npj | quantum information

Article

Published in partnership with The University of New South Wales

https://doi.org/10.1038/s41534-025-01022-z

# Characterizing privacy in quantum machine learning

Check for updates

Jamie Heredge[1,2] ✉, Niraj Kumar[1], Dylan Herman[1], Shouvanik Chakrabarti[1], Romina Yalovetzky[1], Shree Hari Sureshbabu[1], Changhao Li[1] & Marco Pistoia[1]

Ensuring data privacy in machine learning models is critical, especially in distributed settings where model gradients are shared among multiple parties for collaborative learning. Motivated by the increasing success of recovering input data from the gradients of classical models, this study investigates the analogous challenge for variational quantum circuits (VQC) as quantum machine learning models. We highlight the crucial role of the dynamical Lie algebra (DLA) in determining privacy vulnerabilities. While the DLA has been linked to the trainability and simulatability of VQC models, we establish its connection to privacy for the first time. We show that properties conducive to VQC trainability, such as a polynomial-sized DLA, also facilitate extracting detailed snapshots of the input, posing a weak privacy breach. We further investigate conditions for a strong privacy breach, where original input data can be recovered from snapshots by classical or quantum-assisted methods. We establish properties of the encoding map, such as classical simulatability, overlap with DLA basis, and its Fourier frequency characteristics that enable such a privacy breach of VQC models. Our framework thus guides the design of quantum machine learning models, balancing trainability and robust privacy protection.

In the contemporary technological landscape, data privacy concerns command increasing attention, particularly within the domain of machine learning (ML) models that are trained on sensitive datasets. Privacy concerns are widespread in many different applications, including financial records[1,2], healthcare information[3–5], and location data[6], each providing unique considerations. Furthermore, the multi-national adoption of stringent legal frameworks[7] has further amplified the urgency to improve data privacy.

The introduction of distributed learning frameworks, such as federated learning[8–10], not only promises increased computational efficiency but also demonstrates the potential for increased privacy in ML tasks. In federated learning, each user trains a machine learning model, typically a neural network, locally on their device using their confidential data, meaning that they only need to send their model gradients to the central server, which aggregates gradients of all users to calculate the model parameters for the next training step. As the user does not send their confidential data, but rather their training gradients, this was proposed as the first solution to enable collaborative learning while preventing data leakage. However, subsequent works have shown that neural networks are particularly susceptible to gradient inversion-based attacks to recover the original input data[11–15]. To mitigate the above issue, classical techniques have been proposed to enhance the privacy of distributed learning models, ranging from gradient encryption-based methods[16], the addition of artificial noise in the gradients to leverage differential-privacy type techniques[10], or strategies involving the use of batch training to perform gradient mixing[17]. These techniques, although mitigative in nature, are not fully robust since they either still leak some input information, add substantial computational overhead while training the model in the distributed setting, or result in reduced performance of the model.

A natural question that follows is whether quantum machine learning can help mitigate the privacy concerns that their classical counterparts exhibit. Specifically, one is interested in exploring the fundamental question underpinning the privacy of quantum models: *Given the gradients of a quantum machine learning model, how difficult is it to reconstruct the original classical data inputs?* In search of privacy guarantees with quantum techniques, several quantum distributed learning proposals have been previously introduced[18–26]. Within the field of quantum differential privacy, quantum noise[27] and randomized encoding[28] have been reported to have a beneficial effect. Previous methods for improving privacy in a federated learning context have ranged from the use of blind quantum computing[29], high-frequency encoding circuits[30], and hybrid quantum-classical methods that combine pre-trained classical models with quantum neural networks[31]. In particular, the work of[30] considered variational quantum circuits (VQC) as quantum machine learning models and suggested that highly expressive product encoding maps along with an overparameterized hardware efficient ansatz (HEA) would necessitate an exponential amount of resources (in terms of the number of qubits $n$) for an attacker to learn the input from the

[1]Global Technology Applied Research, JP Morgan Chase, New York, NY, USA. [2]School of Physics, The University of Melbourne, Parkville, VIC, Australia. ✉e-mail: jamie.heredge@jpmorgan.com

gradients. Their work, although the first and sole one to date to theoretically analyze the privacy of a specific VQC model architecture, has certain key drawbacks. The first is that overparameterization of a HEA leads to an untrainable model, since it mixes very quickly to a 2-design[32] and thus leads to a barren plateau phenomenon[33]. The authors enforced the requirement of overparameterization to ensure that there are no spurious local minima in the optimization landscape and that all local minima are exponentially concentrated toward global minima[34]. However, this requires the HEA to have an exponential depth and thus an exponential number of parameters, which precludes efficient training due to an exponential memory requirement to store and update the parameters. Secondly, the difficulty of inverting gradients to recover data primarily stems from the high expressivity, characterized in this case by an exponentially large number of non-degenerate frequencies of the generator Hamiltonian of the encoding map. Introducing high-frequency terms in the encoding map may not be an exclusive quantum effect, as classical machine learning models could also be enhanced by initially loading the data with these high-frequency feature maps[35].

While previous studies have aimed to highlight the benefits of employing VQC models in safeguarding input privacy, none have convincingly addressed what sets VQC models apart from classical neural networks in their potential to provide robust privacy guarantees. A critical aspect missing in a comprehensive examination of the privacy benefits offered by VQC models in a privacy framework tailored for them. Such a framework should avoid dependence on specific privacy-enhancing procedures or architectures and instead focus on exploring the fundamental properties of VQC models that result in input privacy.

To address the above concerns, we introduce a framework designed to assess the possibility of retrieving classical inputs from the gradients observed in VQC models. We consider VQCs that satisfy the Lie algebra supported ansatz (LASA) property, which has been key in establishing connections with the trainability and classical simulatability of VQCs[36–38]. Our study systematically differentiates the separate prerequisites for input reconstruction across both the variational ansatz and encoding map architectures of these VQC models as summarized in Table 1. Our first result concerns the properties of the variational ansatz and the measurement operator of the VQC. Specifically, we show that when the VQC satisfies the LASA condition, i.e., when the measurement operator is within the dynamical Lie algebra (DLA) of the ansatz, and when the DLA scales polynomially with the number of qubits, it is possible to efficiently extract meaningful *snapshots* of the input, enabling training and evaluation of VQC models for other learning tasks without having direct access to the original input. We call this the *weak privacy* breach of the model. Further, we investigate conditions for *strong privacy* breach, i.e., recoverability of the original input by classical or quantum-assisted polynomial time methods. Fully reconstructing the input data from these snapshots to perform a strong privacy breach presents a further challenge, which we show is dependent on properties of the encoding map, such as the hardness of classically simulating the encoding, the overlap of the DLA basis with encoding circuit generators, and its Fourier frequency characteristics. The two types of privacy breach we introduce are summarized in Fig. 1, while more specific definitions regarding snapshots, recoverability, and invertibility are provided in the input recoverability definitions section.

This investigation presents a comprehensive picture of strategies to extract the key properties of VQCs to provide robust privacy guarantees while ensuring that they are still trainable. We structure our paper in the following manner. Supplementary file Sec I provides the notation used in this work. The results section starts by providing a general framework for studying privacy with VQC. This includes describing the VQC framework, providing Lie theoretic definitions required for this work, and the privacy definitions in terms of input recoverability. The results section then continues with the snapshot recovery and snapshot invertibility subsections that provide a detailed analysis of the snapshot recoverability from the gradients, and snapshot inversion to recover the input, respectively. The method section establishes the connections between privacy and the well-studied trainability of VQCs, and then consequently highlights the future directions of enabling robust privacy with quantum machine learning models.

## Results

### General Framework

**Variational quantum circuits for machine learning.** A variational quantum circuit (VQC) is described in the following manner. We consider the $d$-dimensional input vector $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$, which is loaded into the quantum encoding circuit $V(\mathbf{x})$ of $n$ qubits to produce a feature map with the input state mapping,

$$\rho(\mathbf{x}) = V(\mathbf{x})|0\rangle^{\otimes n}\langle 0|^{\otimes n} V(\mathbf{x})^{\dagger}. \tag{1}$$

This operation loads the input vector of dimension $d$ to a Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$ of dimension $\dim(\rho(\mathbf{x})) = 2^n$. We will explicitly consider the scenario where $n = \Theta(d)$, which is a common setting in most existing VQC algorithms, and hence the number of qubits in a given algorithm will be of the same order as the input vector dimension $d$. The state $\rho(\mathbf{x})$ is then passed through a variational circuit ansatz $U(\boldsymbol{\theta})$ defined as

$$\mathbf{U}(\boldsymbol{\theta}) = \prod_{k=1}^{D} e^{-i\theta_k \mathbf{H}_{\nu(k)}}, \tag{2}$$

which is parameterized by a vector of variational parameters $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_D]$, where $D$ is the total number of variational parameters. Here $\{\mathbf{H}_1, \cdots, \mathbf{H}_N\}$ are

**Table 1 | Summary of results on the privacy guarantees and complexity provided by the studied attack models on various VQC models**

| Privacy Breach | Description | Complexity | Requirements |
|---|---|---|---|
| Weak | Snapshot recovery | Algorithm 2: $\mathcal{O}(\text{poly}(\dim(\mathfrak{g})))$ | $\mathcal{O}(\text{poly}(\text{n}))$ sized DLA + LASA condition (Def 5) + Slow Pauli Expansion (Def 9) |
| Strong | Snapshot inversion for local Pauli encoding | Algorithm 4: $\mathcal{O}(\text{poly}(\text{n}, 1/\epsilon)$ | Snapshot recovery requirement + Separable state with $\rho_J(\mathbf{x})$ parameterized by subset $\text{x}_J \subseteq \mathbf{x}$<br>• $\dim(\mathbf{x}_{\text{J}}) = \mathcal{O}(1)$<br>• each $x_k$ is encoded at most $R = \mathcal{O}(\text{poly}(\text{n}))$ times<br>• Snapshot components with non-zero overlap w.r.t. $\rho_J(\mathbf{x}_J)$ has cardinality at least $\dim(\mathbf{x}_J)$. |
| Strong | Snapshot inversion for generic encoding | Grid Search : $\mathcal{O}\left(\left(\frac{L}{\epsilon}\right)^d\right)$ | The recovery cost function is $L$-Lipschitz, leading to efficient privacy breach not being possible |

We consider two privacy breach scenarios involving VQCs : *weak privacy* breach and *strong privacy* breach for classical or quantum-assisted polynomial time methods. Weak privacy breach concerns the recovery of the meaningful snapshots of the input encoded state, allowing training VQC models for distinct learning tasks without requiring access to the input. Strong privacy breach concerns subsequently arise when inverting the snapshots to recover the original input. We consider the snapshot invertibility for the local Pauli encoding map, which admits an efficient (polynomial in the number of qubits $n$) algorithm if the requirements stated in the table are met. For the case of generic encoding maps where the VQC is considered as a black-box $L$-Lipschitz function, snapshot invertibility requires performing the grid search, which scales exponentially in the input dimension $d$, and thus it rules out efficient privacy breaches.

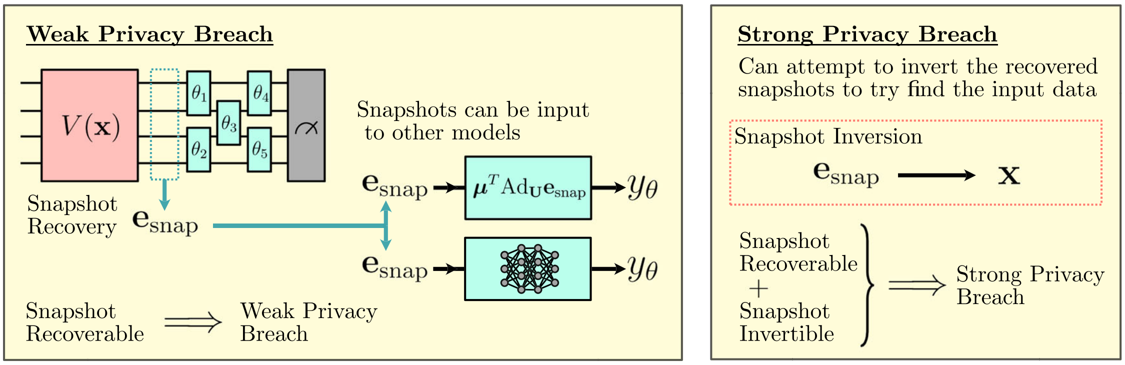

Weak Privacy Breach
$V(\mathbf{x})$
$\theta_1$
$\theta_2$
$\theta_3$
$\theta_4$
$\theta_5$
Snapshot Recovery
$\mathbf{e}_{\text{snap}}$
Snapshots can be input to other models
$\mathbf{e}_{\text{snap}}$
$\boldsymbol{\mu}^T \text{Ad}_{\mathbf{U}} \mathbf{e}_{\text{snap}}$
$y_\theta$
$\mathbf{e}_{\text{snap}}$
$y_\theta$
Snapshot Recoverable $\Longrightarrow$ Weak Privacy Breach
Strong Privacy Breach
Can attempt to invert the recovered snapshots to try find the input data
Snapshot Inversion
$\mathbf{e}_{\text{snap}} \longrightarrow \mathbf{x}$
Snapshot Recoverable + Snapshot Invertible $\Longrightarrow$ Strong Privacy Breach

**Fig. 1 | Overview of the general framework and definitions.** Weak privacy breach corresponds to attacks where snapshots of the data are retrieved. These can be used as inputs to other models, without explicitly needing the exact data, allowing one to potentially learn characteristics of the data. If these snapshots can then be further inverted to retrieve the input data **x** explicitly, we say the attack has succeeded in a strong privacy breach.

the set of $N$ Hermitian generators of the circuit $U$. The generator assignment map $\nu$: $[D] \rightarrow [N]$ is used to assign one the generator $\mathbf{H}_{\nu(k)}$ to the corresponding variational parameter $\theta_k$. Under this notation, multiple distinct variational parameters can use the same generator. This is the case for repeated layers of a variational ansatz, where for $L$ repeated layers, one would have $D = NL$ and $\nu(k) = ((k-1) \bmod N) + 1$. We note that the above structure is quite general since some common ansatz structures such as the hardware efficient ansatz, the quantum alternating operator ansatz, and Hamiltonian variational ansatz, among others, are all encapsulated in this framework as highlighted in ref. 39.

The parameterized state $\rho(\mathbf{x})$ is passed through a variational circuit denoted by $U(\boldsymbol{\theta})$, followed by the measurement of some observable $\mathbf{O} \in \mathcal{H}$. For a given $\boldsymbol{\theta}$, the output of the variational quantum circuit model is expressed as the expectation value of $\mathbf{O}$ with the parameterized state,

$$y_{\boldsymbol{\theta}}(\mathbf{x}) = \operatorname{Tr}(\mathbf{U}^\dagger(\boldsymbol{\theta})\mathbf{O}\mathbf{U}(\boldsymbol{\theta})\rho(\mathbf{x})). \tag{3}$$

For the task of optimizing the variational quantum circuits, the model output is fed into the desired cost function $\texttt{Cost}(\boldsymbol{\theta}, \mathbf{x})$, which is subsequently minimized to obtain,

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \texttt{Cost}(\boldsymbol{\theta}, \mathbf{x}), \tag{4}$$

where $\boldsymbol{\theta}^*$ are the final parameter values after optimization. Typical examples of cost functions include cross-entropy loss, and mean-squared error loss, among others[40].

The typical optimization procedure involves computing the gradient of the cost function with respect to the parameters $\boldsymbol{\theta}$, which in turn, involves computing the gradient with respect to the model output $y_{\boldsymbol{\theta}}(\mathbf{x})$

$$C_j = \frac{\partial y_{\boldsymbol{\theta}}(\mathbf{x})}{\partial \theta_j}, j \in [D]. \tag{5}$$

Going forward, we will directly deal with the recoverability of input $\mathbf{x}$ given $C_j$, instead of working with specific cost functions. Details of how to reconstruct our results when considering gradients with respect to specific cost functions are covered in the Supplementary file Sec II.

**Lie theoretic framework.** We review some introductory as well as recent results on Lie theoretic framework for variational quantum circuits which are relevant to our work. For a more detailed review of this topic, we refer the reader to[39,41]. We provide the Lie theoretic definitions for a periodic ansatz of the form Eq. (2).

**Definition 1.** (Dynamical Lie Algebra). The dynamical Lie algebra (DLA) $\mathfrak{g}$ for an ansatz $\mathbf{U}(\boldsymbol{\theta})$ of the form Eq. (2) is defined as the real span of the Lie closure of the generators of $U$

$$\mathfrak{g} = \operatorname{span}_{\mathbb{R}} \langle i\mathbf{H}_1, \cdots, i\mathbf{H}_N \rangle_{\text{Lie}}, \tag{6}$$

where the closure is defined under taking all possible nested commutators of $S = \{i\mathbf{H}_1, \cdots, i\mathbf{H}_N\}$. In other words, it is the set of elements obtained by taking the commutation between elements of $S$ until no further linearly independent elements are obtained.

**Definition 2.** (Dynamical Lie Group). The dynamical Lie group $\mathcal{G}$ for an ansatz $\mathbf{U}(\boldsymbol{\theta})$ of the form of Eq. (2) is determined by the DLA $\mathfrak{g}$ such that,

$$\mathcal{G} = e^{\mathfrak{g}}, \tag{7}$$

where $e^{\mathfrak{g}} := \{e^{i\mathbf{H}}, i\mathbf{H} \in \mathfrak{g}\}$ and is a subgroup of $\text{SU}(2^n)$. For generators in $\mathfrak{g}$, the set of all $\mathbf{U}(\boldsymbol{\theta})$ of the form Eq (2) generates a dense subgroup of $\mathcal{G}$.

**Definition 3.** (Adjoint representation). The Lie algebra adjoint representation is the following linear action: $\forall \mathbf{K}, \mathbf{H} \in \mathfrak{g}$,

$$\text{ad}_{\mathbf{H}}\mathbf{K} := [\mathbf{H}, \mathbf{K}] \in \mathfrak{g}, \tag{8}$$

and the Lie group adjoint representation is the following linear action $\forall \mathbf{U} \in \mathcal{G}, \forall \mathbf{H} \in \mathfrak{g}$,

$$\text{Ad}_{\mathbf{U}}\mathbf{H} := \mathbf{U}^\dagger \mathbf{H} \mathbf{U} \in \mathfrak{g}. \tag{9}$$

**Definition 4.** (DLA basis). The basis of the DLA is denoted as $\{i\mathbf{B}_\alpha\}_\alpha$, $\alpha \in \{1, \cdots, \dim(\mathfrak{g})\}$, where $\mathbf{B}_\alpha$ are Hermitian operators and form an orthonormal basis of $\mathfrak{g}$ with respect to the Frobenius inner product.

Any observable $\mathbf{O}$ is said to be entirely supported by the DLA whenever $i\mathbf{O} \in \mathfrak{g}$, or in other words

$$\mathbf{O} = \sum_\alpha \mu_\alpha \mathbf{B}_\alpha, \tag{10}$$

where $\mu_\alpha$ is the coefficient of support of $\mathbf{O}$ in the basis $\mathbf{B}_\alpha$.

**Definition 5.** (Lie Algebra Supported Ansatz[36]). A Lie Algebra Supported Ansatz (LASA) is a periodic ansatz of the form Eq. (2) of a VQC where the measurement operator $\boldsymbol{O}$ is completely supported by the DLA $\mathfrak{g}$ associated with the generators of $U(\boldsymbol{\theta})$, that is,

$$i\boldsymbol{O} \in \mathfrak{g}. \tag{11}$$

In addition to its connections to the trainability of a VQC, this condition also implies that $\forall \boldsymbol{\theta}$, $U^\dagger(\boldsymbol{\theta}) i\boldsymbol{O} U(\boldsymbol{\theta}) \in \mathfrak{g}$, which enables us to express the evolution of the observable $\boldsymbol{O}$ in terms of elements of $\mathfrak{g}$. This is key to some simulation algorithms that are possible for polynomial-sized DLAs[37,38].

## Input recoverability definitions

In this section, we provide meaningful definitions of what it means to recover the classical input data given access to the gradients $\{C_j\}_{j=1}^{D}$ of a VQC. Notably, our definitions are motivated in a manner that allows us to consider the encoding and variational portions of a quantum variational model separately.

A useful concept in machine learning is the creation of data *snapshots*. These snapshots are compact and efficient representations of the input data's feature map encoding. Essentially, a snapshot retains enough information to substitute for the full feature map encoded data, enabling the training of a machine learning model for a distinct task with the same data but without the need to explicitly know the input data was passed through the feature map. For example, in methods such as $\mathfrak{g}$-sim[38], these snapshots are used as input vectors for classical simulators. The simulator can then process these vectors efficiently under certain conditions, recreating the operation of a variational quantum circuit.

It will become useful to classify the process of input data $\mathbf{x}$ recovery into two stages; the first concerns recovering snapshots of the quantum state $\rho(\mathbf{x})$ (Eq (1)) from the gradients, which involves only considering the variational part of the circuit.

**Definition 6**. (Snapshot Recovery). Given the gradients $C_j$, $j \in [D]$ as defined in Eq (5) as well as the parameters $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_D]$, we consider a VQC to be snapshot recoverable if there exists an efficient $\mathcal{O}(poly(d, \frac{1}{\epsilon}))$ classical polynomial time algorithm to recover the vector $\mathbf{e}_{\text{snap}}$ such that,

$$|[\mathbf{e}_{\text{snap}}]_\alpha - \text{Tr}(\mathbf{B}_\alpha \rho(\mathbf{x}))| \leq \epsilon, \forall \alpha \in [\dim(\mathfrak{g})], \tag{12}$$

for some $\{\mathbf{B}_\alpha\}$ forming a Frobenius-orthonormal basis of the DLA $\mathfrak{g}$ corresponding to $U(\boldsymbol{\theta})$ in Eq. (2), and the above holds for any $\epsilon > 0$. We call $\mathbf{e}_{\text{snap}}$ the snapshot of $\mathbf{x}$.

In other words, $\mathbf{e}_{\text{snap}}$ is the orthogonal projection of the input state $\rho(\mathbf{x})$ onto the DLA of the ansatz, and thus the elements of $\mathbf{e}_{\text{snap}}$ are the only components of the input state that contribute to the generation of the model output $y_{\boldsymbol{\theta}}(\mathbf{x})$ as defined in Eq. (3). Here, we constitute the retrieval of the snapshot $\mathbf{e}_{\text{snap}}$ of a quantum state $\rho(\mathbf{x})$ as *weak privacy* breach, since the snapshot could be used to train the VQC model for other learning tasks involving the same data $\{\mathbf{x}\}$ but without the need to use the actual data. As an example, consider an adversary that has access to the snapshots corresponding to the data of certain customers. Their task is to train the VQC to learn the distinct behavioral patterns of the customers. It becomes apparent that the adversary can easily carry out this task without ever needing the original data input since the entire contribution of the input $\mathbf{x}$ in the VQC output decision-making $y_{\boldsymbol{\theta}}(\mathbf{x})$ is captured by $\mathbf{e}_{\text{snap}}$.

Next, we consider the stronger notion of privacy breach in which the input data $\mathbf{x}$ must be fully reconstructed. Assuming that the snapshot has been recovered, the second step we therefore consider is inverting the recovered snapshot $\mathbf{e}_{\text{snap}}$ to find the original data $\mathbf{x}$, a process that is primarily dependent on the encoding part of the circuit. Within our snapshot inversion definition, we consider two cases that enable different solution strategies: snapshot inversion utilizing purely classical methods and snapshot inversion methods that can utilize quantum samples.

**Definition 7**. (Classically Snapshot Invertible Model). Given the snapshot $\mathbf{e}_{\text{snap}}$ as the expectation values of the input state $\rho(\mathbf{x})$, we say that VQC admits classical snapshot invertibility if there exists an efficient $\mathcal{O}(\text{poly}(\text{d}, \frac{1}{\epsilon}))$ polynomial time classical randomized algorithm to recover

$$\mathbf{x}' : \|\mathbf{x}' - \mathbf{x}\|_2 \leq \epsilon, \tag{13}$$

with probability at least $p = \frac{2}{3}$, for any user defined $\epsilon > 0$.

**Definition 8**. (Quantum Assisted Snapshot inversion). Given the snapshot $\mathbf{e}_{\text{snap}}$ as the expectation values of the input state $\rho(\mathbf{x})$, and the ability to query $\text{poly}(\text{d}, \frac{1}{\epsilon})$ number of samples from the encoding circuit $V$ to generate snapshots $\mathbf{e}'_{\text{snap}}$ for any given input $\mathbf{x}'$, we say that VQC admits quantum-assisted snapshot invertibility, if there exists an efficient $\mathcal{O}(\text{poly}(\text{d}, \frac{1}{\epsilon}))$ polynomial time classical randomized algorithm to recover

$$\mathbf{x}' : \| \mathbf{x}' - \mathbf{x}\|_2 \leq \epsilon, \tag{14}$$

with probability at least $p = \frac{2}{3}$, for any user defined $\epsilon > 0$.

In this work, we specifically focus on input recoverability by considering the conditions under which VQC would admit snapshot recovery followed by snapshot invertibility. Considering these two steps individually allows us to delineate the exact mechanisms that contribute to the overall recovery of the input.

It is important to mention that it may potentially only be possible to recover the inputs of a VQC up to some periodicity, such that there only exists a classical polynomial time algorithm to recover $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{k}\pi$ up to $\epsilon$-closeness, where $\mathbf{k} \in \mathbb{Z}$. As the encodings generated by quantum feature maps inherently contain trigonometric terms, in the most general case it may therefore only be possible to recover $\mathbf{x}$ up to some periodicity. However, this can be relaxed if the quantum feature map is assumed to be injective.

Figure 2 shows a diagram that highlights the Lie algebraic simulation method[38] along with specifications of the input recovery framework as defined in this work.

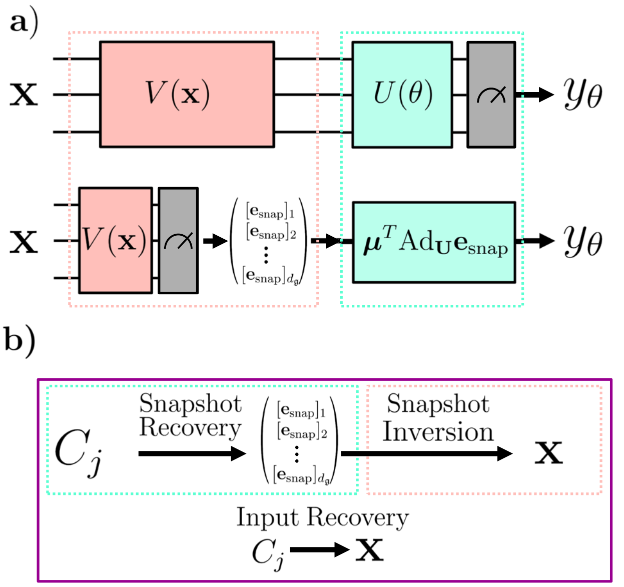

a)
X
V(x)
U(θ)
$y_\theta$
X
V(x)
$[\mathbf{e}_{\text{snap}}]_1$
$[\mathbf{e}_{\text{snap}}]_2$
$[\mathbf{e}_{\text{snap}}]_{d_\mathfrak{g}}$
$\boldsymbol{\mu}^T \text{Ad}_\mathbf{U} \mathbf{e}_{\text{snap}}$
$y_\theta$
b)
$C_j$
Snapshot Recovery
$[\mathbf{e}_{\text{snap}}]_1$
$[\mathbf{e}_{\text{snap}}]_2$
$[\mathbf{e}_{\text{snap}}]_{d_\mathfrak{g}}$
Snapshot Inversion
X
Input Recovery
$C_j \rightarrow$ X

**Fig. 2 | Visualization of the full privacy attack process. a** Visualization of the difference between the circuit implementation of a variational quantum model and a Lie algebraic simulation procedure of the same model[38]. In the Lie algebraic Simulation framework[38], input data $\mathbf{x}$ is encoded into a quantum circuit using $V(\mathbf{x})$, however, the measurements are then performed on this encoded state and used to form a vector of snapshot expectation values. This vector of snapshot expectation values can then be passed as inputs to a classical simulator that uses the adjoint form of $U(\theta)$, which can be performed with resources scaling with the dimension of the DLA formed by the generators of $U(\theta)$. **b** In this work, we assess the ability to recover an input $\mathbf{x}$ from gradients $C_j$. This can be broken into two parts: Firstly, the snapshot $\mathbf{e}_{\text{snap}}$ must be recovered from the gradients $C_j$, which corresponds to reversing the Lie algebraic simulation step. Secondly, the recovered snapshot $\mathbf{e}_{\text{snap}}$ must be inverted to find the original data $\mathbf{x}$, which requires finding the values of $\mathbf{x}$ that when input into $V(\mathbf{x})$ will give the same snapshot values $\mathbf{e}_{\text{snap}}$. If both snapshot recovery and snapshot inversion can be performed, then it admits efficient input recovery.

## Snapshot recovery

This section addresses the *weak privacy* notion of recovering the snapshots of the input as introduced in Def 6. As the name implies, the goal here is to recover the vector $\mathbf{e}_{\text{snap}}$ for some Schmidt orthonormal basis $\{\mathbf{B}_\alpha\}_{\alpha\in\dim(\mathfrak{g})}$ of the DLA corresponding to the VQC ansatz $\mathbf{U}(\boldsymbol{\theta})$, given that the attacker is provided the following information,

1. $D$ gradient information updates $C_j = \frac{\partial y_{\boldsymbol{\theta}}(\mathbf{x})}{\partial\theta_j}, j\in[D]$ as defined in Eq. (5).
2. Ansatz architecture $\mathbf{U}(\boldsymbol{\theta})$ presented as an ordered sequence of Hermitian generators $\{\theta_k, \mathbf{H}_{\nu(k)}\}_{k=1}^{D}$, where $\mathbf{H}_{\nu(k)}$ is expressed as a polynomial (in the number of qubits) linear combination of Pauli strings.
3. Measurement operator $\mathbf{O}$, which satisfies the LASA condition according to Def 5 and is expressed as a polynomial (in the number of qubits) linear combination of Pauli strings

Recovering these snapshots will enable an attacker to train the VQC model for other learning tasks that effectively extract the same information from the input states $\rho(\mathbf{x})$ but without the need to use the actual data. The main component of the snapshot recoverability algorithm makes use of the $\mathfrak{g}$-sim[37,38] framework, which we briefly review in the following subsection while also clarifying some previously implicit assumptions, to construct a system of linear equations that can be solved to recover $\mathbf{e}_{\text{snap}}$ as detailed in Algorithm 2.

**Review of Lie-algebraic simulation framework**. We start by reviewing the $\mathfrak{g}$-sim framework[37,38] for classically computing the cost function and gradients of VQCs, when the observable lies in the DLA of the chosen ansatz. Specifically, this framework evolves the expectation values of observables via the adjoint representation. However, a necessary condition for this procedure to be efficient is that the dimension of the DLA ( $\dim(\mathfrak{g})$) is only polynomially growing in the number of qubits.

The first step of $\mathfrak{g}$-sim consists of building an orthonormal basis for the DLA $\mathfrak{g}$ given $(\{\theta_k, \mathbf{H}_{\nu(k)}\})_{k=1}^{D}$. Algorithm 1 presents a well-known procedure to do this. The procedure simply computes pairwise commutators until no new linearly independent elements are found. Given that all operators are expressed in the Pauli basis, the required orthogonal projectors and norm computations performed by Algorithm 1 can be performed efficiently. If the dimension of DLA is $\mathcal{O}(\text{poly}(\text{n}))$, then the iteration complexity, i.e., the number of sets of commutators that we compute, of this procedure is polynomial in $n$. However, an important caveat is that potentially the elements forming our estimation for the DLA basis could have exponential support on the Pauli basis, which is a result of computing new pairwise commutators at each iteration. Thus, for this overall procedure to be efficient, we effectively require that the nested commutators of the generators $\mathbf{H}_k$ do not have exponential support on the Pauli basis.

**Definition 9**. (Slow Pauli Expansion). A set of Hermitian generators $\{\mathbf{H}_1, \ldots, \mathbf{H}_N\}$ on $n$-qubits expressed as linear combinations of $\mathcal{O}(\text{poly}(\dim(\mathfrak{g})))$ Pauli strings satisfies the slow Pauli expansion condition if $\forall\, r\in[N]$, $[\mathbf{H}_r, [\cdots, [\mathbf{H}_2, \mathbf{H}_1]]]$ can be expressed as a linear combination of $\mathcal{O}(\text{poly}(\dim(\mathfrak{g})))$ Pauli strings.

In general, it is unclear how strong of an assumption this is, which means that the attacks that we present may not be practical for all VQCs that satisfy the polynomial DLA condition, and thus privacy preservation may still be possible. Also, it does not seem to be possible to apply the $\mathfrak{g}$-sim framework without the slow Pauli expansion condition. Lastly, a trivial example of a set of Hermitian generators that satisfies the slow Pauli expansion is those for the quantum compound ansatz discussed in ref. 36.

**Algorithm 1**. Finding DLA basis

```
Require: Hermitian circuit generators $\{H_1, \ldots, H_N\}$, all elements are linear
    combinations of polynomially-many Pauli strings
Ensure: $\mathcal{A}''' = \{\mathbf{B}_1, \ldots, \mathbf{B}_{\dim(\mathfrak{g})}\}$ as the basis for the DLA $\mathfrak{g}$
    1. Let $\mathcal{A} = \{H_1, \ldots, H_N\}$, with all elements represented in the
       Pauli basis.
    2. Repeat until breaks
        (a) Compute pairwise commutators of elements of $\mathcal{A}$ into $\mathcal{A}'$
        (b) Orthogonally project $\mathcal{A}'$ onto the orthogonal complement of $\mathcal{A}$
            in $\mathfrak{g}$
        (c) Set new $\mathcal{A}''$ to be $\mathcal{A}$ plus new orthogonal elements. If no new
            elements, break.
    3. Perform Gram–Schmidt on $\mathcal{A}$ forming $\mathcal{A}'''$.
    4. Return $\mathcal{A}'''$.
```

Given the orthonormal basis $\mathbf{B}_\alpha$ for $\mathfrak{g}$, under the LASA condition, we can express $\mathbf{O} = \sum_{\alpha\in[\dim(\mathfrak{g})]}\mu_\alpha\mathbf{B}_\alpha$, and hence we can write the output as

$$\begin{aligned} y_{\boldsymbol{\theta}}(\mathbf{x}) = {} & \operatorname{Tr}(\mathbf{U}^\dagger(\boldsymbol{\theta})\mathbf{O}\mathbf{U}(\boldsymbol{\theta})\rho(\mathbf{x})) = \sum_\alpha \operatorname{Tr}(\mu_\alpha\mathbf{U}^\dagger\mathbf{B}_\alpha\mathbf{U}\rho(\mathbf{x})) \\ & = \sum_\alpha \operatorname{Tr}(\mu_\alpha \mathrm{Ad}_{\mathbf{U}}(\mathbf{B}_\alpha)\rho(\mathbf{x})). \end{aligned} \tag{15}$$

In addition, given the form of $\mathbf{U}$, we can express $\mathrm{Ad}_{\mathbf{U}}$ as,

$$\mathrm{Ad}_{\mathbf{U}} = \prod_{k=1}^{D} e^{-\theta_k \mathrm{ad}_{i\mathbf{H}_{\nu(k)}}}. \tag{16}$$

We can also compute the structure constants for our basis $\mathbf{B}_\alpha$, which is the collection of $\dim(\mathfrak{g})\times\dim(\mathfrak{g})$ matrices for the operators $\mathrm{ad}_{i\mathbf{B}_\alpha}$. As a result of linearity, we also have the matrix for each $\mathrm{ad}_{i\mathbf{H}}$ for $\mathbf{H}\in\mathfrak{g}$ in the basis $\mathbf{B}_\alpha$. Then, by performing matrix exponentiation and multiplying $\dim(\mathfrak{g})\times\dim(\mathfrak{g})$ we can compute the matrix for $\mathrm{Ad}_{\mathbf{U}}$.

Using the above, the model output may be written,

$$y_\theta = \sum_{\alpha,\beta}\mu_\alpha[\mathrm{Ad}_{\mathbf{U}}]_{\alpha\beta}\operatorname{Tr}(\mathbf{B}_\beta\rho(\mathbf{x})) = \boldsymbol{\mu}^{\mathrm{T}}\mathrm{Ad}_{\mathbf{U}}\mathbf{e}_{\text{snap}}, \tag{17}$$

where $\mathbf{e}_{\text{snap}}$ is a vector of expectation values of the initial state, i.e., $[\mathbf{e}_{\text{snap}}]_\beta = \operatorname{Tr}[\mathbf{B}_\beta\rho(\mathbf{x})]$.

Similar to the cost function, the circuit gradient can also be computed via $\mathfrak{g}$-sim. Let,

$$C_j = \frac{\partial y_\theta}{\partial\theta_j} = \boldsymbol{\mu}^T\frac{\partial \mathrm{Ad}_{\mathbf{U}}}{\partial\theta_j}\mathbf{e}_{\text{snap}} =: \chi^{(j)}\cdot\mathbf{e}_{\text{snap}}, \tag{18}$$

where the adjoint term differentiated with respect to $\theta_j$ can be written as,

$$\frac{\partial \mathrm{Ad}_{\mathbf{U}}}{\partial\theta_j} = \left[\prod_{k=j}^{D} e^{\theta_k \mathrm{ad}_{i\mathbf{H}_{\nu(k)}}}\right]\mathrm{ad}_{i\mathbf{H}_{\nu(j)}}\left[\prod_{k=1}^{j} e^{\theta_k \mathrm{ad}_{i\mathbf{H}_{\nu(k)}}}\right]. \tag{19}$$

The components of $\chi^{(j)}$ can be expressed as,

$$\chi_\beta^{(j)} = \sum_\alpha \mu_\alpha\left[\frac{\partial\mathrm{Ad}_{\mathbf{U}}}{\partial\theta_j}\right]_{\alpha,\beta}, \tag{20}$$

allowing $C_j$ terms to be represented in a simplified manner as

$$C_j = \sum_{\beta=1}^{\dim(\mathfrak{g})}\chi_\beta^{(j)}[\mathbf{e}_{\text{snap}}]_\beta. \tag{21}$$

The key feature of this setup is that the matrices and vectors involved have dimension $\dim(\mathfrak{g})$, therefore for a polynomial-sized DLA, the simulation time will scale polynomially and model outputs can be calculated in polynomial time[38]. Specifically, the matrices for each $\mathrm{ad}_{i\mathbf{H}_k}$ in the basis $\{\mathbf{B}_l\}$ and $\mathrm{Ad}_{\mathbf{U}}$ are polynomial in this case.

This Lie-algebraic simulation technique was introduced in order to show efficient methods of simulating LASA circuits with polynomially sized DLA. In this work, we utilize the framework in order to investigate the snapshot recovery of variational quantum algorithms. Based on the above discussion, the proof of the following theorem is self-evident.

**Theorem 1**. (Complexity of $\mathfrak{g}$-sim). If ansatz family $\mathbf{U}(\boldsymbol{\theta})$ with an observable $\mathbf{O}$ satisfies both the LASA condition and Slow Pauli Expansion, then the cost function and its gradients can be simulated with complexity $\mathcal{O}(\mathrm{poly}(\dim(\mathfrak{g})))$ using a procedure that at most queries a quantum device a polynomial number of times to compute the $\dim(\mathfrak{g})$-dimensional snapshot vector $\mathbf{e}_{\mathrm{snap}}$.

### Snapshot recovery algorithm

**Algorithm 2**. Snapshot Recovery

```
Require: Observable $\mathbf{O}$ such that $i\mathbf{O} \in \mathfrak{g}$, generators $\{\mathbf{H}_{\nu(k)}\}_{k=1}^{D}$, ordered sequence $(\{\theta_k, \mathbf{H}_{\nu(k)}\})_{k=1}^{D}$, and gradients $C_j = \frac{\partial y_{\boldsymbol{\theta}}(\mathbf{x})}{\partial \theta_j}, j \in [D]$ for some unknown classical input $\mathbf{x}$.
Ensure: Snapshot $\mathbf{e}_{\mathrm{snap}}$ for $\mathbf{x}$
    1. Run Algorithm 1 to obtain an orthonormal basis for the DLA $\{\mathbf{B}_\beta\}_{\beta\in[\dim(\mathfrak{g})]}$
    2. For $\beta \in [\dim(\mathfrak{g})]$, compute the $\dim(\mathfrak{g}) \times \dim(\mathfrak{g})$ matrix $\mathrm{ad}_{i\mathbf{B}_\beta}$
    3. For $k \in [D]$, compute the coefficients of $\mathbf{H}_{\nu(k)}$ in the basis $\{\mathbf{B}_\beta\}_{\beta\in[\dim(\mathfrak{g})]}$, which gives us $\mathrm{ad}_{i\mathbf{H}_{\nu(k)}}$
    4. For $k \in [D]$, compute the $\dim(\mathfrak{g}) \times \dim(\mathfrak{g})$ matrix exponential $e^{\theta_k \mathrm{adi}\mathbf{H}_{\nu(\mathrm{k})}}$
    5. For $j \in [D]$ compute the $\dim(\mathfrak{g}) \times \dim(\mathfrak{g})$ matrix
         $$\frac{\partial \mathrm{Ad}_{\mathbf{U}}}{\partial \theta_j} = \left[\prod_{k=j}^{D} e^{\theta_k \mathrm{ad}_{i\mathbf{H}_{\nu(k)}}}\right] \mathrm{ad}_{i\mathbf{H}_{\nu(j)}} \left[\prod_{k=1}^{j} e^{\theta_k \mathrm{ad}_{i\mathbf{H}_{\nu(k)}}}\right]. \qquad (22)$$
    6. For $\beta \in [\dim(\mathfrak{g})]$, compute the coefficients $\mu_\beta$ of $\mathbf{O}$ in the basis $\{\mathbf{B}_\beta\}_{\beta\in[\dim(\mathfrak{g})]}$
    7. For $j \in [D], \beta \in [\dim(\mathfrak{g})]$, compute
         $$\chi_\beta^{(j)} = \sum_\alpha \mu_\alpha \left[\frac{\partial \mathrm{Ad}_{\mathbf{U}}}{\partial \theta_j}\right]_{\alpha,\beta}, \qquad (23)$$
       and construct $D \times \dim(\mathfrak{g})$ matrix $\mathbf{A}$ with $\mathbf{A}_{rs} = \chi_s^{(r)}$.
    8. Solve the following linear system,
         $$[C_1, \ldots, C_D]^{\mathsf{T}} = \mathbf{A}\mathbf{y}, \qquad (24)$$
       and return $\mathbf{y}$ as the snapshot $\mathbf{e}_{\mathrm{snap}}$ ..
```

With the framework for the $\mathfrak{g}$-sim[38] established, we focus on how snapshots $\mathbf{e}_{\mathrm{snap}}$ of the input data can be recovered using the VQC model gradients $C_j$, with the process detailed in Algorithm 2. In particular, the form of Eq (21) allows a set-up leading to the recovery the snapshot vector $\mathbf{e}_{\mathrm{snap}}$ from the gradients $\{C_j\}_{j=1}^{D}$, but requires the ability to solve the system of $D$ linear equations given by $\{C_j\}$ with $\dim(\mathfrak{g})$ unknowns $[\mathbf{e}_{\mathrm{snap}}]_{\beta\in\dim(\mathfrak{g})}$. The following theorem formalizes the complexity of recovering the snapshots from the gradients.

**Theorem 2**. (Snapshot Recovery). Given the requirements specified in Algorithm 2, along with the assumption that the number of variational parameters $D \geq \dim(\mathfrak{g})$, where $\dim(\mathfrak{g})$ is the dimension of the DLA $\mathfrak{g}$, the VQC model admits snapshot $\mathbf{e}_{\mathrm{snap}}$ recovery with complexity scaling as $\mathcal{O}(\mathrm{poly}(\dim(\mathfrak{g})))$.

**Proof**. Firstly, we note that given the gradients $C_j$ and parameters $\theta_{j\in[D]}$, the only unknowns are the components of the vector $\mathbf{e}_{\mathrm{snap}}$ of length $\dim(\mathfrak{g})$. Therefore, it is necessary to have $\dim(\mathfrak{g})$ equations in total; otherwise, the system of equations would be underdetermined, and it would be impossible to find a unique solution. The number of equations depends on the number of gradients and, therefore, the number of variational parameters in the model; hence, the requirement that $D \geq \dim(\mathfrak{g})$.

Assuming now that we deal with the case where there are $D \geq \dim(\mathfrak{g})$ variational parameters of the VQC model, we can therefore arrive at a determined system of equations. The resulting system of simultaneous equations can be written in a matrix form as,

$$\begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_D \end{pmatrix} = \begin{pmatrix} \chi_1^{(1)} & \chi_2^{(1)} & \cdots & \chi_{\dim(\mathfrak{g})}^{(1)} \\ \chi_1^{(2)} & \chi_2^{(2)} & \cdots & \chi_{\dim(\mathfrak{g})}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ \chi_{\dim(\mathfrak{g})}^{(D)} & \chi_{\dim(\mathfrak{g})}^{(D)} & \cdots & \chi_{\dim(\mathfrak{g})}^{(D)} \end{pmatrix} \begin{pmatrix} [\mathbf{e}_{\mathrm{snap}}]_1 \\ [\mathbf{e}_{\mathrm{snap}}]_2 \\ \vdots \\ [\mathbf{e}_{\mathrm{snap}}]_{\dim(\mathfrak{g})} \end{pmatrix} \qquad (25)$$

In order to solve the system of equations highlighted in Eq. (25) to obtain $\mathbf{e}_{\mathrm{snap}}$, we first need to compute the coefficients $\{\chi_\beta^{(j)}\}_{j\in[D],\beta\in[\dim(\mathfrak{g})]}$. This can be done by the $\mathfrak{g}$-sim procedure highlighted in the previous section and in steps 1-7 in Algorithm 2 with complexity $\mathcal{O}(\mathrm{poly}(\dim(\mathfrak{g})))$. The next step is to solve the system of equations, i.e., step 8 of Algorithm 2, which can solved using Gaussian elimination procedure incurring a complexity $\mathcal{O}(\dim(\mathfrak{g})^3)$[42]. Thus, the overall complexity of recovering the snapshots from the gradients is $\mathcal{O}(\mathrm{poly}(\dim(\mathfrak{g})))$. This completes the proof.

In the case that the dimension of DLA is exponentially large $\dim(\mathfrak{g}) = \mathcal{O}(\exp(\mathrm{n}))$, then performing snapshot recovery by solving the system of equations would require an exponential number of gradients and thus an exponential number of total trainable parameters $D = \mathcal{O}(\exp(n))$. However, this would require storing an exponential amount of classical data, as even the variational parameter array $\boldsymbol{\theta}$ would contain $\mathcal{O}(\exp(n))$ many elements, and hence this model would already breach the privacy definition, which only allows for a polynomial (in $n = \Theta(d)$) time attacker. In addition, the complexity of obtaining the coefficients $\chi_\beta^{(j)}$ and subsequently solving the system of linear equations would also incur an exponential cost in $n$. Hence, for the system of simultaneous equations to be determined, it is required that $\dim(\mathfrak{g}) = \mathcal{O}(\mathrm{poly}(\mathrm{n}))$. Under the above requirement, it will also be possible to solve the system of equations in Eq. (25) in polynomial time and retrieve the snapshot vector $\mathbf{e}_{\mathrm{snap}}$. Hence, a model is snapshot recoverable if the dimension of the DLA scales polynomially in $d$.

### Snapshot invertibility

We have shown that in the case that the DLA dimension of the VQC is polynomial in the number of qubits $n$ and the slow Pauli expansion condition (Def 9) is satisfied, then it is possible to reverse engineer the snapshot vector $\mathbf{e}_{\mathrm{snap}}$ from the gradients. As a result, this breaks the weak-privacy criterion. The next step in terms of privacy analysis is to see if a strong privacy breach can also occur. This is true when it is possible to recover the original data $\mathbf{x}$ that was used in the encoding step to generate the state $\rho(\mathbf{x})$; the expectation values of this state with respect to the DLA basis elements form the snapshot $\mathbf{e}_{\mathrm{snap}}$. Hence, even if the DLA is polynomial and snapshot recovery allows the discovery of $\mathbf{e}_{\mathrm{snap}}$, there is still the possibility of achieving some input privacy if $\mathbf{e}_{\mathrm{snap}}$ cannot be efficiently inverted to find $\mathbf{x}$. The overall privacy of the VQC model, therefore, depends on both the data encoding and the variational ansatz.

One common condition that is necessary for our approaches to snapshot inversion is the ability to compute the expectation values $\mathrm{Tr}\,(\rho(\mathbf{x}')\mathbf{B}_\mathrm{k}), \forall \mathrm{k} \in [\,\dim(\mathfrak{g})]$ for some guess input $\mathbf{x}'$. This is the main condition that distinguishes between completely classical snapshot inversion and quantum-assisted snapshot inversion. It is well-known that computing expectation values of specific observables is a weaker condition than $\rho(\mathbf{x})$ being classically simulatable[43]. Hence, it may be possible to classically perform snapshot inversion even if the state $\rho(\mathbf{x})$ overall is hard to classically simulate. In the quantum-assisted case, it is always possible to calculate $\mathrm{Tr}\,(\rho(\mathbf{x}'\mathbf{B}_\mathrm{k})$ values by taking appropriate measurements of the encoding circuit $V(\mathbf{x}')$.

In the first subsection, we present inversion attacks that apply to commonly used feature maps and explicitly make use of knowledge about the locality of the encoding circuit. The common theme among these feature maps is that by restricting to only a subset of the inputs, it is possible to express the $\rho(\mathbf{x})$ or expectations thereof in a simpler way. The second subsection focuses on arbitrary encoding schemes by viewing the problem as black-box optimization. In general, snapshot inversion can be challenging or intractable even if the snapshots can be efficiently recovered and/or the feature map can be classically simulated. Our focus will be on presenting sufficient conditions for performing snapshot inversion, which leads to suggestions for increasing privacy.

**Snapshot inversion for local encodings**. For efficiency reasons, it is common to encode components of the input vector $\mathbf{x}$ in local quantum gates, typically just single-qubit rotations. The majority of the circuit complexity is usually either put into the variational part or via non-parameterized entangling gates in the feature map. In this section, we demonstrate attacks to recover components of $\mathbf{x}$, up to periodicity, given snapshot vectors when the feature map encodes each $x_j$ locally. More specifically, we put bounds on the allowed amount of interaction between qubits that are used to encode each $x_j$. In addition, we also require that the number of times the feature map can encode a single $x_j$ be sufficiently small. While the conditions will appear strict, we note that they are satisfied for some commonly used encodings, e.g., the *Pauli product feature map* or *Fourier tower map*[30], which was previously used in a VQC model that demonstrated resilience to input recovery.

For the Pauli product encoding, we show that a completely classical snapshot inversion attack is possible. An example of a Pauli product encoding is the following:

$$\bigotimes_{j_1}^{n} \rho_j(x_j) = \bigotimes_{j_1}^{n} R_\mathsf{X}(x_j)|0\rangle\langle 0|R_\mathsf{X}(-x_j). \tag{26}$$

where $R_\mathsf{X}$ is the parameterized Pauli $\mathsf{X}$ rotation gate. The Fourier tower map is similar to Eq. (26) but utilizes a parallel data reuploading scheme, i.e.,

$$\bigotimes_{j=1}^{d}\left(\bigotimes_{l=1}^{m} R_\mathsf{X}(5^{l-1}x_j)\right). \tag{27}$$

where $n = dm$, with $m$ being the number of qubits used to encode a single dimension of the input.

**1. Pauli Product Encoding**: The first attack that we present will specifically target Eq. (26). However, the attack does apply to the Fourier tower map as well. More generally, the procedure applies to any parallel data reuploading schemes of the form:

$$\bigotimes_{j=1}^{d}\left(\bigotimes_{l=1}^{m} R_\mathsf{X}(\alpha_l x_j)\right). \tag{28}$$

We explicitly utilize Pauli $\mathsf{X}$ rotations, but a similar result holds for $\mathsf{Y}$ or $\mathsf{Z}$. For a Pauli operator $\mathsf{P}$, let $\mathsf{P}_j := i\mathbb{I}^{\otimes(j-1)} \otimes \mathsf{P} \otimes \mathbb{I}^{\otimes(n-j)}$.

**Algorithm 3**. Classical Snapshot Inversion for Pauli Product Encoding

```
Require: Snapshot vector $\mathbf{e}_{\text{snap}}(\mathbf{x})$ of dimension $\dim(\mathfrak{g}) = \mathcal{O}(\mathrm{poly}\,(\mathrm{n}))$
  corresponding to a basis $(\mathbf{B}_k)_{k=1}^{\dim(\mathfrak{g})}$ of DLA $\mathfrak{g}$. Each $\mathbf{B}_k$ is expressed as a
  linear combination of $\mathcal{O}(\mathrm{poly}\,(\mathrm{n}))$ Pauli strings. Snapshot inversion is
  being performed for a VQC model that utilizes a trainable portion of $\mathbf{U}(\boldsymbol{\theta})$
  with DLA $\mathfrak{g}$ and Pauli product encoding Eq. (26). Index $j \in [d]$, $\epsilon < 1$
Ensure: An $\epsilon$ estimate of the $j$th component $x_j$ of the data input $\mathbf{x} \in \mathbb{R}^d$ up
    to periodicity, or output FAILURE.
  If $i\mathsf{Z}_j \in \mathfrak{g}$ then
    $\alpha \leftarrow 1, \beta \leftarrow 0$
    $\mathbf{W} \leftarrow \mathsf{Y}_j$
  else if $i\mathsf{Y}_j \in \mathfrak{g}$ then
    $\alpha \leftarrow 1, \beta \leftarrow 0$
    $\mathbf{W} \leftarrow \mathsf{Y}_j$
  else
    1.Determine set of Pauli strings required to span elements $(i\mathbf{B}_k)_{k=1}^{\dim(\mathfrak{g})}$
      and denote the set $\mathcal{P}_\mathfrak{g}$.
    2. $\mathcal{P}_\mathfrak{g} \leftarrow \mathcal{P}_\mathfrak{g} \cup \{\mathsf{Z}_j, \mathsf{Y}_j\}$,  $|\mathcal{P}_\mathfrak{g}| = \mathcal{O}(\mathrm{poly}\,(\mathrm{n}))$ by assumption.
       Reduce $\mathcal{P}_\mathfrak{g}$ to a basis.
    3. Let $\mathbf{C}$ be a $|\mathcal{P}_\mathfrak{g}| \times \dim(\mathfrak{g})$ matrix whose $k$-th column corresponds
       to the components of $i\mathbf{B}_k$ in the basis $\mathcal{P}_\mathfrak{g}$.
    4. Let $\mathbf{A}$ be a $|\mathcal{P}_\mathfrak{g}| \times 2$ whose first column contains a 1 in the row
       corresponding to $\mathsf{Z}_j$ and whose second column contains a 1 in the
       row corresponding to $\mathsf{Y}_j$.
    5. Perform a singular value decomposition on $\mathbf{A}^\mathsf{T}\mathbf{C}$, and there are at
       most two nonzero singular values $r_1, r_2$.
    if $r_1 \neq 1$ and $r_2 \neq 1$ then
      return FAILURE
  else
      1. $\mathbf{W} \leftarrow$ singular vector with singular value 1.
      2. Expand $i\mathbf{W}$ in basis $(i\mathsf{Z}_j, i\mathsf{Y}_j)$ record components as $\alpha$ and $\beta$,
         respectively.
    end if
  end if
    1. Expand $i\mathbf{W}$ in basis $(\mathbf{B}_k)_{k=1}^{\dim(\mathfrak{g})}$, and record components as $\gamma_k$.
    2. Compute

  $\tilde{x}_j = \cos^{-1}\left[\frac{2}{\mathrm{sign}\,(\alpha)\sqrt{\alpha^2+\beta^2}}\sum_{k=1}^{\dim(\mathfrak{g})}\gamma_k[\mathbf{e}_{\text{snap}}]_k\right] - \tan^{-1}(\beta/\alpha).$   (29)

    3. return $\tilde{x}_j$.
```

**Theorem 3**. Suppose that the polynomial DLA and slow Pauli expansion (Def 9) conditions are satisfied. Also, suppose that we are given a snapshot vector $\mathbf{e}_{\text{snap}}(\mathbf{x})$ for a VQC with trainable portion $\mathbf{U}(\boldsymbol{\theta})$ with DLA $\mathfrak{g}$ and Pauli product feature encoding (Eq. (26)) and the corresponding DLA basis elements $(\mathbf{B}_k)_{k=1}^{\dim(\mathfrak{g})}$. The classical Algorithm 3 outputs an $\epsilon$ estimate of $x_j$, up to periodicity, or outputs FAILURE, with time $\mathcal{O}(\mathrm{poly}\,(\mathrm{n})\log(1/\epsilon))$.

**Proof**. We provide the proof in the methods section.

For illustrative purposes, we show in Fig. 3 the snapshot inversion process for the special case where $i\mathsf{Z}_j \in \mathfrak{g}$, i.e.,

$$x_j = \cos^{-1}\left(2\boldsymbol{\gamma}^{(j)} \cdot \mathbf{e}_{\text{snap}}\right), \tag{30}$$

for $i\mathsf{Z}_j = \sum_{k=1}^{\dim(\mathfrak{g})} \gamma_k^{(j)}\mathbf{B}_k$. The general parallel data reuploading case can be handled by applying the procedure to only one of the rotations that encodes at $x_j$ at a time, checking to find one that does not cause the algorithm to return FAILURE.

**2. General Pauli Encoding**: We now present a more general procedure that applies to feature maps that use serial data reuploading and multi-qubit Paulis. However, we introduce a condition that ensures that each $x_j$ is locally encoded. More generally, we focus our discussion on encoding states that may be written as a tensor product of $\Omega$ subsystems, i.e., multipartite states.

$$\rho(\mathbf{x}) = \bigotimes_{J\in\mathcal{P}} \rho_J(\mathbf{x}), \tag{31}$$

where $\dim(\mathsf{x}_J)$ is constant. The procedure is highlighted in Algorithm 4 and requires solving a system of polynomial equations.

In addition, the procedure may not be completely classical as quantum assistance may be required to compute certain expectation values of $\rho_J(\mathbf{x})$, specifically with respect to the DLA basis elements. For simplicity, the algorithm and the theorem characterizing the runtime ignore potential errors in estimating these expectations. If classical estimation is possible, then we can potentially achieve a $\mathcal{O}(\text{poly}(\log(1/\epsilon)))$ scaling. However, if we must use quantum, then we will incur a $\mathcal{O}(1/\epsilon)$ (due to amplitude estimation) dependence, which can be significant. Theorem 4 presents the attack complexity, ignoring these errors.

```
Algorithm 4. Snapshot Inversion for General Pauli Encodings
Require: Snapshot vector $\mathbf{e}_{\text{snap}}(\mathbf{x})$ of dimension $\dim(\mathfrak{g}) = \mathcal{O}(\text{poly}(n))$
    corresponding to a basis $(\mathbf{B}_k)_{k=1}^{\dim(\mathfrak{g})}$ of DLA $\mathfrak{g}$. Each $\mathbf{B}_k$ is expressed as a
    linear combination of $\mathcal{O}(\text{poly}(n))$ Pauli strings. Snapshot inversion is
    being performed for a VQC model that utilizes a trainable portion of
    $\mathbf{U}(\boldsymbol{\theta})$
    with DLA $\mathfrak{g}$ and separable encoding Eq. (31) with qubit partition $\mathcal{P}$.
Index
    $j \in [d]$, $\epsilon < 1$
Ensure: An $\epsilon$ estimate of the $j$th component $x_j$ of the data input $\mathbf{x} \in \mathbb{R}^d$ up
    to periodicity
        1. Find a $\rho_J$ for $J \in \mathcal{P}$ that depends on $x_j$. Let $R$ denote the number of
           Pauli rotations in the circuit for preparing $\rho_J$ that involve $x_j$.
        2. For each $k \in [\dim(g)]$, compute $\text{Tr}(\mathbf{B}_k\rho_J(\mathbf{x}))$ and $\text{Tr}(\mathbf{B}_\text{k}\rho_{\text{J}^c}(\mathbf{x}))$.
        3. Determine the set $\mathcal{S}_J = \{k : \text{Tr}(\mathbf{B}_\text{k}\rho_\text{J}(\mathbf{x})) \neq 0$ &
           $\text{Tr}(\mathbf{B}_\text{k}\rho_{\text{J}^c}(\mathbf{x})) = 0\}$, $J^c := [n] - J$.
    if $\mathcal{S}_J < \dim(\mathbf{x}_J)$ then
        return FAILURE
    else
        1. For each $k \in \mathcal{S}_J$ evaluate $\text{Tr}(\mathbf{B}_k\rho_J(\mathbf{x}))$ at $M = 2R^{\dim(\mathsf{x}_J)} + 1$
points, $\mathbf{x}_r \in \{\frac{2\pi r}{2R+1} : r = -R, \ldots R\}^{\dim(\mathsf{x}_J)}$
        2. For each $k$, solve a linear system

            $$\text{Tr}(\mathbf{B}_k\rho_J(\mathbf{x}_r)) = \alpha_0 + \sum_{r\in[R]^{\dim(\mathsf{x}_J)}} \alpha_r e^{i\mathbf{r}\cdot\mathbf{x}_r}$$

        for $\alpha$'s.
        3. Consider the polynomial system:

            $$[\mathbf{e}_{\text{snap}}]_k = \text{Re}\left[\alpha_0 + \sum_{r\in[R]^{\dim(\mathsf{x}_J)}} \alpha_\mathbf{r} \prod_{j=1}^{\dim(\mathsf{x}_J)} (T_{r_j}(u_j) + iv_j U_{r_j-1}(u_j))\right], \tag{32}$$

with $k \in \mathcal{S}_J$,

            $$u_j^2 + v_j^2 = 1, j \in J, \tag{33}$$

 where $u_j = \cos(x_j)$, $v_k = \sin(x_j)$ and $T_r$, $U_r$ relate to Chebyshev
polynomials.
        4. Apply Buchberger's algorithm to obtain a Gröbner basis for the
system.
        5. Back substitution and univariable root-finding algorithm[44] (e.g.,
Jenkins-Traub[45]) to obtain $\tilde{\mathbf{x}}_J$.
        6. return $\tilde{\mathbf{x}}_j$
    end if
```

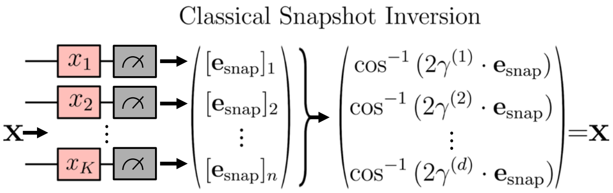

Classical Snapshot Inversion
$x_1$
$x_2$
$x_K$
$\mathbf{X}$
$[\mathbf{e}_{\text{snap}}]_1$
$[\mathbf{e}_{\text{snap}}]_2$
$[\mathbf{e}_{\text{snap}}]_n$
$\cos^{-1}(2\gamma^{(1)} \cdot \mathbf{e}_{\text{snap}})$
$\cos^{-1}(2\gamma^{(2)} \cdot \mathbf{e}_{\text{snap}})$
$\cos^{-1}(2\gamma^{(d)} \cdot \mathbf{e}_{\text{snap}})$
$=\mathbf{X}$

**Fig. 3 | Product map encoding and inversion.** A product map encoding, whereby each input variable $x_j$ is encoded into an individual qubit, and the snapshot used by the model corresponds to single-qubit measurements of the DLA basis elements. In this setting, the snapshot is trivial to invert and find the original data using the relation $x_j = \cos^{-1}\left(2\boldsymbol{\gamma}^{(j)} \cdot \mathbf{e}_{\text{snap}}\right)$.

**Theorem 4**. Suppose that the feature encoding state $\rho(\mathbf{x})$ is a multipartite state, specifically, there exists a partition $\mathcal{P}$ of qubits $[n]$ such that

$$\rho(\mathbf{x}) = \bigotimes_{J\in\mathcal{P}} \rho_J(\mathbf{x}),$$

where we define $\mathsf{x}_J \subseteq \mathbf{x}$ to be components of $\mathbf{x}$ on which $\rho_J$ depends. In addition, we have as input an $\mathcal{O}(\text{poly}(n))$-dimensional snapshot vector $\mathbf{e}_{\text{snap}}$ with respect to a known basis $\mathbf{B}_k$ for the DLA of the VQC.

Suppose that for $\rho_J(\mathbf{x})$ the following conditions are satisfied:

- $\dim(\mathbf{x}_\text{J}) = \mathcal{O}(1)$,
- each $x_k$ is encoded at most $R = \mathcal{O}(\text{poly}(n))$ times in, potentially multi-qubit, Pauli rotations.
- and the set $\mathcal{S}_J = \{k : \text{Tr}(\mathbf{B}_\text{k}\rho_\text{J}(\mathbf{x}))\neq 0$ & $\text{Tr}(\mathbf{B}_\text{k}\rho_{\text{J}^c}(\mathbf{x})) = 0\}$ has cardinality at least $\dim(\mathsf{x}_\text{J})$, where $J^c := [n] - J$.

Then the model admits quantum-assisted snapshot inversion for recovering $\mathsf{x}_J$. Furthermore, a classical snapshot inversion can be performed if $\forall\, k$, $\text{Tr}(\mathbf{B}_k\rho_J(\mathbf{x}))$ can be evaluated classically for all $\mathbf{x}$. Overall, ignoring error in estimating $\text{Tr}(\mathbf{B}_k\rho_J(\mathbf{x}))$, with the chosen parameters, this leads to a $\mathcal{O}(\text{poly}(n, \log(1/\epsilon)))$ algorithm.

**Proof**. We provide the proof in the methods section.

In the case a circuit has an encoding structure that leads to a separable state, we have indicated conditions that guarantee snapshot inversion can be performed. If the model is also snapshot recoverable, by having a polynomially sized DLA, then this means the initial data input can be fully recovered from the gradients, and hence the attack constitutes a strong privacy breach.

**Snapshot Inversion for Generic Encodings**. In the general case, but still $\dim(\mathfrak{g}) = \mathcal{O}(\text{poly}(n))$, where it is unclear how to make efficient use of our knowledge of the circuit, we attempt to find an $\mathbf{x}$ via black-box optimization methods that produces the desired snapshot signature. More specifically, suppose, for simplicity we restrict our search to $[-1, 1]^d$. We start with an initial guess for the input parameters, denoted as $\mathbf{x}'$, and use these to calculate expected snapshot values $\text{Tr}[\mathbf{B}_\text{k}\rho(\mathbf{x}')]$. A cost function can then be calculated that compares this to the true snapshot, denoted $\mathbf{e}_{\text{snap}}$. As an example, one can use the mean squared error as the cost function,

$$\begin{aligned} f(\mathbf{x}') &= \| \mathbf{e}_{\text{snap}} - (\text{Tr}[\mathbf{B}_\text{k}\rho(\mathbf{x}')])_{k=1}^{\dim(\mathfrak{g})} \|_2^2 \\ &= \sum_{k\in[\dim(\mathfrak{g})]} \left([\mathbf{e}_{\text{snap}}]_k - \text{Tr}[\mathbf{B}_\text{k}\rho(\mathbf{x}')]\right)^2. \end{aligned} \tag{34}$$

The goal will be to solve the optimization problem $\min_{\mathbf{x}'\in[-1,1]^d} f(\mathbf{x}')$. For general encoding maps, it appears that we need to treat this as a black-box optimization problem, where we evaluate the complexity in terms of the evaluations of $f$ or, potentially, its gradient. However, in our setting, it is unclear what is the significance of finding approximate local minimum, and thus it seems for privacy breakage, we must resort to an exhaustive grid search. For completeness, we still state results on first-order methods that can produce approximate local minima.

We start by reviewing some of the well-known results for black-box optimization. We recall Lipschitz continuity by,

**Definition 10**. ($L$-Lipschitz Continuous Function). A function $f:\mathbb{R}^d\to\mathbb{R}$ is said be $L$-Lipschitz continuous if there exists a real positive constant $L>0$ for which,

$$|f(\mathbf{x})-f(\mathbf{y})|\le L\parallel\mathbf{x}-\mathbf{y}\|_2.$$

If we consider the quantum circuit as a black-box $L$-Lipschitz function and $\mathbf{x}'$ in some convex, compact set with diameter $P$ (e.g., $[-1,1]^d$ with diameter $2\sqrt{d}$). One can roughly upper bound $L$ by the highest frequency component of the multidimensional trig series for $f$, which can be an exponential in $n$ quantity. In this case, the amount of function evaluations that would be required to find $\mathbf{x}'$ such that $\|\mathbf{x}-\mathbf{x}'\|_2\le\epsilon$ would scale as

$$\mathcal{O}\left(P\left(\frac{L}{\epsilon}\right)^d\right), \tag{35}$$

which is the complexity of grid search[46]. Thus if for constant $L$ this is a computationally daunting task, i.e., exponential in $d=\Theta(n)$.

As mentioned earlier, it is possible to resort to first-order methods to obtain an effectively dimension-independent algorithm for finding an approximate local minimum. We recall the definition of $\beta$-smoothness as,

**Definition 11**. ($\beta$-Smooth Function). A differentiable function $f:\mathbb{R}^d\to\mathbb{R}$ is said be be $\beta$-smooth if there exists a real positive constant $\beta>0$ for which

$$\|\nabla f(x)-\nabla f(y)\|_2\le\beta\|x-y\|_2.$$

If we have access to gradients of the cost function with respect to each parameter, then using perturbed gradient descent[47] would roughly require

$$\tilde{\mathcal{O}}\left(\frac{PL\beta}{\epsilon^2}\right), \tag{36}$$

function and gradient evaluations for an $L$-Lipschitz function that is $\beta$-smooth to find an approximate local min. With regards to first-order optimization, computing the gradient of $f$ can be expressed in terms of computing certain expectation values of $\rho$, either via finite-difference approximation or the parameter-shift rule for certain gate sets[48].

Regardless of whether recovering an approximate local min reveals any useful information about $\mathbf{x}$, up to periodicity, it is still possible to make such a task challenging for an adversary. In general, the encoding circuit will generate expectation values with trigonometric terms. To demonstrate, we can consider a univariate case of a single trigonomial $f(x)=\sin(\omega x)$, with frequency $\omega$. This function will be $\omega$-Lipschitz continuous with $\omega^2$-Lipchitz continuous gradient. Hence, when considering the scaling of gradient-based approach in Eq (36) we see that the frequency of the trigonometric terms will directly impact the ability to find a solution. Hence, if selecting a frequency that scales exponentially $\omega=\mathcal{O}(\exp(n))$, then snapshot inversion appears to be exponentially difficult with this technique.

Importantly, if the feature map includes high frequency terms, for example the Fourier Tower map of[30], then $\beta$ and $L$ can be $\mathcal{O}(\exp(n))$. However, as noted in the snapshot inversion for local encoding part of the results section it is possible to make use of the circuit structure to obtain more efficient attacks. In addition, a poor local minimum may not leak any information about $\mathbf{x}$.

**Direct input recovery**. Note that it also may be possible to completely skip the snapshot recovery procedure and instead variationally adjust $\mathbf{x}'$ so that the measured gradients of the quantum circuit $C_j'$, match the known gradients $C_j$ with respect to the actual input data. This approach requires consideration of the same scaling characteristics explained in Eq. (36), particularly focusing on identifying the highest frequency component in the gradient spectrum. If the highest frequency term in the gradient $C_j$ scales exponentially, $\omega=\mathcal{O}(\exp(n))$, then even gradient descent based methods are not expected to find an approximate local min in polynomial time.

Further privacy insights can be gained from Eq. (21), where a direct relationship between the gradients and the expectation value snapshot is shown, which in general can be written as

$$C_j(\mathbf{x})=\chi_t^{(j)}\cdot\mathbf{e}_{\mathrm{snap}}(\mathbf{x}). \tag{37}$$

This indicates that the highest frequency terms of any $\mathbf{e}_{\mathrm{snap}}$ component will also correspond to the highest frequency terms in $C_j(\mathbf{x})$, as long as its respective coefficient is non-zero $\chi_t^{(j)}\neq0$.

This underscores scenarios where direct input recovery may prove more challenging compared to snapshot inversion, particularly in a VQC model. Consider a subset $\tilde{\mathbf{e}}_{\mathrm{snap}}\subseteq\mathbf{e}_{\mathrm{snap}}$ where each component has the highest frequency that scales polynomially with $n$. If there are sufficiently many values in $\tilde{\mathbf{e}}_{\mathrm{snap}}$ then recovering the approximate local min to Eq. (34) may be feasible for these components. However, for gradient terms $C_j(\mathbf{x})$ that depend on all values of $\mathbf{e}_{\mathrm{snap}}$, including terms outside of $\tilde{\mathbf{e}}_{\mathrm{snap}}$ that exhibit exponential frequency scaling, then gradient descent methods may take exponentially long when attempting direct input inversion, even if recovering approximate local minima to the snapshot inversion task can be performed in polynomial time.

Investigations into direct input recovery have been covered in previous work[30] where the findings concluded that the gradients generated by $C_j(\mathbf{x})$ would form a loss landscape dependent on the highest frequency $\omega$ generated by the encoding circuit, indicating that exponentially scaling frequencies led to models that take exponential time to recover the input using quantum-assisted direct input recovery. The Fourier tower map encoding circuit used in ref. 30 was designed such that $\omega$ scales exponentially to provide privacy; this was done by using $m$ qubits in a sub-register per data input $x_j$, with the single qubit rotation gates parameterized by an exponentially scaling amount. The encoding can be defined as

$$\bigotimes_{j=1}^{d}\left(\bigotimes_{l=1}^{m}R_X(5^{l-1}x_j)\right). \tag{38}$$

Hence, the gradient contained exponentially scaling highest frequency terms, leading to a model where gradient descent techniques took exponential time. However, if considering the expectation value of the first qubit in a sub-register of this model, we note this corresponds to a frequency $\omega=1$, and hence the respective expectation value for the first qubit would be snapshot invertible. However, in the case of ref. 30, the DLA was exponentially large, meaning the model was not snapshot recoverable, hence these snapshots could not be found to then be invertible. Hence, from our new insights, we can conclude that the privacy demonstrated in ref. 30 was dependent on having an exponential DLA dimension. However, an exponentially large DLA also led to an untrainable model, limiting the real-world applicability of this previous work. Lastly, recall that Algorithm 3 in the case of poly DLA and slow Pauli expansion is a completely classical snapshot inversion attack for the Fourier tower map. Further, highlighting how snapshot inversion can be easier than direct inversion.

We show that both direct input recovery and snapshot inversion are dependent on frequencies $\omega$ generated by the encoding circuit, highlighting that this is a key consideration when constructing VQC models. The introduction of high-frequency components can be used to slow down methods that obtain approximate local minimum to Eq. (34). However, for true privacy breakage, it appears that, in general, we still need to resort to grid search, which becomes exponentially hard with dimension regardless of

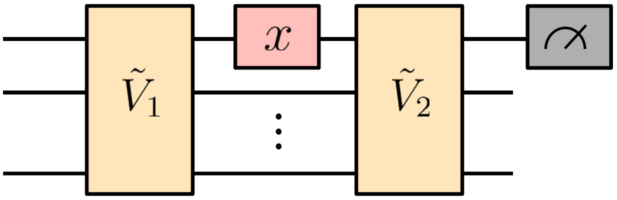


**Fig. 4 | Single qubit rotation encoding circuit.** Encoding circuit diagram showing a single qubit $R_X$ rotation gate parameterized by the univariate parameter $x$, but with arbitrary $2^n$ dimensional unitaries applied before and after the $x$ parameterized gate. Despite being hard to simulate analytically, the expectation value $e_{\text{in}}$ varies as a simple sinusoidal function in $x$, regardless of the total number of qubits $n$.

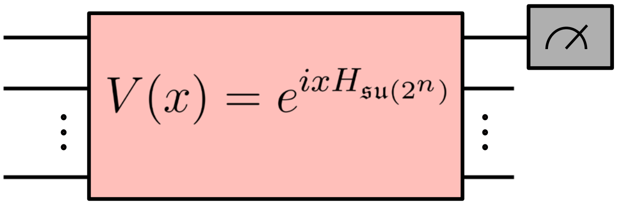


**Fig. 5 | Generic unitary encoding circuit.** Encoding circuit diagram showing a SU($2^n$) gate parameterized by a univariate parameter $x$.

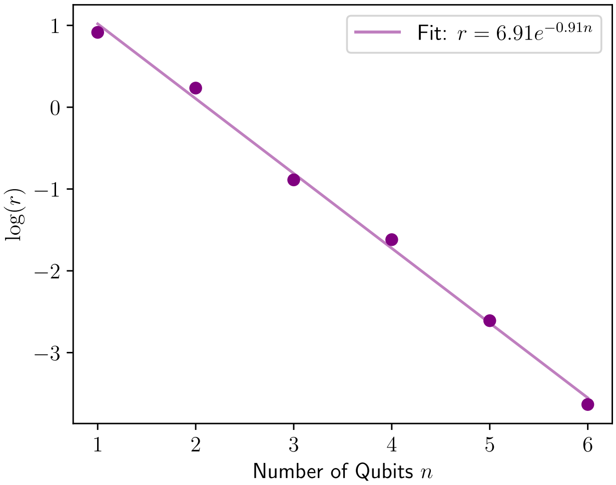


**Fig. 6** | Scaling of average minimum distance between stationary points. Plot showing the relationship between the average minimum distance $r$ between stationary points of the expectation value $Z \otimes \mathbb{I}^{\otimes n-1}$ as a function of a univariate $x$ input. The encoding circuit considered is a parameterized SU($2^n$) gate which is parameterized by a univariate input $x$ as $U = e^{iHx}$, where $H$ is a randomly generated Hermitian matrix. The average was taken over ten repeated experiments where $H$ was regenerated each time.

high-frequency terms. However, for problems with a small amount of input, introducing high-frequency terms can be used to also make grid search harder. The idea of introducing large frequencies is a proxy for the more general condition that our results hint at *for privacy*, which is that *the feature map* $\rho(\mathbf{x}')$ *should be untrainable in terms of varying* $\mathbf{x}'$.

Notably, cases exist where the same model can have an exponential frequency gradient, but can still contain a certain number of expectation snapshot values with polynomial scaling frequencies. Hence, it is also important to note that merely showing that a model is not directly input recoverable does not guarantee privacy, as one needs to also consider that if the model is snapshot recoverable, and that these snapshots may be invertible if sufficient polynomial scaling frequency terms can be recovered. This duality highlights the complexity of ensuring privacy in quantum computing models and stresses the need for a comprehensive analysis of the frequency spectrum in both model construction and evaluation of privacy safeguards.

**Expectation value landscape numerical results**. In this section, we provide a numerical investigation of the impact of high-frequency components in the encoding circuit on the landscape of Eq. (34) for snapshot inversion. The idea is to present examples that move beyond the Fourier tower map. We present two cases of encodings that would generally be difficult to simulate classically. By plotting a given expectation value against a univariate $x$, we can numerically investigate the frequencies produced by both models.

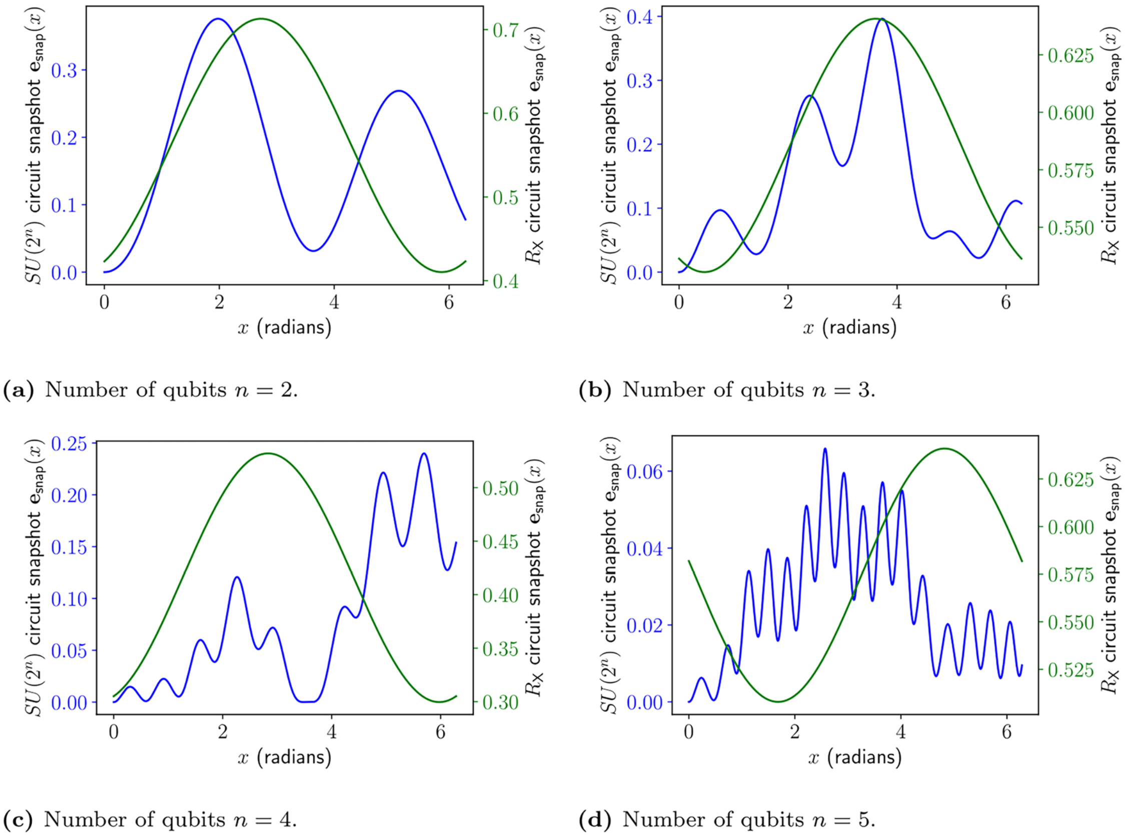
In Fig. 4 we demonstrate a circuit in which $x$ parameterizes a single $R_X$ rotation gate, but on either side of this is an unknown arbitrary unitary matrix acting on $n$ qubits. This would be classically hard to simulate due to the arbitrary unitary matrices; however, the result effectively corresponds to taking measurements on an unknown basis, and using only a few samples of $x$ it is possible to recreate the graph as a single frequency sinusoidal relationship. This results in the distance between the stationary points being $r = \pi$ for any value of $n$. This corresponds to a frequency $\omega = \frac{r}{\pi} = 1$, regardless of the value of $n$. This circuit, therefore, exhibits constant frequency scaling independent of $n$ and hence could be easy for gradient-based methods to recover an approximate local min.

We briefly give an example of a type of circuit that can generate high-frequency expectation values. Figure 5 demonstrates a circuit where $x$ parameterizes an SU($2^n$) gate. The result when measuring the same expectation value corresponds to the highest frequency term that is exponentially increasing. This is shown in the plot in Fig. 6 in which the distance between stationary points $r$ shrinks exponentially as the number of qubits increases for the SU($2^n$) parameterized model, which roughly corresponds to an exponentially increasing highest frequency term. A comparison between the expectation value landscape of the two different encoding architectures, is shown in Fig. 7, demonstrating that the single rotation gate parameterization, as shown in Fig. 4, produces a sinusoidal single-frequency distribution, even as the number of qubits is increased; while the SU($2^n$) gate parameterization, shown in Fig. 5, contains exponentially increasing frequency terms. A visual representation for the multivariate case is also demonstrated in Fig. 8 which shows the expectation value landscape when two input parameters are adjusted, for a model comprised of two different SU($2^n$) parameterized gates parameterized by the variables $x_1$ and $x_2$ respectively, demonstrating that as more qubits are used, the frequencies of the model increase and hence so does the difficulty of finding a solution using gradient descent techniques.

The two example circuits demonstrate encoding circuits that are hard to simulate, and hence, no analytical expression for the expectation values can be easily found. These models do not admit classical snapshot inversion; however, by sampling expectation values, it may be possible to variationally perform quantum-assisted snapshot inversion. Whether numerical snapshot inversion can be performed efficiently will likely be affected by the highest frequency $\omega$ inherent in the encoding, which will itself depend on the architecture of the encoding circuit. This suggests that designing encoding circuits such that they contain high-frequency components is beneficial in high-privacy designs. We have shown that SU($2^n$) parameterized gates can produce high-frequency terms, whereas single-qubit encoding gates will be severely limited in the frequencies they produce.

## Discussion

In this research, we conduct a detailed exploration of the privacy safeguards inherent in VQC models regarding the recovery of original input data from observed gradient information. Our primary objective was to develop a systematic framework capable of assessing the vulnerability of these quantum models to a general class of inversion attacks, specifically through introducing the snapshot recovery and snapshot inversion attack techniques, which primarily depend on the variational and encoding architectures, respectively.



**(a)** Number of qubits $n = 2$.

**(b)** Number of qubits $n = 3$.

**(c)** Number of qubits $n = 4$.

**(d)** Number of qubits $n = 5$.

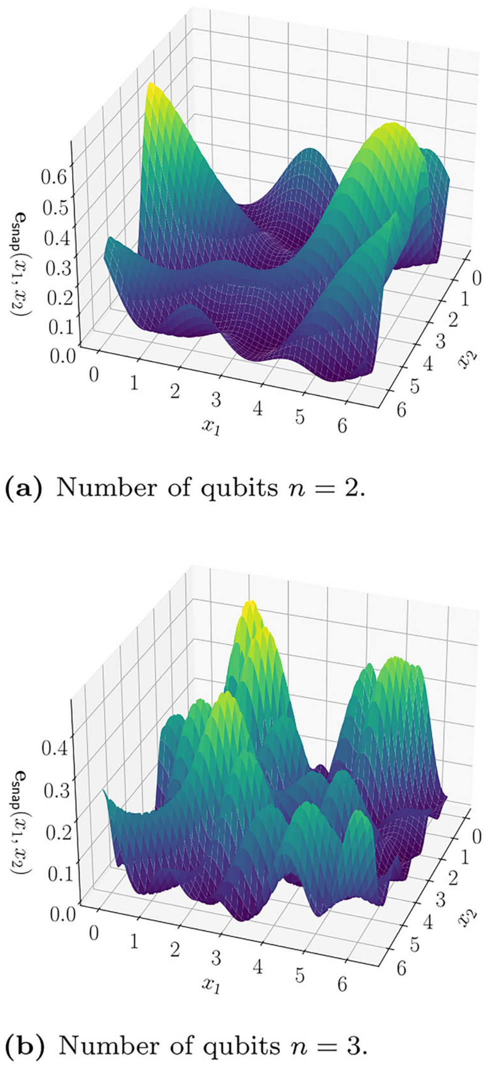
**Fig. 7 | Snapshot landscape visualization with one dimensional input.** Comparison of how the expectation value of the measurement of $Z_1$ varies with $x$ for both the model parameterized using a single $R_X$ rotation gate as detailed in Fig. 4 and the model parameterized using an $SU(2^n)$ gate as detailed in Fig. 5, for varying amounts of qubits. **a** Landscape with two qubits in the encoding. **b** Landscape with three qubits in the encoding. **c** Landscape with four qubits in the encoding. **d** Landscape with five qubits in the encoding.

Our analysis began by establishing the feasibility of recovering snapshot expectation values from the model gradients under the LASA assumption. We demonstrated that such recovery is viable when the Lie algebra dimension of the variational circuit exhibits polynomial scaling in the number of qubits. This result underscores the importance of algebraic structure in determining the potential for privacy breaches in quantum computational models. Furthermore, due to the fact that a polynomial scaling DLA dimension is commonly required for models to be trainable, our results suggest that a trade-off may exist between privacy and the trainability of VQC models. Assuming one insists on a polynomial-sized DLA, our framework suggests that a weak privacy breach will always be possible for the type of VQC model studied. To ensure the privacy of the model overall, one cannot rely on the variational circuit and needs to instead focus more on the encoding architecture and ensuring snapshot inversion cannot be performed. If snapshot inversion is not possible, then at least strong privacy breaches can be prevented.

We then explored snapshot inversion, where the task is to find the original input from the snapshot expectation values, effectively inverting the encoding procedure. Studying widely used encoding ansatz, such as the local multiqubit Pauli encoding, we found that under the conditions that a fixed subset of the data paramaterizes a constituent state which has sufficient overlap with the DLA, and the number of gates used to encode each dimension of the input $\mathbf{x}$ was polynomial, that snapshot inversion was possible in $\mathcal{O}(\text{poly}\,(n, \log(1/\epsilon))$ time. This shows that a potentially wide range of encoding circuits are vulnerable to strong privacy breaches and brings their usage in privacy-focused models into question. For the most general encoding, which we approached as a black-box optimization problem, we demonstrated that using perturbed gradient descent to find a solution is constrained by the frequency terms within the expectation value Fourier spectrum. In general for exactly finding $\mathbf{x}$ it appears that a grid search would be required. Although we cannot provide strictly sufficient conditions due to the possibility of unfavorable local minima with perturbed gradients, we note that gradient descent for snapshot inversion may, in some cases, be easier to perform than for direct input data recovery from the gradients. This simplification arises because gradients can inherit the highest frequency term from the snapshots, potentially leading to scenarios where the gradient term contains exponentially large frequencies. However, there may still be sufficient polynomial frequency snapshots to permit snapshot inversion. This shifts the focus in attack models away from direct input recovery from gradients, a common approach in classical privacy analysis, towards performing snapshot inversion as detailed in this study, as a potentially more efficient attack method.

The dual investigation allowed us to construct a robust evaluative framework that not only facilitates the assessment of existing VQC models for privacy vulnerabilities but also aids in the conceptualization and development of new models where privacy is a critical concern. Our reevaluation of previous studies, such as those cited in ref. 30, through the lens of our new framework, reveals that the privacy mechanisms employed, namely the utilization of high-frequency components and exponentially large DLA, effectively prevent input data recovery via a lack of snapshot recoverability, but at the same time contribute to an untrainable model of limited practical use.

In conclusion, we offer a methodological approach for classifying and analyzing the privacy features of VQC models, presenting conditions for weak and strong privacy breaches for a broad spectrum of possible VQC architectures. Our findings not only enhance the understanding of quantum privacy mechanisms but also offer strategic guidelines for the design of quantum circuits that prioritize security while at the same time maintaining



**(a)** Number of qubits $n = 2$.

**(b)** Number of qubits $n = 3$.

**Fig. 8 | Snapshot landscape visualization with two dimensional input.** Comparison of how the expectation value of the measurement of $Z_1$ varies with $x$ for both the model parameterized using a single $R_X$ rotation gate as detailed in Fig. 4 and the model parameterized using an $SU(2^n)$ gate as detailed in Fig. 5, for varying amounts of qubits. **a** Landscape with two qubits in the encoding. **b** Landscape with three qubits in the encoding.

trainability. Looking ahead, this research paves the way for more robust quantum machine learning model designs, where privacy and functionality are balanced. This knowledge offers the potential to deliver effective machine learning models that simultaneously demonstrate a privacy advantage over conventional classical methods.

## Methods

We utilize this section to draw the connections between the two key properties of VQC: *trainability*, i.e., the lack of barren plateaus[36,49], and the ability to retain privacy of input. Building upon this connection, we discuss the prospects and future of achieving robust privacy guarantees with VQC models.

### Connections between trainability and privacy in VQC

Solely requiring a machine learning model to be private is not sufficient to deploy it for a practical use case of distributed learning, such as federated learning. A key requirement in this collaborative learning scenario is also to ensure that the model remains trainable. A plethora of works have gone into exactly characterizing the trainability of VQC models by analyzing the presence of barren plateaus in the VQC model, starting from the work of McClean et al.[33] and culminating in the works of Fontana and Ragone[36,49]. Especially, the work of Fontana[36] provides an exact expression of the variance of the gradient of the model when the VQC is constrained to be in the LASA case, the details of which we also provide in Supplementary file Sec IV for completeness. A key insight into these works suggests that LASA models, with exponentially-sized DLA, may lead to the presence of barren plateaus, drastically deteriorating the trainability of such models[36,49].

Within our privacy framework centered around snapshot recoverability, we also show via Theorem 2 that LASA models with an exponential size DLA are not classically snapshot recoverable, although this may lead to untrainable models. We can therefore conclude that a possible condition for protection against classical input recovery using gradients in a VQC model is to choose an ansatz that exhibits an exponentially large dynamical Lie algebra dimension, as this would render snapshot recovery difficult. Through our framework, we can see that previous works[30] effectively relied on this property to ensure privacy. Combining the concept of trainability leads to the following corollary on the privacy of VQC models:

**Corollary 5**. Any trainable VQC on $n$ qubits that satisfies the LASA condition in Def 5, fulfills the slow Pauli expansion condition as highlighted in Def 9, and has a DLA $\mathfrak{g}$ whose dimension scales as $\mathcal{O}(\text{ poly }(\text{n}))$, would admit snapshot recoverability with complexity $\mathcal{O}(\text{ poly }(\text{n}))$.

Hence, we can conclude that, at least in the LASA case of VQC, the privacy of the model is linked to the DLA dimension, and furthermore, that there is a direct tradeoff between privacy and trainability of the model. As exponentially sized DLA models are expected to be untrainable in the LASA case, this means that for realistic applications, it does not seem feasible to rely on quantum privacy derived from an exponential DLA, precluding snapshot recoverability in the model. This suggests that any privacy enhancement from quantum VQCs should not derive from the variational part of the circuit for LASA-type models that are intended to be trainable. In other words, we expect the majority of trainable VQC models to be vulnerable to weak privacy breaches. The privacy of variational models beyond the LASA case becomes linked to a larger question within the field, notably, whether there exist quantum variational models that are not classically simulatable and do not have barren plateaus[50].

It is also worth noting that if one attempted to create a model that is not snapshot recoverable by ensuring that $D < \dim(\mathfrak{g})$, and hence an underparameterized system of equations, it would effectively lead to an underparameterized model. A model is underparmeterized when there are not enough variational parameters to fully explore the space generated by the DLA of the ansatz, which is a property that may not be desirable for machine learning models[51].

### Future direction of VQC quantum privacy

Due to the above argument suggesting that achieving privacy via an exponentially large DLA may cause trainability issues in the underlying model, it appears that future improvements in privacy using VQC may primarily focus on preventing the snapshot inversion step, as we highlight in the input recoverability definitions part of the results section. This promotes a focus on the encoding circuit architectures of the VQC in order to prevent the model from admitting snapshot inversion to facilitate input recovery.

We have explicitly shown the necessary condition required to achieve privacy from purely classical attacks. If it is not possible to classically simulate the expectation values of the quantum encoded state with respect to the DLA basis elements of the variational circuit, then it will not be possible to attempt classical analytical or numerical inversion attacks. Any VQC designed where these expectation values cannot be simulated will, therefore, be protected against any purely classical snapshot inversion attempts. This condition can therefore prevent strong privacy breaches, as long as the attacking agent only has access to a classical device.

In the case where the attacker can simulate expectation values of the DLA basis or has access to a quantum device to obtain the expectation values, then numerical classical snapshot inversion or numerical quantum-assisted snapshot inversion can be attempted, respectively. We have shown that in this case, an important factor in preventing these techniques is that the expectation values have exponentially scaling frequency terms, resulting in the attacks requiring solving a system of high-degree polynomial equations. The implication of this is that to achieve a useful privacy benefit in VQC, it may require that the encoding circuit is constructed in such a way that the expectation values of the DLA basis elements of the variational circuit contain frequency terms that scale exponentially. Notably, we find that having high frequency terms in the gradients, as suggested in the encoding circuit of ref. 30, does not necessarily protect against numerical snapshot inversion attacks. This is because the gradients inherit the highest frequency term from all expectation values, but there may be a sufficient number of polynomial frequency expectation values to perform snapshot inversion, even if direct input inversion is not possible.

Unlike the variational case, where a connection between DLA dimension and trainability has been established, the effect that privacy-enhancing quantum encodings would have on the trainability of a model is less clear. If the majority of expectation values used in the model contain exponentially large frequencies, then this potentially restricts the model to certain datasets. In classical machine learning, there have been positive results using trigonometric feature maps to classify high-frequency data in low dimensions[35]. It remains a question for future research, the types of data that may be trained appropriately using the privacy-preserving high-frequency feature maps proposed. If models of this form are indeed limited in number, then the prospects for achieving input privacy from VQC models appear to be limited. More generally, the prospect for quantum privacy rests on feature maps that are *untrainable* with regard to adjusting $\mathbf{x}'$ to recover expectation values $\mathbf{e}_{\text{snap}}$, while at the same time remaining useful feature maps with respect to the underlying dataset and overall model.

## Proof of Theorem 3

**Proof**. Steps 1–5 in Algorithm 3 can be performed in $\mathcal{O}(\text{ poly }(\text{n}))$ classical time due to the polynomial DLA and slow Pauli expansion conditions. The purpose of step 5 is to compute the angles between the linear subspaces $\mathfrak{g}$ and $\text{span}_{\mathbb{R}}\{i\mathsf{Z}_j, i\mathsf{Y}_j\}$. This is to identify if there is any intersection, i.e., if $\exists\, \alpha$, $\beta$ such that $\alpha i\mathsf{Z}_j + \beta i\mathsf{Y}_j \in \mathfrak{g}$, which is identified by singular values equal to 1. The algorithm cannot proceed if the intersection is trivially empty, as the snapshot vector does not provide the required measurement to obtain $x_j$ efficiently with this scheme. From now on, we suppose that such an element has been found.

We can, without loss of generality, just focus on the one-qubit reduced density matrix for $x_j$. In this case, using Bloch sphere representation:

$$\rho_j(x_j) = R_{\mathsf{X}}(x_j)|0\rangle\langle 0|R_{\mathsf{X}}(-x_j) = \frac{\mathbb{I} - \sin(x_j)\mathsf{Y} + \cos(x_j)\mathsf{Z}}{2}, \tag{39}$$

such that

$$\begin{aligned} \text{Tr}\,([\alpha\mathsf{Z}_{\text{j}} + \beta\mathsf{Y}_{\text{j}}]\rho_{\text{j}}(\text{x}_{\text{j}})) &= \tfrac{\alpha}{2}\cos(\text{x}_{\text{j}}) - \tfrac{\beta}{2}\sin(\text{x}_{\text{j}}) \\ &= \tfrac{\text{sign}\,(\alpha)}{2}\sqrt{\alpha^2 + \beta^2}\cos(x_j + \tan^{-1}(\beta/\alpha)). \end{aligned} \tag{40}$$

However, by assumption, $\gamma_k \in \mathbb{R}$ such that

$$\begin{aligned} \alpha i\mathsf{Z}_j + \beta i\mathsf{Y}_j &= \sum_{k=1}^{\dim(\mathfrak{g})} \gamma_i \mathbf{B}_k \\ \Rightarrow\quad \text{Tr}\,([\alpha\mathsf{Z}_{\text{j}} + \beta\mathsf{Y}_{\text{j}}]\rho_{\text{j}}(\text{x}_{\text{j}})) &= \sum_{\text{k}=1}^{\dim(\mathfrak{g})} \gamma_{\text{k}}[\mathbf{e}_{\text{snap}}]_{\text{k}}. \end{aligned} \tag{41}$$

So to recover $x_j$, we only need to solve:

$$\sum_{k=1}^{\dim(\mathfrak{g})} \gamma_k[\mathbf{e}_{\text{snap}}]_k = \frac{\text{sign}\,(\alpha)}{2}\sqrt{\alpha^2 + \beta^2}\cos(x_j + \tan^{-1}(\beta/\alpha)), \tag{42}$$

which, after rearranging, allows the recovery of

$$\begin{aligned} x_j \quad = \quad & \cos^{-1}\left[\tfrac{2}{\text{sign}\,(\alpha)\sqrt{\alpha^2+\beta^2}} \sum_{k=1}^{\dim(\mathfrak{g})} \gamma_k[\mathbf{e}_{\text{snap}}]_k\right] \\ & -\tan^{-1}(\beta/\alpha), \end{aligned} \tag{43}$$

up to periodicity. By the polynomial DLA and slow Pauli expansion conditions (i.e., all DLA basis elements are expressed as linear combinations of Paulis), we can compute $\gamma_k$ in $\mathcal{O}(\text{ poly }(\text{n})\log(1/\epsilon))$ time.

## Proof of Theorem 4

**Proof**. Given that each $x_k$ is encoded with multiqubit Pauli rotations, i.-e., possible eigenvalues are 1 and $-1$, it is well known[48] that the following holds:

$$f_k(\mathsf{x}_J) = \text{ Tr}\,(\mathbf{B}_{\text{k}}\rho_{\text{J}}(\mathbf{x})) = \alpha_0 + \sum_{\text{r}\in[\text{R}]^{\dim(\mathsf{x}_{\text{J}})}} \alpha_{\text{r}} \text{e}^{\mathbf{ir}\cdot\mathbf{x}_{\text{r}}}, \forall\text{k} \in \mathcal{S}, \tag{44}$$

and $\text{Tr}(\mathbf{B}_k\rho_J(\mathbf{x}))$ is real. The set $\mathcal{S}_J$ is to ensure that we can isolate a subsystem where $\dim(\mathsf{x}_J)$ is constant.

To ensure that the number of terms is $\mathcal{O}(\text{ poly }(\text{n}))$ it suffices to restrict to $\dim(\mathsf{x}_{\text{J}}) = \mathcal{O}(\log(\text{n}))$, $\text{R} = \mathcal{O}(\log(\text{n}))$. The $\alpha$ coefficients can be computed by evaluating $\text{Tr}(\mathbf{B}_k\rho_J(\mathbf{x}))$ at $2R^{\dim(\mathsf{x}_J)} + 1 = \mathcal{O}(\text{ poly }(\text{n}))$ different points $\mathbf{x}'$. Depending on whether $\text{Tr}(\mathbf{B}_k\rho_J(\mathbf{x}))$ can be evaluated classically or quantumly implies whether this falls under classical or quantum-assisted snapshot inversion. This leads to a system of $\dim(\mathsf{x}_{\text{J}})$ equations in $\mathsf{x}_J$:

$$[\mathbf{e}_{\text{snap}}]_k = f_k(\mathsf{x}_J), k = 1, \ldots, \dim(\mathfrak{g}). \tag{45}$$

Using the Chebyshev polynomials $T_n$, $U_n$ of the first and second kind, respectively, we can express the system as a system of polynomial equations with additional constraints:

$$[\mathbf{e}_{\text{snap}}]_k = \text{Re}\left[\alpha_0 + \sum_{r\in[R]^{\dim(\mathsf{x}_J)}} \alpha_{\mathbf{r}} \prod_{j=1}^{\dim(\mathsf{x}_J)} (T_{r_j}(u_j) + iv_jU_{r_j-1}(u_j))\right], \tag{46}$$

with $k \in \mathcal{S}_J$,

$$u_j^2 + v_j^2 = 1, j \in J, \tag{47}$$

where $u_j = \cos(x_j)$, $v_k = \sin(x_j)$. In addition, we use the Chebyshev polynomials defined as $\cos(n\theta) = T_n(\cos(\theta))$ and $\sin(\theta)U_{n-1}(\cos(\theta)) = \sin(n\theta)$. By our assumption that the DLA is polynomial, we have $\mathcal{O}(\text{ poly }(\text{n}))$ equations in $2\dim(\mathsf{x}_J) = \mathcal{O}(\log\log(n))$ unknowns.

If all conditions until now are satisfied, we will have successfully written down a system of determined simultaneous equations. Considering bounds from computational geometry, we note that in the worst-case of Buchberger's algorithm[52] the degrees of a reduced Gröbner basis are bounded by

$$M = 2\left(\frac{\Delta^2}{2} + \Delta\right)^{2^{Q-2}}, \tag{48}$$

where $\Delta$ is the maximum degree of any polynomial and $Q$ is the number of unknown variables[53]. For a system of linear equations, it was shown that a worst-case degree bound grows double exponentially in the number of variables[54]. The maximum degree of any equation in Eq (46) is $\Delta = R^{\dim(\mathsf{x}_J)}$, and $Q = 2\dim(\mathsf{x}_J)$ so that

$$M = \mathcal{O}(R^{2\dim(\mathsf{x}_J)2^{\dim(\mathsf{x}_J)}}), \tag{49}$$

so for our chosen conditions the maximum degree is bounded by $M = \mathcal{O}(\text{ poly }(\text{n}))$.

Buchberger's algorithm provides a Gröbner basis in which back-substitution could be used to solve equations in one variable. Numerical methods for solving polynomials in one variable generally scale polynomially in the degree. For solving each univariate polynomial at each step of the back substitution, we can apply a polynomial root-finding method, such that Jenkins–Traub[45], which can achieve at least quadratic global convergence (converge from any initial point and at a rate that is at least $\log\log(1/\epsilon)$). This leads to an overall $\mathcal{O}(\,\mathrm{poly}\,(\mathrm{n}, \log(1/\epsilon))$ algorithm, ignoring the error in estimating $\mathrm{Tr}(\mathbf{B}_j\rho_J)$.

## Data availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

## Code availability

The code supporting the findings of this study is available from the corresponding author upon reasonable request.

Received: 4 August 2024; Accepted: 3 April 2025;
Published online: 19 May 2025

## References

1. Liu, T. et al. Efficient and secure federated learning for financial applications. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2303.08355 (2023).
2. Awosika, T., Shukla, R. M. & Pranggono, B. Transparency and Privacy: The Role of Explainable AI and Federated Learning in Financial Fraud Detection, Vol. 12, 64551–64560 https://doi.org/10.1109/ACCESS.2024.3394528 (IEEE Access, 2024).
3. Kaissis, G., Makowski, M. R., Rückert, D. & Braren, R. F. Secure, privacy-preserving and federated machine learning in medical imaging. *Nat. Mach. Intell.* **2**, 305 – 311 (2020).
4. Ahamed, S. et al. Investigating privacy-preserving machine learning for healthcare data sharing through federated learning. *Sci. Temper.* **14**, 1308–1315 (2023).
5. Aguiar Jr, E. C. & Traina, A. Security and privacy in machine learning for health systems: strategies and challenges. *Yearb. Med. Inform.* **32**, 269–281 (2023).
6. Park, C. et al. FedGeo: Privacy-preserving user next location prediction with federated learning. *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, Vol. 39, 1–10, https://doi.org/10.1145/3589132.3625582 (2023).
7. Albrecht, J. P. How the gdpr will change the world. *Eur. Data Prot. Law Rev.* **2**, 287–289 (2016).
8. Brauneck, A. et al. Federated machine learning in data-protection-compliant research. *Nat. Mach. Intell.* **5**, 2–4 (2023).
9. Tong, J. et al. Distributed learning for heterogeneous clinical data with application to integrating COVID-19 data across 230 sites. *npj Digital Med.* **5**, 76 (2022).
10. McMahan, H. B., Moore, E., Ramage, D., Hampson, S. & y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Vol. 54, 1273–1282 https://doi.org/10.48550/arXiv.1602.05629 (PMLR, 2017).
11. Zhu, L., Liu, Z. & Han, S. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, vol. 32 (Curran Associates, Inc., 2019). https://proceedings.neurips.cc/paper/2019/file/60a6c4002cc7b29142def8871531281a-Paper.pdf.
12. Huang, Y., Gupta, S., Song, Z., Li, K. & Arora, S. Evaluating gradient inversion attacks and defenses in federated learning. In (eds Beygelzimer, A., Dauphin, Y., Liang, P. & Vaughan, J. W.) *Advances in Neural Information Processing Systems*. https://openreview.net/forum?id=0CDKgyYaxC8 (2021).
13. Zhao, B., Mopuri, K. R. & Bilen, H. iDLG: improved deep leakage from gradients. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2001.02610 (2020).
14. Geiping, J., Bauermeister, H., Dröge, H. & Moeller, M. Inverting gradients—how easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6–12, 2020, virtual* (2020). https://proceedings.neurips.cc/paper/2020/hash/c4ede56bbd98819ae6112b20ac6bf145-Abstract.html.
15. Yin, H. et al. See through Gradients: Image Batch Recovery via GradInversion. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16332–16341 (2021).
16. Phong, L. T., Aono, Y., Hayashi, T., Wang, L. & Moriai, S. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Trans. Inf. Forensics Secur.* **13**, 1333–1345 (2018).
17. Eloul, S., Silavong, F., Kamthe, S., Georgiadis, A. & Moran, S. J. Enhancing privacy against inversion attacks in federated learning by using mixing gradients strategies. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2204.12495 (2022).
18. Huang, R., Tan, X. & Xu, Q. Quantum federated learning with decentralized data. *IEEE J. Sel. Top. Quantum Electron.* **28**, 1–10 (2022).
19. Qi, J., Zhang, X. & Tejedor, J. Optimizing quantum federated learning based on federated quantum natural gradient descent. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 1–5 (2023). https://api.semanticscholar.org/CorpusID:257505021.
20. Chehimi, M. & Saad, W. Quantum federated learning with quantum data. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 8617–8621 (2021). https://api.semanticscholar.org/CorpusID:235265617.
21. Li, C. et al. Blind quantum machine learning with quantum bipartite correlator. *Phys. Rev. Lett.* **133**, 120602 (2024).
22. Gurung, D., Pokhrel, S. R. & Li, G. Decentralized quantum federated learning for metaverse: analysis, design and implementation. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2306.11297 (2023).
23. Lusnig, L. et al. Hybrid quantum image classification and federated learning for hepatic steatosis diagnosis. *Diagnostics* **14**, 558 (2024).
24. Gilboa, D. & McClean, J. R. Exponential quantum communication advantage in distributed learning. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2310.07136 (2023).
25. Li, C., Kumar, N., Song, Z., Chakrabarti, S. & Pistoia, M. Privacy-preserving quantum federated learning via gradient hiding. *Quantum Sci. Technol.* **9**, 035028 (2024).
26. Koyasu, I., Raymond, R. & Imai, H. Distributed coordinate descent algorithm for variational quantum classification. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 1, 457–467 (IEEE, 2023).
27. Du, Y., Hsieh, M.-H., Liu, T., Tao, D. & Liu, N. Quantum noise protects quantum classifiers against adversaries. *Phys. Rev. Res.* **3**, 023153 (2021).
28. Gong, W., Yuan, D., Li, W. & Deng, D.-L. Enhancing quantum adversarial robustness by randomized encodings. *Phys. Rev. Res.* **6**, 023020 (2024).
29. Li, W., Lu, S. & Deng, D.-L. Quantum federated learning through blind quantum computing. *Sci. China Phys. Mech. Astron.* https://api.semanticscholar.org/CorpusID:237396275 (2021).
30. Kumar, N. et al. Expressive variational quantum circuits provide inherent privacy in federated learning. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2309.13002 (2023).
31. Chen, S. Y.-C. & Yoo, S. Federated quantum machine learning. *Entropy* https://www.mdpi.com/1099-4300/23/4/460 (2021).
32. Haah, J., Liu, Y. & Tan, X. Efficient approximate unitary designs from random Pauli rotations. *IEEE 65th Annual Symposium on Foundations*

*of Computer Science (FOCS)* https://doi.org/10.1109/FOCS61266.2024.00036 (2024).
33. McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R. & Neven, H. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.* **9**, 4812 (2018).
34. Anschuetz, E. & Kiani, B. Quantum variational algorithms are swamped with traps. *Nat. Commun.* **13**, 7760 (2022).
35. Tancik, M. et al. Fourier features let networks learn high frequency functions in low dimensional domains. NIPS'20: *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Vol. 632, 7537–7547 https://doi.org/10.5555/3495724.3496356 (2020).
36. Fontana, E. et al. Characterizing barren plateaus in quantum ansätze with the adjoint representation. *Nat. Commun.* **15**, 7171 (2024).
37. Somma, R., Ortiz, G., Barnum, H., Knill, E. & Viola, L. Nature and measure of entanglement in quantum phase transitions. *Phys. Rev. A* **70**, 042311 (2004).
38. Goh, M. L., Larocca, M., Cincio, L., Cerezo, M. & Sauvage, F. Lie-algebraic classical simulations for variational quantum computing. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2308.01432 (2023).
39. Larocca, M. et al. Diagnosing barren plateaus with tools from quantum optimal control. *Quantum* **6**, 824 (2022).
40. Wang, Q., Ma, Y., Zhao, K. & Tian, Y. A comprehensive survey of loss functions in machine learning. *Ann. Data Sci.* **9**, 187–212 (2022).
41. Ragone, M. et al. Representation theory for geometric quantum machine learning. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2210.07980 (2022).
42. Grcar, J. F. Mathematicians of Gaussian elimination. *Not. AMS* **58**, 782–792 (2011).
43. Suzuki, R., Mitarai, K. & Fujii, K. Computational power of one- and two-dimensional dual-unitary quantum circuits. *Quantum* **6**, 631 (2022).
44. Nocedal, J. & Wright, S. J. *Numerical Optimization* (Springer, 1999).
45. Jenkins, M. A. & Traub, J. F. A three-stage variable-shift iteration for polynomial zeros and its relation to generalized Rayleigh iteration. *Numerische Math.* **14**, 252–263 (1970).
46. Nesterov, Y. et al. *Lectures on Convex Optimization*, vol. 137 (Springer, 2018).
47. Jin, C., Ge, R., Netrapalli, P., Kakade, S. M. & Jordan, M. I. How to escape saddle points efficiently. *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70, 1724–1732 https://doi.org/10.48550/arXiv.1703.00887 (PMLR, 2017).
48. Wierichs, D., Izaac, J., Wang, C. & Lin, C. Y.-Y. General parameter-shift rules for quantum gradients. *Quantum* **6**, 677 (2022).
49. Ragone, M. et al. A unified theory of barren plateaus for deep parametrized quantum circuits. *Nat. Commun.* **15**, 7172 (2024).
50. Cerezo, M. et al. Does provable absence of barren plateaus imply classical simulability? Or, why we need to rethink variational quantum computing. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2312.09121 (2023).
51. Shoham, N. & Avron, H. Experimental design for overparameterized learning with application to single shot deep active learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**, 11766–11777 (2023).
52. Buchberger, B. *Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory*, 184–232 (Reidel Publishing Company, 1985).
53. Dubé, T. W. The structure of polynomial ideals and gröbner bases. *SIAM J. Comput.* **19**, 750–773 (1990).
54. Mayr, E. W. & Meyer, A. R. The complexity of the word problems for commutative semigroups and polynomial ideals. *Adv. Math.* **46**, 305–329 (1982).

## Acknowledgements

The authors thank Brandon Augustino, Raymond Rudy Putra, and the rest of our colleagues at the Global Technology Applied Research Center of JPMorganChase for helpful comments and discussions. This paper was prepared for informational purposes by the Global Technology Applied Research Center of JPMorgan Chase. This paper is not a product of the Research Department of JPMorganChase or its affiliates. Neither JPMorgan Chase nor any of its affiliates makes any explicit or implied representation or warranty and none of them accept any liability in connection with this paper, including, without limitation, with respect to the completeness, accuracy, or reliability of the information contained herein and the potential legal, compliance, tax, or accounting effects thereof. This document is not intended as investment research or investment advice, or as a recommendation, offer, or solicitation for the purchase or sale of any security, financial instrument, financial product, or service, or to be used in any way for evaluating the merits of participating in any transaction.

## Author contributions

N. Kumar and J. Heredge devised the project. J. Heredge, N. Kumar, D. Herman, and S. Chakrabarti developed the connection between the privacy of QML models and the dynamical Lie Algebra. J. Heredge and S.H. Sureshbabu created the diagrams of the method. J. Heredge performed numerical simulations in the paper. M. Pistoia led the overall project. All authors contributed in writing the paper.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at https://doi.org/10.1038/s41534-025-01022-z.

**Correspondence** and requests for materials should be addressed to Jamie Heredge.

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

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