

Quantum-Secure Multiparty Deep Learning

Kfir Sulimany^{1,*}, Sri Krishna Vadlamani,¹ Ryan Hamerly^{1,2}, Prahlad Iyengar,^{1,2} and Dirk Englund¹

¹*Research Laboratory of Electronics, Massachusetts Institute of Technology,
Cambridge, Massachusetts, USA*

²*Physics and Informatics Laboratories, NTT Research, Inc., Sunnyvale, California, USA*



(Received 10 October 2024; revised 9 August 2025; accepted 31 October 2025; published 24 December 2025)

Secure multiparty computation enables the joint evaluation of multivariate functions across distributed users while ensuring the privacy of their local inputs. This field has become urgent due to the demand for computationally intensive deep learning inference. These computations are typically offloaded to cloud servers, leading to vulnerabilities. To solve this problem, we introduce a linear algebra engine that leverages the quantum nature of light for information-theoretically secure multiparty inference using telecommunication components. We apply this linear algebra engine to deep learning and derive rigorous upper bounds on the information leakage of both the deep neural network weights and the client's data, enabling double-blind operations. Applied to the modified National Institute of Standards and Technology classification task, we obtain test accuracies exceeding 95% while guaranteeing leakage of less than 0.1 bits per weight and data element. This leakage is an order of magnitude below the minimum bit precision required for accurate deep learning using state-of-the-art quantization techniques. Our work lays the foundation for practical quantum-secure computation and unlocks secure cloud deep learning as a field.

DOI: [10.1103/k8wg-qmbh](https://doi.org/10.1103/k8wg-qmbh)

Subject Areas: Optics, Quantum Physics,
Quantum Information

I. INTRODUCTION

Although deep learning has revolutionized multiple domains [1–3], its applications are constrained by the increasing computational demands on the hardware [4–6]. Given the high energy consumption required for state-of-the-art deep neural network (DNN) inference [7–9], it is common to delegate inference workloads from the edge to centralized server clusters. Unfortunately, this paradigm introduces vulnerabilities that compromise data security, which is crucial in applications such as business, finance, and health care [4]. This situation is reminiscent of the central challenge in secure computation, where multiple parties perform a joint evaluation of multivariate functions across distributed resources while preserving the privacy of their inputs [10] (Fig. 1).

Modern secure computation schemes are built on homomorphic encryption, which allows universal computation on encrypted data and preserves the security of the input and output of the computation [11]. Recent developments have adapted state-of-the-art homomorphic encryption

schemes for secure machine learning [12], but real-world applications are limited due to their massive computational overhead [6] and recently discovered security vulnerabilities [13]. Moreover, these encryption schemes typically depend on computational assumptions and are not information-theoretically secure.

Information-theoretically secure computation has been extensively studied through the use of quantum primitives [14], including superposition and entanglement, and has encompassed various protocols such as bit commitment [15,16], coin flipping [17,18], two-party function evaluation and oblivious transfer [19–24], multiparty quantum computation [25–29], and blind quantum computing [30–37]. Despite these significant advancements, their practical applicability for real-world computational tasks, particularly in the domain of deep learning, remains elusive due to strict requirements on the quantum hardware.

In this work, we introduce a linear algebra engine for secure computation supported by quantum information theorems. We apply our engine to multiparty deep learning and derive a rigorous security analysis of the information leakage for all parties. Applied to the modified National Institute of Standards and Technology (MNIST) classification task, our hardware simulations obtain test accuracies of more than 95% while leaking less than 0.1 bits per weight and data element. This weight leakage is an order of magnitude below the state-of-the-art minimum bit precision required for deep learning [38].

*Contact author: kfir@mit.edu

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

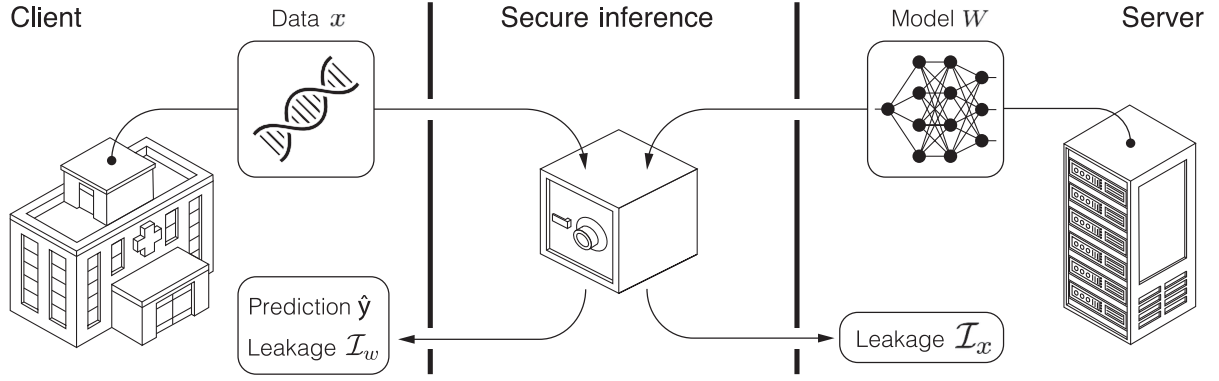


FIG. 1. Secure inference. Secure multiparty computation enables joint evaluation of a function across distributed users while ensuring the privacy of the local inputs W and x . In multiparty deep learning inference, a client such as a hospital wishes to classify a patient's confidential data x using a confidential deep learning model W belonging to the server. The client receives the prediction $\hat{y}$ and the weight leakage $\mathcal{I}_w$, whereas the server receives the data leakage $\mathcal{I}_x$.

Our protocol relies on a photonic computation architecture to realize efficient optical matrix-vector multiplication [39–41]. State-of-the-art optical DNNs have been demonstrated using integrated photonics [42–51], free-space optics [52–59], and fiber optics [60,61]. Recently, a delocalized optical DNN using low-power devices was introduced and demonstrated [62,63]. This protocol, which operates in an edge computing setting [5] where the DNN weights are streamed from a central server to clients that perform inference, shields client data from exposure but reveals the weights to the client.

To address this asymmetry, the client in our protocol performs the inference and then returns the residual light to the server as a certificate of the security of the weights. Leveraging the quantum nature of light, we show that the server can compute an upper bound on their weight leakage using the Holevo bound [64] and that the client can upper bound their data leakage using the Cramér-Rao bound [65]. This approach enables double-blind operations, which form the cornerstone of our quantum-secure multiparty deep learning protocol.

II. SECURE MULTIPARTY DEEP LEARNING

In this section, we formalize the secure inference problem. Consider a K -class classification task with training and tests sets: $(x_i, y_i)_{i=1}^n$ and $(x_i, y_i)_{i=n+1}^{n+m}$, where $x_i \in \mathbb{R}^M$ and labels $y_i \in \{1, \dots, K\}$. The server holds a deep neural network specified by weight matrices $W^{(\ell)} \in \mathbb{R}^{M_\ell \times M_{\ell-1}}$, $\ell = 1, \dots, L$.

For an input x_i , the layerwise activations are

$$x_i^{(0)} = x_i, \quad x_i^{(\ell)} = \sigma(W^{(\ell)} x_i^{(\ell-1)}), \quad x_i^{(L)} = W^{(L)} x_i^{(L-1)},$$

where $\sigma: \mathbb{R} \rightarrow \mathbb{R}$ is a public nonlinearity [e.g., rectified linear unit (ReLU) applied coordinatewise]. The predicted class for x_i is

$$\hat{y}(x_i) = \arg \max_{c \in \{1, \dots, K\}} [x_i^{(L)}]_c.$$

We measure performance by the classification accuracy on the test set:

$$\text{Acc} = \frac{1}{m} \sum_{i=n+1}^{n+m} \mathbf{1}\{\hat{y}(x_i) = y_i\},$$

where $\mathbf{1}\{\cdot\}$ is the indicator function.

Suppose that we require the system to obey strict leakage upper bounds on the information of both the client (data) and the server (weights). Define $\mathcal{I}_w$ as the information leakage per weight element, averaged over all the elements of W , and $\mathcal{I}_x$ as the information leakage per data element, averaged over all the elements of x . The reasoning behind these definitions is to provide a convenient comparison with the bit precision required to achieve high classification accuracy in standard deep learning models [38].

In this setting, we ask, and answer, the following question: *What classification accuracy is achievable as a function of the information leakages $\text{Acc}(\mathcal{I}_w, \mathcal{I}_x)$?*

Figure 1 depicts a typical scenario: A client (e.g., a hospital) holds sensitive data x , and a server holds a confidential DNN. Only the client learns the prediction; the server observes no output. Our method (next section) additionally returns a return state to the server to certify security, without affecting the inference outcome.

III. COHERENT LINEAR ALGEBRA ENGINE

Our protocol implements the neural network of the previous section in a layer-by-layer fashion using a coherent optical linear algebra engine utilized by the two parties, the server and the client, in a network. To implement the k th layer, the server holds matrix $W^{(k)}$ of dimensions $M_k \times M_{k-1}$ while the client holds the activation vector $\sigma(\vec{x}^{(k-1)})$ of length M_{k-1} . For ease of exposition in

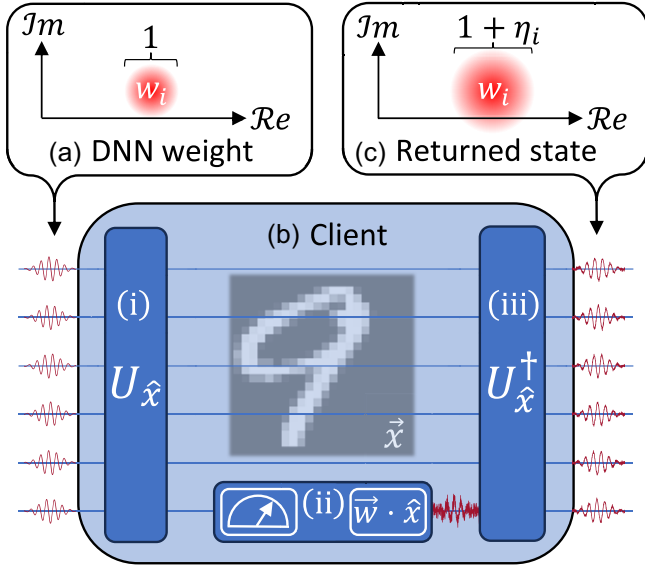


FIG. 2. Coherent linear algebra engine. (a) DNN weights are encoded into the complex amplitudes of coherent states w_i (of variance 1SNU). (b) The client uses these weights for inference with their local data $\vec{x}$ and transmits the residual light, called the return state, back to the server. The client calculates the inner product $\vec{w} \cdot \vec{x}$ using (i) unitary transformation $U_{\hat{x}}$, (ii) measurement and feedforward of the complex amplitude $\vec{w} \cdot \hat{x}$, which adds noise (marked in red) to the last mode, and (iii) the unitary $U_{\hat{x}}^\dagger$. After applying $U_{\hat{x}}^\dagger$, the excess noise from the measure and feedforward step is spread over the output modes (right). (c) The return state has the same mean as the original DNN weight w_i but a higher variance of $(1 + \eta_i)$ SNU, where η_i is the excess noise.

the following, we shall use notation that is independent of layer k , since the procedure is the same for all k . We shall denote M_k by M , M_{k-1} by N , $W^{(k)}$ by W , and $\sigma(\vec{x}^{(k-1)})$ by $\vec{x}$.

The matrix-vector product $W\vec{x}$ is computed through M inner products $W_i \cdot \vec{x}$ for $i \in \{1, \dots, M\}$, each of which involves three steps.

- (1) The server transmits a weight vector W_i of length N by encoding the components into the complex amplitudes of N coherent states denoted jointly by $\vec{w}$ (see Appendix B). The amplitudes of these physical states, w_j , encode the numerical weights as $w_j = \sqrt{\mu} W_{ij} / \|W\|$, where $\|W\|$ denotes the root mean square of the elements in W . This encoding ensures that the average photon number is μ , a value chosen by the server. These coherent states have a variance of 1 shot noise unit (SNU) in each quadrature, as illustrated in Fig. 2(a).
- (2) The client operates on these amplitudes as follows [Fig. 2(b)].
 - (i) The client calculates the inner product of the incoming weight vector $\vec{w}$ with its local input data vector $\vec{x}$ using a unitary transformation $U_{\hat{x}}$ that is designed to yield $\vec{w} \cdot \hat{x}$ into one mode of $U_{\hat{x}}\vec{w}$, where $\hat{x}$ is obtained by the ℓ^2

normalization of $\vec{x}$. We call this mode the “result” mode.

- (ii) The client diverts the result mode and measures both quadratures to obtain $\vec{w} \cdot \hat{x}$. Then, the client performs a feedforward protocol by reinjecting light with the same quadratures back into the result mode. The feedforwarded state preserves the measured state’s expectation value $\vec{w} \cdot \hat{x}$ but, due to quantum noise, consists of additional Gaussian noise of variance 1 SNU. The output of this measure and feedforward step is labeled $\mathcal{M}(U_{\hat{x}}\vec{w})$.
 - (iii) The client finally applies the unitary transformation $U_{\hat{x}}^\dagger$ on $\mathcal{M}(U_{\hat{x}}\vec{w})$ and returns $\rho_v = U_{\hat{x}}^\dagger \mathcal{M}(U_{\hat{x}}\vec{w})$ as a return state to the server for security checks. By the definition of $\mathcal{M}(U_{\hat{x}}\vec{w})$ from the previous point, the first moment of $U_{\hat{x}}^\dagger \mathcal{M}(U_{\hat{x}}\vec{w})$ is $\vec{w}$, while the variance of each mode i has increased by η_i ; see Fig. 2(c).
- (3) Finally, the server measures the variance of each mode i of the return state in both quadratures, $(1 + \eta_i)$ SNU, and calculates an upper bound for the leakage of each weight w_i to the client. Note that, before this security check, the return state can be reused by other clients to perform their own inference as discussed in Sec. VI.

The measure and feedforward operation $\mathcal{M}$ outlined above can be generalized to an operation $\mathcal{G}$ that consists of a combination of phase-insensitive amplification G and a beam splitter with a split ratio $1 - (1/G) : (1/G)$; the former approach is obtained from the latter in the limit $G \gg 1$ (see Appendix A) [66]. In this general case, the client applies a gain G to the component $\vec{w} \cdot \hat{x}$, passes it through a beam splitter with the above splitting ratio, and uses the output of the $1 - (1/G)$ port for its inner product computation readout while reinjecting the output of the $1/G$ port back into the result mode that originally carried $\vec{w} \cdot \hat{x}$. Note that, in order to obtain the desired inner product, the client scales the result up digitally by $[(\|\vec{x}\|)/(\sqrt{G-1})](\|W\|/\sqrt{\mu})$. However, for the purpose of classification tasks, the scaling step for the client is superfluous (see Appendix B).

The gain G contributes amplification noise to the result mode. We calculate the variance of this mode in Appendix A. This variance is larger than the standard shot noise by a quantity we refer to as the “excess noise.” This excess noise is then spread via the operation of $U_{\hat{x}}^\dagger$ in the return states modes, such that the excess noise in the i th mode of the return state, η_i , is weighted by $|\hat{x}_i|^2$:

$$\eta_i = \left(2 - \frac{2}{G}\right) |\hat{x}_i|^2, \quad (1)$$

where $\hat{x}_i$ is the i th element of the normalized data vector $\hat{x}$.

The signal-to-noise ratio (SNR) in the client's measurement after applying the amplification and beam splitting is $\text{SNR} = \{[G(G-1)]/(2G^2 - 3G + 2)\}|\vec{w} \cdot \hat{x}|^2$ (see Appendix A). Since $|\vec{w}|^2$ is proportional to the average photon occupation number μ , the SNR is proportional to μ .

The gain G , controlled by the client, and the average weight occupation μ , controlled by the server, are the only hardware configuration parameters in the system. These parameters dictate the return state excess noise, Eq. (1), and the SNR of the inner product measurement. In the large gain limit $G \gg 1$, the excess noise η_i and the SNR reduce to $2\hat{x}_i^2$ and $\frac{1}{2}|\vec{w} \cdot \hat{x}|^2$, respectively. This result aligns with the measure and feedforward special case. In the small gain limit $G \approx 1$, the excess noise η_i and the SNR reduce to $2(G-1)\hat{x}_i^2$ and $(G-1)|\vec{w} \cdot \hat{x}|^2$, respectively. Reducing the gain disturbs the return state less but also reduces the SNR of the inner product obtained by the client. Therefore, the client's actions can be described by the theory of weak measurements [67–70]. More details of the amplification-and-split procedure and the hardware implementation of the protocol using standard telecommunication components are presented in Appendix A.

IV. CLASSIFICATION ACCURACY

In this section, we calculate the classification accuracy of a secure neural network that uses our coherent linear algebra engine on the standard MNIST classification task. To this end, we first trained a digital noiseless neural network on the MNIST dataset and obtained a test classification accuracy of 98%. Then, we fed the trained weights into a PYTORCH model of our optical architecture to evaluate the test accuracy of our secure optical neural network; see Appendix B for more details. The complete code used in this work is available online [71].

We calculate the classification accuracy as a function of the average photon number occupation μ and the amplification gain G ; see Appendix B. We obtain $> 95\%$ test accuracy with an average photon number of less than $\mu = 6$ and amplification gain of $G = 1.3$. Here, we use $\mu = 6$ to meet high-accuracy targets; tighter bounds or higher-SNR receivers (e.g., squeezed probes or Kennedy receiver) would allow lower μ at the same accuracy.

In contrast to previous studies in optical machine learning [55,63], we do not measure all of the light sent to the client but only the portion corresponding to $|\vec{w} \cdot \hat{x}|^2$ and yet obtain similar high accuracies. This fundamentally follows from the fact that the SNR in the inner product is similar whether one optically routes the inner product amplitude to one mode and measures it versus measuring all the modes and calculating the inner product digitally.

V. SECURITY ANALYSIS

In Sec. VA, we analyze the security of the server's information, assuming an honest server and a malicious

client. In Sec. VB, we analyze the security of the client's data, assuming an honest client and a malicious server. These two results allow us to analyze the classification accuracy as a function of the information leakage of each party $\text{Acc}(\mathcal{I}_w, \mathcal{I}_x)$ in Sec. VI.

A. Weight leakage

In this section, we use the return state ρ_v returned to the server by the client to calculate an upper bound on the amount of information about $\vec{w}$, in number of bits, that could be learned by a malicious client. A malicious client tries to learn the weights using arbitrary operations other than the operations defined by our protocol.

Here, we assume that dishonest operations are independent and identically distributed over all the incoming modes of $\vec{w}$. Our security analysis for the weight leakage relies on results for quantum-secure communication [72–77] and proceeds through the Holevo theorem [64].

We reduce the weight leakage analysis in the secure computation problem to key information leakage analysis in secure communication by mapping our two computational parties, namely, the server and the client, to the three communication parties, namely, Alice (transmitter), Bob (reviewer), and Eve (eavesdropper). Alice and Bob are both on the server side, while Eve represents the client. Alice is the part of the server that sends the weights. The client is either honestly using the transmitted information (the weights) to perform some computations or maliciously trying to extract extra information from them, so, from the perspective of the server, the client acts like an eavesdropper, Eve. Bob is the part of the server that receives the return state to check if Eve obtained more information than they should have in an honest inner-product computation.

In secure communication, Bob checks for any leakage of the message sent by Alice, while in our secure computation protocol, Bob checks if the leakage exceeds what is to be expected from an honest inner-product computation. Any malicious client deviating from the protocol will be immediately detected by the Bob part of the server, and the computation will be aborted immediately. For example, if the client measures the weights, they will be unable to provide a low-noise return state, and, consequently, the client's dishonesty will be immediately detected by the server. See Appendix C 1 for the complete analysis.

We present here the final mathematical result for the weight leakage. Representing the weight occupation by μ and defining the quantities

$$\begin{aligned} 2a &:= 2\mu + 1, & b &:= 2\mu + 1 + \eta_i \\ c &:= \sqrt{4\mu^2 + 2\mu + 1}, & z &:= \sqrt{(a+b)^2 - 4c^2}, \end{aligned}$$

the weight leakage $\mathcal{I}_w$ is bounded by the Holevo theorem:

$$\mathcal{I}_w \leq \frac{1}{2}[g(\nu_1) + g(\nu_2) - g(\nu_3)] \text{ (bits)}, \quad (2)$$

where

$$g(\nu) = \left(\frac{\nu+1}{2}\right) \log_2 \left(\frac{\nu+1}{2}\right) - \left(\frac{\nu-1}{2}\right) \log_2 \left(\frac{\nu-1}{2}\right),$$

$$\nu_{1,2} = \frac{1}{2}(z \pm [b-a]), \quad \nu_3 = b - \frac{c^2}{a+1}$$

This inequality is shown to be tight by analyzing the entangling cloner attack [74]. The derivation of Eq. (2), and more details, can be found in Appendix C 1.

The leakage is calculated for one query, that is, a single exposure of the client to the weights. To control information leakage after multiple uses of the model, the server sends the client affine transformations versions of the DNN model, which consist of different individual weights but implement the same model function (therefore, conserving the inference output); see Appendix C 1 a. Intuitively, this method associates with each query a “one-time pad” that was randomly selected from all of the transformations under which the function is invariant. Because of this invariance, the client can run the computation without knowing the one-time pad and will never obtain the same model twice.

B. Data leakage

To calculate the data leakage, we consider an honest client and a dishonest server. In this setting, the server sends a weight matrix $\mathbf{W}$ of shape $M \times N$ to the client and aims to evaluate the client’s data vector $\vec{x}$ of length N using M measurements of the M return states.

We recall from Sec. III that the mode i of the return state $U_{\vec{x}}^\dagger \mathcal{G}(U_{\vec{x}} \vec{w})$ has the mean w_i and variance $(1 + \eta_i \text{ SNU})$ [Eq. (1)]. In other words, information about the client’s data $\vec{x}$ is leaked to the server not through the mean of the return state but through its variance. The client may control this leakage by reducing the gain G in its amplification-and-splitting step. We also note that, since the client is honest, it announces the true gain G to the server.

We phrase dishonest attempts of the server to learn $\vec{x}$ through M measurements as the following concrete statistical estimation problem [78]: Let ρ_v , the state sent back to the server by the client, be a multimode Gaussian state with mean $\vec{w}$ and a covariance matrix whose i th diagonal element is $\sigma_i^2 = 1 + \eta_i = 1 + [2 - (2/G)] \cdot |\hat{x}_i|^2$. What is the amount of information, in terms of bits, that the server can obtain about $\vec{x}$ from M independent measurements of ρ_v , given that it knows G ?

We answer this question by using the Cramér-Rao inequality [65] to compute a lower bound on the variance

$\sigma_{\hat{x}_i}$ of the server’s estimator $\hat{X}_i$ of the true $\hat{x}_i$: $\sigma_{\hat{x}_i} \geq \{1/[M \cdot I(\hat{x}_i)]\}$, where $I(\hat{x}_i)$ is the Fisher information of the true normalized data element $\hat{x}_i$, i.e., individual attacks.

Next, we assume that the server’s estimator $\hat{X}_i$ is a Gaussian random variable with mean $\hat{x}_i$ and variance given the above lower bound and use the formula for the communication capacity of a Gaussian channel to calculate the amount of mutual information $\mathcal{I}_x$, in number of bits, between the true unnormalized x_i and the estimator $\hat{X}_i$. By the data processing inequality, the information $\mathcal{I}_x$ leaked about x_i is upper bounded by the information $\mathcal{I}_{\hat{x}}$ leaked about $\hat{x}_i$. We present the derivation in Appendix C 2 a.

Next, we consider collective attacks, where the server measures all M return states jointly. Writing $\gamma := 2 - 2/G$ so that the covariance reads $\Sigma_x = I + \gamma x x^\dagger$, the Sherman-Morrison inversion and the quantum-Fisher formula give $F(x) = \gamma^2 (1 + \gamma)^{-1} (I_N - x x^\dagger)$. Because the quantum-Fisher information (QFI) is additive, the total information scales linearly with M , and the quantum Cramér-Rao bound sets the corresponding lower limit on the variance of any unbiased estimator of $\vec{x}$. This bound is grater or equal to the Cramér-Rao bound for individual attacks; see Appendix C 2 b for the full derivation.

In the high-dimensional (asymptotic) regime, this Fisher information can be used to evaluate the mutual information between the client’s input and the return state received by the server analytically: The entropies of both the uniform prior on the sphere and the Bayesian posterior are approximated by the entropies of Gaussians with the corresponding covariances. This yields the leakage bound (see Appendix C 2 b)

$$\mathcal{I}_x \leq \frac{1}{2} \log \left(1 + \frac{M_t \gamma^2}{N(1+\gamma)} \right) \text{ (bits)}, \quad \gamma = 2 - \frac{2}{G}. \quad (3)$$

where M_t is the total number of return states obtained during the inference. Finite-block corrections vanish with increasing dimensions, as in finite-key quantum key distribution (QKD) analyses, so an eavesdropper, even when using optimal collective measurements, can gain only a bounded amount of information about the direction.

In hidden layers, the return state is a server-controlled Gaussian that depends on the activation direction via a unit-norm map $f(x)$. A dishonest server could amplify leakage by engineering large derivatives of f . We derive a closed-form quantum Fisher information that quantifies this attack (Appendix C 2 c) and show that the client can counter it by dithering—tiny perturbations around x ; comparing the outputs reveals any oversized derivative, and the client aborts the protocol if a preset threshold is crossed, thereby keeping the client’s data leakage bounded.

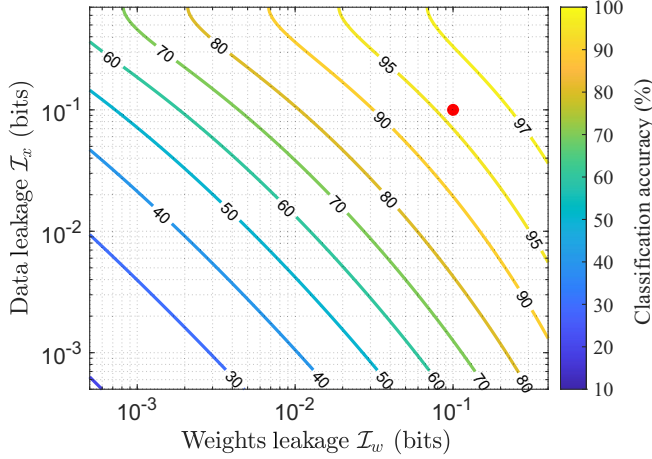


FIG. 3. Classification accuracy versus weight and data leakage. The classification accuracy of the secure optical neural network, which uses our coherent linear algebra engine (Fig. 2), is numerically calculated as a function of the average photon occupation and the amplification gain. We also use these parameters to upper bound the weight leakage $\mathcal{I}_w$ [Eq. (2)] and the data leakage $\mathcal{I}_x$ [Eq. (3)]. The classification accuracy increases with both leakages and achieves the digital noiseless accuracy. For small data leakages, the weight leakage is inversely proportional to the data leakage for any given fixed classification accuracy. Our protocol achieves a classification accuracy of $> 95\%$ with weight leakage of $\mathcal{I}_w = 0.1$ bits and data leakage of $\mathcal{I}_x = 0.1$ bits, (red marker).

VI. SECURE CLASSIFICATION

Now that we have a mathematical handle on both the weight [Eq. (2)] and the data leakage [Eq. (3)], we can ask how they trade off against each other if one requires a constant test accuracy. For this purpose, we revisit the classification accuracy that we computed numerically as a function of the two hardware configuration parameters of the system, the server average photon occupation and the client gain G (see Fig. 7 in Appendix B). Recalling that the weight leakage and the data leakage are both functions of the hardware configuration parameters, we perform a change of variables and replace the axes in Fig. 7 with weight and data leakage to obtain the trade-off result in Fig. 3. Note that the data leakage is calculated for the quantum adversary capable of correlated measurement.

The figure depicts a clear trade-off of the data leakage and weight leakage in order to maintain classification accuracy. This is because, when the server sends less energy mode in order to limit weight leakage, the client has to increase the gain to preserve accuracy of the computation, thereby exposing more of its data. The data leakage increases monotonically with the gain and saturates, achieving the measure and feedforward leakage. Therefore, as the server laser power is reduced and the client gain is increased, one moves from right to left on the classification accuracy isocontours in Fig. 3. For example,

the red marker above the 95% classification accuracy isocontour indicates that a classification accuracy of $> 95\%$ is achieved with weight leakage of $\mathcal{I}_w = 0.1$ bits and data leakage of $\mathcal{I}_x = 0.1$ bits.

Low-precision inference is extensively studied in academia and industry, with research indicating that reliable inference typically requires at least 8 bits per weight or as few as 1 bit using state-of-the-art quantization techniques [79]. This requirement is an order of magnitude higher than the upper bound on weight leakage for our protocol. Furthermore, assuming that a dishonest client does not have access to the training dataset, they would be unable to use the leaked information to infer the rest of the model through training.

The optical loss present between the different parties in real-life communication networks affects the leakage calculations we present above. We analyze the effect of loss on security in Appendix C 1 and show that weight leakage increases with the channel loss, while data leakage is independent of the channel loss. To conserve the classification accuracy, one can adjust the photon occupation number; the loss-adjusted weight leakage and data leakage are presented in Fig. 4(a) as a function of the round-trip channel loss. Here, we chose an amplification gain of $G = 1.3$ to obtain an MNIST classification accuracy of 95%. For local area networks or metropolitan area networks with fiber losses of up to 3 dB [80], the model leakage is below 2 bits per weight.

Since the number of neurons per layer of state-of-the-art DNNs is much larger than the size presented in our study, we present the dependence of the data and weight leakages on the number of neurons per layer in Fig. 4(b). We chose an average photon number of $\mu = 6$ and an amplification gain of $G = 1.3$ to obtain an MNIST classification accuracy of 95%. Interestingly, the leakage vanishes with the number of neurons per layer. This is because the amplification excess noise is spread over more modes in the return state, leading to less weight and data leakages. Therefore, advanced DNNs, which are much larger than the model used in this work, are expected to suffer even smaller information leakage.

Next, we analyzed the impact of excess technical noise (see Appendix C 1). We assume the same technical noise for both the server and the client, and we adjust the server power to compensate for the clients noise in order to conserve the classification accuracy. We plotted the information leakage as a function of loss and technical excess noise for values ranging from 0 to 0.2 SNU. Laudenbach *et al.* [74] derived a model for excess noise considering multiple additive sources, including electronic noise, limited detection efficiency, imperfections in signal preparation, and Raman scattering. Their findings indicate that realistic excess noise values are around 2% SNU. We highlight a horizontal line at 2% SNU in Fig. 4(c), representing realistic noise in real-world scenarios. At this

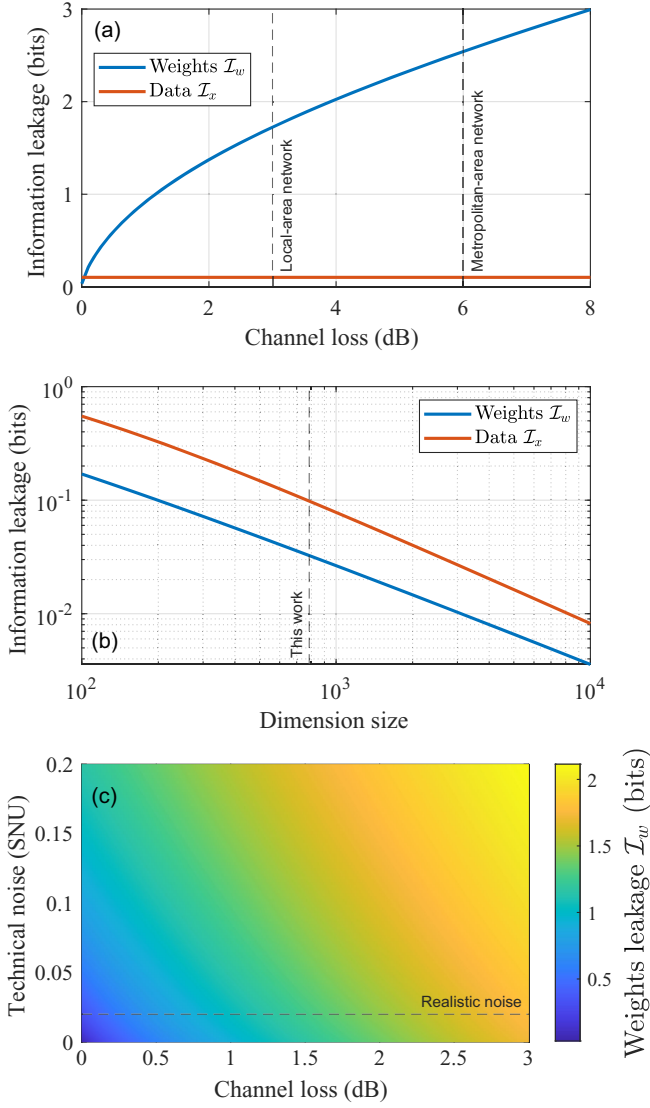


FIG. 4. Information leakage versus channel loss, neurons per layer, and technical noise. (a) In the presence of loss, the server can increase the average photon occupation to conserve classification accuracy. This increases the weights leakage while the data leakage remains unaffected. The client cannot compensate for high losses, as the classification accuracy saturates as a function of the gain. A weight leakage of up to 4 bits per weight is obtained for standard losses in local-area networks (LAN) and metropolitan-area networks (MAN). (b) Both weight and data leakages diminish with an increasing number of neurons per layer. Therefore, advanced DNNs, which are much larger than the model used in this work, are expected to suffer from even smaller leakages. (c) Weight leakage as a function of channel losses and technical noise. At realistic technical noise of 2% SNU, the technical noise does not contribute significantly to the leakage.

realistic noise level, losses are more important than technical noise in determining the weight leakage. This analysis considers strict assumptions, while a lower leakage could be obtained in loose assumptions where the malicious party has limited access to the system components.

Finally, we briefly introduce an extension of our protocol to a multiuser network consisting of a server and two or more clients, where each party assumes that all other parties are dishonest and collaborate against them.

We propose two network architectures: symmetric and asymmetric. In the symmetric network, the server splits the light encoding of the weights in half and sends each portion to a different client. The clients use the same gain for inference and transmit return states to the server. The server coherently combines the return states and measures the noise in the output. The added excess noise is doubled relative to a single client, resulting in an increase in weight leakage [Eq. (2)]. However, the data leakage for each client does not change compared to a single client [Eq. (3)]. Since the light received by each client is halved in power, the SNR drops by a factor of 2 for each client, leading to a reduction in classification accuracy. In the asymmetric scheme, client 1 uses all the light for inference and transmits its entire return state to client 2. Client 2, after performing its own inference, sends the doubly disturbed return state back to the server, which measures the excess noise. The advantage of the asymmetric architecture is that it eliminates the need for coherent combining of the return states from both clients at the server.

VII. DISCUSSION

Our current protocol forces a trade-off between security for classification accuracy; future generations could relax this trade-off by joint optimization of DNN model parameters and protocol steps. Additionally, better performance may be achieved in a hardware-aware environment by optimizing the weights through custom physical layers instead of noiseless layers. Furthermore, future generations could leverage artificial intelligence-based theorem provers to obtain tighter upper bounds for information leakage [81,82].

Performance may be further improved by incorporating squeezing [83,84] or non-Gaussian operations into the protocol [85–87]. For example, the client could use a photon-number-resolving detector, and the server could encode the weights using the photon number. The use of quantum resources will also enable the extension of our protocol beyond classical computation [88], such as quantum machine learning [89–92] or blind quantum computation [24,30,32,33]. For instance, an exponential speedup in linear algebra operations [88] combined with quantum security can unlock the realization of larger DNNs in the cloud without compromising privacy.

Our current security analysis assumes the weights follow a Gaussian distribution and the asymptotic limit of infinitely many channel uses. Recent work, however, supplies both asymptotic and finite-block security proofs for discretely modulated continuous-variable QKD under general attacks [93–95]. Incorporating these techniques would allow us to extend our protocol's guarantees to realistic

settings with discrete modulation alphabets and modest signal blocks. We note that, in finite-block analysis, the server gains sufficient confidence for parameter estimation by analyzing the weights within a single layer such that multiple inference queries are not required. Moreover, to prevent leakage during parameter estimation in the finite-block size analysis, the server can first send a one-time pad matrix OTP followed by a matrix $A = W - \text{OTP}$, for each weight matrix W . The calibrated transmission of $A = W - \text{OTP}$ after the OTP phase acts operationally like privacy amplification in QKD: After parameter estimation, the server selects the A -phase photon occupation to bound the total information about W within a prescribed budget while preserving correctness for an honest client.

While our analysis of the server's weights security is applicable to all possible operations from the client, at present our analysis of the client's data security rests on the assumption of Gaussian probe states sent by the malicious server. An extended analysis of the client's data security against non-Gaussian probe attacks from the server is left for future work.

Recent experimental advances support the feasibility of our protocol. On the client side, large unitary transformations and optical computations have recently been demonstrated [55,96–100]. On the server side, continuous-variable quantum key distribution has been successfully implemented [101–103]. The combination of these technologies paves the way for the near-term realization of our quantum-secure multiparty deep learning. Consequently, we expect that our protocol can be readily implemented using existing quantum key distribution systems, extending their capabilities from merely establishing cryptographic keys to secure computation.

Although this work focuses on secure inference, the secure inner-product engine presented here applies equally to learning. The inner product primitive can be used to compute the partial derivatives of the loss with respect to the hidden layer activations $\nabla_{x^{(i)}} \mathcal{L}$ in backpropagation, since that step involves matrix-vector products; only the roles of the matrix and vector operands change. Encoding the operands in backpropagation in our protocol, therefore, yields a private training step without changing the security proof. Any algorithm dominated by inner products, from principal-component analysis to linear regression, inherits the same guarantees. Hence, the inner-product engine is a general quantum-secure multiparty primitive whose first use case is inference but whose reach extends to full training and other data-analytic tasks.

Our approach solves a fundamental security challenge in multiparty computation, paving the way for information-theoretical security at various stages of the machine-learning pipeline. The introduction of our protocol enables the secure application of deep learning on private datasets and models, establishing a rigorous standard of

secure computation in sectors such as finance, health care, and business.

ACKNOWLEDGMENTS

The authors kindly thank Saumil Bandyopadhyay, Leshem Choshen, Rom Dudkiewicz, and Michael Ben-Or for fruitful discussions and suggestions, and we thank Alessandro Buzzi for providing Fig. 1 illustrations. K. S. acknowledges the support of the Israeli Council for Higher Education, the Zuckerman STEM Leadership Program, and the European Union's Horizon program under Marie Skłodowska-Curie Grant No. 101202109. This work was supported by the Quantum Augmented Network (QuANET) under Award No. HR001124C0405.

K. S. proposed the protocol, performed the security analyses and numerical simulations, and wrote the manuscript under the guidance of D. E. S. K. V. designed the PYTORCH package for optical deep neural networks (Appendix B). R. H. assisted with the optical realization of the protocol and its quantum description (Appendix A). P. I. assisted in calculation of the Cramér-Rao bound (Appendix C 2). All authors contributed to the writing.

K. S. and D. E. have filed a U.S. Patent application 63677972 on quantum-secure multiparty deep learning. The other authors declare no competing interests.

DATA AVAILABILITY

The data that support the findings of this article are openly available [71].

APPENDIX A: OPTICAL IMPLEMENTATION

The server consists of two modules, the transmitter and the receiver [see Fig. 5(a)]. The transmitter module I/Q modulates the neural network weights onto a train of weak coherent states which are produced by attenuating a continuous-wave laser to the few-photon limit. The receiver module measures the modulated quadratures of the incoming return state using homodyne detection with a reference local oscillator. The optical power difference between the two output arms is proportional to either the I or the Q quadrature of the return state, depending on the phase of the local oscillator.

We propose multiple optical implementations of the unitaries $U_{\hat{x}}$ and $U_{\hat{x}}^\dagger$ in the client. The client could operate via time domain encoding using optical loops [104] [see Fig. 5(b)]. Equivalently, the client could operate by spatial domain encoding using a mesh of interferometers [96,105] [see Fig. 5(c)] or free-space multiplane light converters [106–108]. Time-domain implementations are promising for large-scale universal optical information processing because the number of elements does not scale with the number of optical modes [109–112]. However, because of

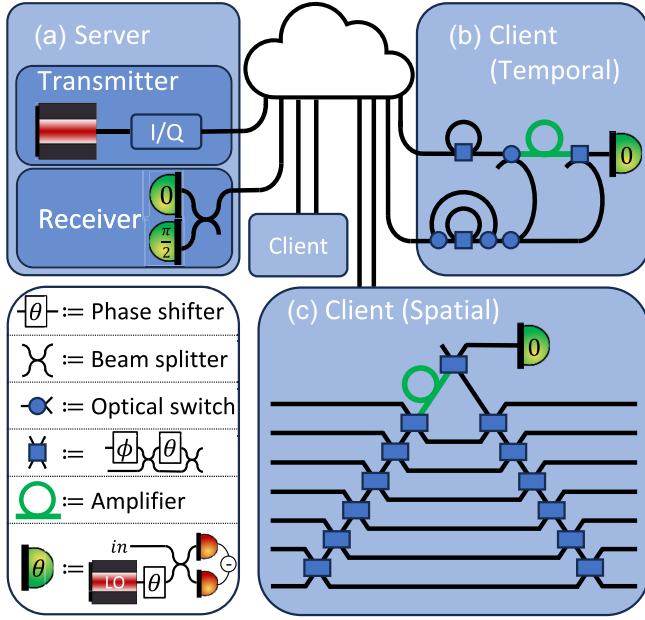


FIG. 5. Optical implementation. (a) The server transmits neural network weights, encoded by complex amplitudes of weak coherent states generated by an attenuated laser. The encoding can be either temporal or spatial. (b),(c) The client performs a series of linear operations on the beam to measure the inner product of a local data vector with the distributed vector (see the main text). After these operations, the client returns the residual light as a return state to the server, which measures the excess noise to compute the leakage of its weights. In the spatial domain (c), the client uses an MZI mesh to interfere spatially adjacent modes, resulting in the uppermost output mode having a complex amplitude with a real value representing the desired inner product. This mode is then amplified (green loop) and split by a weighted beam splitter (top) into a coherent homodyne detector (green detectors) and into another MZI mesh. The second MZI mesh implements the inverse of the first mesh. In the time domain (b), the first MZI mesh is replaced with a single MZI and a fiber loop, while the second MZI mesh is replaced by nested fiber loops.

optical loss, arbitrary unitaries in both technologies are currently limited to a few dozen modes.

Our protocol, however, does not require an arbitrary unitary transformation. Photonic meshes are typically used to implement the multiplication of an $N \times N$ unitary matrix (U , which is programmed into the mesh) and an N length vector $\vec{x}$ (which is supplied as input fields to the mesh) and usually contain $\mathcal{O}(N^2)$ tunable physical degrees of freedom. In our case, we need only a mesh that permits the programming of an N length vector $\vec{x}$ (the client's input) into $\mathcal{O}(N)$ tunable physical degrees of freedom and then computes vector-vector inner products $\vec{w} \cdot \vec{x}$ by taking as input an N length vector $\vec{w}$ (a row of the matrix W).

Moreover, one can split each inner product into k inner products of vectors of size N/k (which can then be fed into present-day optical meshes) and sum the partial results digitally. In this case, the excess noise added to each

computation scales proportionally to k , which increases the lower bound on leakage but allows operation with fewer physical modes. Lastly, our inner product engine is designed to work with complex numbers, compressing the N length real vectors into $N/2$ length complex vectors, as described in Appendix B.

In the spatial domain [Fig. 5(c)], optical modes are defined by spatially separated waveguides. Here, the client implements the same $U_{\hat{x}}$ and $U_{\hat{x}}^\dagger$ via two series of Mach-Zehnder interferometers (MZIs) weighted according to the elements of $\hat{x}$ [105]. These MZIs mix spatially adjacent modes to shift the inner product into the top mode. The light in this mode is then amplified by a factor of G using a phase-insensitive amplifier such as a standard erbium-doped fiber amplifier. Then, a beam splitter divides the light with a splitting ratio of $1 - (1/G):(1/G)$ for the transmitted and reflected ports. The transmitted port is then measured using homodyne detection in the real quadrature to estimate the desired inner product. The reflected port is fed into a the second MZI mesh to apply $U_{\hat{x}}^\dagger$ and routes the resultant return state back to the server.

In the temporal domain design [Fig. 5(b)], optical modes are defined by temporally separate pulses. The first operation $U_{\hat{x}}$ is implemented using a MZI and a fiber loop [113]. This structure allows us to interfere successive pulses from the incoming pulse train $\vec{w}$ with a pulse in the loop that contains the running sum of the inner product, resulting in the final inner product $\vec{w} \cdot \hat{x}$ being written into the complex amplitude of the last output pulse. As in the spatial domain, the light in this mode is then amplified by a factor of G using a phase-insensitive amplifier, and then a beam splitter divides the light with a splitting ratio of $1 - (1/G):(1/G)$ for the transmitted and reflected ports. The transmitted port is then measured using homodyne detection in the real quadrature to estimate the desired inner product. The reflected port is fed into a nested Mach-Zehnder fiber loop that implements the unitary $U_{\hat{x}}^\dagger$ and routes the resultant return state back to the server.

1. Quantum description

a. Phase-insensitive amplifier

We use the quantum electromagnetic field ladder operators (or annihilation and creation operators), denoted by $\hat{a}$ and $\hat{a}^\dagger$, respectively, to analyze the behavior of the system at the quantum level. The action of these operators on the photon number states $|n\rangle$ is given by

$$\hat{a}|n\rangle = \sqrt{n}|n-1\rangle \quad \text{and} \quad \hat{a}^\dagger|n\rangle = \sqrt{n+1}|n+1\rangle.$$

The input-output relationship of the ladder operators in the Heisenberg picture for a phase-insensitive amplifier with gain G is [114]

$$\hat{a}_{\text{amp}} = \sqrt{G}\hat{a}_{\text{in}} + \sqrt{G-1}\hat{a}_v^\dagger,$$

where $\hat{a}_{\text{in}}$ is the annihilation operator for the input mode, $\hat{a}_{\text{amp}}$ is the annihilation operator for the amplified output mode, and $\hat{a}_v$ is the annihilation operator for the auxiliary vacuum noise mode introduced during amplification.

The quadrature operators are defined as

$$\hat{X} = \hat{a} + \hat{a}^\dagger, \quad \hat{P} = \frac{1}{i}(\hat{a} - \hat{a}^\dagger).$$

The variance for an operator $\hat{O}$ is given by

$$\langle (\Delta \hat{O})^2 \rangle = \langle \hat{O}^2 \rangle - \langle \hat{O} \rangle^2.$$

Then, the following is true for the input mode:

$$\langle (\Delta \hat{X}_{\text{in}})^2 \rangle = 1, \quad \langle (\Delta \hat{P}_{\text{in}})^2 \rangle = 1.$$

For the output mode of the phase-insensitive amplifier,

$$\hat{X}_{\text{amp}} = \hat{a}_{\text{amp}} + \hat{a}_{\text{amp}}^\dagger, \quad \hat{P}_{\text{amp}} = \frac{1}{i}(\hat{a}_{\text{amp}} - \hat{a}_{\text{amp}}^\dagger).$$

Substituting the expression for $\hat{a}_{\text{amp}}$:

$$\begin{aligned} \hat{X}_{\text{amp}} &= \sqrt{G}\hat{X}_{\text{in}} + \sqrt{G-1}\hat{X}_v, \\ \hat{P}_{\text{amp}} &= \sqrt{G}\hat{P}_{\text{in}} - \sqrt{G-1}\hat{P}_v, \end{aligned}$$

the variance of the quadratures after amplification is

$$\begin{aligned} \langle (\Delta \hat{X}_{\text{amp}})^2 \rangle &= \langle (\sqrt{G}\hat{X}_{\text{in}} + \sqrt{G-1}\hat{X}_v)^2 \rangle = 2G - 1, \\ \langle (\Delta \hat{P}_{\text{amp}})^2 \rangle &= 2G - 1. \end{aligned}$$

This shows that the noise is amplified along with the signal, and, for large G , the noise variance scales linearly with G .

b. Weighted beam splitter

The beam splitter has a splitting ratio of $[1 - (1/G) : (1/G)]$. In the spatial domain picture, the input modes for the beam splitter are the amplified output $\hat{a}_{\text{amp}}$ and a vacuum mode, labeled $\hat{b}_{\text{vac}}$. The weighted beam splitter transform yields

$$\begin{aligned} \hat{a}_{\text{out1}} &= \sqrt{1 - \frac{1}{G}}\hat{a}_{\text{amp}} - \sqrt{\frac{1}{G}}\hat{b}_{\text{vac}}, \\ \hat{a}_{\text{out2}} &= \sqrt{\frac{1}{G}}\hat{a}_{\text{amp}} + \sqrt{1 - \frac{1}{G}}\hat{b}_{\text{vac}}. \end{aligned}$$

Substituting the above expressions for $\hat{X}_{\text{amp}}$ and $\hat{P}_{\text{amp}}$,

$$\hat{X}_{\text{out2}} = \sqrt{\frac{1}{G}}\hat{X}_{\text{amp}} + \sqrt{1 - \frac{1}{G}}\hat{X}_{\text{vac}}$$

and similarly for $\hat{P}_{\text{out2}}$.

Since $\hat{X}_{\text{vac}}$ is the quadrature of the vacuum mode, it has unit variance: $\langle (\Delta \hat{X}_{\text{vac}})^2 \rangle = 1$.

Notice that two vacuum modes have been introduced: one during amplification to preserve the commutation relation of the amplified output mode and one at the dark input port of the beam splitter. The two vacuum modes are independent and uncorrelated, so they can be summed in quadrature. One can see this directly by considering the reparametrization: $\sqrt{G} \mapsto \cosh(\theta)$ and $\sqrt{G-1} \mapsto \sinh(\theta)$ [note that $\cosh^2(x) = \sinh^2(x) + 1$ for any x]. Their addition in quadrature forms a single vacuum noise mode with unit variance with a coefficient determined by the Pythagorean theorem sum of their composite variances.

Alternatively, we can derive the noise variance directly as follows:

$$\langle (\Delta \hat{X}_{\text{out2}})^2 \rangle = \langle (\Delta \hat{P}_{\text{out2}})^2 \rangle = 1 + \left(2 - \frac{2}{G}\right).$$

The excess noise on the amplified signal after beam splitting is, thus, $[2 - (2/G)]$.

For $\hat{a}_{\text{out1}}$,

$$\hat{X}_{\text{out1}} = \sqrt{1 - \frac{1}{G}}\hat{X}_{\text{amp}} - \sqrt{\frac{1}{G}}\hat{X}_{\text{vac}}$$

and similarly for $\hat{P}_{\text{out1}}$. Thus,

$$\langle (\Delta \hat{X}_{\text{out1}})^2 \rangle = \langle (\Delta \hat{P}_{\text{out1}})^2 \rangle = \frac{2G^2 - 3G + 2}{G}.$$

The optical power at the client's detector, after gain and splitting, is $S = G[1 - (1/G)]|\vec{w} \cdot \hat{x}|^2$. The SNR of the client's measurement is, therefore,

$$\text{SNR} = \frac{G(G-1)}{2G^2 - 3G + 2} |\vec{w} \cdot \hat{x}|^2. \quad (\text{A1})$$

c. Return state excess noise

First, we consider the effect of the overall unitary transform pair $\{U_{\hat{x}}, U_{\hat{x}}^\dagger\}$ on the variance of the state. The initial input $\vec{w}$ is a set of coherent states with a unit shot-noise variance in each mode; that is, the noise itself is uncorrelated and isotropic. We describe the noise in these N modes and the correlations between them, through the covariance matrix of their quadratures Σ^X . The (i, j) th element of this matrix is given by $\Sigma^X(i, j) = \mathbb{E}[X_i X_j] - \mathbb{E}[X_i]\mathbb{E}[X_j]$.

Since a coherent state is equivalent to a displacement operation on the vacuum state, the covariance matrix of $\vec{w}$ is $\Sigma_w^X = \mathbb{1}$. Since we want the first output mode (the result mode) of $U_{\hat{x}}\vec{w}$ to carry the inner product $\vec{w} \cdot \hat{x}$, $U_{\hat{x}}$ and $U_{\hat{x}}^\dagger$ take the form

$$U_{\hat{x}} \sim \begin{bmatrix} \hat{x} \\ \mathbf{v} \end{bmatrix}, \quad U_{\hat{x}}^\dagger \sim \begin{bmatrix} \hat{x}^\dagger & | & \mathbf{v}^\dagger \end{bmatrix},$$

where $\hat{x}$ is represented as a row vector and $\mathbf{v}$ is a (nonunique) rectangular matrix which is chosen to ensure unitarity of $U_{\hat{x}}$.

The covariance matrix of $U_{\hat{x}}\vec{w}$ is

$$U_{\hat{x}}\Sigma_w^X U_{\hat{x}}^\dagger = U_{\hat{x}}U_{\hat{x}}^\dagger = \mathbb{1}.$$

Next, we apply the amplify-and-split operation $\mathcal{G}(\cdot)$ on the vector of modes $U_{\hat{x}}\vec{w}$. This operation involves applying a phase-insensitive amplification with gain G on the result mode and then splitting it with the ratio $1 - (1/G) : (1/G)$. The output of the $1/G$ port is fed back to the mesh that implements $U_{\hat{x}}^\dagger$. The operation $\mathcal{G}(\cdot)$ leaves all the other modes of $U_{\hat{x}}\vec{w}$ untouched. The resultant covariance matrix after amplification and splitting is the diagonal matrix $\Sigma_{\mathcal{G}(U_{\hat{x}}\vec{w})}^X \sim \text{diag}(\sigma^2, 1, 1, \dots, 1)$, where $\sigma^2 = 1 + [2 - (2/G)]$ is the variance of the result mode calculated in Appendix A.

Finally, we operate with the transform $U_{\hat{x}}^\dagger$:

$$\Sigma_{U_{\hat{x}}^\dagger \mathcal{G}(U_{\hat{x}}\vec{w})}^X = U_{\hat{x}}^\dagger \Sigma_{\mathcal{G}(U_{\hat{x}}\vec{w})}^X U_{\hat{x}} = \mathbb{1} + \left(2 - \frac{2}{G}\right) \hat{x} \hat{x}^\dagger.$$

Note that, since $\hat{x} \hat{x}^\dagger$ is a dense matrix, the various components of the return state $U_{\hat{x}}^\dagger \mathcal{G}(U_{\hat{x}}\vec{w})$ are all correlated with one another. However, the variance of the i th output mode contains only information from $\hat{x}_i$; recalling that we had defined the excess variance of the i th output mode as the excess noise η_i in Sec. III, we have

$$\eta_i = \left(2 - \frac{2}{G}\right) \cdot |\hat{x}_i|^2.$$

APPENDIX B: CLASSIFICATION ACCURACY CALCULATION

To analyze our protocol's performance for deep learning, we compute the classification accuracy for the standard MNIST classification task using PYTORCH. To this end, we write custom neural network layers that implement our coherent linear algebra engine presented in Sec. III; several of these custom layers were strung together to form a secure neural network. The homodyne current detected in each custom layer is sampled from a unit Gaussian distribution $\mathcal{N}(0, 1)$ SNU to account for the quantum shot noise in each quadrature. The resultant noisy sample is then fed to the nonlinear ReLU activation function before being passed on to the next custom layer.

Separately, we trained a digital model composed of standard PYTORCH layers on preprocessed MNIST images. The preprocessing of the dataset included the flattening of the 28×28 images to vectors of size 784, followed by a centering and scaling of each vector to the range $[-1, 1]$. Then, we trained a standard two-layer network with

784 inputs, 784 hidden neurons, and ten outputs, obtaining a classification accuracy of 98%. We then transferred these trained digital weights into the custom secure neural network and computed the accuracy achieved by the secure protocol as a function of the weight pulse energy. We found that the accuracy of the digital and analog models were in agreement up to variations caused by quantum shot noise.

The real operations required for standard deep learning tasks can be implemented on our complex-valued hardware by encoding the real input vector $\vec{x}^{(\mathbb{R})}$ of length N and the real weight matrix $W^{(\mathbb{R})}$ of size $M \times N$ into a complex input vector $\vec{x}^{(\mathbb{C})}$ of length $N/2$ and a complex weight matrix $W^{(\mathbb{C})}$ of shape $M \times N/2$ using the following procedure:

$$W_{i,k}^{(\mathbb{C})} = W_{i,2k-1}^{(\mathbb{R})} + iW_{i,2k}^{(\mathbb{R})} \quad \text{and} \quad x_k^{(\mathbb{C})} = x_{2k-1}^{(\mathbb{R})} - ix_{2k}^{(\mathbb{R})}.$$

The hardware computes the real value of the product $W^{(\mathbb{C})}x^{(\mathbb{C})}$, which is precisely the desired matrix-vector product $W^{(\mathbb{R})}x^{(\mathbb{R})}$ required for the DNN computations: $\text{Re}[\sum_{k=1}^{N/2} W_{i,k}^{(\mathbb{C})} x_k^{(\mathbb{C})}] = \sum_{k=1}^N W_{i,k}^{(\mathbb{R})} x_k^{(\mathbb{R})}$. Note that other real-to-complex mappings for machine learning have been studied as well [115].

We define the SNR at the homodyne detector at the neural network outputs as

$$\text{SNR}_i = \frac{G(G-1)}{2G^2 - 3G + 2} |\vec{w} \cdot \hat{x}|^2,$$

where the amplitudes of these physical states, w_j , encode the logical weights as $w_j = \sqrt{\mu} \mathbf{W}_{ij} / \|\mathbf{W}\|$, where $\|\mathbf{W}\|$ denotes the root mean square of the elements in $\mathbf{W}$. Then, we have

$$\text{SNR}_i = \frac{G(G-1)}{2G^2 - 3G + 2} \frac{\mu}{\|\mathbf{W}\|^2} |\mathbf{W}_i \cdot \hat{x}|^2.$$

The prefactor is the result of a beam splitter with a ratio of $1/G : 1 - 1/G$ right after the amplifier G that dictates how much of the power will be used for the computation and how much for the return state. Therefore, SNR does not monotonically increase with G , and, in fact, it increases from 1 to $2 + \sqrt{2}$ and then decreases.

Note that, in order to obtain the desired inner product, the client scales the result digitally by $[(\|\vec{x}\|)/(\sqrt{G-1})]$ ($\|\mathbf{W}\|/\sqrt{\mu}$). However, for the purpose of deep learning, the scaling step for the client is superfluous, because the client normalizes between layers regardless. We define a “physical scaling parameter” F to capture the hardware-dependent prefactor in the SNR equation:

$$F := \sqrt{\frac{G(G-1)}{2G^2 - 3G + 2}} \mu.$$

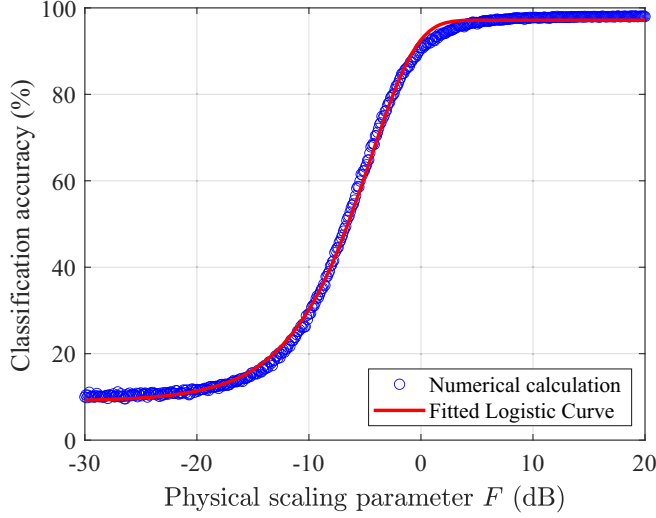


FIG. 6. Classification accuracy versus the physical scaling parameter F . We calculate the classification accuracy considering an additive noise to each inner product, using weights trained for a digital noiseless model. We present the classification accuracy as a function of the physical scaling parameter F . The classification accuracy increases monotonically with F , asymptotically obtaining the digital noiseless accuracy. This parameter captures the hardware-dependent prefactor in the SNR, allowing one to calculate the mutual effect of the average photon number μ and the amplification gain G . We fit the classification accuracy using a logistic curve.

Varying F changes the SNR of the inner product outputs, thus affecting the accuracy of the secure neural network. The purpose of defining this parameter is to enable the calculation of the classification accuracy as a function of the gain G and average photon occupation μ without directly executing the intensive numerical calculations in the two-dimensional space spanned by these two parameters. Instead, we calculate the classification accuracy within the space spanned by F . The dependency of the test accuracy achieved by the secure neural network on F is presented in Fig. 6. We further find that a logistic function faithfully fits the variation of the classification accuracy as a function of F :

$$\text{Acc}(F) = \frac{L}{1 + e^{-k(F-F_0)}} + B$$

with root-mean-square error of 1.1%. This fit allows us to use a convenient analytical formula to predict the classification accuracy as a function of the hardware parameters G and μ instead of using look-up tables.

We present the classification accuracy as a function of the average photon occupation and the amplification gain at the client in Fig. 7. The classification accuracy increases with gain and average occupation number and asymptotically achieves the digital noiseless accuracy. At low gain values, the gain required to maintain a fixed

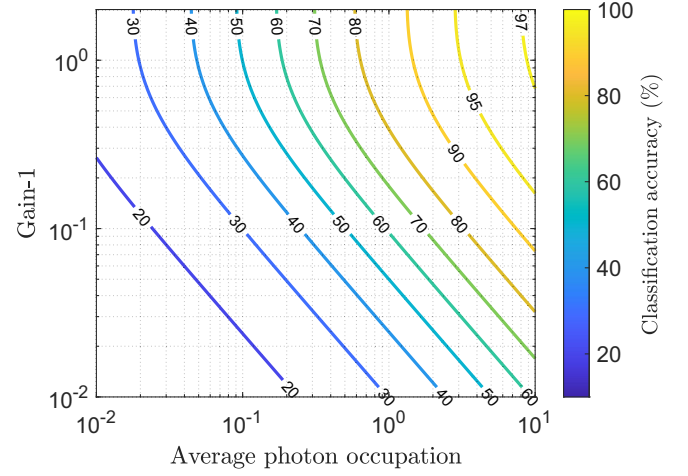


FIG. 7. Classification accuracy versus amplification gain and average photon occupation. The classification accuracy of the physical model, which uses our coherent linear algebra engine (Fig. 2), is numerically calculated as a function of the average photon occupation and the amplification gain at the client. The classification accuracy increases with gain and average occupation number and asymptotically achieves the digital noiseless accuracy. At low gain values, the gain is inversely proportional to the photon occupation number for any given fixed classification accuracy.

classification accuracy is inversely proportional to the average photon occupation.

APPENDIX C: DETAILED SECURITY ANALYSES

As explained in Sec. V, we analyze the security of server following the standard security analysis for continuous-variable quantum key distribution (CVQKD), and we analyze the security of the client by phrasing our problem as a statistical estimation problem.

First, we express the setting of our problem in the CVQKD framework. In a standard CVQKD setting, two legitimate parties Alice and Bob wish to communicate a message under possible attack by a third-party eavesdropper Eve external to both of them. In the CVQKD protocol, normal operation results in Bob receiving the message from Alice, and the mutual information between Alice and Eve (along with Eve's Holevo information) must be bounded below an arbitrary threshold.

In the context of our multiparty quantum-secured DNN protocol, normal operation results in the transfer of a specific amount of information from the server to the client—specifically, the inner product of the server's weights with the client's data must be extracted by the client in order to satisfy the accuracy requirements. Here, both parties (server and client) have the opportunity to act maliciously.

When the server is honest and the client is malicious, the information that is at risk of leakage is the weight vector sent from the server to the client. When the server is

malicious and the client is honest, the information that is at risk of leakage is the client data vector used for the client's inner product. We seek only to protect the information of the honest party, so the case in which both parties act maliciously is not relevant to our protocol.

1. Weight leakage

From the perspective of a malicious client and honest server, the server prepares displaced coherent states with quadrature components q and p that are realizations of two independent and identically distributed random variables Q and P . These random variables are normally distributed as $Q, P \sim \mathcal{N}(0, \tilde{V}_{\text{mod}})$, where $\tilde{V}_{\text{mod}}$ denotes the modulation variance. Here, we use the fact that the information leakage upper bound, quantified by the Holevo information between Eve and Alice, is maximized when the state shared by Alice and Bob is Gaussian [116–118].

The prepare-and-measure covariance matrix, after transmission through a channel with loss T and noise ξ into Bob's lab who performs heterodyne measurement, is [74]

$$\Sigma^{\text{PM}} = \begin{pmatrix} V_{\text{mod}} \mathbb{1}_2 & \sqrt{\frac{T}{\mu}} V_{\text{mod}} \mathbb{1}_2 \\ \sqrt{\frac{T}{2}} V_{\text{mod}} \mathbb{1}_2 & \frac{T}{2} V_{\text{mod}} + \mathbb{1} + \frac{\xi}{2} \mathbb{1}_2 \end{pmatrix},$$

where we defined $V_{\text{mod}} = 4\tilde{V}_{\text{mod}}$.

Using the source replacement method commonly applied in QKD security analyses, the state sent by the server can be described as an entangled state. In this scenario, the server prepares a two-mode squeezed vacuum state (TMSVS), measures both quadratures of one mode, and sends the other mode to the client. In shot-noise units, the TMSVS is characterized by the covariance matrix [74]:

$$\Sigma_{\text{TMSVS}} = \begin{pmatrix} V \mathbb{1} & \sqrt{V^2 - 1} \sigma_z \\ \sqrt{V^2 - 1} \sigma_z & V \mathbb{1} \end{pmatrix}.$$

Here, σ_z is the Pauli matrix, and V represents the variance of the quadrature operators. The upper and lower diagonal 2×2 blocks of Σ_{TMSVS} are the covariance matrices of the quadrature operators of Alice and Bob, respectively, while the off-diagonal block matrices are the covariances between the quadratures of Alice and those of Bob. The variance V is related to the mean photon number per pulse by the relation $\mu = \frac{1}{2}(V - 1)$. This variance is the sum of Alice's actual modulation variance of her two quadrature components and the vacuum shot-noise variance.

After Bob's mode is transmitted through the channel with transmission coefficient T and excess noise ξ , the covariance matrix transforms to

$$\Sigma_{AB} = \begin{pmatrix} V \mathbb{1} & \sqrt{T(V^2 - 1)} \sigma_z \\ \sqrt{T(V^2 - 1)} \sigma_z & [T(V - 1) + 1 + \xi] \mathbb{1} \end{pmatrix}.$$

Without loss of generality, we consider $|\psi\rangle$ as a bipartite state with Alice's and Bob's joint subsystem comprising one part and Eve's subsystem the other. Any pure bipartite state can be expressed using the Schmidt decomposition as $|\psi\rangle = \sum_i \lambda_i |i\rangle_{AB} |i\rangle_E$. Tracing out either subsystem yields a mixed state with a von Neumann entropy that depends only on the Schmidt coefficients λ_i :

$$S_E = S_{AB} = -\sum_i \lambda_i \log(\lambda_i).$$

We use the Holevo theorem to bound the amount of information that Eve can obtain [64]. The Holevo theorem provides an upper bound on the accessible information that can be extracted from a quantum system. Tracing out Eve's subspace from the joint state $|\psi\rangle\langle\psi|$ yields Alice's and Bob's joint substate, that is, $\text{Tr}_E(|\psi\rangle\langle\psi|) = \rho_{AB}$. Mathematically, the Holevo bound χ for the information held by Eve's substate $\rho_E = \text{Tr}_{AB}(|\psi\rangle\langle\psi|)$ about the quantum state ρ_{AB} shared between Alice and Bob is given by [74]

$$\chi = S_E - S_{E|A} = S_{AB} - S_{B|A},$$

where $S_{X|Y}$ is the conditional von Neumann entropy of the density matrix ρ_X given the density matrix ρ_Y .

The von Neumann entropy of a quantum state can be expressed in terms of the symplectic eigenvalues of that state's covariance matrix. The symplectic eigenvalues of a $2N \times 2N$ matrix Σ are defined as the absolute values of the ordinary eigenvalues of the matrix $\tilde{\Sigma} = i\Omega\Sigma$ where Ω is the $2N \times 2N$ matrix given by the direct sum:

$$\Omega = \bigoplus_{i=1}^N \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

It follows that a Gaussian state with covariance matrix Σ has a von Neumann entropy given by $S = \sum_i g(\nu_i)$, where

$$g(\nu) = \left(\frac{\nu+1}{2}\right) \log_2 \left(\frac{\nu+1}{2}\right) - \left(\frac{\nu-1}{2}\right) \log_2 \left(\frac{\nu-1}{2}\right)$$

and ν_i is the i th symplectic eigenvalue of Σ (see Ref. [74] for details).

To calculate S_{AB} we calculate the symplectic eigenvalues of Σ_{AB} :

$$\nu_{1,2} = \frac{1}{2}(z \pm [b - a]),$$

where

$$\begin{aligned} a &:= V, & b &:= T(V - 1) + 1 + \xi, \\ c &:= \sqrt{T(V^2 - 1)}, & z &:= \sqrt{(a + b)^2 - 4c^2}. \end{aligned}$$

Then, we have $S_{AB} = g(\nu_1) + g(\nu_2)$. Note that in Sec. V we presented these quantities using occupations $\mu = [(V - 1)/2]$ and $T = 1$. We analyze the effect of optical loss where the transmission $T < 1$ in Sec. VI.

We calculate $S_{B|A}$ assuming that Bob implements homodyne detection. The covariance matrix $\Sigma_{B|A}$ is given by

$$\Sigma_{B|A} = \Sigma_B - \frac{1}{V_A + 1} \Sigma_C \Sigma_C^T,$$

where $\Sigma_B = b\mathbb{1}$, $\Sigma_C = c\sigma_z$, $V_A = a$, and a , b , and c are defined as above. Using these definitions, we obtain

$$\Sigma_{B|A} = \begin{pmatrix} b & 0 \\ 0 & b \end{pmatrix} - \frac{1}{a+1} \begin{pmatrix} c^2 & 0 \\ 0 & c^2 \end{pmatrix} = \left(b - \frac{c^2}{a+1} \right) \mathbb{1}.$$

Its symplectic eigenvalue is

$$\nu_3 = b - \frac{c^2}{a+1}.$$

This gives us $S_{B|A} = g(\nu_3)$, which finally yields $\chi = g(\nu_1) + g(\nu_2) - g(\nu_3)$. Equation (2) presents half of this quantity, accounting for the weight elements rather than the complex modes.

To calculate the Holevo information in Eq. (2), one should determine the intensity transmittance T from Alice to Bob (server to client and back to the server) and the excess noise in the intensity ξ . To this aim, we follow Ref. [74] and define the variance of the weights sent by Alice, $\tilde{V}_{\text{mod}}$, the return state light amplitude variance V_B measured using homodyne detection, and the covariance between the encoded weights and the return state light amplitudes: C_{AB} . Then, defining $V_{\text{mod}} := 4\tilde{V}_{\text{mod}}$, the transmittance is obtained from $T = 2(C_{AB}/V_{\text{mod}})^2$, and the excess noise is obtained from $\xi = 2(V_B - 1) - TV_{\text{mod}}$. In summary, V_{mod} , V_B , and C_{AB} are measured, while T and ξ are inferred from the measurements using the formulas above.

a. Weight leakage accumulation due to multiple queries

We previously showed that clients gain only limited information on the DNN weights through a single broadcast of the weights. We now discuss how to control the accumulation of this information during multiple queries. Specifically, we propose to manipulate the weights of the model in every broadcast in a way that preserves the neural network function while drastically reducing the accumulation of weight information at the client's end. Although neural network functions are not invariant to general linear transformations (e.g., translation [119,120]), there always exist isomorphisms between weight matrices that do not alter the neural network function.

Let $W_1, W_2 \in \mathbb{R}^{N \times N}$ (this procedure works equally well for rectangular matrices) be two weight matrices with $\sigma(\cdot)$ implementing an elementwise nonlinearity, so $W_2\sigma(W_1)$ implements two successive layers of a neural network. We note that, for any permutation matrix τ and its inverse τ^{-1} , $W_2\sigma(W_1\vec{x}) = (W_2\tau^{-1})\sigma(\tau W_1\vec{x})$. Thus, there are $N!$ different permutations possible that all return the same computation [121]. The server can send any of these τW matrices. This process can be repeated with W_2 and the next matrix of the network W_3 and so on. For the common case of $\sigma(x) = \text{ReLU}(x) = \max(0, x)$, τ may also include multiplication by a random scalar [122] or even multiplication by a random diagonal matrix, limiting the information in one measurement by the entropy of these random manipulations. By choosing a maximally entropic distribution over the transformation space, Alice can limit Eve's ability to infer weight information between broadcasts.

The idea of using permutations to hide information has a rich history in the recent privacy and statistics literature. Permutations were first employed to hide ("obfuscate") neural network functions, to the best of our knowledge, in 2007 [123] in conjunction with homomorphic encryption to achieve computational security. Achieving statistical security with permutations is an area of vigorous current research in the probability and statistics communities where recent results [124] have established that permutation recovery is statistically hard in situations where the SNR is limited. In parallel, in the field of differential privacy (DP), it was proven recently that random shuffling (permutations) amplifies differential privacy guarantees over nonshuffling baselines.

In DP, privacy leakage is measured via two leakage parameters ϵ and δ ; the smaller they are, the more private the protocol is. It was proven by Refs. [125–127] that, if the per-row leakage ϵ_{row} stays below a threshold $\log_2 N$ (where N is the number of rows in the matrix), shuffle DP reduces the effective leakage ϵ_{sh} for an entire query by roughly $1/\sqrt{N}$ [126]. Once ϵ_{row} exceeds the logarithmic threshold, the permutation can be reconstructed and the shuffle no longer helps—precisely the phase transition observed in DP studies. Our protocol enables below-threshold operation via tuning of the server's laser power and the client's amplification gain.

To raise privacy beyond what permutations and positive scalings can deliver, one should select an activation that commutes with a larger group of linear masks. Block-radial gates—where each K -channel group is separately normalized and nonlinearly rescaled—commute with every block-orthogonal matrix, adding $K(K-1)/2$ masking degrees of freedom per block. Another example is hinge activations, which reflect a vector across a learned normal c whenever $\langle x, c \rangle < 0$. The price of these nonelementwise nonlinearities is extra computation, but in exchange they enable a mask space almost as large as the linear layers themselves, tightening information-theoretic leakage bounds.

2. Data leakage

In this scenario, the client operates legitimately and the server acts maliciously. Under our eavesdropper model, we assume that the server can perform any arbitrary individual attack upon receiving the return state ρ_v . In particular, the server could be equipped to make quantum measurements, observe quantum correlations, and transmit non-Gaussian states.

The eavesdropper may operate under different models based on different capabilities; here, we present analysis for individual attacks and collective attacks. Under individual attacks, the attacker performs independent and identically distributed attacks on all incoming modes; that is, they prepare separable ancilla states that interact individually with one of the signal modes in the quantum channel. The ancilla states are then stored in a quantum memory until measurement and are measured independently of each other. Under collective attacks, the attacker similarly stores the states in a quantum memory but now is able to interact with multiple states at a time, including by performing entanglement-based attacks to correlate multiple states prior to collective measurement.

Finally, we consider the case where the server transmits non-Gaussian states. Non-Gaussian states are not fully characterized by their covariance matrices; i.e., the inter-mode correlations of non-Gaussian states are not restricted to their first- and second-order moments. In particular, the client's Gaussian operations on the server's non-Gaussian states also influence the higher-order moments, which could further expose the client's data. Therefore, beyond optimizing measurements (captured by the quantum-Fisher information), a dishonest server might attempt to optimize the probe state to increase information about the client's data. A full characterization of leakage under server state optimization is left for future work.

a. Individual attacks

We recall from Appendix A 1 c that the excess noise in the i th mode in ρ_v is

$$\eta_i = \left(2 - \frac{2}{G}\right) \cdot |x_i|^2.$$

In the following classical leakage analysis, we characterize the leakage of each individual data element x (subscript i suppressed for convenience) under the possibility of an individual classical attack from a malicious server.

Let R be a Gaussian random variable corresponding to the i th mode of the return state with mean w and variance σ^2 , where

$$\sigma^2 = 1 + \left(2 - \frac{2}{G}\right)x^2$$

for a known parameter G and an unknown parameter x .

We aim to calculate the precision in estimating x using the Cramér-Rao bound for M measurements and use this precision to determine the number of bits of information obtained by the server about x .

The probability density function of R is given by

$$f(R; x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(R-w)^2}{2\sigma^2}\right), \quad (\text{C1})$$

where σ^2 is given above.

The Fisher information $\mathcal{I}_R(x)$ for a single measurement with respect to a given distribution on R is given by

$$\mathcal{I}_R(x) = \mathbb{E}\left[\left(\frac{\partial}{\partial x} \log f(R; x)\right)^2\right].$$

Noting that $[(\partial\sigma^2)/(\partial x)] = 2\{[(2G-2)]/G\}x$, we compute the derivative of the log-likelihood function:

$$\frac{\partial}{\partial x} \log f(R; x) = 2 \frac{(2G-2)}{G} x \left(-\frac{1}{2\sigma^2} + \frac{(R-w)^2}{2\sigma^4}\right).$$

Noting that σ and x are constants with respect to the distribution over R , the Fisher information is

$$\mathcal{I}_R(x) = \left(2 \frac{(2G-2)}{G} x\right)^2 \frac{1}{4\sigma^4} \mathbb{E}\left[\left(-1 + \frac{(R-w)^2}{\sigma^2}\right)^2\right].$$

Using the properties of the moments of Gaussian distributions: $\mathbb{E}[(R-w)^2] = \sigma^2$ and $\mathbb{E}[(R-w)^4] = 3\sigma^4$, we obtain

$$\begin{aligned} \mathcal{I}_R(x) &= \left(2 \frac{(2G-2)}{G} x\right)^2 \frac{1}{4\sigma^4} \left(1 - 2 \frac{\sigma^2}{\sigma^2} + 3\right) \\ &= \frac{2(2 - \frac{2}{G})^2 |x|^2}{\sigma^4}. \end{aligned}$$

For M independent measurements, the Fisher information is additive:

$$\mathcal{I}_{R,M}(x) = M\mathcal{I}_R(x) = \frac{2M(2 - \frac{2}{G})^2 |x|^2}{\sigma^4}.$$

The Cramér-Rao bound for the variance of any unbiased estimator $\hat{x}$ of x is

$$\text{Var}(\hat{x}) = \sigma_{\hat{x}}^2 \geq \frac{1}{\mathcal{I}_{R,M}(x)} = \frac{\sigma^4}{2M(2 - \frac{2}{G})^2 |x|^2}.$$

The precision (standard deviation) for estimating x is

$$\sigma_{\hat{x}} \geq \sqrt{\frac{\sigma^4}{2M(2 - \frac{2}{G})^2 x^2}} = \frac{\sigma^2}{(2 - \frac{2}{G})x\sqrt{2M}}.$$

The number of bits of information $\mathcal{I}_x$ that the estimator $\hat{x}$ contains about x can be computed by treating $\hat{x}$ as the received signal when x is sent through an additive Gaussian channel with noise variance $\text{Var}(\hat{x})$:

$$\mathcal{I}_x^{\text{ind}} = \frac{1}{2} \log_2 \left(1 + \frac{|x|^2}{\sigma_{\hat{x}}^2} \right) \leq \frac{1}{2} \log_2 \left(1 + \frac{8M(G-1)^2|x|^4}{G^2\sigma^4} \right).$$

This provides the number of bits of information obtained by the server from M measurements of the client's return state.

b. Collective attacks

We quantify the information an adversary can obtain when she is allowed collective measurements on all M identical copies of the returned quantum state. Let the hidden parameter be a unit vector $X \in \mathbb{C}^N$ with the Haar distribution on the unit sphere. Conditioned on $X = x$ the server obtains the m -mode Gaussian density operator

$$\hat{\rho}_x = \mathcal{N}(w, \Sigma(x)), \quad \Sigma(x) = I + \gamma x x^\dagger, \quad \gamma = 2 - \frac{2}{G},$$

whose displacement w is independent of x , so that all x dependence resides in the rank-one spike of the covariance.

For a zero-mean Gaussian state, the QFI matrix is

$$F_{jk} = \frac{1}{2} \text{Tr}[\Sigma^{-1}(\partial_j \Sigma) \Sigma^{-1}(\partial_k \Sigma)].$$

Because $\Sigma(x)$ differs from the identity by a rank-one term, the Sherman-Morrison formula gives

$$\Sigma^{-1}(x) = I - \frac{\gamma}{1 + \gamma} x x^\dagger.$$

Substituting yields

$$F(x) = \frac{\gamma^2}{1 + \gamma} (I - x x^\dagger),$$

an isotropic block on the $(N-1)$ -dimensional tangent space of the unit sphere. Since F itself is a rank-one perturbation of the identity, the quantum Cramér-Rao inequality becomes

$$\text{Cov}(\hat{x}) \geq \frac{1 + \gamma \|x\|^2}{\gamma^2 \|x\|^2} I - \frac{(1 - \gamma \|x\|^2)(1 + \gamma \|x\|^2)}{2\gamma^2 \|x\|^4} x x^\dagger.$$

With $\|x\| = 1$, the per-coordinate variance simplifies to

$$\text{Var}(\hat{x}_i) \geq \frac{G}{2(G-1)} + \frac{G^2(2 - x_i^2)}{8(G-1)^2} + \frac{x_i^2}{2}.$$

This multivariate variance bound is higher than or equal to the univariate variance bound by the AM-GM inequality.

In the high-dimensional (asymptotic) regime, this Fisher information can be used to evaluate the mutual information between the client's input and the return state received by the server analytically. In the asymptotic limit of high number of modes and return states, one can get a simple leakage formula. First, the uniform-sphere prior has covariance $\Sigma_{\text{prior}} = N^{-1} P_\perp^x$ (where $P_\perp^x = I - x x^\dagger$), and its entropy differs from that of the centered Gaussian with the same covariance only by an $O(\ln N)$ term. Second, for large M , the Bayesian posterior entropy converges to the Gaussian entropy with the same conditional covariance. This phenomenon is a known result in the Bayesian inference literature [128]; the Bernstein-von Mises theorem ensures that the posterior distribution of the sensed parameter converges to a Gaussian with covariance given by (the inverse of) the Fisher information, justifying this approximation. Therefore,

$$\Sigma_{\text{post}} = (MF(x) + \Sigma_{\text{prior}}^{-1})^{-1}, \quad F(x) = \frac{\gamma^2}{1 + \gamma} P_\perp^x,$$

and any finite-block-length correction can be evaluated exactly as in the finite-size analyses of QKD protocols. Substituting these two Gaussian entropies gives the leading mutual-information term

$$I(X; \hat{\rho}_x^{\otimes M}) = \frac{N-1}{2} \log \left(1 + \frac{M\gamma^2}{N(1 + \gamma)} \right),$$

while the combined non-Gaussian and finite- M corrections are upper bounded by $O(\ln N) + O(M^{-1/2})$. Because each mode contributes two real parameters, the leakage per real component obeys

$$\mathcal{I}_x = \frac{1}{2} \log \left(1 + \frac{M\gamma^2}{N(1 + \gamma)} \right), \quad \gamma = 2 - \frac{2}{G},$$

so an eavesdropper, even with optimal collective measurements, gains only a bounded amount of information about each element of the secret direction.

c. Hidden layer leakage control

Having derived the leakage from the first layer, we now turn our consideration to leakage from the hidden layers. We assume that the server inserts a normalized map $f: \mathbb{C}^N \rightarrow \mathbb{C}^N$ with $\|f(x)\| = 1$, representing the function that maps the activations to the next layer. The server then receives

$$\hat{\rho}_x^{(1)} = \mathcal{N}(0, I + \gamma f(x) f(x)^\dagger),$$

and the corresponding quantum-Fisher information matrix with respect to the coordinates of x is

$$F_f(x) = \frac{\gamma^2}{1 + \gamma} J_f(x)^\dagger J_f(x), \quad J_f(x) = \frac{\partial f}{\partial x} \Big|_x.$$

A malicious server could induce more leakage by picking an f that enlarges some directional derivative $\|J_f(x)v\|$ for a tangent vector $v \perp x$. The client thwarts this by making quick, tiny, random perturbations $x \mapsto x + \delta v_k$ ($\delta \ll 1$) in several orthonormal directions during the inference and measuring the finite differences $L_k = \|f(x + \delta v_k) - f(x)\|/\delta$. Small perturbations will not change the classification accuracy but will enable the evaluations of the derivatives. Experimentally, this tangent-dithering step is analogous to the standard dithering used in telecommunications, where small random offsets are injected into a signal to probe and stabilize the system's response. If any L_k exceeds a preset threshold, the client aborts and refuses to send a return state; otherwise, the observed L_k values bound the relevant Jacobian components, so the information leakage that the server can extract from any additional batch of copies remains bounded.

-
- [1] Y. LeCun, Y. Bengio, and G. Hinton, *Deep learning*, *Nature (London)* **521**, 436 (2015).
 - [2] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev *et al.*, *Grandmaster level in starcraft II using multi-agent reinforcement learning*, *Nature (London)* **575**, 350 (2019).
 - [3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, *Language models are few-shot learners*, *Adv. Neural Inf. Process. Syst.* **33**, 1877 (2020).
 - [4] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, and K. Chen, *Multi-key privacy-preserving deep learning in cloud computing*, *Future Gener. Comput. Syst.* **74**, 76 (2017).
 - [5] M. Satyanarayanan, *The emergence of edge computing*, *Computer* **50**, 30 (2017).
 - [6] U. A. Butt, M. Mehmood, S. B. H. Shah, R. Amin, M. W. Shaikat, S. M. Raza, D. Y. Suh, and M. J. Piran, *A review of machine learning algorithms for cloud computing security*, *Electronics* **9**, 1379 (2020).
 - [7] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, *Efficient processing of deep neural networks: A tutorial and survey*, *Proc. IEEE* **105**, 2295 (2017).
 - [8] T.-J. Yang, Y.-H. Chen, J. Emer, and V. Sze, *A method to estimate the energy consumption of deep neural networks*, in *Proceedings of the 2017 51st Asilomar Conference on Signals, Systems, and Computers* (IEEE, New York, 2017), pp. 1916–1920.
 - [9] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean, *Carbon emissions and large neural network training*, *arXiv:2104.10350*.
 - [10] A. C. Yao, *Protocols for secure computations*, in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)* (IEEE, New York, 1982), pp. 160–164.
 - [11] C. Gentry, *Fully homomorphic encryption using ideal lattices*, in *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, 2009), pp. 169–178.
 - [12] J. H. Cheon, A. Kim, M. Kim, and Y. Song, *Homomorphic encryption for arithmetic of approximate numbers*, in *Advances in Cryptology—ASIACRYPT 2017: Proceedings of the 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, 2017, Part I* (Springer, New York, 2017), pp. 409–437.
 - [13] B. Li and D. Micciancio, *On the security of homomorphic encryption on approximate numbers*, in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques* (Springer, New York, 2021), pp. 648–677.
 - [14] C. Portmann and R. Renner, *Security in quantum cryptography*, *Rev. Mod. Phys.* **94**, 025008 (2022).
 - [15] N. H. Y. Ng, S. K. Joshi, C. Chen Ming, C. Kurtsiefer, and S. Wehner, *Experimental implementation of bit commitment in the noisy-storage model*, *Nat. Commun.* **3**, 1326 (2012).
 - [16] V. Vilasini, C. Portmann, and L. Del Rio, *Composable security in relativistic quantum cryptography*, *New J. Phys.* **21**, 043057 (2019).
 - [17] A. Chailloux and I. Kerenidis, *Optimal quantum strong coin flipping*, in *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science* (IEEE, New York, 2009), pp. 527–533.
 - [18] A. Pappa, P. Jouguet, T. Lawson, A. Chailloux, M. Legré, P. Trinkler, I. Kerenidis, and E. Diamanti, *Experimental plug and play quantum coin flipping*, *Nat. Commun.* **5**, 3717 (2014).
 - [19] D. Unruh, *Universally composable quantum multi-party computation*, in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques* (Springer, New York, 2010), pp. 486–505.
 - [20] S. Wehner, C. Schaffner, and B. M. Terhal, *Cryptography from noisy storage*, *Phys. Rev. Lett.* **100**, 220502 (2008).
 - [21] R. König, S. Wehner, and J. Wullschlegel, *Unconditional security from noisy quantum storage*, *IEEE Trans. Inf. Theory* **58**, 1962 (2012).
 - [22] C. Erven, N. Ng, N. Gegov, R. Laflamme, S. Wehner, and G. Weihs, *An experimental implementation of oblivious transfer in the noisy storage model*, *Nat. Commun.* **5**, 3418 (2014).
 - [23] M. B. Santos, P. Mateus, and A. N. Pinto, *Quantum oblivious transfer: A short review*, *Entropy* **24**, 945 (2022).
 - [24] C. Li, B. Li, O. Amer, R. Shaydulin, S. Chakrabarti, G. Wang, H. Xu, H. Tang, I. Schoch, N. Kumar *et al.*, *Blind quantum machine learning with quantum bipartite correlator*, *Phys. Rev. Lett.* **133**, 120602 (2024).
 - [25] C. Crépeau, D. Gottesman, and A. Smith, *Secure multi-party quantum computation*, in *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, 2002), pp. 643–652.
 - [26] M. Ben-Or, C. Crépeau, D. Gottesman, A. Hassidim, and A. Smith, *Secure multiparty quantum computation with (only) a strict honest majority*, in *Proceedings of the 2006 47th Annual IEEE Symposium on Foundations of*

- Computer Science (FOCS'06)* (IEEE, New York, 2006), pp. 249–260.
- [27] F. Dupuis, J. B. Nielsen, and L. Salvail, *Actively secure two-party evaluation of any quantum operation*, in *Proceedings of the Annual Cryptology Conference* (Springer, New York, 2012) pp. 794–811.
- [28] Y. Dulek, A. B. Grilo, S. Jeffery, C. Majenz, and C. Schaffner, *Secure multi-party quantum computation with a dishonest majority*, in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques* (Springer, New York, 2020), pp. 729–758.
- [29] V. Lipinska, J. Ribeiro, and S. Wehner, *Secure multiparty quantum computation with few qubits*, *Phys. Rev. A* **102**, 022405 (2020).
- [30] A. Broadbent, J. Fitzsimons, and E. Kashefi, *Universal blind quantum computation*, in *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science* (IEEE, New York, 2009), pp. 517–526.
- [31] V. Dunjko, E. Kashefi, and A. Leverrier, *Blind quantum computing with weak coherent pulses*, *Phys. Rev. Lett.* **108**, 200502 (2012).
- [32] S. Barz, E. Kashefi, A. Broadbent, J. F. Fitzsimons, A. Zeilinger, and P. Walther, *Demonstration of blind quantum computing*, *Science* **335**, 303 (2012).
- [33] V. Giovannetti, L. Maccone, T. Morimae, and T. G. Rudolph, *Efficient universal blind quantum computation*, *Phys. Rev. Lett.* **111**, 230501 (2013).
- [34] J. F. Fitzsimons and E. Kashefi, *Unconditionally verifiable blind quantum computation*, *Phys. Rev. A* **96**, 012303 (2017).
- [35] J. Zeuner, I. Pitsios, S.-H. Tan, A. N. Sharma, J. F. Fitzsimons, R. Osellame, and P. Walther, *Experimental quantum homomorphic encryption*, *npj Quantum Inf.* **7**, 25 (2021).
- [36] S. Barz, J. F. Fitzsimons, E. Kashefi, and P. Walther, *Experimental verification of quantum computation*, *Nat. Phys.* **9**, 727 (2013).
- [37] H.-L. Huang, Q. Zhao, X. Ma, C. Liu, Z.-E. Su, X.-L. Wang, L. Li, N.-L. Liu, B. C. Sanders, C.-Y. Lu *et al.*, *Experimental blind quantum computing for a classical client*, *Phys. Rev. Lett.* **119**, 050503 (2017).
- [38] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, *Deep learning with limited numerical precision*, in *Proceedings of the International Conference on Machine Learning* (PMLR, Brookline, MA, 2015), pp. 1737–1746.
- [39] G. Wetzstein, A. Ozcan, S. Gigan, S. Fan, D. Englund, M. Soljačić, C. Denz, D. A. Miller, and D. Psaltis, *Inference in artificial intelligence with deep optics and photonics*, *Nature (London)* **588**, 39 (2020).
- [40] B. J. Shastri, A. N. Tait, T. Ferreira de Lima, W. H. Pernice, H. Bhaskaran, C. D. Wright, and P. R. Prucnal, *Photonics for artificial intelligence and neuromorphic computing*, *Nat. Photonics* **15**, 102 (2021).
- [41] P. L. McMahon, *The physics of optical computing*, *Nat. Rev. Phys.* **5**, 717 (2023).
- [42] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund *et al.*, *Deep learning with coherent nanophotonic circuits*, *Nat. Photonics* **11**, 441 (2017).
- [43] A. N. Tait, T. F. De Lima, E. Zhou, A. X. Wu, M. A. Nahmias, B. J. Shastri, and P. R. Prucnal, *Neuromorphic photonic networks using silicon photonic weight banks*, *Sci. Rep.* **7**, 7430 (2017).
- [44] F. Ashtiani, A. J. Geers, and F. Aflatouni, *An on-chip photonic deep neural network for image classification*, *Nature (London)* **606**, 501 (2022).
- [45] J. Feldmann, N. Youngblood, C. D. Wright, H. Bhaskaran, and W. H. Pernice, *All-optical spiking neurosynaptic networks with self-learning capabilities*, *Nature (London)* **569**, 208 (2019).
- [46] J. Feldmann, N. Youngblood, M. Karpov, H. Gehring, X. Li, M. Stappers, M. Le Gallo, X. Fu, A. Lukashchuk, A. S. Raja *et al.*, *Parallel convolutional processing using an integrated photonic tensor core*, *Nature (London)* **589**, 52 (2021).
- [47] Z. Chen, A. Sludds, R. Davis III, I. Christen, L. Bernstein, L. Ateshian, T. Heuser, N. Heermeier, J. A. Lott, S. Reitzenstein *et al.*, *Deep learning with coherent vcsel neural networks*, *Nat. Photonics* **17**, 723 (2023).
- [48] S. Bandyopadhyay, A. Sludds, S. Krastanov, R. Hamerly, N. Harris, D. Bunandar, M. Streshinsky, M. Hochberg, and D. Englund, *Single-chip photonic deep neural network with forward-only training*, *Nat. Photonics* **18**, 1335 (2024).
- [49] T. Wu, M. Menarini, Z. Gao, and L. Feng, *Lithography-free reconfigurable integrated photonic processor*, *Nat. Photonics* **17**, 710 (2023).
- [50] S. Pai, Z. Sun, T. W. Hughes, T. Park, B. Bartlett, I. A. Williamson, M. Minkov, M. Milanizadeh, N. Abebe, F. Morichetti *et al.*, *Experimentally realized in situ back-propagation for deep learning in photonic neural networks*, *Science* **380**, 398 (2023).
- [51] J. R. Basani, M. Heuck, D. R. Englund, and S. Krastanov, *All-photonic artificial-neural-network processor via non-linear optics*, *Phys. Rev. Appl.* **22**, 014009 (2024).
- [52] X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, Y. Luo, M. Jarrahi, and A. Ozcan, *All-optical machine learning using diffractive deep neural networks*, *Science* **361**, 1004 (2018).
- [53] T. Zhou, X. Lin, J. Wu, Y. Chen, H. Xie, Y. Li, J. Fan, H. Wu, L. Fang, and Q. Dai, *Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit*, *Nat. Photonics* **15**, 367 (2021).
- [54] L. G. Wright, T. Onodera, M. M. Stein, T. Wang, D. T. Schachter, Z. Hu, and P. L. McMahon, *Deep physical neural networks trained with backpropagation*, *Nature (London)* **601**, 549 (2022).
- [55] T. Wang, S.-Y. Ma, L. G. Wright, T. Onodera, B. C. Richard, and P. L. McMahon, *An optical neural network using less than 1 photon per multiplication*, *Nat. Commun.* **13**, 123 (2022).
- [56] L. Bernstein, A. Sludds, C. Panuski, S. Trajtenberg-Mills, R. Hamerly, and D. Englund, *Single-shot optical neural network*, *Sci. Adv.* **9**, eadg7904 (2023).
- [57] T. Wang, M. M. Sohoni, L. G. Wright, M. M. Stein, S.-Y. Ma, T. Onodera, M. G. Anderson, and P. L. McMahon, *Image sensing with multilayer nonlinear optical neural networks*, *Nat. Photonics* **17**, 408 (2023).

- [58] S.-Y. Ma, T. Wang, J. Laydevant, L. G. Wright, and P. L. McMahon, *Quantum-limited stochastic optical neural networks operating at a few quanta per activation*, *Nat. Commun.* **16**, 359 (2025).
- [59] S. Choi, Y. Salamin, C. Roques-Carnes, R. Dangovski, D. Luo, Z. Chen, M. Horodynski, J. Sloan, S. Z. Uddin, and M. Soljačić, *Photonic probabilistic machine learning using quantum vacuum noise*, *Nat. Commun.* **15**, 1 (2024).
- [60] X. Xu, M. Tan, B. Corcoran, J. Wu, A. Boes, T. G. Nguyen, S. T. Chu, B. E. Little, D. G. Hicks, R. Morandotti *et al.*, *11 tops photonic convolutional accelerator for optical neural networks*, *Nature (London)* **589**, 44 (2021).
- [61] R. Davis III, Z. Chen, R. Hamerly, and D. Englund, *Rf-photonic deep learning processor with shannon-limited data movement*, *Sci. Adv.* **11**, eadt3558 (2025).
- [62] R. Hamerly, L. Bernstein, A. Sludds, M. Soljačić, and D. Englund, *Large-scale optical neural networks based on photoelectric multiplication*, *Phys. Rev. X* **9**, 021032 (2019).
- [63] A. Sludds, S. Bandyopadhyay, Z. Chen, Z. Zhong, J. Cochrane, L. Bernstein, D. Bunandar, P. B. Dixon, S. A. Hamilton, M. Streshinsky *et al.*, *Delocalized photonic deep learning on the internet's edge*, *Science* **378**, 270 (2022).
- [64] A. S. Holevo, *Bounds for the quantity of information transmitted by a quantum communication channel*, *Probl. Peredachi Inf.* **9**, 3 (1973). [*Probl. Inf. Transm.* **9**, 177 (1973)].
- [65] C. R. Rao, *Information and the accuracy attainable in the estimation of statistical parameters*, in *Breakthroughs in Statistics: Foundations and Basic Theory* (Springer, New York, 1992), pp. 235–247.
- [66] T. C. Ralph, *All-optical quantum teleportation*, *Opt. Lett.* **24**, 348 (1999).
- [67] O. Oreshkov and T. A. Brun, *Weak measurements are universal*, *Phys. Rev. Lett.* **95**, 110409 (2005).
- [68] J. S. Lundeen and C. Bamber, *Procedure for direct measurement of general quantum states using weak measurement*, *Phys. Rev. Lett.* **108**, 070402 (2012).
- [69] Y.-S. Kim, J.-C. Lee, O. Kwon, and Y.-H. Kim, *Protecting entanglement from decoherence using weak measurement and quantum measurement reversal*, *Nat. Phys.* **8**, 117 (2012).
- [70] L. Zhang, A. Datta, and I. A. Walmsley, *Precision metrology using weak measurements*, *Phys. Rev. Lett.* **114**, 210801 (2015).
- [71] K. Sulimany, S. K. Vadlamani, R. Hamerly, P. Iyengar, and D. Englund, *Quantum-secure multiparty deep learning*, <https://github.com/kfirsuli/Quantum-secure-multiparty-deep-learning> (2024).
- [72] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, *The security of practical quantum key distribution*, *Rev. Mod. Phys.* **81**, 1301 (2009).
- [73] F. Xu, X. Ma, Q. Zhang, H.-K. Lo, and J.-W. Pan, *Secure quantum key distribution with realistic devices*, *Rev. Mod. Phys.* **92**, 025002 (2020).
- [74] F. Laudenbach, C. Pacher, C.-H. F. Fung, A. Poppe, M. Peev, B. Schrenk, M. Hentschel, P. Walther, and H. Hübel, *Continuous-variable quantum key distribution with Gaussian modulation—The theory of practical implementations*, *Adv. Quantum Technol.* **1**, 1800011 (2018).
- [75] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani *et al.*, *Advances in quantum cryptography*, *Adv. Opt. Photonics* **12**, 1012 (2020).
- [76] K. Sulimany, G. Pelc, R. Dudkiewicz, S. Korenblit, H. S. Eisenberg, Y. Bromberg, and M. Ben-Or, *High-dimensional coherent one-way quantum key distribution*, *npj Quantum Inf.* **11**, 16 (2025).
- [77] B. A. Bash, A. H. Gheorghe, M. Patel, J. L. Habif, D. Goeckel, D. Towsley, and S. Guha, *Quantum-secure covert communication on bosonic channels*, *Nat. Commun.* **6**, 8626 (2015).
- [78] E. Polino, M. Valeri, N. Spagnolo, and F. Sciarrino, *Photonic quantum metrology*, *AVS Quantum Sci.* **2**, 024703 (2020).
- [79] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, *A survey of quantization methods for efficient neural network inference*, in *Low-Power Computer Vision* (Chapman and Hall/CRC, Boca Raton, 2022), pp. 291–326.
- [80] G. P. Agrawal, *Fiber-Optic Communication Systems* (John Wiley & Sons, New York, 2012).
- [81] L. d. Moura and S. Ullrich, *The Lean 4 theorem prover and programming language*, in *Automated Deduction—CADE 28: Proceedings of the 28th International Conference on Automated Deduction, Virtual Event, 2021* (Springer, New York, 2021), pp. 625–635.
- [82] B. Breen, M. Del Tredici, J. McCarran, J. Aspuru Mijares, W. W. Yin, K. Sulimany, J. M. Taylor, F. H. L. Koppens, and D. Englund, *AX-Prover: A deep reasoning agentic framework for theorem proving in mathematics and quantum physics*, [arXiv:2510.12787](https://arxiv.org/abs/2510.12787).
- [83] K. Sulimany and Y. Bromberg, *All-fiber source and sorter for multimode correlated photons*, *npj Quantum Inf.* **8**, 4 (2022).
- [84] R. Nehra, R. Sekine, L. Ledezma, Q. Guo, R. M. Gray, A. Roy, and A. Marandi, *Few-cycle vacuum squeezing in nanophotonics*, *Science* **377**, 1333 (2022).
- [85] Y. Xia, W. Li, Q. Zhuang, and Z. Zhang, *Quantum-enhanced data classification with a variational entangled sensor network*, *Phys. Rev. X* **11**, 021047 (2021).
- [86] Q. Zhuang and Z. Zhang, *Physical-layer supervised learning assisted by an entangled sensor network*, *Phys. Rev. X* **9**, 041023 (2019).
- [87] Y. Bloom, Y. Ordan, T. Levin, K. Sulimany, E. G. Bowes, J. A. Hollingsworth, and R. Rapaport, *Decoy-state and purification protocols for superior quantum key distribution with imperfect quantum-dot-based single-photon sources: Theory and experiment*, *PRX Quantum* **6**, 030332 (2025).
- [88] A. W. Harrow, A. Hassidim, and S. Lloyd, *Quantum algorithm for linear systems of equations*, *Phys. Rev. Lett.* **103**, 150502 (2009).
- [89] P. Rebertost, M. Mohseni, and S. Lloyd, *Quantum support vector machine for big data classification*, *Phys. Rev. Lett.* **113**, 130503 (2014).

- [90] A. Skolik, J. R. McClean, M. Mohseni, P. Van Der Smagt, and M. Leib, *Layerwise learning for quantum neural networks*, *Quantum Mach. Intell.* **3**, 1 (2021).
- [91] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, *Quantum machine learning*, *Nature (London)* **549**, 195 (2017).
- [92] I. Cong, S. Choi, and M. D. Lukin, *Quantum convolutional neural networks*, *Nat. Phys.* **15**, 1273 (2019).
- [93] J. Lin, T. Upadhyaya, and N. Lütkenhaus, *Asymptotic security analysis of discrete-modulated continuous-variable quantum key distribution*, *Phys. Rev. X* **9**, 041064 (2019).
- [94] F. Kanitschar, I. George, J. Lin, T. Upadhyaya, and N. Lütkenhaus, *Finite-size security for discrete-modulated continuous-variable quantum key distribution protocols*, *PRX Quantum* **4**, 040306 (2023).
- [95] S. Ghorai, P. Grangier, E. Diamanti, and A. Leverrier, *Asymptotic security of continuous-variable quantum key distribution with a discrete modulation*, *Phys. Rev. X* **9**, 021059 (2019).
- [96] W. Bogaerts, D. Pérez, J. Capmany, D. A. Miller, J. Poon, D. Englund, F. Morichetti, and A. Melloni, *Programmable photonic circuits*, *Nature (London)* **586**, 207 (2020).
- [97] E. Polino, M. Riva, M. Valeri, R. Silvestri, G. Corrielli, A. Crespi, N. Spagnolo, R. Osellame, and F. Sciarrino, *Experimental multiphase estimation on a chip*, *Optica* **6**, 288 (2019).
- [98] V. Saggio, B. E. Asenbeck, A. Hamann, T. Strömberg, P. Schiansky, V. Dunjko, N. Friis, N. C. Harris, M. Hochberg, D. Englund *et al.*, *Experimental quantum speed-up in reinforcement learning agents*, *Nature (London)* **591**, 229 (2021).
- [99] I. Agresti, N. Viggianiello, F. Flamini, N. Spagnolo, A. Crespi, R. Osellame, N. Wiebe, and F. Sciarrino, *Pattern recognition techniques for boson sampling validation*, *Phys. Rev. X* **9**, 011013 (2019).
- [100] D. Marpaung, J. Yao, and J. Capmany, *Integrated micro-wave photonics*, *Nat. Photonics* **13**, 80 (2019).
- [101] P. Jouguet, S. Kunz-Jacques, A. Leverrier, P. Grangier, and E. Diamanti, *Experimental demonstration of long-distance continuous-variable quantum key distribution*, *Nat. Photonics* **7**, 378 (2013).
- [102] N. Jain, H.-M. Chin, H. Mani, C. Lupo, D. S. Nikolic, A. Kordts, S. Pirandola, T. B. Pedersen, M. Kolb, B. Ömer *et al.*, *Practical continuous-variable quantum key distribution with composable security*, *Nat. Commun.* **13**, 4740 (2022).
- [103] Y. Zhang, Z. Chen, S. Pirandola, X. Wang, C. Zhou, B. Chu, Y. Zhao, B. Xu, S. Yu, and H. Guo, *Long-distance continuous-variable quantum key distribution over 202.81 km of fiber*, *Phys. Rev. Lett.* **125**, 010502 (2020).
- [104] K. R. Motes, A. Gilchrist, J. P. Dowling, and P. P. Rohde, *Scalable boson sampling with time-bin encoding using a loop-based architecture*, *Phys. Rev. Lett.* **113**, 120501 (2014).
- [105] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, *Experimental realization of any discrete unitary operator*, *Phys. Rev. Lett.* **73**, 58 (1994).
- [106] N. K. Fontaine, R. Ryf, H. Chen, D. T. Neilson, K. Kim, and J. Carpenter, *Laguerre-gaussian mode sorter*, *Nat. Commun.* **10**, 1865 (2019).
- [107] O. Lib, K. Sulimany, and Y. Bromberg, *Processing entangled photons in high dimensions with a programmable light converter*, *Phys. Rev. Appl.* **18**, 014063 (2022).
- [108] O. Lib, K. Sulimany, M. Araújo, M. Ben-Or, and Y. Bromberg, *High-dimensional quantum key distribution using a multi-plane light converter*, *Opt. Quantum* **3**, 182 (2025).
- [109] Y. He, X. Ding, Z.-E. Su, H.-L. Huang, J. Qin, C. Wang, S. Unsleber, C. Chen, H. Wang, Y.-M. He *et al.*, *Time-bin-encoded boson sampling with a single-photon device*, *Phys. Rev. Lett.* **118**, 190501 (2017).
- [110] J. M. Arrazola, V. Bergholm, K. Brádler, T. R. Bromley, M. J. Collins, I. Dhand, A. Fumagalli, T. Gerrits, A. Goussev, L. G. Helt *et al.*, *Quantum circuits with many photons on a programmable nanophotonic chip*, *Nature (London)* **591**, 54 (2021).
- [111] K. Yonezu, Y. Enomoto, T. Yoshida, and S. Takeda, *Time-domain universal linear-optical operations for universal quantum information processing*, *Phys. Rev. Lett.* **131**, 040601 (2023).
- [112] C. Cui, J. Postlewaite, B. N. Saif, L. Fan, and S. Guha, *Superadditive communication with the green machine as a practical demonstration of nonlocality without entanglement*, *Nat. Commun.* **16**, 3760 (2025).
- [113] C. Cui, J. R. Basani, J. Postlewaite, B. N. Saif, L. Fan, E. Waks, and S. Guha, *Arbitrary unitary transformation over time-binned optical modes and the demonstration of superadditive communication*, in *Quantum Computing, Communication, and Simulation V* (SPIE, Bellingham, WA, 2025), Vol. 13391, pp. 11–17.
- [114] M. O. Scully and M. S. Zubairy, *Quantum Optics* (Cambridge University Press, Cambridge, England, 1997).
- [115] M. G. Baccvanski, S. K. Vadlamani, K. Sulimany, and D. R. Englund, *Qamnet: Fast and efficient optical QAM neural networks*, *arXiv:2409.12305*.
- [116] A. Leverrier and P. Grangier, *Unconditional security proof of long-distance continuous-variable quantum key distribution with discrete modulation*, *Phys. Rev. Lett.* **102**, 180504 (2009).
- [117] R. García-Patrón and N. J. Cerf, *Unconditional optimality of Gaussian attacks against continuous-variable format quantum key distribution*, *Phys. Rev. Lett.* **97**, 190503 (2006).
- [118] M. M. Wolf, G. Giedke, and J. I. Cirac, *Extremality of Gaussian quantum states*, *Phys. Rev. Lett.* **96**, 080502 (2006).
- [119] C. Lyle, M. van der Wilk, M. Kwiatkowska, Y. Gal, and B. Bloem-Reddy, *On the benefits of invariance in neural networks*, *arXiv:2005.00178*.
- [120] E. Kauderer-Abrams, *Quantifying translation-invariance in convolutional neural networks*, *arXiv:1801.01450*.
- [121] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, *Exploring generalization in deep learning*, in *Advances in Neural Information Processing Systems*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., Red Hook, NY, 2017), Vol. 30.
- [122] B. Neyshabur, R. R. Salakhutdinov, and N. Srebro, *Pathsgd: Path-normalized optimization in deep neural networks*, in *Advances in Neural Information Processing*

- Systems, Vol. 28, edited by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Curran Associates, Inc., 2015).
- [123] C. Orlandi, A. Piva, and M. Barni, *Oblivious neural network computing via homomorphic encryption*, *EURASIP J. Inf. Secur.* **2007**, 1 (2007).
- [124] A. Pananjady, M. J. Wainwright, and T. A. Courtade, *Linear regression with shuffled data: Statistical and computational limits of permutation recovery*, *IEEE Trans. Inf. Theory* **64**, 3286 (2017).
- [125] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta, *Amplification by shuffling: From local to central differential privacy via anonymity*, in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms* (SIAM, Philadelphia, 2019), pp. 2468–2479.
- [126] V. Feldman, A. McMillan, and K. Talwar, *Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling*, in *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, New York, 2022), pp. 954–964.
- [127] V. Feldman, A. McMillan, and K. Talwar, *Stronger privacy amplification by shuffling for rényi and approximate differential privacy*, in *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (SIAM, Philadelphia, 2023), pp. 4966–4981.
- [128] A. Van der Vaart, 10.2 Bernstein–von Mises theorem, *Asymptotic Statistics* (1998).