i and ϕ_i angle of the point is encoded into the respective angles of a single qubit in the Bloch-sphere representation.

Rotation of a point cloud corresponds to rotating every individual point by the same amount and in the same direction such that $(r_i, \theta_i, \phi_i) \to (r_i, \theta_i + \Delta\theta, \phi_i + \Delta\phi) \forall i \in [1, n]$. Considering this in terms of the quantum encoded state above, it would correspond to the application of the same rotation $U_{SU(2)} \in SU(2)$ on each qubit $|\mathbf{p}_i\rangle$, rotating every qubit state about the Bloch sphere by the same amount. In order to have a quantum encoded point cloud that is invariant to rotations of the point cloud, it would require the following invariance,

$$\begin{aligned}&|\mathbf{p}_1\rangle|\mathbf{p}_2\rangle|\mathbf{p}_3\rangle|\mathbf{p}_4\rangle\\ &\quad = U_{SU(2)}|\mathbf{p}_1\rangle U_{SU(2)}|\mathbf{p}_2\rangle U_{SU(2)}|\mathbf{p}_3\rangle U_{SU(2)}|\mathbf{p}_4\rangle,\end{aligned} \tag{110}$$

which would be invariance under the action $SU(2)^{\otimes 4}$.

We consider irreducible representations of the group S_4, where the representations of group elements $\sigma \in S_4$ correspond to the SWAP gates $U_\sigma = \text{SWAP}_\sigma$ that permute the constituent points $|\mathbf{p}_i\rangle$ accordingly. We are able to use this because Schur-Weyl duality (for a detailed discussion, see Appendix B) provides a relation between irreducible finite-dimensional representations between the symmetric group and general linear group. This means that for the one-dimensional trivial representation of $SU(2)^{\otimes n}$ of multiplicity m, there will correspond an m-dimensional representation of S_n covering the same subspace [68,77,78]. The character table for S_4 is shown in Table II.

TABLE II. The character table for the group S_4, which corresponds to the group of all permutations of the set $\{1, 2, 3, 4\}$. Each conjugacy class is characterized by the cycle structure of its permutations, where a cycle of length k corresponds to a sequence of k elements being permuted cyclically. The identity is represented by (I), 2-cycles are represented by (12) and correspond to swapping any two elements. 3-cycles and 4-cycles are represented by (123) and (1234), respectively. Disjoint cycles are those that affect different sets of elements and can be represented together, such as (12)(34). There are five irreducible representations, with their characters denoted by χ_1, χ_2, χ_3, χ_4, and χ_5.

	Conjugacy class				
	$C_{(I)}$	$C_{(12)}$	$C_{(12)(34)}$	$C_{(123)}$	$C_{(1234)}$
Number of elements	1	6	3	8	6
χ_1	1	1	1	1	1
χ_2	1	−1	1	1	−1
χ_3	2	0	2	−1	0
χ_4	3	−1	−1	0	1
χ_5	3	1	−1	0	−1

If we utilize the irreducible-representation projection procedure to project the state $|\mathbf{p}_1\rangle \otimes |\mathbf{p}_2\rangle \otimes |\mathbf{p}_3\rangle \otimes |\mathbf{p}_4\rangle$ to the subspace of irreducible representation $r = 3$, with characters χ_3, then this corresponds to projecting to the basis states [68,77,78]

$$\begin{aligned}|d_1\rangle = \frac{\sqrt{3}}{6}(&2|0011\rangle + 2|1100\rangle - |0101\rangle\\ &- |1010\rangle - |0110\rangle - |1001\rangle)\end{aligned} \tag{111}$$

and

$$|d_2\rangle = \frac{1}{2}(|0101\rangle + |1010\rangle - |0110\rangle - |1001\rangle). \tag{112}$$

When writing the action of $(U \otimes U \otimes U \otimes U)$ in the Schur basis, as shown in Fig. 21, it can be seen that the above basis states, $|d_1\rangle$ and $|d_2\rangle$, are invariant under the action of $(U \otimes U \otimes U \otimes U)$, as they are in the subspace associated with the trivial representation of $SU(2)^{\otimes 4}$ [77,78]. We are able to reach this space through projecting in S_4 due to Schur-Weyl duality, since the irreducible subspaces of S_4 and $SU(2)^{\otimes 4}$ are simultaneously block diagonalized. This means that we have effectively projected the quantum encoded point-cloud state to a subspace that is invariant with respect to 3D rotations of the point cloud. This is a desirable symmetry for point clouds, as they are naturally invariant to 3D rotations.

In order to demonstrate this numerically, we show in Fig. 17 that the rotationally invariant encoding introduced by this projection gives a constant overlap equal to unity when comparing two identical but rotated point clouds. This constant overlap of unity is maintained as the degree of rotation is increased, demonstrating that the encoding is producing rotationally invariant encoded states. It is also demonstrated that without applying this symmetry projection, a noninvariant encoding produces an overlap between identical but rotated point clouds that decreases as the magnitude of the rotation is increased. This is significant, as it means that without a rotationally invariant encoding, the exact same point cloud, rotated by π radians, would produce a quantum state that appears to be completely different and will give zero overlap with the original orientation of the point cloud.

D. Symmetric subspace amplification

The rotationally invariant encoding discussed previously, as well as previous work on permutation-invariant encodings [20], have been implemented in a binary fashion: the initial quantum state is either projected to a fully permutation-invariant state or remains unchanged. Such an implementation significantly reduces the dimensionality

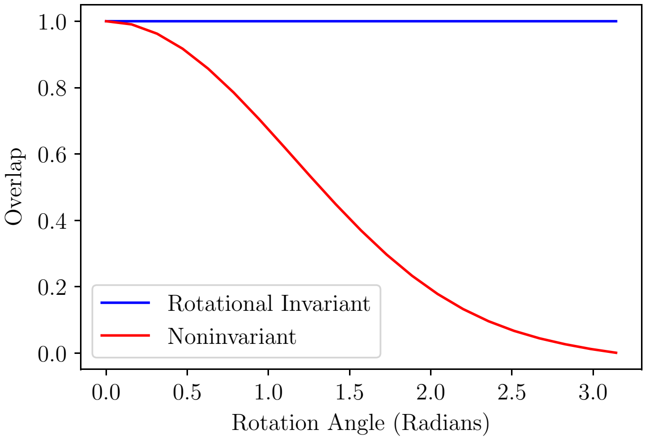

FIG. 17. The overlap between two identical point clouds that have been rotated about a random direction in three dimensions as the amount of rotation is increased, for the rotational invariant encoding and generic noninvariant encoding. To generate this plot, we have randomly generated $n = 4$ points on a sphere of radius 1 to act as the point cloud. We have then encoded the (θ, ϕ) radial coordinates of each point into a qubit using the specification of Eq. (109). This results in an initial state $|P\rangle_{\text{init}} = |\mathbf{p}_1\rangle|\mathbf{p}_2\rangle|\mathbf{p}_3\rangle|\mathbf{p}_4\rangle$. We have then repeated the procedure after the point-cloud data have been transformed by a 3D rotation in a random direction by some angle Θ to produce noninvariant state $|P\rangle_\Theta$. To generate rotationally invariant states $|P_{\text{rot}}\rangle$, we have applied the result in Theorem 3 to project $|P\rangle_\Theta$ states to the irreducible representation denoted by $r = 3$ as specified in Table II, which via Schur-Weyl duality lie in the trivial representation subspace of $SU(2)^{\otimes 4}$ and are hence rotationally invariant under our data-encoding setup. This means that we now have rotationally invariant encoded states $|P_{\text{rot}}\rangle_\Theta$ for all Θ values. The blue line plots the overlap $\langle P_{\text{init}}|P_{\text{rot}}\rangle_\Theta$ and the red line plots the overlap $\langle P_{\text{init}}|P\rangle_\Theta$ as the point-cloud data rotation angle Θ is varied over $[0, \pi]$. As the overlap $\langle P_{\text{init}}|P_{\text{rot}}\rangle_\Theta$ is constant, we therefore numerically confirm that projection to the irreducible representation denoted by $r = 3$ creates rotationally invariant states.

of the encoding, yielding benefits in certain scenarios through improved generalization of the model. This section explores the potential advantages of introducing a continuous spectrum of invariance with respect to some symmetry, rather than adhering to binary extremes, and investigates whether optimizing this aspect as a hyperparameter could enhance classification performance in quantum machine-learning (QML) models.

Recent studies suggest that permutation-equivariant variational circuits may be classically tractable under specific conditions [74,75], indicating that a drastic reduction in dimensionality might increase the likelihood of classical simulatability. This raises concerns that fully projecting to a polynomially sized subspace could render algorithms susceptible to classical simulation. To address these issues, we propose a novel approach that involves partially amplifying the portion of the quantum state into a given invariant space. Specifically, we will consider amplifying the permutation-invariant subspace, although any irreducible subspace could be chosen. Apart from edge cases, the quantum state will still, in general, be exponential in dimension but the permutation-symmetric subspace will have a higher weighting compared to all other subspaces; the amount of permutation-symmetry amplification is governed by a hyperparameter α that will therefore also affect the expressivity of the encoding. This method provides enhanced control over training performance and facilitates the tuning of the model to achieve optimal results. For data types with inherent permutation symmetry, such as point-cloud data, the optimal global-classification function must also exhibit permutation invariance. However, the specific outcome of a QML model depends on the circuit architecture, classification protocol, and inherent limitations of QML models, suggesting that a purely permutation-invariant QML algorithm may not necessarily be the closest to the global solution. We hypothesize that by incrementally adjusting the degree of permutation invariance in the model, we can converge toward a value that is closer to the true global optimum performance than is possible with an noninvariant, or fully permutation invariant, model. This concept is illustrated in Fig. 18.

The framework introduced previously allows the implementation of any linear combination of projections to an irreducible-representation subspace, with full control of the ratios of these projections. One can consider decomposing a state into its irreducible-representation components, as shown in Fig. 19, and gaining control over the relative ratios between the amplitude components, allowing them to be adjusted to suit the model and data.

Via the LCU framework, we have previously shown in Theorem 4 that it is possible to implement a projection onto a quantum state in the following form:

$$\sum_r a_r \hat{P}_r |\psi\rangle, \tag{113}$$

where the a_r parameters can be controlled via the initial state of the ancilla qubits as

$$|c\rangle = \frac{1}{\Omega} \sum_{r=1}^{R} a_r n_r |b_r\rangle. \tag{114}$$

In order to implement the symmetric-subspace-amplification algorithm, one can define a parameter α that parametrizes the amount of symmetry in the encoding. Then we can assign the following a_r values:

$$a_r = \begin{cases} a_1 = 1, & \text{for } r = 1, \\ a_r = (1 - \alpha), & \text{for } r \neq 1. \end{cases} \tag{115}$$

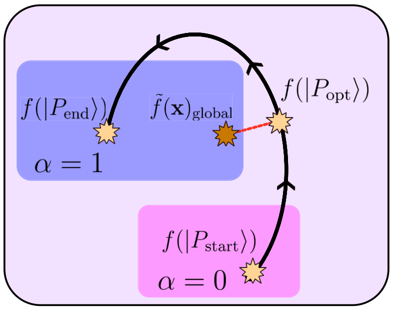

FIG. 18. A diagram providing a conceptual overview of the symmetric-subspace-amplification motivation. We start at $\alpha = 0$ with a state encoding $|P_{\text{start}}\rangle = \bigotimes_{i=1}^{n} |\mathbf{p}_i\rangle$ corresponding to an initial state without symmetry amplification, which is input to some classification model $f(\cdot)$. As α increases, the state moves through the solution space until finally we have a fully symmetric state at $\alpha = 1$ where the input state will be equal to $|P_{\text{end}}\rangle = 1/\Omega' \sum_{\sigma \in S_n} |\mathbf{p}_{\sigma_1}\rangle |\mathbf{p}_{\sigma_2}\rangle \ldots |\mathbf{p}_{\sigma_n}\rangle$, as initially proposed in Ref. [20]. If we are dealing with data that are known to be permutation symmetric, then the global optimal model for the input data $\tilde{f}(\mathbf{x})_{\text{global}}$ (which could be classical or quantum) must be within the fully permutation-symmetric subspace of models. However, the precise model that yields the global optimum remains unknown and may not correspond to the function $f(\cdot)$ used in this particular case, suggesting that an intermediate value of α might align more closely with the global optimum than the fully symmetric projection at $\alpha = 1$. In the figure depicted, the red line illustrates the deviation between the global model solution and the nearest solution obtained at an intermediate value of α, in which the input state is denoted $|P_{\text{opt}}\rangle$.

Hence, the initial ancilla state will be of the form

$$|c\rangle = \frac{1}{\Omega}\left(n_1|b_1\rangle + (1-\alpha)\sum_{r=2}^{R} n_r|b_r\rangle\right). \tag{116}$$

The α term can be varied to adjust the amount of permutation symmetry. When $\alpha = 0$, this corresponds to an equal projection on all irreducible representations, since the resulting operation is an equal projection onto all subspaces (weighted by the dimension of the subspace) and hence will leave the state unchanged. In contrast, the case in which $\alpha = 1$ will result in projection onto the symmetric subspace only. As α increases, this corresponds to continuously amplifying the portion of the state that lies in the symmetric subspace.

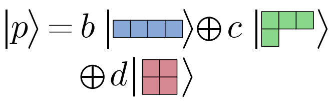

FIG. 19. A state can be viewed as a linear combination of its projection onto its irreducible representations. Utilizing this projection algorithm, we are able to change the relative weightings of b, c, and d, effectively amplifying and reducing the portion of the original state that lies in the respective irreducible-representation subspace.

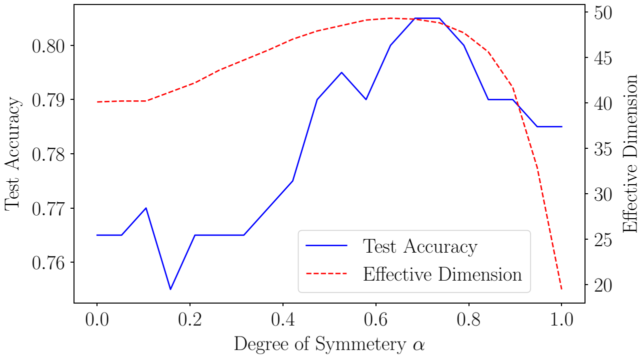

FIG. 20. The test accuracy and effective dimension as the degree of permutation symmetry is varied. The result is the average over ten repeated experiments, each with 100 data points each of the sphere and torus classification data set using quantum kernel estimation followed by a support vector machine (SVM) classification [20,80,81]. The effective dimension is found by explicitly calculating the higher-dimensional encoding, performing principal-component analysis, and calculating the number of dimensions required to explain 95% of the variance of the data set. We show that an intermediate value of the permutation symmetry around $\alpha = 0.7$ results in the best performance for this classification.

In order to investigate the potential of symmetric subspace amplification, we have implemented the model proposed here utilizing the aforementioned technique alongside QISKIT STATEVECTOR_SIMULATOR [79]. This has been used to encode point-cloud data and perform classification on the sphere and torus point-cloud data set as specified in Ref. [20]. More details can be found in Appendix C.

We record the average test accuracy over ten experiments for a given α and then vary α to see which value is optimal for the data. The results presented in Fig. 20 illustrate that increasing the degree of symmetry yields the highest test accuracy at an intermediate value, demonstrating the potential advantage of this technique. Moreover, it is observed that while a fully symmetric projection ($\alpha = 1$) surpasses the performance of the nonsymmetrized configuration ($\alpha = 0$), there is a significant reduction in dimensionality as we approach a fully symmetric state. This reduction corresponds to projecting onto a polynomial-size space for the case of symmetric subspaces [20]. The α parameter is the ratio of the symmetric subspace to all other subspaces. In general, one could select any relative weighting for all the irreducible subspaces, at the cost of having more parameters to optimize. We envisage that in practice, a grid search would be used to find these hyperparameter values and that they would be optimized using a cross-validation set, allowing the final classification performance to be assessed on an unseen testing set.

A key concern with fully symmetric states is their potential for efficient classical simulation due to this polynomially sized space. In contrast, the technique of partial symmetrization not only achieves higher accuracy but also resides in a higher-dimensional space, possibly making it more challenging to simulate classically. While computer-vision data can consist of thousands of points, there are example data sets that may be more suitable in the near term, such as particle-physics collision-classification tasks in which there may be fewer than ten particles (points) in a given event [80]. This would likely be a nearer-term goal for the application of point-cloud-specific algorithms, although the method described in this paper could be applied to many symmetries across various different data types.

V. CONCLUSIONS

In this study, we have demonstrated the applicability and flexibility of the LCU method in enhancing QML architectures. Our work provides several implementations of classical machine-learning structures within the quantum domain, not only achieving a foundational translation of these concepts but also demonstrating potential for computational advantages.

The implementation of quantum ResNet demonstrates a procedure that can avoid the trainability issues associated with barren plateaus by allowing shallow-depth components to survive in the final loss function. Understanding the exact dynamics of the nonunitary terms is still a question for further research. We have demonstrated that by parametrizing the strength of the residual connections through β_l terms, one can increase the lower-bound probability of success of the LCU method. This provides a possible avenue to tackling the key problem of the LCU method, which is that the implementations can only be performed probabilistically.

We have also demonstrated an implementation of quantum native average-pooling layers commonly applied in CNNs. These layers have demonstrated success in classical techniques and therefore have the potential to improve quantum CNNs. While classically one would need to calculate $\mathcal{O}(N^2)$ averages, the quantum parallelism inherent in our technique applies the averaging to all pixels (which are amplitude encoded into a quantum state) simultaneously. By generalizing our work further, we can recreate quantum convolutional filters as has been previously proposed [16], while utilizing exponentially fewer controlled unitary operations.

Finally, we have demonstrated an integration of irreducible-representation projections into a quantum framework, allowing for the encoding of data symmetries directly into the quantum state, which has the potential to enhance the generalization capabilities of the model across various data structures. This general irreducible-subspace-projection framework includes previous work regarding permutation-invariant encodings for point clouds [20] as a special case, while in Sec. IV C 2 we have introduced a novel rotationally invariant encoding for point-cloud data using the S_4 group. We have further introduced, in Sec. IV D, a method for parametrizing the amount of symmetry in an encoding as a tunable hyperparameter. This allows the quantum state to remain in an exponentially large space, while certain subspaces are amplified relative to each other, helping to mitigate some of the simultability concerns associated with fully projecting to a polynomially sized subspace. This added flexibility has resulted in an improved performance for point-cloud data when encoding an intermediate amount of permutation symmetry as compared to either a fully permutation-invariant or noninvariant encoding. These have been illustrative examples of how the framework could be utilized, and there could be many possible symmetries and data sets for which this framework could be similarly adapted and optimized.

As the field of quantum computing continues to mature, the methods and frameworks presented here offer promising avenues for developing more sophisticated quantum algorithms that leverage both the computational benefits of quantum mechanics and the established successes of classical machine learning architectures. By utilizing LCU methods, we have shown that nonunitary operations can be applied in a QML setting; this research not only extends the theoretical possibilities and flexibility of QML algorithms but also provides practical frameworks for their application, setting the stage for further evolution of QML models.

ACKNOWLEDGMENTS

This research was supported by the Australian Research Council through Grant No. DP210102831. J.H. acknowledges the support of the Australian Government Research Training Program Scholarship and the support of the N. D. Goldsworthy Scholarship. This work was supported by the University of Melbourne through the establishment of an IBM Quantum Network Hub at the University. We thank Dougal Davis for early discussions on representation theory that helped aid the development of the irreducible-subspace-projection work. We also thank Giulio Crognaletti for discussions on current quantum ResNet literature. This work was supported in part by the Australian Research Council Centre of Excellence for Quantum Biotechnology (CE230100021).

J.H. envisaged the project, initially focused on irreducible-subspace projections, with M.S. and L.H., as an extension of their previous work [20]. J.H. devised

the ResNet and average-pooling-layers applications. The theorems and numerical simulations were developed by J.H., with M.W.'s help. M.S. and L.H. supervised the overall project. All authors contributed in reviewing the manuscript.

APPENDIX A: CONJUGACY-CLASS IMPLEMENTATION

In the implementation of irreducible-subspace projections detailed in Sec. IV, an operator $\hat{\chi}$ was required that contained the character of every element in the group G. This meant that the dimension of the matrix scaled with $|G|$, which can be problematic for some groups, such as the permutation group S_n, where $|S_n| = n!$. To tackle this, in this appendix we will consider a unitary operator $\tilde{\chi}$ that contains the character for each conjugacy class in the group, with additional ancilla-qubit registers per conjugacy class prepared in an equal superposition that then acts to distribute these characters to the corresponding group elements. This effectively only requires encoding the character table of the group into a unitary operator and hence $\tilde{\chi}$ scales with the number of conjugacy classes of G. This has the potential to drastically reduce the size of the unitary that must be applied, at the expense of requiring additional ancilla qubits. We show that utilizing this circuit structure recreates the result of Theorem 4 and is therefore a valid alternative approach.

The character orthogonality theorem [72] stated previously in Eq. (91) relates to a sum over all the elements of the group. As the character is the same for all elements in a conjugacy class, this can be adjusted to a sum over all conjugacy classes $\nu_i \in \mathcal{C}$ by introducing a term $d_{\nu_i} = |\nu_i|$ that represents the number of group elements in the conjugacy class ν_i. Therefore, the character orthogonality theorem can be written as

$$\frac{1}{|G|}\sum_{\nu_i \in \mathcal{C}} d_{\nu_i}\chi_j(\nu_i)\chi_k(\nu_i)^* = \delta_{jk}. \tag{A1}$$

Therefore, to create a matrix with orthonormal columns (to ensure that $\tilde{\chi}$ is unitary), the characters for a conjugacy class ν must be weighted by the square root of the number of elements in the conjugacy class $\sqrt{d_{\nu_i}}$. Hence, we require that

$$\tilde{\chi}|b_r\rangle = \frac{1}{\sqrt{|G|}}\sum_{\nu_i \in \mathcal{C}} \sqrt{d_{\nu_i}}\chi_r(\nu_i)^*|\nu_i\rangle, \tag{A2}$$

where in this case the conjugacy classes $\nu_i \in \mathcal{C}$ are labeling the basis states $|\nu_i\rangle \equiv |b_i\rangle$, defined as $\{|b_j\rangle\}_{j\in[1,2^k]} \in (\mathbb{C}^2)^{\otimes k}$ the basis states of the 2^k-dimensional Hilbert space for the k-qubit ancilla register denoted by $\mathcal{H} = (\mathbb{C}^2)^{\otimes k}$. These can be taken to be the computational basis states.

TABLE III. The character table for the group S_3. There are three irreducible representations, with their characters denoted by χ_1, χ_2, and χ_3. The conjugacy classes of S_3 are $C_{(I)} = \{I\}$, $C_{(12)} = \{\sigma_{12}, \sigma_{13}, \sigma_{23}\}$, and $C_{(123)} = \{\sigma_{231}, \sigma_{312}\}$.

	Conjugacy class ν_i		
	$C_{(I)}$	$C_{(12)}$	$C_{(123)}$
Number of elements d_{ν_i}	1	3	2
χ_1	1	1	1
χ_2	2	0	−1
χ_3	1	−1	1

If we consider S_3 with its character table given in Table III, then we see that in this case the appropriate unitary matrix to be constructed would be

$$\tilde{\chi} = \frac{1}{\sqrt{6}}\begin{bmatrix} 1 & 2 & 1 & 0 \\ \sqrt{3} & 0 & -\sqrt{3} & 0 \\ \sqrt{2} & -\sqrt{2} & \sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{A3}$$

This successfully allows for the construction of a unitary matrix $\tilde{\chi}$ that scales in size with the number of conjugacy classes as opposed to the number of group elements. We have therefore satisfied the constraint that $\tilde{\chi}$ is unitary and can be implemented on a quantum device. In general, $\tilde{\chi}$ can be written

$$\tilde{\chi} = \frac{1}{\sqrt{|G|}}\sum_{r=1}^{R}\sum_{\nu_i \in \mathcal{C}} \sqrt{d_{\nu_i}}\chi_r(\nu_i)^*|\nu_i\rangle\langle b_r| + \sum_{r=R+1}^{2^k}(\ldots)\langle b_r|, \tag{A4}$$

where $(\ldots)$ collects terms associated with $|b_r\rangle, r > R$ that will not be used in the construction.

Similarly to the main text, the first step is the preinitialization of the ancilla qubits to some combination of states that index the different irreducible representations

$$\hat{\gamma}|b_1\rangle|\psi\rangle = \frac{1}{\Omega}\sum_{r=1}^{R} a_r n_r |b_r\rangle|\psi\rangle, \tag{A5}$$

where $|b_1\rangle$ is the initial state of the ancilla-qubit register, usually assumed to be $|0\rangle^{\otimes k}$. By applying $\tilde{\chi}$ to this preinitialized ancilla state, one can write

$$\begin{aligned}\tilde{\chi}\hat{\gamma}|b_1\rangle|\psi\rangle &= \tilde{\chi}\frac{1}{\Omega}\sum_{r=1}^{R} a_r n_r |b_r\rangle|\psi\rangle \\ &= \frac{1}{\Omega\sqrt{|G|}}\sum_{r=1}^{R} a_r n_r \sum_{\nu_i \in \mathcal{C}} \sqrt{d_{\nu_i}}\chi_r(\nu_i)^*|\nu_i\rangle|\psi\rangle,\end{aligned} \tag{A6}$$

where, as previously stated, $|\nu_i\rangle \equiv |b_i\rangle$ are the same basis states, usually taken to be the computational basis states, but we have switched from $|b_r\rangle$ being indexed by the irreducible representation r to the basis states being labeled by the conjugacy class ν_i. This contrasts with $\hat{\chi}$ in the main text, where the group elements label the ancilla states. After application of $\tilde{\chi}$, the ancilla qubits now have basis states corresponding to the conjugacy classes $|\nu_i\rangle$. The selection operator requires the creation of additional ancilla-qubit registers, denoted by $|\omega_{\nu_i}\rangle$, for each conjugacy class, which are encoded into an equal superposition of $d_{\nu_i} = |\nu_i|$ states

$$|\omega_{\nu_i}\rangle = \frac{1}{\sqrt{d_{\nu_i}}}\sum_{l_{\nu_i}=1}^{d_{\nu_i}} |l_{\nu_i}\rangle, \tag{A7}$$

where $|l_{\nu_i}\rangle$ are basis states $\{|l_{\nu_i}\rangle\} \in (\mathbb{C}^2)^{\otimes k_{\nu_i}}$ of the $2^{k_{\nu_i}}$-dimensional Hilbert space for the k_{ν_i}-qubit ancilla register $|\omega_{\nu_i}\rangle$, which can be taken to be the computational basis states. The group elements l_{ν_i} that are contained in the conjugacy class ν_i now label the basis states in register $|\omega_{\nu_i}\rangle$. Each register has the requirement $k_{\nu_i} \geq \log(d_{\nu_i})$ to ensure that there are sufficient qubits in the register such that each element in ν_i has a corresponding basis state $|l_{\nu_i}\rangle$. The $|\omega_{\nu_i}\rangle$ are prepared on a k_{ν_i} qubit register that is initially in a basis state $|b_1^{\nu_i}\rangle$ through the operation

$$\hat{\omega}_{\nu_i} = \frac{1}{\sqrt{d_{\nu_i}}}\sum_{l_{\nu_i}=1}^{d_{\nu_i}} |l_{\nu_i}\rangle\langle b_1^{\nu_i}| + \sum_j (\ldots)\langle b_j^{\nu_i}|, \tag{A8}$$

where $(\ldots)$ summarizes terms associated with $\langle b_j^{\nu_i}|, j \geq 2$ that will not be present in any calculations.

Overall, the operations up to this point can be written as

$$\begin{aligned}&\bigotimes_{\nu_i'} |\omega_{\nu_i'}\rangle \otimes \tilde{\chi}\hat{\gamma}|b_1\rangle \otimes |\psi\rangle \\ &\quad= \frac{1}{\Omega\sqrt{|G|}}\left(\left(\bigotimes_{\nu_i'}\frac{1}{\sqrt{d_{\nu_i'}}}\sum_{l_{\nu_i'}=1}^{d_{\nu_i'}}|l_{\nu_i'}\rangle\right)\right. \\ &\qquad\left.\otimes\sum_{r=1}^{R} a_r n_r \sum_{\nu_i\in\mathcal{C}}\sqrt{d_{\nu_i}}\,\chi_r(\nu_i)^*|\nu_i\rangle|\psi\rangle\right).\end{aligned} \tag{A9}$$

The controlled operations are performed similarly to those in the main text, except that in this case the group element $g \in G$ is indexed by both ν_i, which determines the conjugacy class, and l_{ν_i}, which indexes the element within the given conjugacy class. In this case, $S_{\text{SELECT}} \equiv \hat{S}$ will be controlled by both the conjugacy-class state $|\nu_i\rangle$ and the element selector within the conjugacy class $|l_{\nu_i}\rangle$ of the register $|\omega_{\nu_i}\rangle$ associated with the conjugacy class ν_i. The action of this selection operator can be defined as

$$\hat{S}|l_{\nu_i}\rangle|\nu_i\rangle|\psi\rangle = |l_{\nu_i}\rangle|\nu_i\rangle U_{\nu_i,l_{\nu_i}}|\psi\rangle, \tag{A10}$$

which is implemented by each unitary $U_{\nu_i,l_{\nu_i}}$ being controlled by both the $|l_{\nu_i}\rangle$ and $|\nu_i\rangle$ states. Applying this operator, one can see that the effect is

$$\begin{aligned}&\hat{S}\left(\bigotimes_{\nu_i'}|\omega_{\nu_i'}\rangle\otimes\tilde{\chi}\hat{\gamma}|b_1\rangle\otimes|\psi\rangle\right) \\ &\quad= \frac{1}{\Omega\sqrt{|G|}}\left(\sum_{\nu_i}\left(\bigotimes_{\nu_i'\neq\nu_i}\frac{1}{\sqrt{d_{\nu_i'}}}\sum_{l_{\nu_i'}=1}^{d_{\nu_i'}}|l_{\nu_i'}\rangle\right)\right. \\ &\qquad\left.\otimes\sum_{r=1}^{R}\sum_{l_{\nu_i}=1}^{d_{\nu_i}} a_r n_r\chi_r(\nu_i)^*|l_{\nu_i}\rangle|\nu_i\rangle U_{\nu_i,l_{\nu_i}}|\psi\rangle\right) \\ &\quad= \sum_{\nu_i}\bigotimes_{\nu_i'\neq\nu_i}|\omega_{\nu_i'}\rangle\sum_{r=1}^{R}\sum_{l_{\nu_i}=1}^{d_{\nu_i}}\frac{a_r n_r\chi_r(\nu_i)^*}{\Omega\sqrt{|G|}}|l_{\nu_i}\rangle|\nu_i\rangle \\ &\qquad\otimes U_{\nu_i,l_{\nu_i}}|\psi\rangle,\end{aligned} \tag{A11}$$

where we are being flexible with the precise positioning in the tensor product of the unused registers $|\omega_{\nu_i'\neq\nu_i}\rangle$ for each ν_i' term for the sake of readability. The notation we are using is that for a given ν_i element in the sum over all conjugacy classes, all $|\omega_{\nu_i'}\rangle$ registers that will not affect the S_{SELECT} operator because $\nu_i' \neq \nu_i$ are written first, with the case $\nu_i' = \nu_i$ written to the right of this. In reality, the register position will be different for each ν_i but this becomes difficult to denote in the notation.

We can now uncompute all $\hat{\omega}_{\nu_i}$ registers by noting that

$$\hat{\omega}_{\nu_i}^{\dagger} = \frac{1}{\sqrt{d_{\nu_i}}}\sum_{l_{\nu_i}=1}^{d_{\nu_i}}|b_1^{\nu_i}\rangle\langle l_{\nu_i}| + \sum_j(\ldots)|b_j^{\nu_i}\rangle. \tag{A12}$$

This means that if we require these registers to be measured in the $|b_1^{\nu_i}\rangle$, then we can ignore the terms summarized by $(\ldots)$ associated with $|b_j^{\nu_i}\rangle, j \geq 2$. Application of $\hat{\omega}_{\nu_i}^{\dagger}$ to all such registers gives

$$\begin{aligned}&\bigotimes_{\nu_i}\hat{\omega}_{\nu_i}^{\dagger}\hat{S}\left(\bigotimes_{\nu_i'}|\omega_{\nu_i'}\rangle\otimes\tilde{\chi}\hat{\gamma}|b_1\rangle\otimes|\psi\rangle\right) \\ &\quad= \sum_{\nu_i}\bigotimes_{\nu_i}|b_1^{\nu_i}\rangle\sum_{r=1}^{R}\sum_{l_{\nu_i}=1}^{d_{\nu_i}}\frac{a_r n_r\chi_r(\nu_i)^*}{\Omega\sqrt{d_{\nu_i}|G|}}|\nu_i\rangle\otimes U_{\nu_i,l_{\nu_i}}|\psi\rangle \\ &\qquad+\sum_{j\geq2}(\ldots)|b_j^{\nu_i}\rangle.\end{aligned} \tag{A13}$$

We measure the registers that previously contained $\hat{\omega}_{\nu_i}$ states and discard unless we measure all $|b_1^{\nu_i}\rangle$ states $\forall \nu_i$. This discards the terms collected by the $(\ldots)$ in the previous equation.

The final steps consist of applying $\tilde{\chi}^\dagger$ and measuring the conjugacy class $|\nu_i\rangle$ ancillas to be in the initial state $|b_1\rangle$. Due to the fact that the state $|b_1\rangle$, which is commonly assumed to be $|0\rangle^{\otimes k}$, is used to index the first representation, we can write

$$\langle b_1|\tilde{\chi}^\dagger = \frac{1}{\sqrt{|G|}} \sum_{\nu_i \in \mathcal{C}} \sqrt{d_{\nu_i}}\chi_1(\nu_i)\langle \nu_i|, \tag{A14}$$

where, noticeably, $\chi_1(\nu_i) = 1, \forall \nu_i$ due to the fact that we are free to define $r = 1$ to always correspond to the trivial representation when constructing $\tilde{\chi}$. Applying this operator to the conjugacy-class ancilla qubits in the circuit therefore results in

$$\left(\langle b_1|\tilde{\chi}^\dagger\right) \cdot \left(\bigotimes_{\nu_i} \langle b_1^{\nu_i}|\hat{\omega}^\dagger_{\nu_i}\hat{S}\left(\bigotimes_{\nu_i'}|\omega_{\nu_i'}\rangle \otimes \tilde{\chi}\hat{\gamma}|b_1\rangle \otimes |\psi\rangle\right)\right)$$
$$= \frac{1}{\Omega'}\sum_{\nu_i}\sum_{r=1}^{R}\sum_{l_{\nu_i}=1}^{d_{\nu_i}} a_r \frac{n_r}{|G|}\chi_r(\nu_i)^* U_{\nu_i, l_{\nu_i}}|\psi\rangle, \tag{A15}$$

where $\Omega' = \Omega\sqrt{\pi_S}$ is the normalization constant of the final state. Due to the fact that conjugacy classes partition the group G, the labels of the conjugacy class ν_i along with the labels of the elements within each conjugacy class l_{ν_i} will uniquely index each group element. In addition, all elements in the same conjugacy class have the same character, i.e., $\chi_r(g_1) = \chi_r(g_2)$ for $g_1, g_2 \in \nu_i$. Hence, we can relabel the above expression in terms of group elements g, rather than ν_i and l_{ν_i}, to explicitly see that it can be written as

$$\frac{1}{\Omega}\sum_{r=1}^{R} a_r \frac{n_r}{|G|}\sum_{g\in G}\chi_r(g)^* U_g|\psi\rangle. \tag{A16}$$

This is the precise form of Eq. (86) in Theorem 4. Hence, we have provided another framework that satisfies the theorem in the main text. As the final step only consisted of reordering the indexing, it follows that the probability of success π_S will be the same as in the main text.

The dimension of $\tilde{\chi}$ scales with the number of conjugacy classes of the group G as opposed to $\hat{\chi}$ in the main text, which scales with the number of elements in the group. This means that $\tilde{\chi}$ could potentially be much easier to implement than $\hat{\chi}$. However, a caveat is that an additional ancilla-qubit register $|\omega_{\nu_i}\rangle$ is required for each conjugacy class ν_i, where each register will contain a number of qubits of the order $\mathcal{O}(\log(d_{\nu_i}))$. This presents a trade-off between unitary implementation difficulty and the number of ancilla qubits required. The true difficulty in implementing either unitary will ultimately depend on the group, but we provide both of these general techniques as possible starting points for future implementations.

APPENDIX B: ROTATIONAL INVARIANCE VIA SCHUR-WEYL

In this appendix, we demonstrate how certain irreducible representations of S_n could correspond to rotationally invariant encoding states when using the encoding specified in Sec. IV C 2. Following the work and explanations of Refs. [68,77] and, in particular, Ref. [78], we reproduce an overview of Schur-Weyl duality and the concept of the Schur basis and add discussion at points linking this to how certain irreducible-representation subspaces would correspond to rotationally invariant point-cloud encodings in the case considered in Eq. (109).

1. Introduction to Schur transforms

The rotational invariant point-cloud encoding that we propose in Sec. IV C 2 relies on a property of the probabilistic irreducible-subspace-projection circuit when considering the group S_n, which in effect performs a projection to basis states of the Schur basis. Circuits that implement quantum Schur transforms have previously been introduced, such as those that implement them through the unitary group by iteratively applying the Clebsch-Gordan transforms [82] and alternative methods that implement it by considering the action of the symmetric group using quantum Fourier transforms [83]. Indeed, further investigation on whether the use of these existing algorithms could be adapted to more efficiently refine our procedure would be an interesting further research direction, as our implementation derives from the most general case of Theorem 4, which applies to any group G and therefore may not be the most efficient in practice.

The Schur transform is related to the concept of Schur-Weyl duality. In general, we can consider n qudits of dimension d inside a vector space $(\mathbb{C}^d)^{\otimes n}$, with a computational basis written as $|i_1\rangle \otimes |i_2\rangle \otimes \ldots \otimes |i_n\rangle$. There are two representations on this space that are related to each other by Schur-Weyl duality. One of the representations is that of the symmetric group S_n, the elements of which consist of all possible permutations of n objects. The representation of S_n on this vector space would simply consist of permuting the qudits. For a given group element $\sigma \in S_n$, one can write the action of the representation $P(\sigma)$ on this vector space as

$$P(\sigma)|i_1\rangle \otimes \ldots \otimes |i_n\rangle = |i_{\sigma^{-1}(1)}\rangle \otimes \ldots \otimes |i_{\sigma^{-1}(n)}\rangle. \tag{B1}$$

Hence, each $\sigma \in S_n$ simply corresponds to a different permutation of the n qudits.

The second group to consider is the group of all $d \times d$ unitary operators $U \in \mathcal{U}_d$. As each qudit forms a d-dimensional vector, we see that the natural representation $Q(U)$ of the group $\mathcal{U}_d$ acting on the vector space $(\mathbb{C}^d)^{\otimes n}$ would correspond to the n-fold product action, where the same U is applied to each qudit. This can be written as

$$Q(U)|i_1\rangle \otimes \ldots \otimes |i_n\rangle = U|i_1\rangle \otimes \ldots \otimes U|i_n\rangle. \tag{B2}$$

The two actions $Q(U)$ and $P(\sigma)$ are fully reducible and hence can be written as a direct sum of their irreducible representations:

$$P(\sigma) \rightarrow \bigoplus \boldsymbol{I}_{n_\alpha} \otimes p_\alpha(\sigma), \tag{B3}$$

$$Q(U) \rightarrow \bigoplus \boldsymbol{I}_{m_\beta} \otimes q_\beta(U). \tag{B4}$$

Note that the actions $P(\sigma)$ and $Q(U)$ commute with each other. The Schur-Weyl duality therefore states that there exists a basis that simultaneously decomposes the action of $P(\sigma)$ and $Q(U)$ into irreducible representations as

$$Q(U)P(\sigma) \rightarrow \bigoplus_{\lambda} q_\lambda(U)p_\lambda(\sigma). \tag{B5}$$

We can write this Schur basis as $|\lambda\rangle|q_\lambda\rangle|p_\lambda\rangle_{\text{Sch}}$, which decomposes the action of $P(\sigma)$ and $Q(U)$ as

$$Q(U)|\lambda\rangle|q_\lambda\rangle|p_\lambda\rangle_{\text{Sch}} = |\lambda\rangle(q_\lambda^d(U)|q_\lambda\rangle)|p_\lambda\rangle_{\text{Sch}}, \tag{B6}$$

$$P(\sigma)|\lambda\rangle|q_\lambda\rangle|p_\lambda\rangle_{\text{Sch}} = |\lambda\rangle|q_\lambda\rangle(p_\lambda(\sigma)|p_\lambda\rangle)_{\text{Sch}}. \tag{B7}$$

This action will decompose the vector space $(\mathbb{C}^d)^{\otimes n}$ as

$$(\mathbb{C}^d)^{\otimes n} \rightarrow \bigoplus_{\lambda\in\Lambda} \mathcal{Q}_\lambda^d \otimes \mathcal{P}_\lambda. \tag{B8}$$

Note that since these subspaces $\mathcal{Q}_\lambda^d$ and $\mathcal{P}_\lambda$ are irreducible, the action of the representation $Q(U)$ or $P(\sigma)$ on a vector in spaces will keep the vector in that space. We will focus on building the Schur basis through considering the action of $P(\sigma)$ for $\forall\sigma \in S_n$.

2. Rotational invariance for two points is trivial

If we wish to find states that are rotationally invariant as we rotate points in a 3D point cloud, we should first consider what this means on the point-cloud data. Each point can be represented by a vector in $\mathcal{R}^3$. If the entire point cloud is rotated, this amounts to every point being transformed by the same rotation, which is equivalent to applying a rotation matrix from the group $SO(3)$ to each point.

A natural way in which we choose to represent rotations of points in a quantum encoding is as rotations on the Bloch-sphere representation of a qubit, as has been detailed in Sec. IV C 2. This is further motivated by the fact that $SU(2)$ is the double cover of $SO(3)$, which means that every point in $SO(3)$ corresponds to two points in $SU(2)$, which means that we can use $SU(2)$ to model rotations in 3D space classified by $SO(3)$. To produce an encoding like this, we simply need to apply the parametrized rotation gates $R_x(\theta)$ and $R_z(\phi)$ to an initial $|0\rangle$ state, where (θ, ϕ) are the angle coordinates of a point in radial coordinates. We are now left with the angular direction of the point represented in the Bloch sphere of a single qubit. If we do this for all n points in the point cloud, we will have n qubits. A rotation of the entire point cloud can now be represented by the application of any $SU(2)$ unitary to every qubit simultaneously. This is a tensor representation of the $SU(2)$ group and can be written explicitly as $\bigotimes_{i=1}^n U$ for $U \in \mathrm{SU}(2)$.

To further study this, we shall resort to the case consisting of only two points $n = 2$. We again follow standard explanations from reference texts [68,77,78] while relating these points back to the rotationally invariant encoding that we propose throughout the discussion. In this case, rotation of the point cloud corresponds to applying the representation $U \otimes U; U \in SU(2)$ to the two qubits. Note that this construction commutes with the SWAP operator $[U \otimes U, \text{SWAP}] = 0$, where SWAP corresponds to permuting the two qubits. This fact means that $U \otimes U$ and SWAP can be simultaneously diagonalized in a common basis.

If we consider two qubits such that $V = (\mathbb{C}^2)^{\otimes 2}$ and the group S_2, then a natural representation on two qubits would be $\{\mathbb{I}, \text{SWAP}\}$. If we change from the computational basis to the Schur basis $\{|00\rangle, \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), |11\rangle, \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)\}$, then we can see that this is a basis of eigenvectors for the SWAP operator that is completely diagonalized in this basis. We can show this explicitly as

$$\begin{aligned}
&\text{SWAP}|00\rangle = |00\rangle, \\
&\text{SWAP}\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \\
&\text{SWAP}|11\rangle = |11\rangle, \\
&\text{SWAP}\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) = -\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle).
\end{aligned} \tag{B9}$$

As both $\{\mathbb{I}, \text{SWAP}\}$ have the same action when applied to the basis states $\{|00\rangle, |11\rangle, \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)\}$, this means that it corresponds to the trivial representation **1** of S_2. The action on this subspace leaves states unchanged and hence they have an eigenvalue of 1. The action of SWAP when applied to the basis state $\{|01\rangle - |10\rangle\}$, leads to a sign flip (an eigenvalue of -1) that is therefore denoted as the **sign** representation of S_2, where the sign of the state gets flipped by SWAP and remains unchanged under $\mathbb{I}$.

We can simultaneously block diagonalize $U \otimes U$ and SWAP, as they both commute: $[U \otimes U, \text{SWAP}] = 0$. Hence $U \otimes U$ is block diagonalized in this basis. This means that if we have a state in the space spanned by $\{|00\rangle, \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), |11\rangle\}$ that it will remain in this space under the operation of $U \otimes U$. It is possible to identify for $U \otimes U$ a symmetric eigenspace, denoted $\text{Sym}^2(\mathbb{C}^2)$, with eigenvalue $\lambda_{\text{sym}} = 1$, along with an antisymmetric eigenspace, denoted $\wedge(\mathbb{C}^2)$, with an eigenvalue $\lambda_{\text{ant}} = -1$.

If we want to find a state that is invariant under S_2, which in this case corresponds to permuting the order of the qubits, then the state needs to be in the eigenspace of the trivial representation of S_2. This corresponds to the state formed with the basis $\{|00\rangle, |01\rangle + |10\rangle, |11\rangle\}$, as in this basis both $\mathbb{I}$ and SWAP both leave the state unchanged and hence the state is permutation invariant. Projections to this subspace are how permutation-invariant encodings are performed, as initially suggested in Ref. [20].

However, while the symmetric space is closed under the action $U \otimes U$, the application of the unitaries will still change the state. Consider a unitary matrix given by

$$U = \begin{pmatrix} a & -b^* \\ b & a* \end{pmatrix}, \tag{B10}$$

where $|a|^2+|b|^2 = 1$. If one applies a Schur transform to $U \otimes U$, this will take it to the basis discussed previously. It has been shown that in this basis U /otimes U takes the following block-diagonal form [77,78]

$$\text{Sch}(U \otimes U)\text{Sch}^\dagger = \begin{pmatrix} a^2 & -\sqrt{2}ab^* & (b^*)^2 & 0 \\ \sqrt{2}ab & |a|^2-|b|^2 & -\sqrt{2}a^*b^* & 0 \\ b^2 & \sqrt{2}a^*b & (a^*)^2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{B11}$$

The SWAP operator in the same basis can be written as

$$\text{Sch}(\text{SWAP})\text{Sch}^\dagger = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \tag{B12}$$

The symmetric subspace consists of the upper-left block. $U \otimes U$ will transform the state but keep it within the symmetric subspace. This also means that any variational circuit of the form $U \otimes U$ will similarly keep the state symmetric. This is the intuition behind geometric QML techniques in which variational layers can be constructed, such as $U \otimes U$ in this case, that preserve the symmetry of the state they act on while still allowing variational parameters to be adjusted [40]. In this work, we do not consider creating variational circuits that preserve symmetry; instead, we focus on projecting quantum encoded states to irreducible subspaces, which strictly enforces the symmetry invariance in the model, as all information in the quantum state is deleted except for the portion lying in the chosen irreducible subspace before the training and classification step are considered. It is important to note that the multiplicity of the subspace of S_n corresponds to the dimension of the representation of $U \otimes U$ in that subspace and vice versa.

If we require a state to be rotationally invariant using the encoding previously described, we require that the state lies in the subspace corresponding to the trivial representation of $SU(2)$, as under the trivial representation the state will be unchanged by the action $U \otimes U$. In the case $n = 2$, this would correspond to the basis consisting of a single eigenvector $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$. This is now the trivial representation for $SU(2)$ but corresponds to the **sign** representation in S_2. This means that the state is invariant under $SU(2)$ but the SWAP operator changes its sign; hence it is not invariant under permutation. As the dimension and multiplicity of the space is 1, the only normalized state that can exist in this space would be the state $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$. Therefore, any projection onto the antisymmetric space for $n = 2$ would correspond to projection to the state $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$, which would provide rotational invariance but result in a trivial encoding, as all information about the data would be lost. Hence we show that as $U \otimes U$ and S_2 can be simultaneously block diagonalized, we can perform projections for S_2 that take the state to a space that is the trivial representation of $U \otimes U$ and is therefore a rotationally invariant state. Note that it would not be possible in this setup to have a state that is both permutation invariant and rotationally invariant, as they correspond to orthogonal irreducible subspaces.

We also note that if we consider the case in which one attempts to create a equivariant variational quantum model that is rotationally invariant in a similar manner to how permutation-equivariant models are created in this context, then the variational layer would have to be composed of SWAP gates. These clearly do not have the ability to be parametrized by any variational parameters and therefore it is not possible to create such a variational circuit if strictly following the line of reasoning presented here, although they could be implemented with a different encoding and approach. This highlights a further difference between the rotationally invariant projections proposed here, which enforce symmetry in the state itself, and equivariant VQC models.

3. Rotationally invariant encoding for S_4

The case presented for $n = 2$ has demonstrated that a rotationally invariant projection is possible, but it always corresponds to the state $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$ and hence is deleting all information about the encoded state. The space is technically one-dimensional but when considering the normalization condition of quantum states, the projection is effectively to a zero-dimensional space in terms of the information that is retained. Although the motivation for performing these projections is to reduce the dimensionality of the encoding to aid the generalization ability of the overall model, this is a trivially large reduction in dimensionality that has no use in practice. We should now investigate the irreducible representations of S_n for $n > 2$ to find a less trivial example. We would like to find a subspace where, for S_n, the dimension of the irreducible representation is d and the multiplicity is 1. This means that the corresponding subspace of $U^{\otimes n}$ will have dimension 1 and multiplicity d. This would mean that $U^{\otimes n}$ is still in the trivial representation, meaning that the state is unchanged under the action $U^{\otimes n}$; however, it gives a wider range of possible states within this subspace and hence some information about the encoded state is preserved.

Using the work of Refs. [77,78], we consider the group S_4. In this case, the action of $U \otimes U \otimes U \otimes U$ when written in the Schur basis is shown in Fig. 21. Within this decomposition, we can identify that there are two basis elements in the Schur basis that are invariant under the action of $(U \otimes U \otimes U \otimes U)$. This is the trivial representation of $SU(2)^{\otimes 4}$, which has dimension equal to 1 but a multiplicity equal to 2. This corresponds to the basis states

$$\begin{aligned}|d_1\rangle = \frac{\sqrt{3}}{6}(2|0011\rangle + 2|1100\rangle - |0101\rangle \\ - |1010\rangle - |0110\rangle - |1001\rangle) \qquad \text{(B13)}\end{aligned}$$

and

$$|d_2\rangle = \frac{1}{2}(|0101\rangle + |1010\rangle - |0110\rangle - |1001\rangle). \qquad \text{(B14)}$$

Due to the Schur-Weyl duality, we know that one of the irreducible subspaces of S_n will correspond to the same basis states. However, in this case, the multiplicity of the irreducible representation of S_n will be 1 and the dimension will equal 2. One can find that by projecting onto the $r = 3$ representation of S_4 (as specified in Table II) using the projection circuit proposed in Sec. IV, the quantum input state $|\psi\rangle$ will be projected onto some state $|\psi_{\text{rot}}\rangle = \lambda_1|d_1\rangle + \lambda_2|d_2\rangle$. In this case, there are now two basis states that can be used. However, due to the normalization constraint $|\lambda_1|^2+|\lambda_2|^2 = 1$, the effective dimension is reduced to be equal to 1 overall. This means that at least some information from the original input state is retained when the projection is performed and it is therefore a nontrivial example, although effective projection to a one-dimensional space may be too great a dimensionality reduction in practice.

$$\left[\begin{array}{cccccccccccccccc}
a^4 & -2a^3\overline{b} & \sqrt{6}a^2\overline{b}^2 & -2a\overline{b}^3 & \overline{b}^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2a^3b & a^2\left(4|a|^2-3\right) & \sqrt{6}a\left(1-2|a|^2\right)\overline{b} & \left(4|a|^2-1\right)\overline{b}^2 & -2\overline{a}\overline{b}^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\sqrt{6}a^2b^2 & \sqrt{6}ab\left(2|a|^2-1\right) & 6|a|^4-6|a|^2+1 & \sqrt{6}\left(1-2|a|^2\right)\overline{a}\overline{b} & \sqrt{6}\overline{a}^2\overline{b}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2ab^3 & b^2\left(4|a|^2-1\right) & \sqrt{6}b\left(2|a|^2-1\right)\overline{a} & \left(4|a|^2-3\right)\overline{a}^2 & -2\overline{a}^3\overline{b} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
b^4 & 2b^3\overline{a} & \sqrt{6}b^2\overline{a}^2 & 2b\overline{a}^3 & \overline{a}^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & a^2 & -\sqrt{2}a\overline{b} & \overline{b}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \sqrt{2}ab & 2|a|^2-1 & -\sqrt{2}\overline{a}\overline{b} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & b^2 & \sqrt{2}b\overline{a} & \overline{a}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a^2 & -\sqrt{2}a\overline{b} & \overline{b}^2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{2}ab & 2|a|^2-1 & -\sqrt{2}\overline{a}\overline{b} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b^2 & \sqrt{2}b\overline{a} & \overline{a}^2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a^2 & -\sqrt{2}a\overline{b} & \overline{b}^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{2}ab & 2|a|^2-1 & -\sqrt{2}\overline{a}\overline{b} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b^2 & \sqrt{2}b\overline{a} & \overline{a}^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right]$$

FIG. 21. The action $(U \otimes U \otimes U \otimes U)$ written in the Schur basis [77,78]. The blocks correspond to irreducible subspaces of $SU(2)^{\otimes 4}$. The dimension of a block is the dimension of the irreducible subspace and the number of repeats is the multiplicity. The trivial representation of $SU(2)^{\otimes 4}$ corresponds to blocks of [1], with multiplicity 2. This corresponds to a two-dimensional subspace in S_n with multiplicity 1.

In general, if it is possible to calculate the multiplicity of the trivial representation of dimension 1 of $SU(2)^{\otimes n}$ to be m_{SU}, then by Schur-Weyl duality there exists an irreducible subspace of S_n of dimension m_{SU} and multiplicity 1. Projecting to this subspace will therefore result in reducing the dimensionality of the encoded data to

$m_{SU} - 1$, after considering the normalization condition. We provide this rotationally invariant encoding as an example to demonstrate how the irreducible-subspace-projection circuit could be utilized for point-cloud data, complementing previous work on permutation-invariant encodings of point-cloud data [20]. It may be the case that this is not feasible or useful for $n > 4$ and that such investigations could be subject to further research. However, the main motivation has been to provide an additional example of how the projection framework could be used to encode symmetry in a model. In general, the irreducible-subspace-projection framework that we have introduced can be used for any representation of any finite group G along with any encoding procedure for $|\psi\rangle$ and hence there exists ample flexibility for application to a wide array of problems beyond those mentioned as examples in this work.

APPENDIX C: POINT-CLOUD NUMERIC DETAILS

This appendix specifies the details of the numerical results reported in Sec. IV D. To generate the data, $n = 3$ points have been sampled from the surface of a shape—either a sphere or a torus, as originally used in Ref. [20]—to create point clouds. The process is repeated with an equal number of point clouds from each shape until the desired number of total samples is reached. These are split into training sets (80%) and testing sets (20%). We have evaluated the performance of the model on the test set and averaged the accuracy in ten experiments, in which a newly generated data set is used each time. Both the sphere and torus are centered at the origin, with the torus scaled to match the average point magnitude of the sphere. All data are normalized between $-\pi/2$ and $\pi/2$ to be encoded as rotation angles.

To encode a point cloud, each point $\mathbf{p}_i = (x_i, y_i, z_i)$ is first encoded into a two-qubit quantum state $|\mathbf{p}_i\rangle$ using two repeated layers of the encoding circuit shown in Fig. 22. Then, the corresponding state vector is found using QISKIT STATEVECTOR_SIMULATOR [79]. From here, we can write the entire point cloud as $|P\rangle = |\mathbf{p}_1\rangle \otimes |\mathbf{p}_2\rangle \otimes |\mathbf{p}_3\rangle$. In order to perform the simulations, we have then found the projection of $|P\rangle$ onto each irreducible subspace classically utilizing the result specified in Theorem 3 and the character table for S_3. From here, the a_r parameters specified in Eq. (115) are used to produce the correct linear combination of projections and arrive at the partially symmetric subspace amplified state $|P\rangle_\alpha$, with the amount of symmetry parametrized by α. We then implement a quantum support vector machine (SVM) classification by calculating the inner product between the data points to form a kernel matrix, which is then input to a classical SVM for the final classification [20,81]. We have recorded the average test accuracy over ten experiments for a given α and then repeated with different α values and plotted the results to see which value of α is optimal for the data.

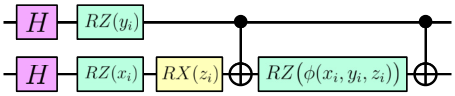

FIG. 22. One layer of the encoding circuit used to encode an individual point $\mathbf{p}_i = (x_i, y_i, z_i)$ in the symmetric-subspace-amplification model. The function $\phi(x_i, y_i, z_i) = 2/\pi^2(\pi - x_i)(\pi - y_i)(\pi - z_i)$ means that the entanglement of the state is dependent on some polynomial of all input variables. To generate the encoded state $|\mathbf{p}_i\rangle$, two repeated layers of this circuit are used [80,81].

APPENDIX D: DIMENSIONALITY REDUCTION IN POOLING LAYERS

Regarding the average-pooling layers implemented in this work in Sec. III, it is crucial to understand that we have confined our operations to averaging quantum states across image translations. Notably, binary addition operates cyclically, where adding 1 to the highest value loops back to the lowest value, meaning that our tiling would wrap around the edges of the image to perform the averaging. This may be an undesirable effect, unless the image has some periodic feature. It is also the case that the average pooling described does not provide any dimensionality reduction, as we can calculate the pooling-window average for every pixel at no additional computational cost.

To perform the dimensionality reduction, and to prevent the image being treated as periodic, one method could involve deleting any states in which $x_i > N - L$ or $y_j > N - L$ after the averaging part of the pooling layer is implemented by implementing

$$|x_i\rangle|y_j\rangle \rightarrow \text{DELETE } \textit{if } \; x_i \text{ or } y_j > N - L. \tag{D1}$$

To perform this without any *a priori* information on the data or architecture, one can define the $F_{\mathrm{FLAG},x}$ and $F_{\mathrm{FLAG},y}$ operators alongside two ancilla flag qubits, one for the x and y flags, respectively:

$$F_{\mathrm{FLAG},x}|x_i\rangle|y_j\rangle|0\rangle_x|0\rangle_y = \begin{cases} |x_i\rangle|y_j\rangle|0\rangle_x|0\rangle_y, & \text{if } x_j \leq N - L, \\ |x_i\rangle|y_j\rangle|1\rangle_x|0\rangle_y, & \text{if } x_j > N - L, \end{cases} \tag{D2}$$

and

$$F_{\mathrm{FLAG},y}|x_i\rangle|y_j\rangle|0\rangle_x|0\rangle_y = \begin{cases} |x_i\rangle|y_j\rangle|0\rangle_x|0\rangle_y, & \text{if } y_j \leq N - L, \\ |x_i\rangle|y_j\rangle|0\rangle_x|1\rangle_y, & \text{if } y_j > N - L. \end{cases} \tag{D3}$$

One can now effectively remove unwanted states to reduce the dimensionality by measuring the ancilla warning qubits and disregarding results unless $|0\rangle_x|0\rangle_y$ is measured. This will delete any states in which the average pooling layer will have wrapped around the sides of the image.

Clearly, as this is a probabilistic procedure, we expect that significantly more efficient implementations are possible depending on the exact specifics of the problem. For example, if it is known that some of the most significant qubits correspond to pixels that should not be used, then they can simply be discarded without needing to implement flag operators and extra ancilla qubits. Implementation of the discarding part of the pooling process will therefore be dependent on the exact requirements of the model, although in the worst-case scenario resorting to using F_{FLAG} operators, as described above, to remove specific states would succeed in reducing the image dimensionality (for more details on implementing pooling within a full quantum CNN framework, see Ref. [16]).

APPENDIX E: SUBTRACTION-CIRCUIT SCALING

The subtraction or decrement operator that we consider is controlled by a single ancilla qubit, meaning that overall we consider $n+1$ qubits consisting of a total of n multicontrolled Toffolli gates, with the largest being controlled by n qubits, and the number of controlling qubits decreasing by 1 each time such that the final gate is a controlled-NOT (CNOT) gate. For $n+1$ total qubits, the n-qubit multicontrolled Toffoli can be decomposed into at worst $\mathcal{O}(n^2)$ basic operations [84], which may be improved if ancillas are used [85]. The m-qubit controlled Toffoli with $m < n-1$ can be created by $\mathcal{O}(m)$ basic operations [84]. As there are $n-1$ gates of this type in total, their overall contribution will be bounded by $\mathcal{O}(n^2)$. Hence, overall, we can upper bound the required number of basic operations at $\mathcal{O}(n^2)$ to create the basic controlled-subtraction gate (although it has also been suggested linear scaling may be possible for increment or decrement gates using extra ancillas [86–88]). Note that as we consider subtraction operators where the subtraction amount doubles each time, this means that subsequent subtraction operators can be implemented by ignoring an additional least significant qubit as the subtraction amount doubles. This means that in practice, subsequent subtraction operators will use even fewer gates, although we will not include this in our complexity calculation. As $n = \log(N)$ and as we require $\mathcal{O}(\log(D))$ operators in total, then, overall, we require $\mathcal{O}(\log(D)(\log(N))^2)$ basic operations to implement the average pooling.

APPENDIX F: DEGENERACY AVOIDANCE IN AVERAGE POOLING

If window length $D < 2^L$, where L is the number of ancilla qubits and controlled operators, then we need to change the final subtraction operator value. For example, if $D = 8$, then we would generate all eight operators $\mathbb{I} + \hat{T}_1 + \hat{T}_2 + \ldots + \hat{T}_7$ by using only three controlled unitaries, $\hat{T}_1$, $\hat{T}_2$, and $\hat{T}_3$. However, if we had $D = 7$, then we could attempt to implement this using $\hat{T}_1$, $\hat{T}_2$, and $\hat{T}_3$, but writing this out in full, one will see that this implements

$$\mathbb{I} + \hat{T}_1 + \hat{T}_2 + 2\hat{T}_3 + \hat{T}_4 + \hat{T}_5 + \hat{T}_6, \tag{F1}$$

which correctly has only $D = 7$ terms but there is a degeneracy for $\hat{T}_3$, meaning that it has double weighting and that we do not have an equal superposition. This may, however, be preferable if one wishes to create a general convolutional filter, where in this case the pixel corresponding to $\hat{T}_3$ in the convolutional filter is supposed to be given a weighting double that of all others.

If one wishes to avoid this degeneracy, though, then this can be achieved by avoiding an equal superposition of all ancilla qubits and instead initializing them to avoid repeating terms. In the above example, both the ancilla states $|100\rangle$ and $|110\rangle$ result in an implementation of $\hat{T}_3$. Setting the amplitude of either of these states to zero would remove the degeneracy and correctly implement the average-pooling layer again.

APPENDIX G: INPUT-SKIP-CONNECTIONS-ONLY ResNet

In this appendix, we shall discuss a less general but more qubit-efficient implementation of quantum ResNet in which the skipped connection is not between each layer but only between the input and every other layer in the circuit (until immediately before the final layer).

Theorem 5 (Input-Skip-Only Quantum ResNet). It is possible to probabilistically implement a quantum ResNet that only includes skipped connections between the input and every other layer by utilizing the LCU method within a quantum variational model. In this context, the implementation of a quantum native ResNet for L layers is defined as the implementation of the operator

$$R_{\mathrm{RES},L}|\phi(\mathbf{x})\rangle = \frac{1}{\Omega'}\sum_{f=1}^{L}\prod_{l=f}^{L}|\gamma_f|^2 W_l(\theta_l)|\phi(\mathbf{x})\rangle,$$

where $W_l(\theta_l)$ corresponds to a unitary variational gate implemented in layer l and Ω' is a normalization constant. The coefficients γ_f correspond to the desired weighting contribution for each layer and can be freely adjusted subject to the requirement that $\sum_f |\gamma_f|^2 = 1$.

Proof. The preparation operator can be implemented on k ancilla qubits as a unitary operator

$$P_{\text{PREP}} = \sum_{f=1}^{L} \gamma_f |b_f\rangle\langle b_1| + \sum_{j=2}^{L}\sum_{f=1}^{L} u_{j,f} |b_f\rangle\langle b_j|, \tag{G1}$$

where the states $\{|b_f\rangle\}_{f\in[1,2^k]} \in (\mathbb{C}^2)^{\otimes k}$ are basis states of the 2^k-dimensional Hilbert space for the k-qubit ancilla register denoted by $\mathcal{H} = (\mathbb{C}^2)^{\otimes k}$. This is usually taken to be the computational basis with $|b_1\rangle \equiv |0\rangle^{\otimes k}$. We are only concerned with the $\langle b_1|$ term, as that operator is only applied to $|b_1\rangle$ such that

$$P_{\text{PREP}}|b_1\rangle = \sum_{f=1}^{L} \gamma_f |b_f\rangle, \tag{G2}$$

subject to the condition that $\sum_f |\gamma_f|^2 = 1$. Subsequently, the selection operator can be applied as

$$S_{\text{SELECT}}|b_f\rangle|\phi(\mathbf{x})\rangle = |b_f\rangle \left(\prod_{l=f}^{L} W_l(\theta_l)|\phi(\mathbf{x})\rangle \right), \tag{G3}$$

where $\prod_{l=f}^{L} W_l(\theta_l)$ is a product of unitary operators $W_l(\theta_l)$ and hence clearly implementable on a quantum circuit. Applying the LCU framework, we can see that

$$\begin{aligned} &S_{\text{SELECT}} P_{\text{PREP}}|b_1\rangle|\phi(\mathbf{x})\rangle \\ &\quad = \sum_{f=1}^{L} \gamma_f |b_f\rangle \left(\prod_{l=f}^{L} W_l(\theta_l)|\phi(\mathbf{x})\rangle \right). \end{aligned} \tag{G4}$$

We can apply the conjugate preparation operator, which in general can be written as

$$(P_{\text{PREP}})^{\dagger} = \sum_{f=1}^{L} \gamma_f^{*} |b_1\rangle\langle b_f| + \sum_{j=2}^{L}\sum_{f=1}^{L} u_{j\,f}^{*} |b_j\rangle\langle b_f|, \tag{G5}$$

to find that

$$\begin{aligned} &P^{\dagger}_{\text{PREP}} S_{\text{SELECT}} P_{\text{PREP}}|b_1\rangle|\phi(\mathbf{x})\rangle \\ &\quad = \sum_{f=1}^{L} |\gamma_f|^2 |b_1\rangle \prod_{l=f}^{L} W_l(\theta_l)|\phi(\mathbf{x})\rangle + \sum_{j=2}^{L} |b_j\rangle(\ldots). \end{aligned} \tag{G6}$$

The next step is to measure the ancilla qubits and discard results when the ancilla is not in the $|b_1\rangle$ state; hence we can ignore terms $|b_j\rangle$ for $j \geq 2$. The probability π_S of measuring the ancillas in the $|b_1\rangle$ state corresponds to

$$\pi_S = \left| \sum_{f=1}^{L} |\gamma_f|^2 \prod_{l=f}^{L} W_l(\theta_l)|\phi(\mathbf{x})\rangle \right|^2 \tag{G7}$$

and will hence be dependent on the architecture and initial data encoding. Requiring the ancillas to be in the $|b_1\rangle$ state, we see that

$$\begin{aligned} &\langle b_1|P^{\dagger}_{\text{PREP}} S_{\text{SELECT}} P_{\text{PREP}}|b_1\rangle|\phi(\mathbf{x})\rangle \\ &\quad = \frac{1}{\Omega'} \sum_{f=1}^{L} |\gamma_f|^2 \prod_{l=f}^{L} W_l(\theta_l)|\phi(\mathbf{x})\rangle, \end{aligned} \tag{G8}$$

where $\Omega' = \sqrt{\pi_S}$, as required. ■

This LCU method will successfully prepare the desired residual network. Hence we have shown that VQC models can be built with skipped connections under the quantum ResNet framework by utilizing the LCU method to implement these nonunitary operations. An example circuit implementation for the four-layer case $L = 4$ is shown in Fig. 23. Notably, this method is more qubit efficient, requiring $\mathcal{O}(\log(L))$ qubits for L layers, as opposed to $\mathcal{O}(L)$ qubits in the previous case.

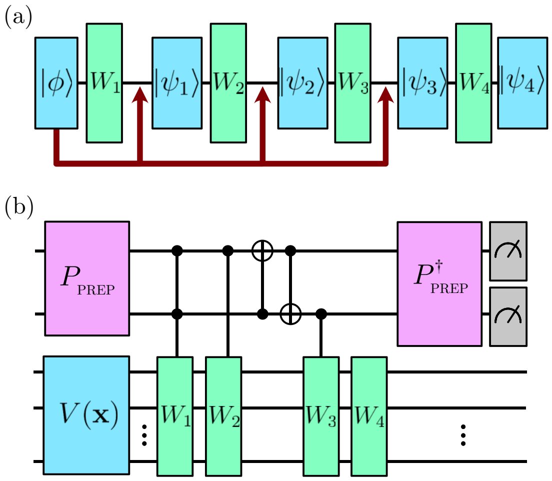

FIG. 23. (a) A ResNet conceptual illustration in which residual connections are only performed from the input layer to all subsequent layers (stopping before the final layer). (b) The circuit implementation of this ResNet using the LCU method, showing that the four-gate case can be implemented with only two ancilla qubits.

In terms of the probabilistic scaling, we see that

$$\begin{aligned}\pi_S =& \Big|\sum_{f=1}^{L}|\gamma_f|^2\prod_{l=f}^{L}W_l(\theta_l)|\phi(\mathbf{x})\rangle\Big|^2\\ =& \sum_{f,h}|\gamma_h\gamma_f|^2\Bigg(\langle\phi(x)|\Big(\prod_{l=h}^{L}W_{L-l+h}(\theta_h)^{\dagger}\Big)\\ &\times\Big(\prod_{l=f}^{L}W_l(\theta_l)\Big)|\phi(\mathbf{x})\rangle\Bigg).\end{aligned} \tag{G9}$$

Hence the probability of success depends on the encoding state $|\phi(\mathbf{x})\rangle$ and the variational layers, but can also be adjusted through the strength of the residual connection using the γ parameters.

1. Two-gate-input worked example

In this section, we give a brief worked example for the two-layer case of our LCU quantum ResNet with equal weightings.

Consider the encoded quantum state as $|\phi(\mathbf{x})\rangle$, which will be evolved by the first layer of a variational circuit $W_1(\theta_1)$, resulting in the state $|\psi_1(x,\theta_1)\rangle = W_1(\theta_1)|\phi(\mathbf{x})\rangle$.

If we wish to introduce a quantum ResNet-type skipped connection from the input layer to the output of the first layer, then one realization of this would be to implement the evolution

$$W_2(\theta_2)\left(|\phi(\mathbf{x})\rangle + |\psi_1(x,\theta_1)\rangle\right). \tag{G10}$$

In general, this is not a unitary operation. However, the desired ResNet skipped connection can indeed be written as a linear combination of unitary operations:

$$|\psi_2(x,\theta_1,\theta_2)\rangle = \left(W_2(\theta_2) + W_2(\theta_2)W_1(\theta_1)\right)|\phi(\mathbf{x})\rangle. \tag{G11}$$

This can be implemented using the LCU method by defining P_{PREP} equal to the Hadamard gate such that

$$P_{\text{PREP}}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{G12}$$

and

$$S_{\text{SELECT}}|0\rangle|\phi(\mathbf{x})\rangle = |0\rangle W_2(\theta_2)|\phi(\mathbf{x})\rangle, \tag{G13}$$

$$S_{\text{SELECT}}|1\rangle|\phi(\mathbf{x})\rangle = |1\rangle W_2(\theta_2)W_1(\theta_1)|\phi(\mathbf{x})\rangle. \tag{G14}$$

After applying the P_{PREP} and S_{SELECT} operators and measuring the ancilla qubit to be in the $|0\rangle$ state, then the appropriate $|\psi_2(x,\theta_1,\theta_2)\rangle$ can be prepared. We show this explicitly through each step. Starting with the preparation operator applied to the initial state we have

$$P_{\text{PREP}}|0\rangle|\phi(\mathbf{x})\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|\phi(\mathbf{x})\rangle. \tag{G15}$$

Applying the selection operator then results in

$$\begin{aligned}&S_{\text{SELECT}}P_{\text{PREP}}|0\rangle|\phi(\mathbf{x})\rangle\\ &\quad= \frac{1}{\sqrt{2}}\left(|0\rangle W_2(\theta_2)|\phi(\mathbf{x})\rangle + |1\rangle W_2(\theta_2)W_1(\theta_1)|\phi(\mathbf{x})\rangle\right).\end{aligned} \tag{G16}$$

Applying the inverse preparation operator, which is a Hadamard gate, results in

$$\begin{aligned}&P^{\dagger}_{\text{PREP}}S_{\text{SELECT}}P_{\text{PREP}}|0\rangle|\phi(\mathbf{x})\rangle\\ &\quad= \frac{1}{2}\left(|0\rangle\left((W_2(\theta_2) + W_2(\theta_2)W_1(\theta_1)\right)|\phi(\mathbf{x})\rangle\right.\\ &\qquad\left.+|1\rangle\left(W_2(\theta_2) - W_2(\theta_2)W_1(\theta_1)\right)|\phi(\mathbf{x})\rangle\right).\end{aligned} \tag{G17}$$

Finally, measuring the ancilla and requiring it to be in the $|0\rangle$ state gives

$$\begin{aligned}&\langle 0|P^{\dagger}_{\text{PREP}}S_{\text{SELECT}}P_{\text{PREP}}|0\rangle|\phi(\mathbf{x})\rangle\\ &\quad= \frac{1}{\sqrt{\pi_S}}\left(W_2(\theta_2) + W_2(\theta_2)W_1(\theta_1)\right)|\phi(\mathbf{x})\rangle,\end{aligned} \tag{G18}$$

which is probabilistically prepared with a probability equal to

$$\begin{aligned}\pi_S &= \frac{1}{4}\langle\phi(x)|(2\mathbb{I} + W_1(\theta_1) + W_1^{\dagger}(\theta_1))|\phi(\mathbf{x})\rangle\\ &= \frac{1}{2}\left(1 + \text{Re}\left(\langle\phi(x)|W_1(\theta_1)|\phi(\mathbf{x})\rangle\right)\right).\end{aligned} \tag{G19}$$

Note that the important part of the probability scaling is the real component of $\text{Re}\left(\langle\phi(x)|W_1(\theta_1)|\phi(\mathbf{x})\rangle\right)$. In the case in which $W_1(\theta_1) = \mathbb{I}$, success is guaranteed, and in the case $W_1(\theta_1) = -\mathbb{I}$, success is impossible. For general $W_1(\theta_1)$ matrices and states $|\phi(\mathbf{x})\rangle$, the success will vary. Therefore, in practice, it may be worth placing restrictions on $W_1(\theta_1)$ to ensure that $\text{Re}\left(\langle\phi(x)|W_1(\theta_1)|\phi(\mathbf{x})\rangle\right)$ does not approach too close to -1, resulting in extremely low success probabilities. However, as mentioned in the main text, if the strength of the residual connections are adjusted, it can be possible to increase the lower bound. This example has used equal strength in the residual connections for simplicity, which corresponds to the worst-case lower bound.

[1] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Sci. Comput. **26**, 1484 (1997).

[2] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik, Quantum chemistry in the age of quantum computing, Chem. Rev. **119**, 10856 (2019).

[3] D. Herman, C. Googin, X. Liu, A. Galda, I. Safro, Y. Sun, M. Pistoia, and Y. Alexeev, A survey of quantum computing for finance, arXiv:2201.02773.

[4] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, LLaMA: Open and efficient foundation language models, arXiv:2302.13971.

[5] H. Chang, H. Zhang, J. Barber, A. Maschinot, J. Lezama, L. Jiang, M.-H. Yang, K. Murphy, W. T. Freeman, M. Rubinstein, Y. Li, and D. Krishnan, MUSE: Text-to-image generation via masked generative transformers, arXiv:2301.00704.

[6] S. Frieder, L. Pinchetti, A. Chevalier, R.-R. Griffiths, T. Salvatori, T. Lukasiewicz, P. C. Petersen, and J. Berner, Mathematical capabilities of ChatGPT, arXiv:2301.13867.

[7] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, The power of quantum neural networks, Nat. Comput. Sci. **1**, 403 (2021).

[8] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann, and R. Wolf, Training deep quantum neural networks, Nat. Commun. **11**, 808 (2020).

[9] N. W. Andrew Childs, Hamiltonian simulation using linear combinations of unitary operations, Quantum Inf. Comput. **12**, 901 (2012).

[10] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Simulating Hamiltonian dynamics with a truncated Taylor series, Phys. Rev. Lett. **114**, 090502 (2015).

[11] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, arXiv:1512.03385.

[12] P. Marion, A. Fermanian, G. Biau, and J.-P. Vert, Scaling ResNets in the large-depth regime, arXiv:2206.06929.

[13] W. Rawat and Z. Wang, Deep convolutional neural networks for image classification: A comprehensive review, Neural Comput. **29**, 2352 (2017).

[14] N. Sharma, V. Jain, and A. Mishra, An analysis of convolutional neural networks for image classification, Procedia Comput. Sci. **132**, 377 (2018).

[15] A. A. M. Al-Saffar, H. Tao, and M. A. Talab, in *2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)* (Jakarta, Indonesia, 2017), p. 26.

[16] S. Wei, Y. Chen, Z. Zhou, and G. Long, A quantum convolutional neural network on NISQ devices, AAPPS Bull **32**, 2 (2022).

[17] Y. Chen, C. Wang, H. Guo, Jianwu, Novel architecture of parameterized quantum circuit for graph convolutional network, arXiv:2203.03251.

[18] L. Deng, The MNIST database of handwritten digit images for machine learning research, IEEE Signal Process. Mag. **29**, 141 (2012).

[19] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, Power of data in quantum machine learning, Nat. Commun. **12**, 2631 (2021).

[20] J. Heredge, C. Hill, L. Hollenberg, and M. Sevior, Permutation invariant encodings for quantum machine learning with point cloud data, Quantum Mach. Intell. **6**, 25 (2024).

[21] M. Larocca, S. Thanasilp, S. Wang, K. Sharma, J. Biamonte, P. J. Coles, L. Cincio, J. R. McClean, Z. Holmes, and M. Cerezo, A review of barren plateaus in variational quantum computing, arXiv:2405.00781.

[22] E. Hassan, M. S. Hossain, A. Saber, S. Elmougy, A. Ghoneim, and G. Muhammad, A quantum convolutional network and ResNet (50)-based classification architecture for the MNIST medical dataset, Biomed. Signal Process. Control **87**, 105560 (2024).

[23] A. Sagingalieva, M. Kordzanganeh, A. Kurkin, A. Melnikov, D. Kuhmistrov, M. Perelshtein, A. Melnikov, A. Skolik, and D. V. Dollen, Hybrid quantum ResNet for car classification and its hyperparameter optimization, Quantum Mach. Intell. **5**, 38 (2023).

[24] K. Zaman, T. Ahmed, M. A. Hanif, A. Marchisio, and M. Shafique, A comparative analysis of hybrid-quantum classical neural networks, arXiv:2402.10540.

[25] M. Kashif and S. Al-Kuwari, ResQNets: A residual approach for mitigating barren plateaus in quantum neural networks, EPJ Quantum Technol. **11**, 4 (2024).

[26] G. Crognaletti, M. Grossi, and A. Bassi, Estimates of loss function concentration in noisy parametrized quantum circuits, arXiv:2410.01893.

[27] G. Crognaletti, M. Grossi, S. Vallecorsa, and A. Bassi, QResNet: A variational entanglement skipping algorithm, presented at Quantum Techniques in Machine Learning Conference 2023. https://indico.cern.ch/event/1288979/contributions/5654644/attachments/2756066/4798864/QTML23_QResNet.pdf.

[28] J. Wen, Z. Huang, D. Cai, and L. Qian, Enhancing the expressivity of quantum neural networks with residual connections, arXiv:2401.15871.

[29] 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, Does provable absence of barren plateaus imply classical simulability? Or, why we need to rethink variational quantum computing, arXiv:2312.09121.

[30] T. Vu, L. Le, and T. Pham, Exploring the features of quanvolutional neural networks for improved image classification, Quantum Mach. Intell. **6**, 29 (2024).

[31] I. Cong, S. Choi, and M. D. Lukin, Quantum convolutional neural networks, Nat. Phys. **15**, 1273 (2019).

[32] C. Umeano, A. E. Paine, V. E. Elfving, and O. Kyriienko, What can we learn from quantum convolutional neural networks? arXiv:2308.16664.

[33] T. Hur, L. Kim, and D. K. Park, Quantum convolutional neural network for classical data classification, Quantum Mach. Intell. **4**, 3 (2022).

[34] Y. Li, R.-G. Zhou, R. Xu, J. Luo, and W. Hu, A quantum deep convolutional neural network for image recognition, Quantum Sci. Technol. **5**, 044003 (2020).

[35] A. Pesah, M. Cerezo, S. Wang, T. Volkoff, A. T. Sornborger, and P. J. Coles, Absence of barren plateaus in quantum

convolutional neural networks, Phys. Rev. X **11,** 041011 (2021).

[36] J. Liu, K. H. Lim, K. L. Wood, W. Huang, C. Guo, and H.-L. Huang, Hybrid quantum-classical convolutional neural networks, Science China Physics, Mech. Astron. **64,** 290311 (2021).

[37] X.-W. Yao, H. Wang, Z. Liao, M.-C. Chen, J. Pan, J. Li, K. Zhang, X. Lin, Z. Wang, Z. Luo, W. Zheng, J. Li, M. Zhao, X. Peng, and D. Suter, Quantum image processing and its application to edge detection: Theory and experiment, Phys. Rev. X **7,** 031041 (2017).

[38] J. M. Kübler, S. Buchholz, and B. Schölkopf, The inductive bias of quantum kernels, arXiv:2106.03747.

[39] J. Bowles, V. J. Wright, M. Farkas, N. Killoran, and M. Schuld, Contextuality and inductive bias in quantum machine learning, arXiv:2302.01365.

[40] J. J. Meyer, M. Mularski, E. Gil-Fuster, A. A. Mele, F. Arzani, A. Wilms, and J. Eisert, Exploiting symmetry in variational quantum machine learning, arXiv:2205.06217.

[41] Q. T. Nguyen, L. Schatzki, P. Braccia, M. Ragone, P. J. Coles, F. Sauvage, M. Larocca, and M. Cerezo, Theory for equivariant quantum neural networks, arXiv:2210.08566.

[42] M. T. West, J. Heredge, M. Sevior, and M. Usman, Provably trainable rotationally equivariant quantum machine learning, PRX Quantum **5,** 030320 (2024).

[43] L. Schatzki, M. Larocca, Q. T. Nguyen, F. Sauvage, and M. Cerezo, Theoretical guarantees for permutation-equivariant quantum neural networks, npj Quantum Inf. **10,** 12 (2024).

[44] C. Tüysüz, S. Y. Chang, M. Demidik, K. Jansen, S. Vallecorsa, and M. Grossi, Symmetry breaking in geometric quantum machine learning in the presence of noise, PRX Quantum **5,** 030314 (2024).

[45] V. Buzek and M. Hillery, Optimal manipulations with qubits: Universal quantum entanglers, Phys. Rev. A **62,** 022303 (2000).

[46] A. Barenco, A. Berthiaume, D. Deutsch, A. Ekert, R. Jozsa, and C. Macchiavello, Stabilisation of quantum computations by symmetrisation, arXiv:9604028.

[47] Z. Li, L. Nagano, and K. Terashi, Enforcing exact permutation and rotational symmetries in the application of quantum neural network on point cloud datasets, arXiv:2405.11150.

[48] D. Guala, J. Soni, and J. M. Arrazola, Pennylane tutorial: Linear combination of unitaries and block encodings, https://pennylane.ai/qml/demos/tutorial_lcu_block encoding/ (accessed: May 24, 2024).

[49] D. Camps, L. Lin, R. V. Beeumen, and C. Yang, Explicit quantum circuits for block encodings of certain sparse matrices, arXiv:2203.10236.

[50] M. Larocca, F. Sauvage, F. M. Sbahi, G. Verdon, P. J. Coles, and M. Cerezo, Group-invariant quantum machine learning, PRX Quantum **3,** 030341 (2022).

[51] E. Fontana, D. Herman, S. Chakrabarti, N. Kumar, R. Yalovetzky, J. Heredge, S. H. Sureshbabu, and M. Pistoia, The adjoint is all you need: Characterizing barren plateaus in quantum ansätze, Nat. Commun. **15,** 7171 (2023).

[52] M. Ragone, B. Bakalov, F. Sauvage, A. Kemper, C. Marrero, M. Larocca, and M. Cerezo, A Lie algebraic theory of barren plateaus for deep parameterized quantum circuits, Nat. Commun. **15,** 7172 (2024).

[53] L. Friedrich and J. Maziero, Quantum neural network with ensemble learning to mitigate barren plateaus and cost function concentration, arXiv:2402.06026.

[54] R. Wiersema, E. Kökcü, A. F. Kemper, and B. N. Bakalov, Classification of dynamical lie algebras for translation-invariant 2-local spin systems in one dimension, arXiv:2309.05690.

[55] N. L. Diaz, D. García-Martín, S. Kazi, M. Larocca, and M. Cerezo, Showcasing a barren plateau theory beyond the dynamical Lie algebra, arXiv:2310.11505.

[56] S. Gudder, Mathematical theory of duality quantum computers, Quantum Inf. Process. **6,** 37 (2007).

[57] A. Arrasmith, Z. Holmes, M. Cerezo, and P. J. Coles, Equivalence of quantum barren plateaus to cost concentration and narrow gorges, Quantum Sci. Technol. **7,** 045015 (2022).

[58] A. Veit, M. Wilber, and S. Belongie, Residual networks behave like ensembles of relatively shallow networks, arXiv:1605.06431.

[59] C. Yu, P.-H. Hung, J.-H. Hong, and H.-Y. Chiang, in *2023 IEEE 12th Global Conference on Consumer Electronics (GCCE)* (2023), p. 747.

[60] V. Vedral, A. Barenco, and A. Ekert, Quantum networks for elementary arithmetic operations, Phys. Rev. A **54,** 147 (1996).

[61] N. I. Galanis, P. Vafiadis, K.-G. Mirzaev, and G. A. Papakostas, Convolutional neural networks: A roundup and benchmark of their pooling layer variants, Algorithms **15,** 391 (2022).

[62] H. Gholamalinezhad and H. Khosravi, Pooling methods in deep neural networks, a review, arXiv:2009.07485.

[63] A. Zafar, M. Aamir, N. Mohd Nawi, A. Arshad, S. Riaz, A. Alruban, A. Dutta, and S. Alaybani, A comparison of pooling methods for convolutional neural networks, Appl. Sci. **12,** 8643 (2022).

[64] M. T. West, S. M. Erfani, C. Leckie, M. Sevior, L. C. L. Hollenberg, and M. Usman, Benchmarking adversarially robust quantum machine learning at scale, Phys. Rev. Res. **5,** 023186 (2023).

[65] M. T. West, A. C. Nakhl, J. Heredge, F. M. Creevey, L. C. L. Hollenberg, M. Sevior, and M. Usman, Drastic circuit depth reductions with preserved adversarial robustness by approximate encoding for quantum machine learning, Intell. Comput. **3,** 0100 (2024).

[66] N. Dowling, M. T. West, A. Southwell, A. C. Nakhl, M. Sevior, M. Usman, and K. Modi, Adversarial robustness guarantees for quantum classifiers, arXiv:2405.10360.

[67] W. Fulton and J. W. Harris, Representation theory: A first course (1991), https://api.semanticscholar.org/CorpusID:118744956.

[68] M. Ragone, P. Braccia, Q. T. Nguyen, L. Schatzki, P. J. Coles, F. Sauvage, M. Larocca, and M. Cerezo, Representation theory for geometric quantum machine learning, arXiv:2210.07980.

[69] H. Maschke, Ueber den arithmetischen Charakter der Coefficienten der Substitutionen endlicher linearer Substitutionsgruppen, Math. Ann. **50,** 492 (1898).

[70] B. C. Hall, *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*, Graduate Texts in Mathematics, Vol. 222 (Springer International Publishing, Cham, 2015).

[71] S. Jean-Pierre, *Linear Representations of Finite Groups* (Springer, New York, 1998).

[72] B. Huppert, *Character Theory of Finite Groups* (De Gruyter, Berlin, 1998).

[73] A. Deitmar, *A First Course in Harmonic Analysis* (Springer New York, 2002).

[74] E. R. Anschuetz, A. Bauer, B. T. Kiani, and S. Lloyd, Efficient classical algorithms for simulating symmetric quantum systems, Quantum **7**, 1189 (2023).

[75] M. L. Goh, M. Larocca, L. Cincio, M. Cerezo, and F. Sauvage, Lie-algebraic classical simulations for variational quantum computing, arXiv:2308.01432.

[76] G. Raut and A. Patole, End-to-end 3D object detection using LiDAR point cloud, arXiv:2312.15377.

[77] W. M. Kirby and F. W. Strauch, A practical quantum algorithm for the Schur transform, Quantum Inf. Comput. **18**, 721 (2018).

[78] W. M. Kirby, A practical quantum Schur transform, https://static1.squarespace.com/static/5a1461108fd4d28cad8dae2c/t/5af30798575d1f7240339f36/1525876638766/William+M+Kirby+Thesis+v2 (accessed: May 24, 2024).

[79] G. Aleksandrowicz, *et al.*, QISKIT: An open-source framework for quantum computing (2019).

[80] J. Heredge, C. Hill, L. Hollenberg, and M. Sevior, Quantum support vector machines for continuum suppression in B meson decays, Comput. Softw. Big Sci. **5**, 27 (2021).

[81] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature **567**, 209 (2019).

[82] D. Bacon, I. L. Chuang, and A. W. Harrow, Efficient quantum circuits for Schur and Clebsch-Gordan transforms, Phys. Rev. Lett. **97**, 170502 (2006).

[83] H. Krovi, An efficient high dimensional quantum Schur transform, Quantum **3**, 122 (2019).

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

[85] J. M. Baker, C. Duckering, A. Hoover, and F. T. Chong, Decomposing quantum generalized Toffoli with an arbitrary number of ancilla, arXiv:1904.01671.

[86] C. Gidney, Constructing large increment gates, https://algassert.com/circuits/2015/06/12/Constructing-Large-Increment-Gates.html (accessed: August 30, 2024).

[87] T. Khattar and C. Gidney, Rise of conditionally clean ancillae for optimizing quantum circuits, arXiv:2407.17966.

[88] C. Gidney, Factoring with $n+2$ clean qubits and $n-1$ dirty qubits, arXiv:1706.07884.