

Efficient privacy-preserving training of quantum neural networks through ensemble encoding of quantum states

Gaoyuan Wang^{1,2}, Jonathan Warrell^{1,2}, and Mark Gerstein^{1,2,3,4,5,*}

¹*Program in Computational Biology and Bioinformatics, Yale University, New Haven, Connecticut 06520, USA*

²*Department of Molecular Biophysics and Biochemistry, Yale University, New Haven, Connecticut 06520, USA*

³*Department of Computer Science, Yale University, New Haven, Connecticut 06520, USA*

⁴*Department of Statistics and Data Science, Yale University, New Haven, Connecticut 06520, USA*

⁵*Department of Biomedical Informatics and Data Science, Yale University, New Haven, Connecticut 06520, USA*



(Received 2 January 2026; accepted 3 June 2026; published 6 July 2026)

Quantum computing is gaining popularity due to its potential to detect complex patterns in data by leveraging unique quantum phenomena. It is particularly promising for complex data applications, such as those in biomedicine. However, in these contexts, it is often necessary to share and then aggregate data from multiple participants to achieve sufficient statistical power. Sharing data, especially sensitive information such as personal medical records, raises significant privacy concerns. To address these challenges, we propose a quantum-native method for encoding entire data cohorts directly into special quantum states, which we refer to as composite states. These quantum states provide a generic encoding suitable for a wide variety of downstream computations while preventing the inference of individual-level information. Quantum computations can be performed directly on composite states without accessing the underlying raw data. Building on this, we present the theoretical foundations of our scheme's utility and privacy guarantees, demonstrating resistance to membership inference attacks as measured via differential privacy. Furthermore, we introduce protocols that support multiparty collaborative quantum neural network training across diverse domains. Finally, we validate the effectiveness of composite states on three different datasets, focusing on genomic studies while also indicating how the approach can be applied in other domains without adaptation.

DOI: [10.1103/81xl-nr9q](https://doi.org/10.1103/81xl-nr9q)

I. INTRODUCTION

Quantum neural networks (QNNs) are gaining increasing interest for their potential to outperform classical neural networks in certain tasks. Leveraging uniquely quantum phenomena such as superposition and entanglement they may be well suited to identify patterns and correlations in data that might be difficult to detect with classical methods [1–6].

In many fields, such as biomedical research, it is highly beneficial for multiple parties to collaborate by sharing data to jointly train QNN models, as this increases statistical power. However, the sensitive nature of much of these data demands strong privacy protections. Consequently, privacy-preserving quantum computing is essential for secure collaborative analysis. Blind quantum computing–based [7,8] quantum federated learning (QFL) [9–11] protocols provide strategies for multiple clients to jointly contribute to QNN model training without exposing private data. However, their practical utility remains limited due to significant computational overhead,

particularly from frequent client-server communication required at each optimization step.

To address these limitations, but also to introduce fundamentally different security schemes to provide robust fallback options in the event that vulnerabilities are discovered in current methods due to unforeseen cryptanalytic advances, we propose a scheme that leverages the nonuniqueness of quantum mixed states [12] to enable secure quantum information sharing and privacy-preserving QNN model training. In quantum mechanics, mixed states are probabilistic mixtures of pure states. When subjected to completely positive trace-preserving maps, including unitary operations or measurement, the outcomes reflect the weighted average of those obtained from the corresponding pure states. Building on this, we developed the concept of composite quantum states—a representative set of mixed states, each constructed from an ensemble of pure quantum states. These states encode the full ensemble structure while concealing individual data points. Importantly, the encoding is generic, allowing a single representation to be reused across multiple problems or tasks, as opposed to task-specific summary statistics. Based on this concept, we introduce a protocol for privacy-preserving QNN training applicable across diverse domains. In our framework, a QNN is trained by minimizing a composite-level loss function, defined as the error between the model's predictions on these composite states and their corresponding composite-level target values. Unlike QFL-based approaches,

*Contact author: mark@gersteinlab.org

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](#) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

this approach requires only one communication from each participating client and offers significant speedup by reducing the total number of quantum circuit executions. A detailed discussion of the additional limitations of QFL and a comparison with our approach are provided in Sec. II F.

Our scheme satisfies key privacy requirements by eliminating the need for direct access to individual-level data. The nonuniqueness of quantum mixtures [12] prevents reliable reconstruction of individual compositions from the composite quantum state, providing strong protection against composition recovery. Additionally, our approach mitigates the risk of membership inference by effectively “hiding” each individual within a group, offering privacy guarantees similar to those in (quantum) differential privacy [13–15].

Beyond privacy, evaluating the utility of composite quantum states is crucial for balancing the privacy-utility trade-off. We consider multiple aspects of utility, including the encoding’s generalizability across various tasks, computational efficiency, and the QNN’s predictive performance on future individual data when trained on composite states. Below, we highlight three key aspects that offer utility advantages over existing distributed learning schemes. (1) Essential quantum features of the individual states are preserved in the composite state, thereby maintaining the utility of the data for effective QNN training. For example, entangling subsystems within a composite state corresponds to entangling the same subsystems in each individual state. (2) The representational capacity of mixed states is richer than that of pure states, enabling more expressive modeling. (3) Any mixed state can be represented as a pure state in an enlarged Hilbert space [16]. Thus, our method effectively embeds multiple data points into a single, higher-dimensional quantum state for downstream model training. This projection mechanism is conceptually aligned with classical techniques such as support vector machines and quantum kernel methods, both of which aim to map data into higher-dimensional spaces to enhance separability and learning performance [17–20]. In addition, by reducing the number of quantum states and corresponding QNN evaluations, our approach significantly accelerates the training process, improving the computation efficiency.

We demonstrate the effectiveness of our protocol, both in terms of utility and privacy protection, in the context of genomic research, where collaborative training of neural network models is desirable for increased statistical power and privacy concerns are especially critical. Anonymized genomic datasets, previously assumed to safeguard privacy, have been shown to be susceptible to reidentification, data leakages, and other privacy attacks [21,22], underscoring the necessity of more robust privacy-preserving mechanisms. Genomic data serves as a unique, immutable identifier for individuals and carries strong predictive power for various health risks. Moreover, because genomic information is shared among family members, privacy breaches can impact not only individuals but also their relatives. Although demonstrated in the context of genomic research, this method is not limited to that domain; it can be directly applied to other fields, such as finance, for privacy-preserving collaborative or delegated QNN training without modification.

This paper is structured as follows: In Sec. II, we present the mathematical foundations of QNN training utility using

composite quantum states, establish the theoretical basis for privacy protection against composition recovery and membership inference, and introduce our protocol alongside its application to genomic data scenarios. In Sec. II F, we provide a more detailed comparison between our method and existing approaches, such as QFL.

In Sec. III, we evaluate the correctness of our approach and assess its utility by analyzing the performance across different model configurations and benchmarking against both classical and quantum models trained on individual-level data. We also quantify the privacy guarantees of our protocol. Section V provides the methodological details underlying our approach.

II. THEORETICAL FRAMEWORK

A. Composite quantum states

We demonstrate the concept of encoding data cohorts into composite quantum states in Fig. 1. Throughout this paper, we use the terms cohort and ensemble interchangeably, with cohort primarily employed when explicitly referring to groups of individuals. In Fig. 1(a), patient A’s real-world record contains genetic data, imaging data, and treatment record. It is stored as a real-valued feature vector x_A , where different data modalities are stored in different segments of the vector. Importantly, this representation is fully generic, allowing many disparate data types to be encoded without bias toward any modality. Using amplitude encoding, x_A is mapped to a quantum state ρ_A , which serves as the patient’s quantum twin. We then construct a composite quantum state ρ_{comp} by combining the quantum twins of multiple patients. As shown in Fig. 1(b), each cohort is represented by a single composite state that can be safely shared. Downstream analysis is performed directly on the composite states without access to individual records or labels. Unlike traditional summary statistics, quantum composites require no predefined target features and encode the full ensemble structure in a generic representation suitable for diverse downstream tasks.

B. Method overview

The overall structure of a gate-based QNN typically consists of three main components: (1) an encoder, which transforms classical data into quantum states; (2) a variational circuit, a parameterized unitary quantum circuit used to process the encoded information; and (3) a measurement step, which produces real-valued outputs from quantum states. In this work, we focus on a specific class of QNNs known as variational quantum classifiers. Variational quantum classifiers [23,24] represent one of the earliest and most well-established applications in quantum machine learning.

In this work, we consider a binary quantum classifier as a test case. The input data are encoded using amplitude encoding $\mathcal{A}$, and RealAmplitudes is used as the ansatz for the variational quantum circuit $\mathcal{U}(\theta)$. The final measurement is performed by evaluating the expectation value of the Pauli-Z operator P_z . The RealAmplitudes ansatz consists of single-qubit rotation gates with two-qubit CNOT gates. This forms a universal set of operations, meaning that any quantum algorithm can, in principle, be implemented using gates from this set. Notably, our scheme is not limited to amplitude encoding;

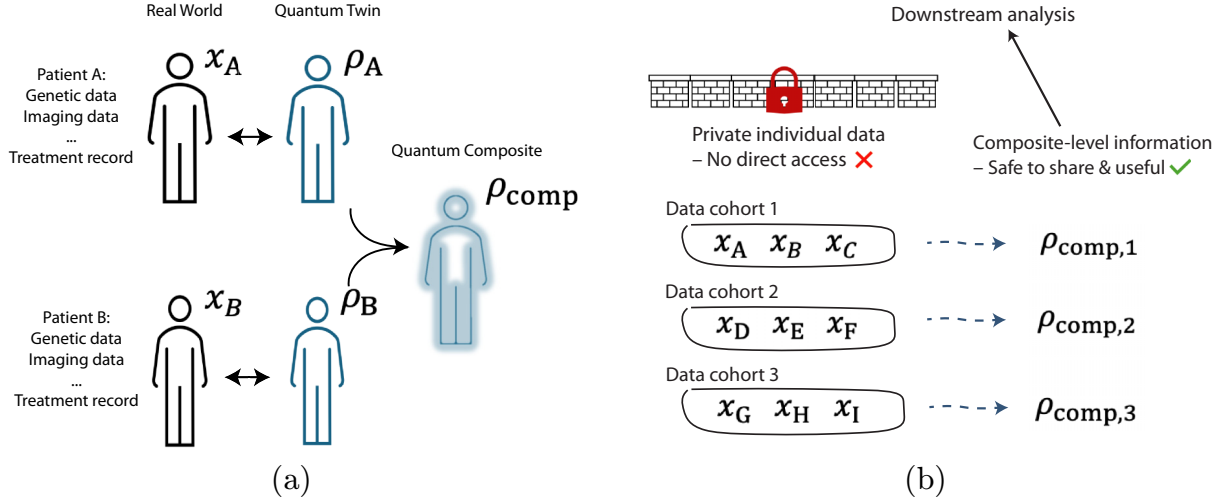


FIG. 1. Demonstration of the quantum composite concept. (a) Quantum composites as generic data cohort representations and (b) Secure analysis with composite quantum states.

it remains applicable if amplitude encoding is replaced with angle encoding.

Given a classical input vector x_u , where u denotes the index of the data point, amplitude encoding generates a quantum state represented by the density matrix $\rho_u = \mathcal{A}(x_u)$. The variational quantum circuit then transforms this input state into an output state $\sigma_u = \mathcal{U}(\theta)\rho_u\mathcal{U}^\dagger(\theta)$. The measurement step computes the expectation value of the Pauli-Z operator P_z with respect to the output state, yielding the model prediction:

$$p_u = \text{Tr}(P_z \sigma_u). \quad (1)$$

To train the quantum classifier using a classical optimizer, a loss function can be defined based on the expectation values p_u . For example, when using the L1 loss (i.e., the absolute error), the loss function is given by

$$\mathcal{L}_{1,\text{inst}} = \frac{1}{N_i} \sum_{u=1}^{N_i} |p_u - y_u| \quad (2)$$

$$= \frac{1}{N_i} \sum_{u=1}^{N_i} |\text{Tr}(P_z \mathcal{U}(\theta) \rho_u \mathcal{U}^\dagger(\theta)) - y_u|, \quad (3)$$

where y_u represents the ground-truth label. We will refer to this conventional type of loss function as the instance-level loss function and illustrate the workflow of QNNs using this loss function in Fig. 2(c). It is important to note that the instance-level loss function aggregates the loss computed for each individual data point.

Popular optimizers for QNNs are typically those suited for optimization problems where the gradient of the objective function is unavailable or costly to compute, due to the lack of efficient gradient measurement algorithms in QNNs. Many of them, such as COBYLA, rely solely on the average loss across the entire training dataset rather than losses computed for individual data points.

Leveraging this property, along with the fact that both the variational circuit and measurement components (i.e., parts 2 and 3 of the QNN architecture) are linear operations, we propose a method that operates on *composite quantum states*

representing batches of data points, defined as follows:

$$\rho_{\text{comp},i} = \frac{1}{N_i} \sum_{u \in C_i} \rho_u, \quad (4)$$

where C_i denotes a batch of data points sharing the same label and N_i is the number of samples in that batch. For example, C_0 consists of data points labeled as signals, while C_1 contains those labeled as controls. The construction of $\rho_{\text{comp},i}$ is illustrated in Fig. 2(a). The composite quantum states $\rho_{\text{comp},i}$ serve as representations of the ensemble of original input states and is used during QNN training.

The composite output state $\sigma_{\text{comp},i}$, obtained by applying the variational quantum circuit to the composite input state $\rho_{\text{comp},i}$ is

$$\sigma_{\text{comp},i} = \mathcal{U}(\theta) \rho_{\text{comp},i} \mathcal{U}^\dagger(\theta). \quad (5)$$

The measurement output of a composite-level QNN is the expectation value of the Pauli-Z operator with respect to the composite output state:

$$p_{\text{comp},i} = \text{Tr}(P_z \sigma_{\text{comp},i}). \quad (6)$$

When training the quantum classifier with composite quantum states, the loss function is given by the accuracy of probabilistic predictions, analogously to an L1 variant of the Brier score:

$$\mathcal{L}_{1,\text{comp}} = \sum_i |p_{\text{comp},i} - y_{\text{comp},i}| \quad (7)$$

$$= \sum_i |\text{Tr}(P_z \mathcal{U}(\theta) \rho_{\text{comp},i} \mathcal{U}^\dagger(\theta)) - y_{\text{comp},i}|, \quad (8)$$

with $y_{\text{comp},i}$ the label of the batch C_i . The index i runs over all batches. We will refer to this type of loss function as the composite-level loss function. We illustrate the workflow of the QNNs using the composite-level loss function in Fig. 2(b).

One cannot trace back the individual data points $x_u \in C_i$ from the composite quantum state $\rho_{\text{comp},i}$ because different probabilistic mixtures of individual quantum states can lead to the same composite quantum state. However, optimizing

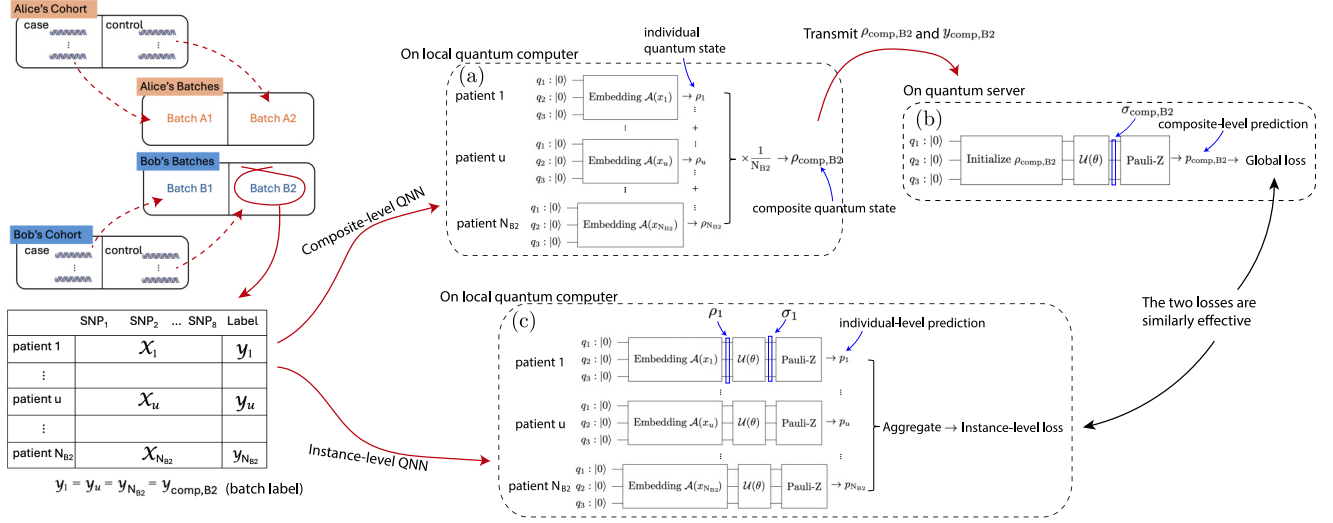


FIG. 2. Workflow for batching and processing batch B2 with either composite-level QNN or instance-level QNN: (a) Construction of the composite quantum state, (b) composite-level QNN processing, and (c) instance-level QNN processing.

the composite-level loss function, which is computed solely from the composite quantum state, is as effective as optimizing the instance-level loss function presented in Eq. (3). Henceforth, we will refer to QNN models trained using the composite-level loss function as composite-level QNNs, and those trained with the instance-level loss function as instance-level QNNs. In Sec. II C, we provide the mathematical details that demonstrate the effectiveness of the composite-level loss function and thus the utility of the composite quantum states. In Secs. II D and II E, we analytically examine the privacy properties of composite quantum states with respect to composition recovery and membership inference, respectively.

C. Utility of composite quantum states

Recall the output of the composite-level QNN for batch C_i ,

$$p_{\text{comp},i} = \text{Tr}(P_z \sigma_{\text{comp},i}). \quad (9)$$

And the averaged instance-level QNN output considering all data points in this batch is

$$p_{\text{inst},i} = \frac{1}{N_i} \sum_{u \in C_i} (\text{Tr}[P_z \sigma_u]). \quad (10)$$

We notice that

$$p_{\text{comp},i} = \text{Tr}[P_z \sigma_{\text{comp},i}] = \text{Tr}[P_z \mathcal{U}(\theta) \rho_{\text{comp},i} \mathcal{U}(\theta)^\dagger] \quad (11)$$

$$= \text{Tr} \left[P_z \mathcal{U}(\theta) \left(\frac{1}{N_i} \sum_{u \in C_i} \rho_u \right) \mathcal{U}(\theta)^\dagger \right] \quad (12)$$

$$= \text{Tr} \left[P_z \frac{1}{N_i} \sum_{u \in C_i} \mathcal{U}(\theta) \rho_u \mathcal{U}(\theta)^\dagger \right] \quad (13)$$

$$= \text{inst},i. \quad (14)$$

The above result holds because both the unitary transformation in the QNN and the computation of expectation values are linear operations.

Since the expectation value with respect to the Pauli-Z operator is bounded within the interval $-1 < \text{Tr}[\dots] < 1$, we rescale it to the range $[0,1]$ using the transformation $(\text{Tr}[\dots] + 1)/2$. We structure the dataset such that the label y takes binary values in $\{0,1\}$. Since we have grouped the data points with the same label, the label y_i is identical for all $u \in C_i$. Either all probability predictions $(\text{Tr}[P_z \sigma_u] + 1)/2$ for $u \in C_i$ are greater than or equal to y_i , or all are less than or equal to y_i . Therefore, the rescaled loss for each data point

$$(\text{Tr}[P_z \sigma_u] + 1)/2 - y_i$$

has the same sign across every $u \in C_i$. This uniformity in sign allows us to move the absolute value outside the summation. As a result, the rescaled, instance-level loss averaged over all data points in the batch becomes

$$\begin{aligned} \mathcal{L}_{1,\text{inst},i} &= \frac{1}{N_i} \sum_{u \in C_i} |(\text{Tr}[P_z \sigma_u] + 1)/2 - y_u| \\ &= \frac{1}{N_i} \left| \sum_{u \in C_i} [(\text{Tr}[P_z \sigma_u] + 1)/2 - y_u] \right|. \end{aligned} \quad (15)$$

The composite-level L1 loss function for the entire dataset summed over all batches is given by

$$\begin{aligned} \mathcal{L}_{1,\text{comp}} &= \sum_i |(\text{Tr}[P_z \sigma_{\text{comp},i}] + 1)/2 - y_i| \\ &= \sum_i \left| \frac{1}{N_i} \sum_{u \in C_i} (\text{Tr}[P_z \sigma_u] + 1)/2 - y_i \right|. \end{aligned} \quad (16)$$

The corresponding instance-level loss for the entire dataset summed over all batches is

$$\mathcal{L}_{1,\text{inst}} = \sum_i \sum_{u \in C_i} \frac{1}{N_i} |(\text{Tr}[P_z \sigma_u] + 1)/2 - y_u|. \quad (17)$$

The two equations above are equivalent because the L1 loss remains a linear operation when the model output is bounded between 0 and 1, and the true labels are either 0 or 1. Standard neural networks train classifiers over cohorts by optimizing a loss function that aggregates information across all individuals or a suitable approximation. As a result, when using optimizers that rely on the average loss across data points, such as COBYLA, the instance-level loss function conveys the same information during optimization as the composite-level loss function.

We have shown that, in the case of the L_1 loss, the composite-level loss function is equivalent to the averaged instance-level loss function. Moreover, the composite-quantum-state-based approach can be extended to settings involving nonlinearities, such as the application of activation functions or nonlinear loss functions.

Let f be an arbitrary monotonic nonlinear function. The classical output of the QNN is then given by

$$f(\text{Tr}[P_z \sigma_u] - y_u). \quad (18)$$

For example, in the case of L_2 loss, $f(\cdot) = (\cdot)^2$, while for a sigmoid activation function, $f(\cdot) = \text{sig}(\cdot)$.

During the optimization procedure, a parameter update from θ to θ' induces a corresponding change in the output quantum state. We define

$$\begin{aligned} \delta \sigma_u &:= \sigma'_u - \sigma_u \\ &= \mathcal{U}(\theta') \rho_u \mathcal{U}^\dagger(\theta') - \mathcal{U}(\theta) \rho_u \mathcal{U}^\dagger(\theta), \end{aligned} \quad (19)$$

and introduce

$$\begin{aligned} x_u &= \text{Tr}[P_z \sigma_u] - y_u, \quad \delta x_u = \text{Tr}[P_z \delta \sigma_u], \\ x_{\text{comp},i} &= \text{Tr}[P_z \sigma_{\text{comp},i}] - y_i, \\ \delta x_{\text{comp},i} &= \text{Tr}[P_z \delta \sigma_{\text{comp},i}] = \frac{1}{N_i} \sum_{u \in C_i} \delta x_u. \end{aligned} \quad (20)$$

The model output before the update is $f(x)$, and after the update it is $f(x + \delta x)$.

Assuming that parameter updates are small, the corresponding changes $\delta \sigma_u$ are also small. We approximate the loss function to its first-order term using power expansion. The change in the composite-level loss for batch C_i is

$$\begin{aligned} \Delta f_{\text{comp},i} &= f(x_{\text{comp},i} + \delta x_{\text{comp},i}) - f(x_{\text{comp},i}) \\ &\approx f'(\bar{x}_{\text{comp},i}) \delta x_{\text{comp},i}, \\ \exists \bar{x}_{\text{comp},i} &\in [x_{\text{comp},i}, x_{\text{comp},i} + \delta x_{\text{comp},i}]. \end{aligned} \quad (21)$$

Similarly, the average change in the instance-level loss over the same batch is

$$\begin{aligned} \frac{1}{N_i} \sum_{u \in C_i} \Delta f_u &= \frac{1}{N_i} \sum_{u \in C_i} (f(x_u + \delta x_u) - f(x_u)) \\ &\approx \frac{1}{N_i} \sum_{u \in C_i} (f'(\bar{x}_u) \delta x_u), \\ \exists \bar{x}_u &\in [x_u, x_u + \delta x_u], \end{aligned} \quad (22)$$

where we have applied the mean value theorem.

Since f is monotonic, its derivative has a constant sign over the entire domain. Denoting this sign by s_i , we have

$$\text{sgn}(f'(\bar{x}_u)) = \text{sgn}(f'(\bar{x}_{\text{comp},i})) = s_i, \quad \forall u \in C_i. \quad (23)$$

And hence

$$\begin{aligned} \text{sgn}(\Delta f_{\text{comp},i}) &= \text{sgn}(f'(\bar{x}_{\text{comp},i}) \delta x_{\text{comp},i}) \\ &= \text{sgn}\left(\frac{1}{N_i} \sum_{u \in C_i} f'(\bar{x}_{\text{comp},i}) \delta x_u\right) \\ &= s_i \cdot \text{sgn}\left(\sum_{u \in C_i} \delta x_u\right) \\ &= \text{sgn}\left(\frac{1}{N_i} \sum_{u \in C_i} f'(\bar{x}_u) \delta x_u\right) \\ &= \text{sgn}\left(\frac{1}{N_i} \sum_{u \in C_i} \Delta f_u\right). \end{aligned} \quad (24)$$

Therefore, while the composite-level and instance-level losses may differ in magnitude, their changes share the same sign and their gradients point in the same direction during optimization. Any parameter update that improves the composite-level loss will also improve the instance-level loss in the same direction.

We numerically validate the correlation between changes in the composite-level and instance-level losses during optimization; see Sec. V A for details.

D. Privacy of composite quantum states against composition recovery

We aim to quantify the ability to recover the individual constituent states from an observed composite quantum state ρ_{comp} . To this end, we estimate the uncertainty that remains about the composition of these underlying individual states after observing ρ_{comp} . This uncertainty is evaluated using conditional entropy as an information-theoretic measure. For practical purposes, we assume that the individual states considered in our problems come from a finite discrete set. This helps discretize the problem and simplifies the calculation of the quantification of information and uncertainty. Importantly, a higher level of uncertainty about the composition indicates less information leakage and thus implies a greater degree of privacy.

We start with a finite discrete set of basis quantum states $\{\rho_j^B\}_{j=1}^{N_{\text{state}}}$, with N_{state} the size of the set. This set contains all feasible individual states and the individual states are not necessarily linearly independent. For example, without loss of generality, if each individual state represents a person's single nucleotide polymorphisms (SNP) vector, and each SNP can take one of three values, 0, 1, or 2, then an individual state is a vector of length N_{SNP} with entries from $\{0, 1, 2\}$. The total number of feasible individual states is then $N_{\text{state}} = 3^{N_{\text{SNP}}}$. This set of all feasible SNP conformations can be

enumerated as

$$\begin{aligned} S_1^B &= (0, 0, \dots, 0), & S_2^B &= (0, 0, \dots, 1), \\ S_3^B &= (0, 0, \dots, 2), & \dots, & S_{N_{\text{state}}}^B = (2, 2, \dots, 2). \end{aligned} \quad (25)$$

Each such vector S_j^B corresponds to a pure quantum state represented as a density matrix ρ_j^B . The density matrix ρ_j^B is obtained via amplitude encoding by first normalizing S_j^B , converting it into a state vector $|S_j^B\rangle$, and then taking the outer product of this state vector with its conjugate transpose $\rho_j^B = |S_j^B\rangle\langle S_j^B|$.

We also specify a batch size N_i , which is the total count of individual data points in the composite quantum state. The composite quantum state can be represented as

$$\rho_{\text{comp}} = \sum_j^{N_{\text{state}}} b_j \rho_j^B, \quad (26)$$

where $\mathbf{b} = (b_1, b_2, \dots, b_{N_{\text{state}}})$ are non-negative integers ($\mathbb{N}^{N_{\text{state}}}$) satisfying $\sum_{j=1}^{N_{\text{state}}} b_j = N_i$. Each coefficient b_j indicates how many times the basis state ρ_j^B appears in the batch. If there exists only a single vector $\mathbf{b}$ compatible with the observed ρ_{comp} , then the individual composition of the batch can be fully inferred, including how many times each SNP conformation occurs. In such cases, privacy is reduced. A large value of b_j , i.e., frequent occurrence of a particular SNP conformation, typically reflects population-level information. However, small values of b_j , particularly $b_j = 1$, could risk exposing the presence of a specific individual with a particular SNP conformation in the batch, potentially revealing sensitive metadata such as the label associated with that batch. Conversely, if many different vectors $\mathbf{b}$ are consistent with the observed ρ_{comp} , then it becomes difficult or impossible to infer the exact composition of the batch. In such cases, the appearance of a specific SNP conformation becomes untraceable, thereby ensuring the privacy of individual-level data.

To this end, we estimate the total number of feasible $\mathbf{b}$ compositions, both with and without the constraint of a fixed composite state ρ_{comp} , as a function of the batch size N_i and the number of basis states N_{state} . A larger number of feasible $\mathbf{b}$ vectors, i.e., higher conditional entropy, corresponds to greater uncertainty about the underlying composition and, consequently, a higher level of privacy.

Since the ρ_j^B are not linearly independent, multiple different $\mathbf{b}$ vectors may result in the same ρ_{comp} . We define the set of all feasible such vectors as

$$B = \left\{ \mathbf{b} \in \mathbb{N}^{N_{\text{state}}} \left| \rho_{\text{comp}} = \sum_{j=1}^{N_{\text{state}}} b_j \rho_j^B, \sum_{j=1}^{N_{\text{state}}} b_j = N_i \right. \right\}. \quad (27)$$

The size of the set $|B|$ reflects the number of distinct $\mathbf{b}$ consistent with ρ_{comp} and thus measures the uncertainty in inferring $\mathbf{b}$ from ρ_{comp} .

We first ignore the constraint imposed by ρ_{comp} and estimate the number of integer vectors $\mathbf{b}$ summing to N_i . This can be remodeled into a *balls into bins problem*, and the total

number of feasible combinations is

$$|B_{\text{all}}| = \binom{N_i + N_{\text{state}} - 1}{N_{\text{state}} - 1} = \frac{(N_i + N_{\text{state}} - 1)!}{(N_{\text{state}} - 1)! N_i!} \quad (28)$$

$$\sim \frac{(N_i + N_{\text{state}} - 1)^{N_i + N_{\text{state}} - 1}}{(N_{\text{state}} - 1)^{N_{\text{state}} - 1} N_i^{N_i}}. \quad (29)$$

In the second step, we used the Stirling approximation for the asymptotic behavior of $n!$ for large N_{state} .

We now consider the constraint imposed by fixing ρ_{comp} . Although the components of $\mathbf{b}$ are discrete integers, the construction of density matrices ρ_j^B involves normalizing the corresponding state S_j^B . This normalization allows the components of ρ_j^B to take on a wide range of values, particularly when large N_{state} . As a result, the composite state ρ_{comp} , enriched by the variability in $\mathbf{b}$, can be considered to span a continuous region of the state space rather than being restricted to a discrete, inhomogeneous lattice of points. Since the ρ_j are linearly dependent, they span only a d -dimensional subspace. Thus, the constraint $\rho_{\text{comp}} = \sum_j b_j \rho_j^B$ imposes d -independent linear conditions, reducing the original N_{state} -dimensional solution space by d degrees of freedom. Consequently, the set of feasible $\mathbf{b}$ consistent with both the total count and ρ_{comp} lies on the intersection of the discrete N_{state} -dimensional lattice [as described in Eq. (29)] with a $(N_{\text{state}} - d)$ -dimensional hyperplane.

Hence, we estimate the number of feasible $\mathbf{b}$ vectors to be

$$|B| \approx |B_{\text{all}}|^{\frac{N_{\text{state}} - d}{N_{\text{state}}}} \sim \left[\frac{(N_i + N_{\text{state}} - 1)^{N_i + N_{\text{state}} - 1}}{(N_{\text{state}} - 1)^{N_{\text{state}} - 1} N_i^{N_i}} \right]^{\frac{N_{\text{state}} - d}{N_{\text{state}}}}. \quad (30)$$

In the example of SNPs, we have $d = N_{\text{SNP}}$, $N_{\text{state}} = 3^d$.

We can then approximate the remaining uncertainty about the individual composition, i.e., conditional entropy, as

$$\begin{aligned} H(\mathbf{b} \mid \rho_{\text{comp}}) &\propto \ln |B_{\rho}| \sim \left(\frac{N_{\text{state}} - d}{N_{\text{state}}} \right) \ln \\ &\times \left[\frac{(N_i + N_{\text{state}} - 1)^{N_i + N_{\text{state}} - 1}}{(N_{\text{state}} - 1)^{N_{\text{state}} - 1} N_i^{N_i}} \right]. \end{aligned} \quad (31)$$

This suggests that the number of feasible solutions for $\mathbf{b}$, and hence the uncertainty about the individual composition, increases with larger values of batch size N_i and N_{state} . As a result, inferring the exact composition vector $\mathbf{b}$ from the observed composite state ρ_{comp} becomes increasingly difficult. For instance, even in a relatively small setting with $N_{\text{SNP}} = 4$ and $N_i = 2$, the estimated number of feasible $\mathbf{b}$ configurations exceeds 7000.

E. Privacy of composite quantum states against membership inference

The membership inference considered in this work is similar to the well-known challenge of determining whether an individual's data are included in a dataset, either from aggregate statistics or from model outputs under black-box access [21,25,26]. In our setting, the publicly available information are the composite quantum states. Given an observed composite quantum state ρ_{comp} , we aim to assess whether it is

possible to infer if a specific individual state ρ^* , representing one individual's data, contributed to the construction of ρ_{comp} (i.e., whether it belongs to the underlying data cohort). Since batches of quantum data often carry associated meta-data (such as class labels with disease status), identifying the presence of a particular individual in the batch may lead to privacy breaches by leaking sensitive information. Intuitively, as the batch size increases, the influence of any single individual's quantum state becomes increasingly “lost in the crowd” [27]—a principle at the heart of many differential privacy mechanisms designed to ensure that no single data point significantly affects the outcome. This is also reflected by the fact the adjacent quantum states, with and without the target individual, become less distinguishable as the batch size grows.

To formalize this intuition, we adopt concepts from quantum differential privacy [14]. Instead of directly comparing quantum states, we evaluate the classical information obtained through measurements. Specifically, we consider a positive operator-valued measure (POVM), which extracts classical outcomes from quantum states. In this framework, the composite quantum state is a mixture of the target individual state ρ^* and other states in the batch. Because the batch composition is unknown to anyone except the trusted local party who owns the data and can thus be treated as effectively random, the measurement outcomes provide inherent randomness that helps conceal the contribution of the target state ρ^* . Here we establish the formalism and calculate the privacy risk for a fixed ρ^* . To quantitatively estimate the privacy risk for a given dataset, one can iterate over all individuals by considering each ρ^* in turn to identify the worst-case privacy risk.

We compare two neighboring composite states: one including the individual state ρ^* and one excluding it:

$$\rho_{\text{comp-1}} = \frac{1}{N_i - 1} \sum_{u=1}^{N_i-1} \rho_u, \quad (32)$$

$$\rho_{\text{comp}} = \frac{N_i - 1}{N_i} \rho_{\text{comp-1}} + \frac{1}{N_i} \rho^*. \quad (33)$$

We consider here a specific measurement defined by the projection operator corresponding to the individual state ρ^* . As we will show later, and as is also intuitively expected, any measurement other than the projection onto ρ^* yields a less distinct signal regarding the presence of ρ^* in the batch.

The probability p_{comp}^* of obtaining ρ^* when performing a POVM defined by the corresponding projection operator of ρ^* on the quantum state ρ_{comp} is

$$p_{\text{comp}}^* = \text{Tr}(\rho^* \rho_{\text{comp}}) = \frac{1}{N_i} \text{Tr}(\rho^* \rho^*) + \frac{N_i - 1}{N_i} \text{Tr}(\rho^* \rho_{\text{comp-1}}) \quad (34)$$

$$= \frac{1}{N_i} \text{Tr}(\rho^* \rho^*) + \frac{1}{N_i} \sum_{u=1}^{N_i-1} \text{Tr}(\rho^* \rho_u) \quad (35)$$

$$= \frac{1}{N_i} + \frac{1}{N_i} \sum_{u=1}^{N_i-1} \text{Tr}(\rho^* \rho_u), \quad (36)$$

$$= \left(\frac{N_i - 1}{N_i} p_{\text{comp-1}}^* + \frac{1}{N_i} \right) = \alpha p_{\text{comp-1}}^* + \beta, \quad (37)$$

with $\alpha = \frac{N_i-1}{N_i}$ and $\beta = \frac{1}{N_i}$. Assuming that all states ρ_u are mutually independent and also independent of ρ^* , from the perspective an attacker, the overall overlap $\sum_{u=1}^{N_i-1} \text{Tr}(\rho^* \rho_u)$ can be assumed to be a random variable with the distribution determined by the underlying statistics of the population. As a result, p_{comp}^* and $p_{\text{comp-1}}^*$ are also random variables with known relationships. If $g_1(p_{\text{comp-1}}^*)$ denotes the probability density function (pdf) of $p_{\text{comp-1}}^*$, then the pdf of p_{comp}^* is given by a related pdf g_2 via

$$g_2(p_{\text{comp}}^*) = \frac{1}{|\alpha|} g_1((p_{\text{comp}}^* - \beta)/\alpha). \quad (38)$$

The closer g_1 and g_2 are, the harder it is to distinguish whether ρ^* was included, thereby implying stronger privacy guarantees. In the quantum (ϵ, δ) -differential privacy framework [14], the mechanism is considered private if

$$\Pr[\mathcal{M}^*(\rho_{\text{comp}}) \in S] \leq e^{\epsilon^*} \Pr[\mathcal{M}^*(\rho_{\text{comp-1}}) \in S] + \delta, \quad (39)$$

for all measurable subsets S , where $\mathcal{M}^*$ denotes the projection measurement corresponding to ρ^* . The corresponding privacy parameter measuring the level of privacy for a fixed ρ^* is

$$\epsilon^* = \sup_p \left(\ln \left[\frac{g_2(p) - \delta}{g_1(p)} \right] \right) = \sup_p \left(\ln \left[\frac{g_1(\frac{p-\beta}{\alpha})/\alpha - \delta}{g_1(p)} \right] \right). \quad (40)$$

For small δ , as $N_i \rightarrow \infty$, we have $\alpha \rightarrow 1$ and $\beta \rightarrow 0$, implying $\epsilon^* \rightarrow 0$. This confirms our expectation: The measurement results on ρ_{comp} and $\rho_{\text{comp-1}}$ become indistinguishable with increasing batch size. To quantitatively estimate ϵ for a dataset without fixing ρ^* in advance, we iterate over all ρ^* and compute

$$\epsilon = \max_{\rho^*} \{\epsilon^*\}. \quad (41)$$

Suppose now that the measurement is not perfectly aligned with ρ^* but contains a random term $\rho' = (1 - a_{\text{noise}})\rho^* + a_{\text{noise}}\rho_{\text{noise}}$, we have

$$p'_{\text{comp}} = \frac{1 - a_{\text{noise}}}{N_i} + \frac{a_{\text{noise}}}{N_i} \text{Tr}(\rho_{\text{noise}} \rho^*) + \frac{1 - a_{\text{noise}}}{N_i} \sum_{u=1}^{N_i-1} \text{Tr}(\rho^* \rho_u) + \frac{a_{\text{noise}}}{N_i} \sum_{u=1}^{N_i-1} \text{Tr}(\rho_{\text{noise}} \rho_u) \quad (42)$$

and

$$p'_{\text{comp-1}} = \frac{1 - a_{\text{noise}}}{N_i - 1} \sum_{u=1}^{N_i-1} \text{Tr}(\rho^* \rho_u) + \frac{a_{\text{noise}}}{N_i - 1} \sum_{u=1}^{N_i-1} \text{Tr}(\rho_{\text{noise}} \rho_u). \quad (43)$$

As before, we can express the relationship as

$$p'_{\text{comp}} = \alpha' p'_{\text{comp-1}} + \beta', \quad (44)$$

with $\alpha' = \alpha = \frac{N_i-1}{N_i}$ and $\beta' = -\frac{1-a+a\text{Tr}(\rho_{\text{noise}}\rho^*)}{N_i} \leq \beta$.

Since $\text{Tr}(\rho^* \rho_u)$ and $\text{Tr}(\rho_{\text{noise}} \rho_u)$ are identically distributed, the same base distribution g_1 applies. Thus,

$$\epsilon' = \sup_{p'} \left(\ln \left[\frac{g_1(\frac{p'-\beta'}{\alpha})/\alpha - \delta}{g_1(p')} \right] \right). \quad (45)$$

Given that $\beta' < \beta^*$, the argument $\frac{p'-\beta'}{\alpha}$ lies closer to p than $\frac{p-\beta}{\alpha}$. As a result, $g_1(\frac{p'-\beta'}{\alpha})$ is generally closer to $g_1(p)$ than $g_1(\frac{p-\beta}{\alpha})$. Under the assumptions of small δ and that g_1 changes gradually without significant fluctuations at the scale $\mathcal{O}(\frac{1}{N_i})$, as with a Gaussian distribution, we expect $\epsilon' < \epsilon^*$.

In practice, an attack might proceed as follows: The malicious party begins by conducting population studies to prepare. They obtain population statistical information that approximates the distribution of ρ_u and use it to simulate many random ρ_u states. From this distribution, they select random ρ^* states and compute the corresponding distributions for g_1 and g_2 . These precomputed distributions serve as background knowledge. When the attacker receives a composite state they wish to analyze, they perform a projection measurement using ρ^* , obtain an outcome, and compare it against the two distributions to draw conclusions.

The exact value of the privacy parameter can be estimated using random matrix theory or empirically, provided the distribution of the ensemble ρ_u is known. We will evaluate this value in the results section (Sec. III) for the specific dataset used in our experiments.

F. Motivation and alternatives

Alternative to the approach proposed in this work, QFL [28–31] is one existing approach that enables multiple clients to collaboratively train a composite machine learning model without exposing their private data. The architecture of QFL is similar to that of classical federated learning. In a QFL protocol designed for collaborative QNN training, each client obtains an initial share QNN as a starting point and uses their own quantum devices to perform local training on their own dataset. Then, instead of sending raw data to a central server, they share only model parameter updates, such as gradient. The central server or a designated client collect the updates and combines them to update the parameters of the shared QNN. The updated QNN model is then sent back to the clients as a new starting point for the next round of training. This process repeats until the model reaches the desired performance. One critical risk for QFL is gradient leakage, where sensitive data can be inferred from shared gradients during training. To protect the privacy of data, additional techniques such as blind quantum computation [8,9], gradient hiding [11], and differential privacy [32] have been introduced to be integrated with QFL schemes.

A major unresolved limitation of QFL is its high communication overhead, which is further exacerbated by the integration of blind quantum computation. This overhead stems from the frequent client-server communication at each optimization step. Although promising results have been demonstrated on benchmarks such as MNIST and the Breast Cancer Wisconsin dataset [9], the practical scalability of current QFL implementations remains limited. Existing approaches typically involve fewer than 10 qubits

[9], which significantly restricts their applicability to large-scale genomic datasets and other data-intensive domains. Our approach eliminate the need of frequent client-server communication at each optimization step and requires only one communication from each participating client.

In addition, current QFL capabilities, along with QNN models more broadly, are limited by the relatively low capacity of noisy intermediate-scale quantum devices. Our approach addresses this constraint by significantly speedup by reducing the total number of quantum circuit executions.

Another advantage of our approach is its generality. Unlike many QFL methods that are specifically tailored to a particular use case, our approach is broadly applicable and can be used with any QNN architecture. If a new analysis needs to be performed, then the composite quantum states can be directly reused without the need to reacquire information from the participating clients who owned the individual level data.

Moreover, QFL faces another potential security vulnerabilities include scenarios where an attacker gains control over a participating device and maliciously modifies model parameters which will remain unnoticed because the aggregation procedure prevents the server from inspecting each user's model update [33]. In contrast, our approach requires trust in only a single party, the central server responsible for training, regarding model security, thereby reducing the attack surface.

Finally, exploring new approaches not only offers better computational efficiency than the existing QFL framework but also introduces fundamentally different security paradigms in the case that vulnerabilities are discovered in QFL due to unforeseen attacks.

G. Protocol

The effective training of QNN models using composite quantum states can be leveraged to develop a quantum-native protocol for private collaborative training, as well as privacy-preserving QNN training on remote quantum servers. We begin by a discussion on batch construction, then we introduce a protocol for single-party delegated QNN training, as it is easier to explain, and then extend it to the multiparty collaborative learning setting. We illustrate practical deployment scenarios in Fig. 3. For concreteness, we consider classification tasks in which researchers aim to predict whether a given sequence of SNPs is associated with a particular disease. Beyond privacy protection, this protocol also offers potential gains in training efficiency. Since composite quantum states summarize batches of data, they reduce the total number of quantum circuit executions required, significantly cutting computation time.

(1) *Batching*. The way data are grouped into batches is typically determined by the party holding the data. In collaborative settings with small cohorts, a single cohort often forms a batch. When a cohort is split into multiple batches, different batching strategies can be considered to balance utility and privacy. Constructing composite states from overlapping data points helps preserve information, producing purer states and, for structured datasets, enhancing differential privacy by making individual states harder to distinguish. However, exploiting data structure can also reduce privacy, as restricting individual states to be close to the composite state reduces the

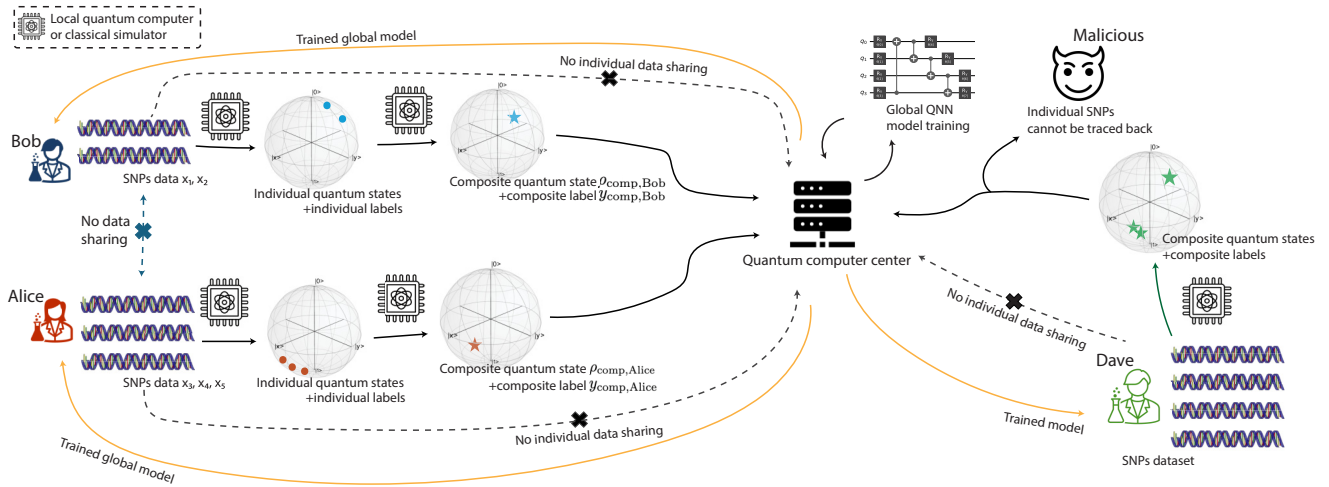


FIG. 3. Illustration of protocol demonstrating the use of composite quantum states for secure private single-party delegated QNN model training and collaborative QNN model training without exposing individual datasets.

number of possible compositions. Uneven batch sizes must also be accounted for, as they introduce implicit weighting of individual data points; this can be mitigated by weighting each composite state according to batch size in the loss function. Additionally, it is easier to infer individual contributions in smaller batches. Details on the two strategies considered in this work are provided under Methods (Sec. V D).

(2) *Assumptions on information access.* When evaluating privacy in Secs. II D and II E, we assume a worst-case scenario in which the attacker (e.g., the quantum server or a malicious party) has fully recovered the quantum state ρ_{comp} . This would require quantum state tomography, which in general needs a number of copies growing exponentially with the number of qubits. In our protocol, however, the number of copies provided by the client corresponds to a standard number of shots used to execute quantum circuits (e.g., a few thousand). While this may enable shallow tomography on the attacker side, our privacy estimates derived under the assumption of full tomography serves as upper bounds. When uneven batch sizes are used and the loss is reweighted, the batch size must be shared with the server. This does not introduce additional privacy risk, as it is already treated as public in our analysis. In classification tasks, the client also shares the composite-level label, revealing the label of all instances in the batch; however, our differential privacy analysis ensures that the resulting membership inference risk remains low. We also assume the attacker has access to population-level information, including subgroup statistics associated with certain metainformation (e.g., specific diseases), as such aggregate data are generally accessible under weak access control.

(3) *Single-party delegated QNN training.* We first consider the single-party delegated training scenario as illustrated on the right side of Fig. 3. Dave wants to train a quantum classifier and offload resource-intensive tasks to a remote quantum server while preserving data privacy. Using his local, less powerful quantum device, Dave begins by encoding his classical individual SNP data into pure quantum states via a standard quantum encoding technique. He then partitions his training dataset, composed of data from multiple patients,

into batches. Each batch contains multiple patients who share the same disease status but retain distinct SNP profiles. For each batch, Dave constructs a composite quantum state using the individual quantum states of the patients in that batch. This results in a “composite-level patient” represented by a mixed quantum state that captures the aggregate statistical features of the group, paired with a corresponding composite (batch) label indicating the shared disease status. Next, Dave specifies the desired QNN architecture and transmits both the composite quantum states and their associated composite labels to the remote quantum server for training. For both the server and any malicious actor who might intercept the composite quantum states during transmission, it would be infeasible to reconstruct the original individual data due to the inherent nonuniqueness of quantum mixtures. After the training, the quantum server sends back the trained model to Dave.

(4) *Multiparty collaborative QNN model training.* The multiparty collaborative QNN training scenario is illustrated on the left side of Fig. 3. In this setting, researchers Alice and Bob, working at different institutions, each possess SNPs profiles from patients diagnosed with a specific disease. Their objective is to collaboratively train a QNN model using a remote quantum server capable of handling the resource-intensive training process. But they wish to keep their individual data private from both other researchers and the remote quantum server while ensuring protection against potential security breaches during data transmission. Following the same principle as in the single-party setting, Alice and Bob independently preprocess their local data using standard quantum encoding techniques on their own, less-powerful quantum devices. Each party then constructs one or more composite quantum states that statistically represent subsets of their datasets, with each composite state associated with a corresponding composite label. Once prepared, the composite quantum states and their associated labels are sent to the remote quantum server. The server aggregates the inputs from all participating parties and performs QNN training using the specified model architecture. Upon completion, the trained

QNN model is returned to all participating entities. At no point does the server gain access to raw individual data, ensuring that privacy is preserved throughout the training process. Although demonstrated with two clients, the protocol can be easily extended to support multiple contributing parties.

In the experimental sections of this paper, although not explicitly stated, it is assumed that data batching and composite quantum state preparation are performed by trusted local parties, while the QNN training on these composite quantum states is carried out by a remote quantum server. Throughout the experiments, the entire dataset is treated collectively without explicitly identifying the contributing entities. The multiparty setup can be inferred; for example, in an experiment with 10 batches per class, this could correspond to two participating clients, each independently creating five batches and the corresponding composite quantum states, which are then sent to the remote server along with batch size and label metadata. From the perspective of the remote quantum server, single-party and multiparty training scenarios are similar. In both cases, the server receives quantum states from one or multiple clients without insight into their preparation. It does not even know whether the input states represent individual data points or composite quantum states constructed from data batches but simply performs QNN training on the states it receives.

H. Conditions and limitations

Since the correlation between the composite-level loss update and the instance-level loss update is shaped by the approximation in Eqs. (21) and (22), the extent to which optimizing the composite-level loss translates into improvements at the instance level depends on how well the nonlinearity of a small optimization step can be captured by its linear term in a power series expansion. In the linear setting (e.g., with L1 loss), the two are equivalent. In contrast, when nonlinearity is too strong or parameter updates are too large, these approximations may no longer hold, and alignment in the sign of change can become unreliable. Therefore, it is important to use small learning rates when training QNNs with a composite-level loss and to empirically validate the correlation between composite-level and instance-level losses when employing new architectures with strong nonlinear behavior.

We briefly comment on the distinction between the privacy analysis considered in this work, which is grounded in a data science setting, and that adopted in cryptographic settings, such as secure key exchange. In the latter, privacy is typically defined in a strict entropic sense. The keys are completely random with no internal structure, and any information leakage beyond random noise is considered unacceptable. By contrast, in data analysis and machine learning applications, some level of population-level information is generally assumed to be accessible. Therefore, the relevant notion of privacy focuses on protecting sensitive individual-level information rather than eliminating all information content. Our analysis should be understood within this context, where the goal is to balance utility and privacy rather than to achieve information-theoretic secrecy.

III. EXPERIMENTAL RESULTS

A. Experimental setup

To evaluate the correctness of the proposed scheme, we constructed multiple synthetic datasets, including a pure-state dataset and a mixed-state dataset. Additionally, we generated synthetic SNP datasets and utilized real-world SNP data from the PsychENCODE project to evaluate the accuracy of QNN model training using composite quantum states, as well as to assess the utility and privacy of the composite quantum states. Detailed descriptions of all datasets are provided under Sec. V. In preparing the SNP datasets in Secs. III C and III D, the SNP vectors are ultimately transformed into general floating-point vectors of varying dimensionality. Their resemblance to SNP data does not affect the effectiveness of the composite quantum states. As shown in Sec. III B, our method is valid regardless of the data's source or real-world meaning. Consequently, although our privacy and utility evaluations focus on SNP data of classical origin, the results reflect the effectiveness of our approach across diverse domains.

We implemented multiple versions of QNNs. For benchmarking, two QNN models were designed with access to individual data points, each trained using an instance-level loss function. The first, referred to as the *basic QNN*, is based on Qiskit's original quantum classifier and supports only pure states. The second, the *instance-level QNN*, was custom-developed to handle mixed states and also serves as the foundational implementation for the composite-level QNN model. Finally, we implemented a third model, the *composite-level QNN*, which has access only to composite quantum states and is trained using a composite-level loss function. This model simulates the perspective of a remote quantum server. Further details on these models are available under Methods.

We employed two batching strategies in this work: random batching and smart batching, as described under Methods.

We evaluate the utility of composite quantum states under different settings by first training a QNN on a training set composed of the corresponding composite quantum states and then measuring the QNN's classification performance, e.g., the test AUC score, on individual data points from the test set. To assess privacy in the low statistic regime, we estimate the worst-case privacy parameter ϵ as

$$\epsilon = \max_p \left\{ \ln \left(\frac{g_2(p) - \delta}{g_1(p)} \right) \right\}, \quad (46)$$

which represents a slight modification of Eqs. (40) and (41). Further details on the privacy risk assessment procedure are provided under Methods.

B. Results on synthetic toy datasets

We tested all three models (basic, instance-level, and composite-level QNN) on the synthetic toy datasets introduced in Sec. V B 1 and present the results in Tables I and II. We did not use any activation but performed a linear rescaling of the loss function to bring the expectation value of the Pauli-Z operators to the range [0,1] as described in Sec. II C. When constructing the composite quantum states, we used one batch per class, i.e., one composite quantum state per class. We found that for both pure- and mixed-state datasets, all three

TABLE I. Performance of all three models on the pure-state dataset with $e_s = 0.4$.

e_{shift} Fidelity	1.5 0.835	2 0.719	2.5 0.608	3 0.528	3.5 0.423	4 0.413
Basic QNN AUC	0.508/0.503	0.629/0.588	0.697/0.698	0.733/0.752	0.795/0.798	0.801/0.807
Instance-level QNN AUC	0.508/0.503	0.629/0.588	0.696/0.698	0.733/0.751	0.795/0.798	0.801/0.806
Composite-level QNN AUC	0.508/0.503	0.629/0.588	0.696/0.7	0.734/0.752	0.795/0.798	0.801/0.806

models exhibited nearly identical performance. This suggests that our instance-level QNN implementation is correct and that the composite-level and instance-level objective functions are equivalent if no nonlinear activation function is used.

We found that, although the Pauli-Z expectation value distributions of the two classes are well separated across all values of e_{shift} , making the classification task appear straightforward, the performance of all models remained notably sub-optimal. At small e_{shift} , the test AUC is only slightly above 0.5, indicating near-random classification performance. The AUC improves as e_{shift} increases or, equivalently, as the fidelity between the composite quantum states of each class decreases. This suggests that the dissimilarity between the composite quantum states is a key factor in classification performance. Even when the data points of each class form well-separated clusters, classifiers failed to distinguish between them as long as their composite quantum states overlapped. This can occur when data point configurations from different classes result in very similar composite quantum states.

The fact that the well-established implementations of Qiskit quantum classifiers (e.g., Pure-Inst) failed to classify our synthetic toy dataset at small e_{shift} , where the fidelity between the composite quantum states of the input classes is high, suggests that careful consideration is needed when assessing whether QNNs are appropriate for a given task.

Our observation implies two important points: (1) before employing quantum classifiers, one should examine whether the composite quantum states of the input classes significantly overlap and (2) the composite-level objective function may effectively replace instance-level objectives. Finally, we note that our synthetic datasets were intentionally designed to test specific aspects of our model. Real-world classification datasets are unlikely to present such particular conditions for quantum classifiers.

C. Results on synthetic SNPs dataset

We tested instance-level QNN and composite-level QNN with a sigmoid activation function on the synthetic SNPs dataset to demonstrate the effectiveness of using only composite quantum states for QNN training. We used four layers for the RealAmplitude ansatz. For the instance-level QNN

on the 256 SNPs dataset, we reduced the number of epochs to 30 and patience to 8 due to the long running time. We varied the number of batches per class from 1 to 50, using both random and smart batching strategies. This corresponds to batch sizes $N_i \in [325, 6.5]$. To explore a wide spectrum of model sizes and complexities, we considered the input feature dimensionality from 16 to 512 SNPs. With amplitude encoding, these input sizes correspond to quantum systems with 4 to 9 qubits.

To evaluate the utility of the composite quantum states, we measured the test AUC scores of the QNN models. For each set of parameters, we conducted five independent runs using a fixed set of five distinct initial weight configurations. We also tested the logistic regression model as the baseline classical machine learning methods. We used logistic regression for comparison, as it is the core algorithm used in widely adopted Polygenic Risk Score software such as PRSice-2 [34]. Furthermore, logistic regression shares a similar activation function with the QNN models.

In Figs. 4(a) and 4(b), we show the mean test AUC scores for the instance-level QNN, composite-level QNN with varying numbers of batches, and the baseline logistic regression model for two synthetic datasets containing 128 and 256 SNPs, respectively. Shaded areas indicate the error ranges for the composite-level QNN models. Figure 4(c) shows the test AUC scores of instance-level QNN, composite-level QNNs with 10 batches, and the logistic regression model for various SNPs dimensionalities. Due to computational limitations, the more resource-intensive instance-level QNN was not evaluated on the dataset with 512 SNPs. Together, Figs. 4(a)–4(c) demonstrate the utility of different composite quantum state configurations. As shown in Fig. 4(c), the QNN models consistently outperform logistic regression across datasets with varying SNP dimensionalities, suggesting that QNNs are generally well suited for classification tasks across the range of input dimensionalities considered. For the 128- and 256-synthetic SNPs datasets in Figs. 4(a) and 4(b), the test AUC scores of the composite-level QNN fluctuate closely around those of the instance-level QNN across different batch sizes and batching strategies. This indicates a stable utility of the composite quantum states as batch size increases and effectively support QNN training.

TABLE II. Performance of all three models on the mixed-state dataset with $e_s = 0.2$.

e_{shift} Fidelity	0.25 0.816	0.26 0.805	0.27 0.797	0.28 0.789	0.29 0.78	0.3 0.767
Instance-level QNN AUC	0.59/0.578	0.68/0.655	0.784/0.798	0.884/0.883	0.961/0.967	0.989/0.99
Composite-level QNN AUC	0.59/0.578	0.68/0.655	0.784/0.798	0.884/0.883	0.961/0.967	0.989/0.99

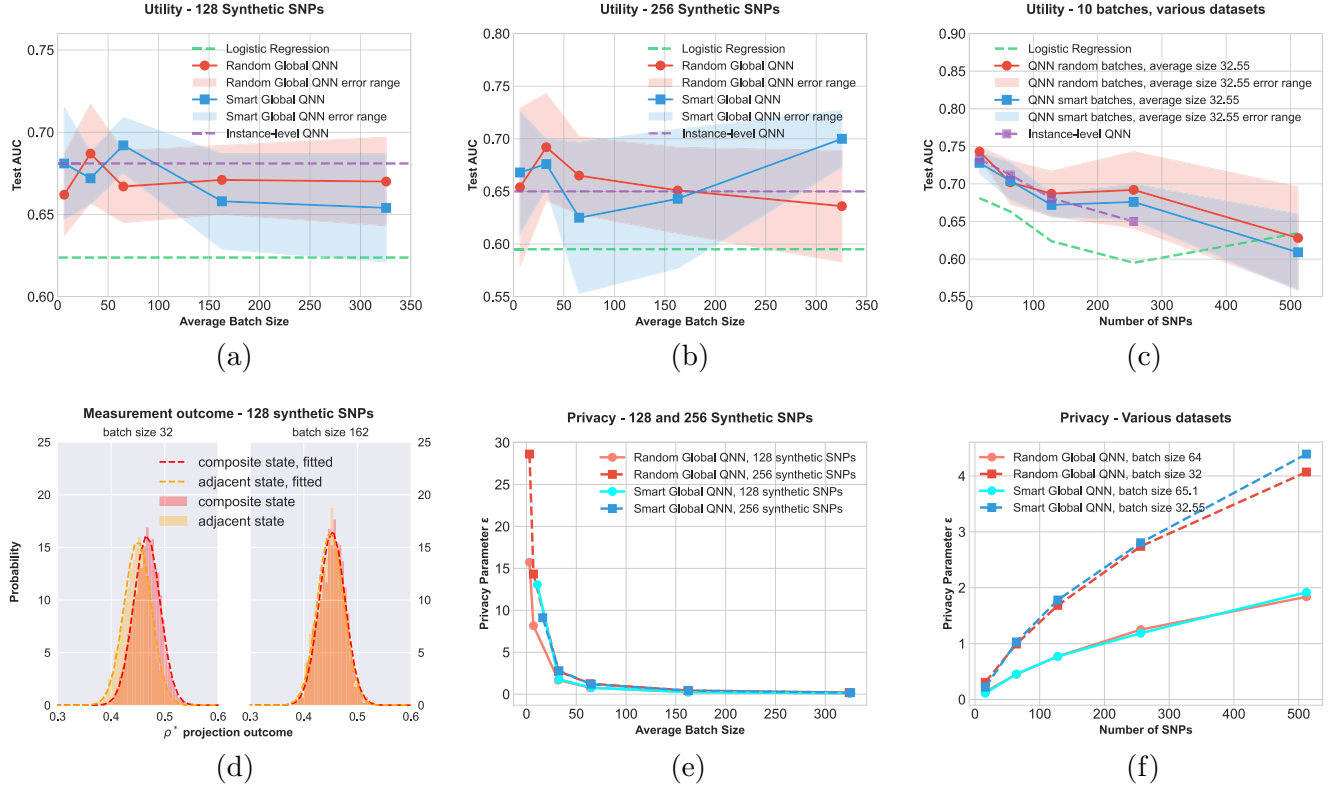


FIG. 4. Utility and privacy evaluation of composite quantum states on synthetic SNPs dataset. [(a) and (b)] Test AUC performance of instance-level and composite-level QNN models compared to a classical baseline, evaluated on synthetic datasets with 128 and 256 SNPs across varying batch sizes. (c) Test AUC of composite-level QNN with fixed batch size, instance-level QNN, and classical baseline as input SNP dimensionality varies. (d) Distribution of ρ^* projection outcomes for random batching. [(e) and (f)] Privacy parameter ϵ as a function of batch size and input feature dimensionality.

To quantify the privacy risk, we estimate the privacy parameter ϵ by evaluating the projection of the target state ρ^* onto the composite state ρ_{comp} (which includes ρ^*) and the adjacent state $\rho_{\text{comp}-1}$ (which excludes ρ^*). For random batching, to improve statistical reliability, we iterate over all individual states and form $\lfloor \# \text{data points} / N_i \rfloor$ nonoverlapping batches from the remaining states of the same class. These batches are used to construct $\rho_{\text{comp}-1}$, while ρ_{comp} is formed by adding ρ^* to each batch. For smart batching, we iterate over all individual states and compute the projection of ρ^* onto ρ_{comp} (the composite state of the batch that contains ρ^*) and onto $\rho_{\text{comp}-1}$, which is constructed by removing ρ^* from that batch. This procedure yields a distribution of projection values, as shown in Fig. 4(d) for $N_i = 32$ and 162 under random batching, where $g_1(p_{\text{comp}-1}^*)$ is labeled as the adjacent state and $g_2(p_{\text{comp}}^*)$ as the composite state. These distributions are well approximated by normal distributions, and we numerically extract the mean (μ) and standard deviation (std) for both the composite-level and adjacent states. The overlap (projection) between ρ^* and the other individual states in the batch can be interpreted as the effective noise added for privacy preservation. The privacy parameter ϵ is computed as the worst-case privacy loss for a fixed $\delta = 0.05$, using Eq. (40). The parameter δ serves as a relaxation term, allowing a small probability of the privacy loss exceeding ϵ . This relaxation is particularly important in low-statistics regimes, where rare but large deviations may occur.

Figure 4(e) shows the privacy parameter ϵ for synthetic datasets with 128 and 256 SNPs, plotted against batch size for both random and smart batching schemes. As observed, ϵ decreases rapidly as the batch size increases. Around a batch size of 60, the small value of ϵ less than one indicates a strong level of privacy protection. The random batching scheme generally yields equal or lower ϵ values compared to smart batching at similar batch sizes. This is because smart batching includes smaller-than-average batches, which yield higher privacy loss due to the worst-case nature of the privacy risk calculation. Moreover, at very small batch sizes, smart batching may form batches containing only a single data point, resulting in infinite privacy loss. Such cases are unsuitable for privacy-preserving computations and are therefore excluded from our analysis. Although smart batching slightly increased privacy risk in this synthetic dataset without showing a clear improvement in utility, this batching strategy is designed to enhance predictive performance while potentially trading off some privacy. Its effects become more apparent in the next section, where we analyze a dataset with a clearer underlying structure. As we will show, smart batching not only demonstrates the utility-privacy tradeoff but can also reduce privacy risk in such settings.

Figure 4(f) presents the privacy parameter ϵ for synthetic datasets with 128 and 256 SNPs, plotted as a function of batch size for both random and smart batching schemes. As shown, ϵ decreases rapidly with increasing batch size.

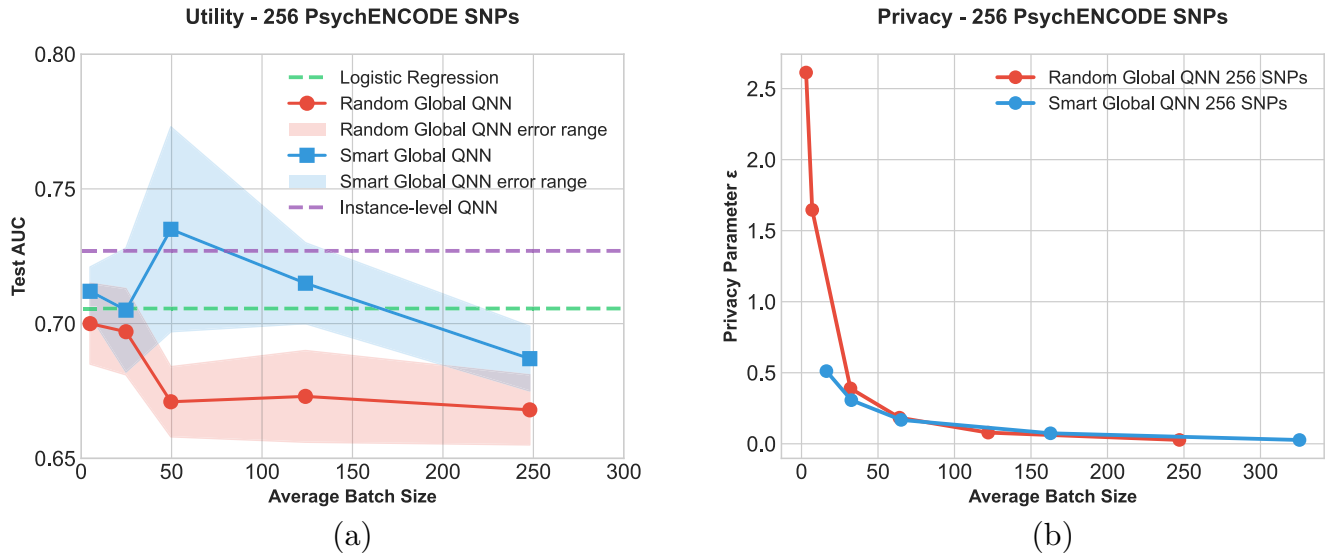


FIG. 5. Utility and privacy evaluation of composite quantum states on PsychENCODE dataset. (a) Test AUC performance of instance-level QNN, composite-level QNNs across varying batch sizes and the classical baseline models on PsychENCODE datasets containing 256 SNPs. Shaded area is the error range. (b) Privacy parameter ϵ as a function of batch size.

At a batch size of around 60, the small value of $\epsilon < 1$ suggests a high degree of privacy protection. The random batching scheme provides smaller or similar ϵ as smart batching for similar batch size, and this is due to the presence of smaller batch sizes in the smart batching scheme; since we are calculating the worst-case privacy risk, those smaller than average batches suffer under higher privacy risk. In addition, at a small batch size, smart batching could construct batches containing only one data point, leading to infinite privacy loss. Such scenarios are unsuitable for privacy-preserving computation and are therefore excluded from the results. In Fig. 4(f), we plot ϵ as a function of the number of SNPs (input feature dimensions). The privacy loss ϵ increases slowly with higher input dimensionality, showing a relatively small effect.

In terms of computational efficiency, using 10 random batches results in each batch containing, on average, 32.5 individual data points, leading to a 32.5-fold reduction in the number of quantum circuit executions compared to instance-level QNN training.

D. Results on PsychEncode SNPs dataset

We focus on the PsychENCODE dataset with 256 SNPs and associated brain disorder status labels, and employ a QNN circuit with five layers. For baseline comparison, we also evaluate the classical logistic regression model on the same task. The test AUC scores of different QNN models and the logistic regression baseline are shown in Fig. 5(a), demonstrating the utility of the composite quantum states under various settings. The instance-level QNN outperforms the classical logistic regression, demonstrating the applicability of quantum classifiers for this task. The test AUC of composite-level QNNs with random batching decreases at larger batch sizes and remains slightly below the baseline models. On the other hand, the performance of the composite-level QNNs with smart batching fluctuates around that of the instance-level QNN,

demonstrating the utility of the composite quantum states constructed via smart batching. We note that although the composite-level QNN does not outperform classical logistic regression in this task, the latter has access to individual-level data, which provides it with an inherent advantage. Therefore, achieving comparable performance under such constraints highlights the effectiveness of the composite-level QNN in privacy-preserving settings.

There is a weak indication, though not statistically conclusive, of a performance peak at an average batch size of approximately 50 for smart batching, followed by a decline for larger batch sizes. The performance advantage offered by smart batching over random batching on this real-world dataset could be due to the clearer underlying structure in the real-world data, which facilitates more effective grouping of samples and thus reduces information loss during the nonlinear postprocessing. The existence of an optimal batch size suggests that certain grouping strategies better preserve essential data patterns. In contrast, synthetic SNP datasets, characterized by high randomness and limited inherent structure, did not exhibit performance gains from smart batching. Overall, QNN models trained on composite quantum states offer a robust framework for disease status classification on real-world genomic datasets, demonstrating the practical utility of composite quantum states for this task.

Figure 5(b) shows the worst-case privacy parameter ϵ as a function of the average batch size. The computation of ϵ follows the same procedure as in the previous section on synthetic data. For batch sizes smaller than 20, smart batching occasionally produces batches containing only a single data point, resulting in infinite privacy loss. Such cases are excluded from the plot. As the batch size increases beyond ≈ 30 , ϵ rapidly decreases to values below 0.5, indicating strong protection against membership inference. We report batch size statistics in the smart batching setting for the regime offering a good balance between utility and privacy. For target batch

sizes around 25, 50, and 124, the observed batch sizes span 13–37 (median 24), 16–83 (median 48.3), and 53–195 (median 124), respectively.

Smart batching yields slightly lower ϵ values than random batching, suggesting improved privacy. This improvement may stem from the clearer structure in the dataset, which leads to greater similarity among data points within the same batch. When the target quantum state is hidden among similar states, it becomes harder to distinguish and thus offers better privacy. Despite the presence of small batches due to uneven batch sizes in smart batching, the “hiding among similar states” effect appears to dominate in contrast to the random batching scenario, where such structure is absent.

For this dataset, using 10 batches results in an approximate 25-fold reduction in the number of quantum circuit executions compared to instance-level training.

IV. DISCUSSION

In this work, we proposed a scheme that uses composite quantum states, constructed as probabilistic mixtures of individual data points, to encode the data cohort directly, and demonstrated the effectiveness of using these composite quantum states to train QNN models. We established the mathematical foundation showing that optimization on a composite-level loss function defined over these composite quantum states simultaneously optimizes the conventional loss function defined over individual data points. We also provided the mathematical basis for the privacy protection offered by our method against both composition recovery and membership inference attacks. Based on this foundation, we introduced protocols for privacy-preserving QNN training in both single-party delegated QNN learning and multiparty collaborative QNN training settings. Our approach eliminates the need to share individual-level data, ensuring data protection during transmission. Additionally, our method reduces the number of forward-pass executions, thereby accelerating QNN training. We validated the protocol on quantum binary classification tasks across three datasets: intrinsically quantum data, synthetic genome sequences generated from classical sources, and a real-world genome sequence dataset. We also discussed the general applicability of our method across domains. That is, it could easily be used for other individual level data for instance such as that product marketing or census determination [35,36]. Overall, our results show that QNN models trained on these composite states can match the performance of models trained on individual data points, while offering strong privacy protection and using only a fraction of the computational resources.

V. METHODS

A. Numerical verification of nonlinear operations

In the following, we numerically validate the correlation between changes in the composite-level and instance-level losses during one update.

We first test the case of sigmoid activation function, i.e., $f(\cdot) = \text{sig}(\cdot)$, i.e., we assess the correlation between

$$\begin{aligned} & \frac{1}{N_i} \sum_{u \in C_i} [\text{sig}(\text{Tr}[P_z \mathcal{U}(\theta') \rho_u \mathcal{U}^\dagger(\theta')])] \\ & - \text{sig}(\text{Tr}[P_z \mathcal{U}(\theta) \rho_u \mathcal{U}^\dagger(\theta)])] \\ \text{and } & \text{sig} \left(\text{Tr}[P_z \mathcal{U}(\theta') \left(\sum_{u \in C_i} \rho_u / N_i \right) \mathcal{U}^\dagger(\theta')] \right) \\ & - \text{sig}(\text{Tr}[P_z \mathcal{U}(\theta) \left(\sum_{u \in C_i} \rho_u / N_i \right) \mathcal{U}^\dagger(\theta)]). \end{aligned} \quad (47)$$

This setup simulates a single step in the QNN optimization process, where a new parameterized circuit $\mathcal{U}(\theta')$ is proposed to replace the current one $\mathcal{U}(\theta)$. During QNN training, a parameter update is accepted if it leads to an improvement in the objective function, i.e., if it decreases the loss in the desired direction.

In our test, each data point ρ_i is a random density matrix. We employ a sigmoid activation function with a temperature parameter $k = 10$, i.e., $\text{sig}(x) = 1/(1 + \exp(-kx))$, consistent with the quantum classifier configuration in the results section. The parameterized quantum circuits $\mathcal{U}(\theta)$ and $\mathcal{U}(\theta')$ are constructed as random circuits by selecting gates uniformly at random from a standard gate set.

In Fig. 6(a), for the scenario of employing a sigmoid activation function and L1 loss, we present the correlation between the composite-level and instance-level expressions from Eq. (47) as a function of batch size N_{batch} and averaged over 10 000 independent trials. The Pearson correlation coefficient (PCC) between the two expressions (red curve) remains consistently high, indicating a strong linear relationship in their responses to circuit updates. Additionally, the strong correlation between the signs of the changes (blue curve), along with the high probability of agreement of the signs of the change (green curve), indicates that during optimization of the composite-level objective function, an accepted update to the QNN parameters, whether it decreases the loss for class zero or increases it for class one, will induce a change of the same sign in the instance-level objective function. This alignment ensures that the instance-level objective is also optimized in the intended direction.

In the case where an L2 loss function is used instead of L1 loss, i.e., $f(\cdot) = (\cdot)^2$, the composite-level loss function over the entire dataset reads

$$\begin{aligned} \mathcal{L}_{2,\text{comp}} &= \sum_i (\text{Tr}[P_z \sigma_{\text{comp},i}] - y_i)^2 \\ &= \sum_i \left(\text{Tr} \left[\sum_{u \in C_i} \frac{1}{N_i} P_z \sigma_u \right] - y_i \right)^2, \end{aligned} \quad (48)$$

while the instance-level loss across the dataset reads

$$\mathcal{L}_{2,\text{inst}} = \sum_i \sum_{u \in C_i} \frac{1}{N_i} (\text{Tr}[P_z \sigma_u] - y_u)^2. \quad (49)$$

We again conduct a numerical evaluation of the correlation between changes in the composite-level and instance-level

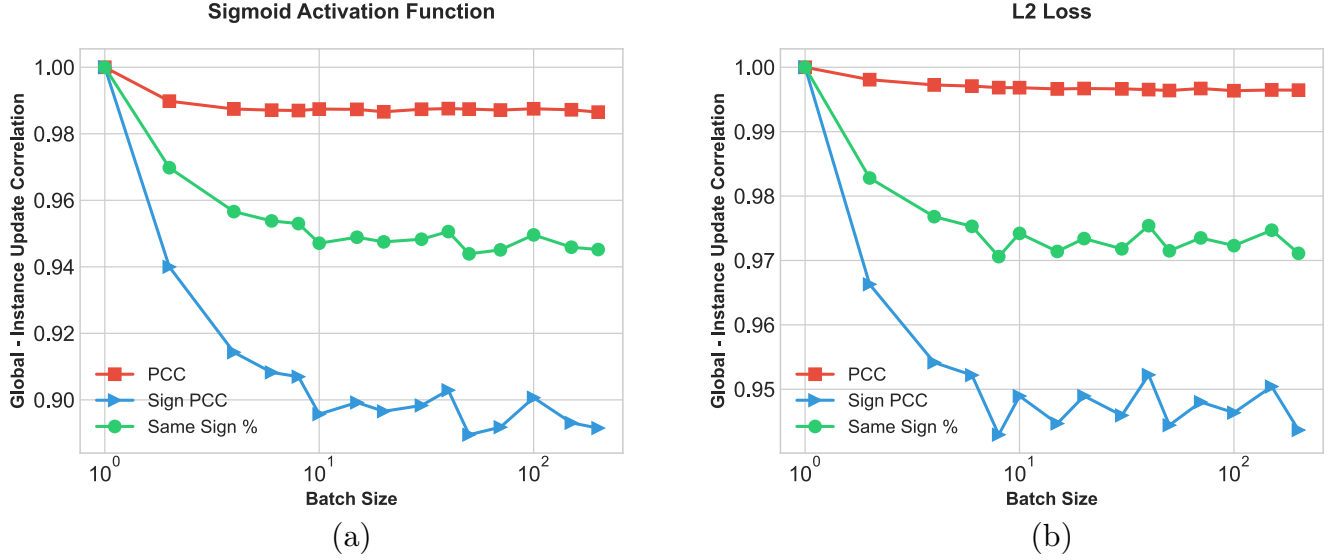


FIG. 6. Correlation between updates to the composite-level and instance-level objective functions induced by random quantum circuits. We report the PCC, the PCC between the signs of the updates, and the percentage of cases where the signs of the composite-level and instance-level updates match for (a) using a sigmoid activation function and (b) using a L2 loss function.

losses induced by a single optimization update, i.e., the correlation between

$$\frac{1}{N_i} \sum_{u \in C_i} [(\text{Tr}[P_z \mathcal{U}(\theta') \rho_u \mathcal{U}^\dagger(\theta')] - y_u)^2 - (\text{Tr}[P_z \mathcal{U}(\theta) \rho_u \mathcal{U}^\dagger(\theta)] - y_u)^2]$$

and

$$\left(\text{Tr} \left[P_z \mathcal{U}(\theta') \left(\sum_{u \in C_i} \rho_u / N_i \right) \mathcal{U}^\dagger(\theta') \right] - y_u \right)^2 - \left(\text{Tr} \left[P_z \mathcal{U}(\theta) \left(\sum_{u \in C_i} \rho_u / N_i \right) \mathcal{U}^\dagger(\theta) \right] - y_u \right)^2. \quad (50)$$

Without loss of generality, we set $y_u = -1$.

We present the PCC between the composite-level and instance-level expressions in Eq. (50), the PCC between their signs and the probability of both expressions having the same sign in Fig. 6(b). The observed high correlation suggests that optimizing the composite-level objective function concurrently improves the instance-level objective function in the same direction when an L2 loss is used. One should note that the correlation is not perfect because of the linearity assumption we adopted in Eqs. (21) and (22) as discussed as one utility limitation. The incorporation the highly nonlinear sigmoid function led to a lower same-sign-rate compared to the case of the L2 loss function.

B. Datasets

1. Synthetic toy dataset

The first dataset we constructed is a synthetic toy dataset aiming to demonstrate the equivalence of the composite-level and instance-level objective functions when no activation function is used, as well as the correctness of our implementation of variational quantum classifier accommodating

mixed states. Since this step is solely intended to validate our theoretical framework and algorithm, we use a small system (four-dimensional Hilbert space with two qubits) and directly prepare the density matrices, bypassing the encoding process. It is composed of two subdatasets focusing on pure states and mixed states.

a. Pure-state dataset. We first selected two initial states for each class and then generated the dataset ensemble by perturbing the initial states. For the positive class, the two initial states are

$$|x_a^{\text{ini},1}\rangle = |00\rangle \quad \text{and} \quad |x_b^{\text{ini},1}\rangle = |01\rangle. \quad (51)$$

The initial states for the negative class are

$$|x_a^{\text{ini},-1}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle) \quad \text{and} \quad |x_b^{\text{ini},-1}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |01\rangle). \quad (52)$$

The ensemble is generated by applying a $R_y(\theta_{y1})$ gate on the initial states. θ_{y1} is a random number drawn from the uniform distribution $\mathcal{U}(-e_s, e_s)$, with e_s a small parameter that controls the overall scale of the perturbation on the first qubit. For data with a positive label, we apply an additional $R_y(\theta_{y2})$ rotation on the second qubit with the random number θ_{y2} drawn from another uniform $\mathcal{U}(-e_{\text{shift}}, e_{\text{shift}})$, with e_{shift} controlling the difference between the two classes. For each data point, we stored both its state vector and density matrix expressions for later use, these two expressions are equivalent.

b. Mixed-state dataset. This dataset simulates intrinsically quantum data, which can only be expressed as density matrices. We use the same initial states as in the pure-state dataset and generated the ensemble by mixing the initial states with small random density matrices. The density matrices of the initial states are denoted as ρ_{ini} . We sample random density matrices, ρ_{random} , from the Hilbert-Schmidt metric. For the

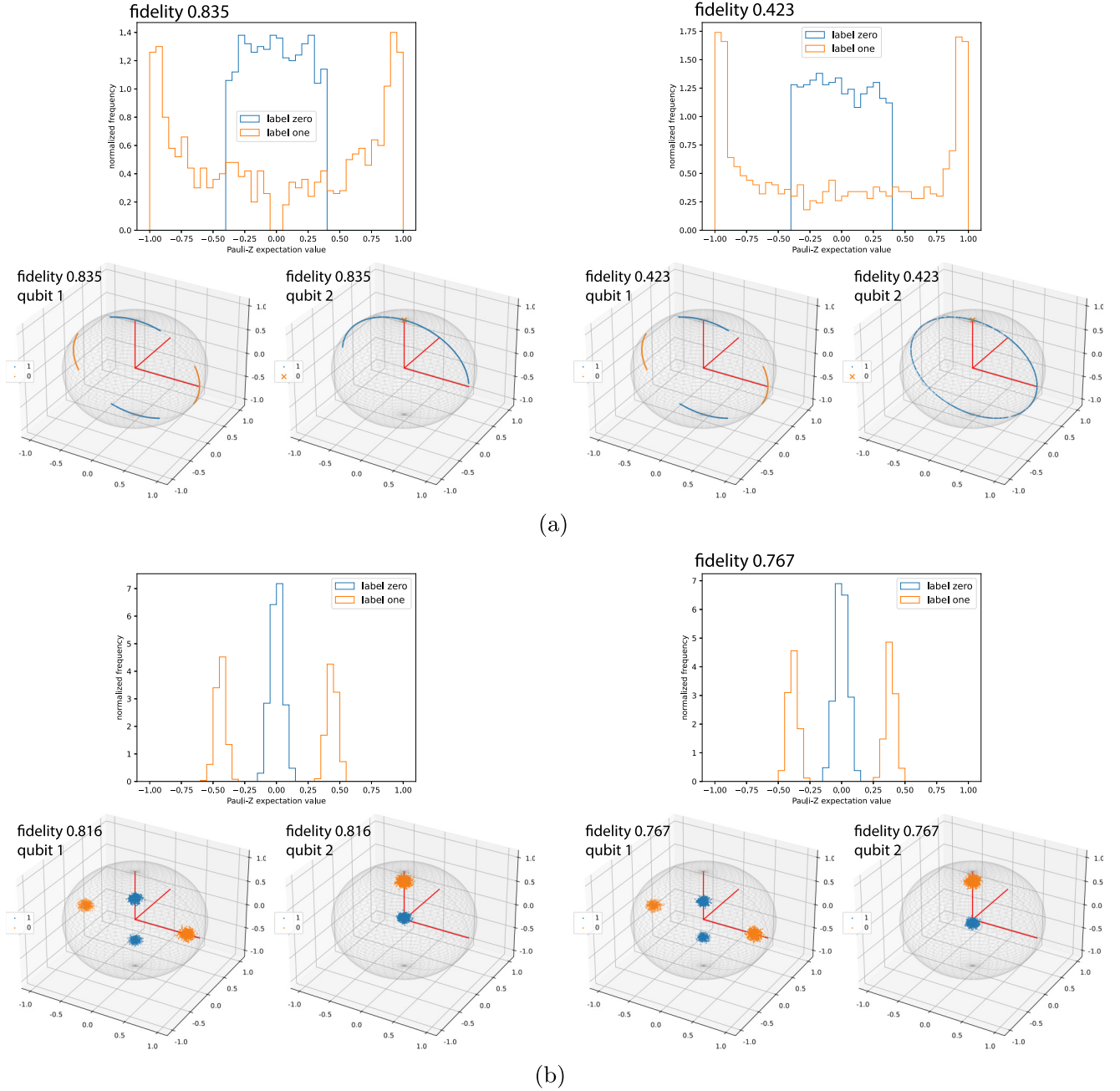


FIG. 7. Features of the synthetic toy dataset. Top in each panel: Pauli-Z expectation values. Bottom in each panel: Quantum states distribution in the Bloch sphere. (a) Features of the pure-state datasets with composite classes fidelity 0.835 and 0.423 and (b) Features of the mixed-state datasets with composite classes fidelity 0.816 and 0.767.

negative class, the ensemble is generated via

$$(1 - e_s)\rho_{\text{ini}} + e_s\rho_{\text{random}}. \quad (53)$$

For the positive class, an additional perturbation is applied, the ensemble is obtained as

$$(1 - e_s - e_{\text{shift}})\rho_{\text{ini}} + e_s\rho_{\text{random}} + e_{\text{shift}} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{pmatrix}. \quad (54)$$

Recall the definition of the composite quantum state $\frac{1}{N_i} \sum_{u \in C_i} \rho_u = \rho_{\text{comp},i}$. For both datasets, we compute one composite quantum state for each class. For instance, C_0 could be the negative class and C_1 the positive class. The composite quantum states of each class are identical when $e_{\text{shift}} = 0$ and gradually diverge as e_{shift} increases. We use fidelity to quantify the distance between the two composite-level density matrices. The Pauli-Z expectation values of the data points from the positive class center around 1 and -1 , while for the negative class it is centered at zero.

Below in Fig. 7, we present the distribution of the data points (their quantum states) in the Bloch sphere, the dis-

tribution of their Pauli-Z expectation values (which is the main classification feature) and the fidelity between the two composite quantum states (each representing one class).

We set the proportion of randomness e_s to a constant and varied e_{shift} to induce a shift between the composite quantum states of class one and class zero. For the pure-state dataset, we set $e_s = 0.4$ and shifted e_{shift} between 1.5 and 4. For the mixed-state dataset, we set $e_s = 0.2$ and shifted e_{shift} between 0.25 and 0.3. Instead of reporting the value of e_{shift} , we reported the fidelity between the composite quantum states of the two classes as a measure of their divergence.

2. Synthetic SNPs dataset

To evaluate the effectiveness of QNN training based solely on composite quantum states, we demonstrate its application in the context of privacy-preserving genomic data analysis. To this end, we simulate a synthetic case-control dataset at SNP markers using *hapgen2*. *Hapgen2* assumes that disease-associated SNPs are in Hardy-Weinberg equilibrium and act independently. To generate phenotypes resulting from interactions between multiple disease-associated SNPs, we first use *hapgen2* to simulate 500 case and 500 control samples for 10000 SNPs on chromosome 1 to obtain an increased probability of risk alleles in the case samples. We selected six disease-associated SNPs based on the GWAS schizophrenia study with accession number GCST002539; their rsID numbers are rs4648845, rs11210892, rs1702294, rs6670165, rs12129573, and rs140505938. We artificially increased the heterozygote-homozygote relative disease risks to (1.5/2.25, 1.2/1.44, 1.6/2.56, 2/4, 1.8/3.24, and 1.3/1.69) to enhance the oversampling of haplotypes carrying risk alleles. We disregard the phenotype output of *hapgen2*, as it only simulates independent disease SNPs. The increased disease risk was used solely to oversample disease-associated haplotypes.

To generate interacting phenotypes, we use *SimulatePhenotypes*. We simulate three phenotype traits based on a two-SNP interaction disease model, referred to as Disease Model 1 in Ref. [37]. In this model, disease odds increase multiplicatively within and between loci, governed by three parameters: baseline disease odds (α), the odds increase for SNP1 (θ_1), and the odds increase for SNP2 (θ_2). The first two SNPs contribute to the first trait with $\alpha = 0.006$, $\theta_1 = 5$, $\theta_2 = 5$. The third and fourth SNPs contribute to the second trait with $\alpha = 0.004$, $\theta_1 = 6$, $\theta_2 = 4$. The last two SNPs contribute to the third trait with $\alpha = 0.003$, $\theta_1 = 7$, $\theta_2 = 7$. These parameters were chosen to generate a high number of cases in the phenotype simulation for genotypes with increased disease-associated haplotypes, while predicting fewer cases in the phenotype simulation for genotypes without increased disease-associated haplotypes. The final genotypes are encoded as vectors containing 0, 1, and 2, and then normalized. The final phenotype is determined by combining the phenotypes from the three traits: If any of the traits is positive, then the data point is considered positive; otherwise, it is classified as negative. We then balance the dataset using NearMiss [38], resulting in 465 disease samples and 465 control samples containing 9993 SNPs. To reduce the dimensionality of the input features, we select a subset of

SNPs with the highest variance across all data points while retaining the disease-associated SNPs.

3. PsychENCODE SNPs dataset

We use a dataset of 355 schizophrenia (SCZ) and 355 control subjects from the PsychENCODE consortium [39], which contains postmortem genomics data from the prefrontal cortex of individuals with psychiatric disorders and matched controls. For each individual, we select all SNPs that were associated with 555 key SCZ genes (containing all transcription factors and a set of high-confidence SCZ genes identified in prior studies) as bulk expression quantitative trait loci, as described in Ref. [39]. This generates a set of 37 215 SNPs, which we further reduce to 256 by selecting those showing the strongest Pearson correlation coefficient with the binary case and control status in the training set. Each subject is therefore represented by a compressed genotype vector of length 256, where each entry may be 0, 1, or 2, indicating the number of copies of the alternative allele contained in the subject's genotype for a specific SNP.

C. Privacy parameter ϵ computation

To calculate the privacy parameter ϵ , we iterate over all individual data points, treating each as ρ^* . For random batches, we randomly select data points from the remainder to construct $\rho_{\text{comp-1}}$ and then add ρ^* to form ρ_{comp} . For smart batches, the data points are those clustered within the same batch as ρ^* . We projecting ρ^* onto $\rho_{\text{comp-1}}$ and ρ_{comp} to obtain $p_{\text{comp-1}}^*$ and p_{comp}^* , respectively.

According to Eqs. (40) and (41), a single ϵ^* can be computed for each ρ^* , and the overall ϵ is defined as the maximum of all ϵ^* values. However, the number of possible batches that can be constructed, and consequently, the number of $p_{\text{comp-1}}^*$ and p_{comp}^* values available to compute the distributions g_1 and g_2 is limited. As a result, especially for larger batch sizes, the estimate of ϵ^* may be unreliable. To address this, we combine $p_{\text{comp-1}}^*$ and p_{comp}^* across all ρ^* to compute aggregate distributions $g_1(p_{\text{comp-1}})$ and $g_2(p_{\text{comp}})$. Specifically, we collect all projection outcomes to form distributions of $p_{\text{comp-1}}$ and p_{comp} . We then fit these distributions with normal probability density functions, obtaining means μ_1 , μ_2 and standard deviations std_1 , std_2 , respectively. The privacy parameter ϵ is then estimated using:

$$\epsilon = \max_p \left\{ \ln \left(\frac{g_2(p) - \delta}{g_1(p)} \right) \right\}, \quad (55)$$

where p are 10 000 samples drawn from $\mathcal{N}(\mu_2, \text{std}_2)$ and $\delta = 0.05$. Values of p that cause division by zero or the logarithm of a negative number are excluded from the calculation.

D. Batching strategies

When constructing the composite quantum states, the individual input data are divided into batches and one composite quantum state is constructed from each batch. We consider two strategies for dividing input data points into batches. Both strategies are class specific, e.g., that data points from each class are batched separately. To maintain a balanced dataset,

we ensure that each class has the same number of batches thus contributes the same number of composite quantum states.

(1) *Random batching.* Data points are randomly divided into nearly equal-sized batches. If the total number of points is not divisible by the desired number of batches, then the first $N_{\text{remainder}}$ batches receive one additional point to account for the remainder. This ensures that batch sizes differ by at most one.

(2) *Smart batching.* Spectral clustering groups similar data points into a target number of batches. The similarity matrix is computed using quantum fidelity between states. This approach often result in uneven batch sizes.

Because density matrices have unit trace, composite states $\frac{1}{N_i} \sum_{u \in C_i} \rho_u$ cannot be renormalized to account for batch size. This effectively reduces the weight of individual points in larger batches. To correct for this, we multiply each composite state's contribution to the loss function by the batch size N_i in the smart batching scheme. For random batching, batch sizes differ by at most one, so this correction was unnecessary.

Smart batching is expected to impact the utility and privacy of the composite states as follows.

Utility. Without nonlinear postprocessing, both batching strategies yield similar utility. With nonlinear operations, smart batching tends to preserve more information and reduces utility loss.

Privacy. Smart batching has mixed effects on privacy. Smaller batches make it easier to infer individual contributions, and restricting individual states to be close to ρ_{comp} reduces the number of possible composition vectors $\mathbf{b}$, weakening privacy. Conversely, clustering similar points together enhances differential privacy in structured datasets, as target states are hidden among similar states.

Overall, these two batching strategies provide different trade-offs between utility and privacy, allowing for flexibility depending on the application's requirements.

E. Code implementation

For benchmarking purposes, we implemented two QNN models with access to individual data points (instance-level loss function), alongside a third QNN model with access to only composite quantum states (composite-level loss function), reflecting the perspective of a remote quantum server.

(1) The unmodified Qiskit implementation of EstimatorQNN + NeuralNetworkClassifier. This model can only accommodate pure states and requires all input vectors to be real valued. Its implementation uses state vectors to represent

quantum states and uses an instance-level loss function. We refer to this model as basic QNN.

(2) The EstimatorQNN and NeuralNetworkClassifier classes from Qiskit are modified to accommodate mixed states. We refer to the combination of these two modified base classes as Mixed_Estimator_Classifier. Its implementation used density matrices to represent quantum states and employs an instance-level loss function. We refer to this model as instance-level QNN.

(3) The modified EstimatorQNN and NeuralNetworkClassifier classes accommodating mixed states (i.e., Mixed_Estimator_Classifier), combined with a composite-level loss function, referred to as composite-level QNN.

We compare basic QNN and instance-level QNN to validate the correctness of our code implementation for Mixed_Estimator_Classifier, while the comparison with composite-level QNN tests the effectiveness of our protocol. As previously noted, optimizers such as COBYLA update parameters based solely on the average loss across all data points. Therefore, when no nonlinear postprocessing, such as activation functions, is applied, all three models are expected to perform nearly identically. When nonlinear elements like a sigmoid activation are introduced, the models' performance should remain highly correlated. Minor differences may still arise due to numerical discrepancies between state-vector-based and density-matrix-based implementations.

In the computational experiments of this work, we use COBYLA optimizer with maxiter 200. We set the maximal number of epochs to 50 with a patience of 10 epochs, unless stated otherwise (i.e., training is terminated if the training score does not improve for 10 consecutive epochs). For the RealAmplitude ansatz, we use a *sca* entanglement structure and repeat the basic block, consisting of a rotation layer followed by an entanglement layer, multiple times. This is referred to as the number of layers in the QNN circuit, we treated it as a hyperparameter and selected the best value for it for each dataset.

ACKNOWLEDGMENTS

This work was supported by the National Institutes of Health under Grant No. R01HG010749. We acknowledge support from the Albert L. Williams Professorship funds.

DATA AVAILABILITY

All code and data supporting the findings of this article are available [40].

- [1] D. Ristè, M. P. da Silva, C. A. Ryan, A. W. Cross, A. D. Córcoles, J. A. Smolin, J. M. Gambetta, J. M. Chow, and B. R. Johnson, Demonstration of quantum advantage in machine learning, *npj Quantum Inf.* **3**, 16 (2017).
- [2] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, Quantum machine learning: A classical perspective, *Proc. R. Soc. A* **474**, 20170551 (2018).

- [3] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, Power of data in quantum machine learning, *Nat. Commun.* **12**, 2631 (2021).
- [4] H.-Y. Huang, R. Kueng, and J. Preskill, Information-theoretic bounds on quantum advantage in machine learning, *Phys. Rev. Lett.* **126**, 190505 (2021).
- [5] H. Yamasaki, N. Isogai, and M. Murao, Advantage of quantum machine learning from general computational advantages *arXiv:2312.03057*.

- [6] M. Cerezo, G. Verdon, H.-Y. Huang, L. Cincio, and P. J. Coles, Challenges and opportunities in quantum machine learning, *Nat. Comput. Sci.* **2**, 567 (2022).
- [7] A. Broadbent, J. Fitzsimons, and E. Kashefi, in *2009 50th Annual IEEE Symposium on Foundations of Computer Science* (IEEE, Los Alamitos, CA, 2009), pp. 517–526.
- [8] J. F. Fitzsimons, Private quantum computation: An introduction to blind quantum computing and related protocols, *npj Quantum Inf.* **3**, 23 (2017).
- [9] W. Li, S. Lu, and D.-L. Deng, Quantum federated learning through blind quantum computing, *Sci. Chin. Phys. Mech. Astron.* **64**, 100312 (2021).
- [10] B. Polacchi, D. Leichtle, L. Limongi, G. Carvacho, G. Milani, N. Spagnolo, M. Kaplan, F. Sciarrino, and E. Kashefi, Multi-client distributed blind quantum computation with the Qline architecture, *Nat. Commun.* **14**, 7743 (2023).
- [11] C. Li, N. Kumar, Z. Song, S. Chakrabarti, and M. Pistoia, Privacy-preserving quantum federated learning via gradient hiding, *Quantum Sci. Technol.* **9**, 035028 (2024).
- [12] E. G. Beltrametti and G. Cassinelli, On the non-unique decomposability of quantum mixtures, in *Current Issues in Quantum Logic*, edited by E. G. Beltrametti and B. C. van Fraassen (Springer US, Boston, MA, 1981), pp. 455–464.
- [13] C. Dwork and A. Roth, The algorithmic foundations of differential privacy, *Found. Trends Theor. Comput. Sci.* **9**, 211 (2014).
- [14] L. Zhou and M. Ying, in *2017 IEEE 30th Computer Security Foundations Symposium (CSF)* (IEEE, Los Alamitos, CA, 2017), pp. 249–262.
- [15] C. Hirche, C. Rouzé, and D. S. França, Quantum differential privacy: An information theory perspective, *IEEE Trans. Inf. Theory* **69**, 5771 (2023).
- [16] M. Kleinmann, H. Kampermann, T. Meyer, and D. Bruß, Physical purification of quantum states, *Phys. Rev. A* **73**, 062309 (2006).
- [17] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond* (The MIT Press, Cambridge, MA, 2001).
- [18] W. S. Noble, What is a support vector machine? *Nat. Biotechnol.* **24**, 1565 (2006).
- [19] M. Schuld and N. Killoran, Quantum machine learning in feature Hilbert spaces, *Phys. Rev. Lett.* **122**, 040504 (2019).
- [20] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, Quantum embeddings for machine learning, [arXiv:2001.03622](https://arxiv.org/abs/2001.03622).
- [21] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig, Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays, *PLoS Genet.* **4**, e1000167 (2008).
- [22] P. S. Emani, M. N. Geradi, G. Gürsoy, M. R. Grasty, A. Miranker, and M. B. Gerstein, Assessing and mitigating privacy risks of sparse, noisy genotypes by local alignment to haplotype databases, *Genome Res.* **33**, 2156 (2023).
- [23] E. Farhi and H. Neven, Classification with quantum neural networks on near term processors, [arXiv:1802.06002](https://arxiv.org/abs/1802.06002).
- [24] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Circuit-centric quantum classifiers, *Phys. Rev. A* **101**, 032308 (2020).
- [25] M. Backes, P. Berrang, M. Humbert, and P. Manoharan, in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)* (Association for Computing Machinery, New York, 2016), pp. 319–330.
- [26] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, Membership inference attacks against machine learning models, in *2017 IEEE Symposium on Security and Privacy (SP)* (IEEE, New York, 2017), pp. 3–18.
- [27] A. Bittau, Ú. Erlingsson, P. Maniatis, I. Mironov, A. Raghunathan, D. Lie, M. Rudominer, U. Kode, J. Tinnés, and B. Seefeld, in *Proceedings of the 26th Symposium on Operating Systems Principles (SOSP '17)* (ACM, New York, 2017), pp. 441–459.
- [28] S. Y.-C. Chen and S. Yoo, Federated quantum machine learning, *Entropy* **23**, 460 (2021).
- [29] C. Ren, R. Yan, H. Zhu, H. Yu, M. Xu, Y. Shen, Y. Xu, M. Xiao, Z. Y. Dong, M. Skoglund, D. Niyato, and L. C. Kwek, Toward quantum federated learning, *IEEE Trans. Neural Netw. Learn. Syst.* **36**, 15580 (2025).
- [30] C. Qiao, M. Li, Y. Liu, and Z. Tian, Transitioning from federated learning to quantum federated learning in internet of things: A comprehensive survey, *IEEE Commun. Surv. Tutor.* **27**, 509 (2025).
- [31] M. Chehimi, S. Y.-C. Chen, W. Saad, D. Towsley, and M. Debbah, Foundations of quantum federated learning over classical and quantum networks, *IEEE Network* **38**, 124 (2024).
- [32] L. Zhu, Z. Liu, and S. Han, Deep Leakage from Gradients, in *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019), Vol. 32.
- [33] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. Nitin Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. El Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, *et al.*, Advances and open problems in federated learning, *Found. Trends Mach. Learn.* **14**, 1 (2021).
- [34] S. W. Choi and P. F. O'Reilly, PRSice-2: Polygenic risk score software for biobank-scale data, *GigaScience* **8**, (2019).
- [35] J. M. Abowd and M. B. Hawes, Confidentiality protection in the 2020 US Census of Population and Housing, *Annu. Rev. Stat. Appl.* **10**, 119 (2023).
- [36] T. Dick, C. Dwork, M. Kearns, T. Liu, A. Roth, G. Vietri, and Z. S. Wu, Confidence-ranked reconstruction of census microdata from published statistics, *Proc. Natl. Acad. Sci. USA* **120**, e2218605120 (2023).
- [37] J. Marchini, P. Donnelly, and L. R. Cardon, Genome-wide strategies for detecting multiple loci that influence complex diseases, *Nat. Genet.* **37**, 413 (2005).
- [38] I. Mani and I. Zhang, kNN approach to unbalanced data distributions: a case study involving information extraction, in *Proceedings of Workshop on Learning from Imbalanced Datasets* (ICML United States, Washington, DC, USA, 2003), Vol. 126, pp. 1–7.
- [39] P. S. Emani, J. J. Liu, D. Clarke, M. Jensen, J. Warrell, C. Gupta, R. Meng, C. Y. Lee, S. Xu, C. Dursun, *et al.*, Single-cell genomics and regulatory networks for 388 human brains, *Science* **384**, eadi5199 (2024).
- [40] https://github.com/gersteinlab/composite_QNN.