Benchmarking quantum tomography completeness and fidelity with machine learning

Yong Siah Teo,[1,*] Seongwook Shin,[1] Hyunseok Jeong,[1,†] Yosep Kim,[2] Yoon-Ho Kim,[2,‡] Gleb I. Struchalin,[3] Egor V. Kovlakov,[3] Stanislav S. Straupe,[3] Sergei P. Kulik,[3] Gerd Leuchs,[4,5] and Luis L. Sánchez-Soto[4,6,§]

[1]*Department of Physics and Astronomy, Seoul National University, 08826 Seoul, South Korea*
[2]*Department of Physics, Pohang University of Science and Technology (POSTECH), 37673 Pohang, Korea*
[3]*Quantum Technologies Centre, and Faculty of Physics, Moscow State University, 119991 Moscow, Russia*
[4]*Max-Planck-Institut für die Physik des Lichts, Staudtstraße 2, 91058 Erlangen, Germany*
[5]*Institute of Applied Physics, Russian Academy of Sciences, 603950 Nizhny Novgorod, Russia*
[6]*Departamento de Óptica, Facultad de Física, Universidad Complutense, 28040 Madrid, Spain*

We train convolutional neural networks to predict whether or not a set of measurements is informationally complete to uniquely reconstruct any given quantum state with no prior information. In addition, we perform fidelity benchmarking based on this measurement set without explicitly carrying out state tomography. The networks are trained to recognize the fidelity and a reliable measure for informational completeness through collective encoding of quantum measurements, data and target states into grayscale images. By gradually accumulating measurements and data, these convolutional networks can efficiently certify a low-measurement-cost quantum-state characterization scheme. We confirm the potential of this machine-learning approach by presenting experimental results for both spatial-mode and multiphoton systems of large dimensions. These predictions are further shown to improve with noise recognition when the networks are trained with additional bootstrapped training sets from real experimental data.

I. INTRODUCTION

Recent advances in quantum algorithms and error correction [1–6] have fueled the development of noisy intermediate-scale quantum computing devices. This progress requires an efficient assessment of the relevant quantum systems [6–8], gates [9–13] and measurements [14–21]. Toolkits developed in quantum tomography [22–33] have concomitantly evolved into modern schemes appropriate for characterizing those components efficiently. A notable branch of schemes attempt to cope with a large number of qubits by directly estimating quantum properties [34–42].

As typical quantum tasks involve pure states, unitary gates, and projective measurements, there also exists a series of compressed-sensing-related proposals [43–50] that fully reconstruct low-rank quantum components with few measurement resources. However, they rely on prior knowledge about the rank, which often turns out to be unreliable in practice because of noise. Very recently, compressive-tomography methods without assuming any prior information has been developed and applied to the individual low-resource characterization of quantum states, processes and measurements [51–55]. A crucial ingredient in these methods is informational completeness certification that determines whether or not a given measurement set and its corresponding data is informationally complete (IC). This is done by computing a uniqueness measure based on the given measurements. Such a computation can be performed with classical semidefinite programs (SDPs) [56] of (worst-case) polynomial-time complexities.

Like any tomography scheme that invokes rounds of optimization routines, an accumulation of errors can occur in real experiments while running SDPs on-the-fly. As a practically feasible solution, we propose to train an artificial neural network to verify the IC property for a set of quantum measurements and corresponding raw data. We further introduce an auxiliary network to be used concurrently for us to validate the fidelity of the unknown state for the given measurement set without carrying out explicit reconstruction. Once a set of IC measurement data is collected, it takes only one final round of state reconstruction to obtain the unique physical estimator, if so desired.

Network training can be done offline using simulated noisy datasets, and the stored network model can later be retrieved and used in real experiments with statistical noise. More specifically, a convolutional neural-network (CNN) architecture shall be used for training and prediction. Among other kinds of networks that have been widely adopted by the quantum-information community [57–60], this is a popular network architecture that is used in image-pattern recognition [61–64], with boosted support by a recent universality proof [65] that such networks can indeed forecast any continuous function mapping. Both its classical application and quantum analog have also gained traction in quantum-information science [66–69].

In this work, we train an *informational completeness certification net* (ICCNet) and a *fidelity prediction net* (FidNet), each made up of a sequence of convolution and pooling neural layers that is reasonably deep. Partnered with FidNet for direct fidelity benchmarking without the need for state tomography, ICCNet constitutes the foundational core for deciding if the given measurement resources are sufficient to uniquely characterize any unknown state in real experimental situations. Neural-network training is versatile in the sense that noise models may be incorporated into the training procedure to improve the predictive power of the networks. After offline training, the network models can heavily reduce the computa-

* yong.siah.teo@gmail.com
† h.jeong37@gmail.com
‡ yoonho72@gmail.com
§ lsanchez@fis.ucm.es

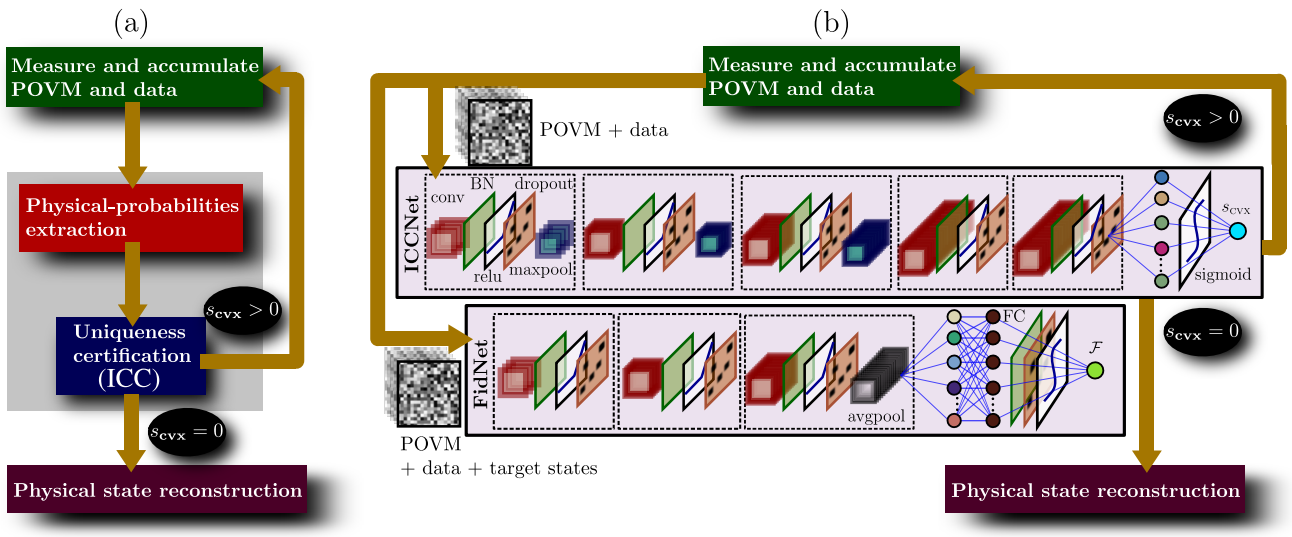


FIG. 1. The physical-probabilities extraction and SDP-based ICC of a resource-efficient quantum-state tomography scheme in (a) may be entirely replaced by the ICCNet and FidNet shown in (b), each of which is a sequence of convolutional blocks (shown here for $d = 16$ as an example). Each convolutional block typically consists of a convolutional layer (conv), a batch normalization layer (BN), the relu activation layer, a dropout layer and a pooling layer (maxpool or avgpool) (more details in Sec. II B). Specific network structures may vary for systems of different dimensions. Numerical values after the convolutional blocks are flattened and activated with the sigmoid function just before the $s_{\rm CVX}$ computation, and passed through a fully-connected (FC) layer before the fidelity $\mathcal{F}$ computation.

tion time of the uniqueness certification by orders of magnitude for large dimensions while running the experiments. This essentially realizes a compressive tomography scheme that is drift-proof, comprising a highly efficient uniqueness certification and fidelity-benchmarking protocols.

Apart from Monte Carlo simulations, we also use real data obtained from two separate groups of experiments to demonstrate that the resulting trained network models can predict the average behaviors of both the IC property and fidelity very well, despite the presence of errors and experimental imperfections. We also show that performances in predicting both properties can be further boosted when the neural networks are trained with additional bootstrapped experimental datasets. Such enhancements highlight the versatility of general neural networks, which can carry out noise recognition tasks to reduce noisy prediction artifacts.

II. BACKGROUND

A. Ascertaining informational completeness

The main procedure for certifying whether a generic measurement dataset is sufficient to unambiguously determine an unknown quantum state can be represented as a simple flowchart in Fig. 1(a). Given a positive operator-valued measure (POVM) that models the measurements performed, the corresponding data counts are noisy due to statistical fluctuation arising from finite data samples. Proper data analysis first entails the extraction of physical probabilities from the accumulated data, which can be done with well-established statistical methods, such as those of maximum likelihood (ML) [23, 24, 70–72] and least squares (LS) [73, 74], subject to the physical constraint of density matrices (refer also to Sec. II A).

After obtaining the physical probabilities, one may proceed to evaluate the measurements and find out whether they are IC. More specifically, a uniqueness indicator $0 \leq s_{\rm CVX} \leq 1$ can be directly computed from the POVM and data with the help of SDPs—the *informational completeness certification* (ICC). When $s_{\rm CVX} > 0$, there is equivalently a convex set of state estimators that are consistent with the physical probabilities. It can be shown [51] that a unique estimator is obtained from the measured POVM and corresponding data if and only if $s_{\rm CVX} = 0$.

Bottom-up resource-efficient quantum-state tomography is thus an iterative program involving rounds of extracting physical probabilities from the measurement data and certifying uniqueness based on these probabilities. At each round, the computed $s_{\rm CVX}$ is used to decide whether new measurements are needed in the next one. In this manner, the POVM outcomes may be accumulated bottom-up until $s_{\rm CVX} = 0$, after which a physical state reconstruction using either the ML or LS scheme is carried out to obtain the unique estimator. The size of the resulting IC POVM is minimized accordingly. We remark that ICC turns out to be the limiting procedure in practical implementation relative to a typical quantum-state reconstruction. This is because an estimation over the space of quantum states can be very efficiently implemented with an iterative scheme, where each step involves a regular gradient update and just one round of convex projection [72]. (The case for quantum processes has also been discussed [75]). On the other hand, satisfying both Born's rule and quantum positivity constraint in ICC requires a separate iterative procedure just to carry out the correct convex projection onto their inter-

section [76]. To date, there exists no efficient way to perform projections of these constraints to the authors' knowledge.

By recalling the results in Refs. [51, 52], we briefly describe the simple procedures that deterministically verify whether a set of POVM outcomes $\{\Pi_j \geq 0\}$ is IC given their corresponding set of relative frequency data $\{\nu_j\}$. The first necessary step is to acquire the physical probabilities from ν_j. To this end, we consider two popular choices often considered in quantum tomography, namely the ML and LS methods. In ML, we maximize the log-likelihood function $\log L$ that best describes the physical scenario over the quantum state space. Since we predominantly discuss von Neumann measurement bases, each basis induces a multinomial distribution such that we have the form $\log L \propto \sum_j \nu_j \log p'_j$, where $p'_j = \mathrm{tr}\{\rho'\Pi_j\}$ are our sought-after physical probabilities to be optimized over the operator space in which $\rho' \geq 0$ and $\mathrm{tr}\{\rho'\} = 1$. In LS, which we have adopted to deal with arbitrary projective measurements that do not sum to the identity operator in general, the distance $\mathcal{D} = \|\nu_j - p'_j\|^2$ is minimized with respect to p'_j over the space of $\rho' \geq 0$, this time with the unit-trace constraint relaxed and later reinstated after the minimization is completed.

Using the obtained physical probability estimators $\widehat{p}_j$ through the aforementioned optimization strategies, we can now define and fix a randomly generated full-rank state Z and conduct the following two SDPs:

$$\begin{aligned} &\text{minimize/maximize } f = \mathrm{tr}\{\rho' Z\} \text{ over } \rho' \\ &\text{subject to:} \\ &\rho' \geq 0\,, \quad \mathrm{tr}\{\rho'\} = 1\,, \quad \mathrm{tr}\{\rho'\Pi_j\} = \widehat{p}_j\,. \end{aligned} \tag{2.1}$$

It is now clear why the SDPs are to be carried out with the physical probabilities $\widehat{p}_j$ instead of raw data ν_j: the relative frequencies ν_j are statistically noisy and in general do not correspond to a feasible solution set in (2.1). It has been shown that when $s_{\mathrm{cvx}} \equiv f_{\max} - f_{\min}$ is zero, this implies that any quantum-state estimator reconstructed from $\{\Pi_j\}$ and $\{\nu_j\}$ is unique and equal to the solution for (2.1).

B. Training the ICCNet and FidNet

We propose to tackle the combined problem of physical probabilities extraction and uniqueness certification by predicting with trained neural networks. We also demonstrate the possibility of performing fidelity evaluation on the reconstruction with such neural networks without explicitly carrying out physical state tomography. To do this, we introduce the ICCNet and FidNet, illustrated in Fig. 1(b), where each possesses a convolutional network architecture that analyzes the given POVM and data in grayscale images. Such a treatment allows one to train the networks with far less trainable parameters to recognize s_{cvx} and fidelity $\mathcal{F}(\widehat{\rho}, \rho_{\mathrm{targ}}) = \mathrm{tr}\left\{\sqrt{\sqrt{\widehat{\rho}}\,\rho_{\mathrm{targ}}\sqrt{\widehat{\rho}}}\right\}^2$ between the state estimator $\widehat{\rho}$ and some target state ρ_{targ} as compared to using, for instance, the multilayer perceptron (feed-forward) architecture [77, 78] that consists only of fully-connected or dense layers. For consistency, $\widehat{\rho}$ shall always be taken as the ML estimator that minimizes the linear function in (2.1).

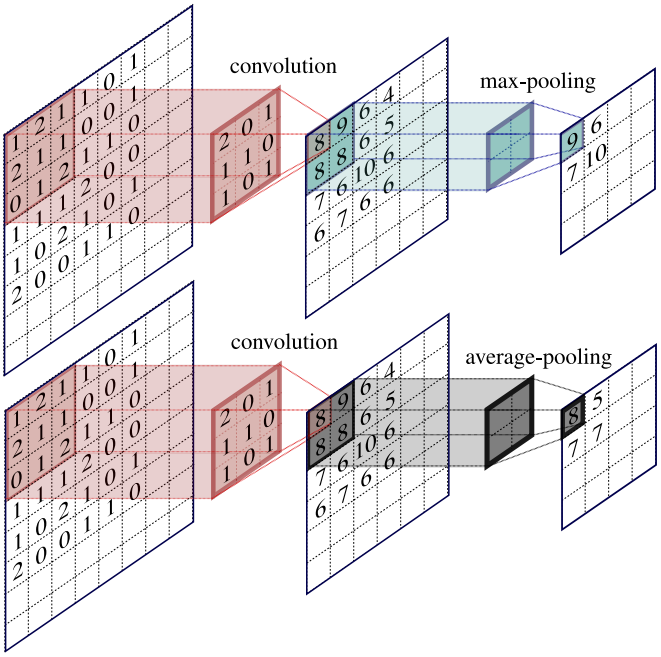
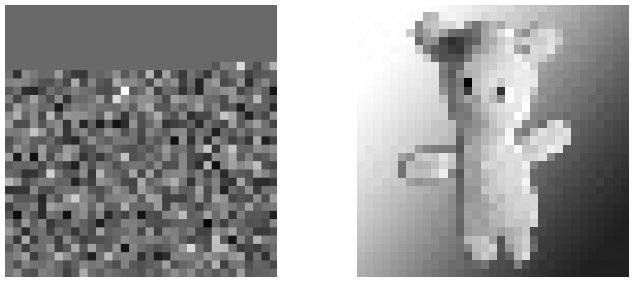


FIG. 2. Operations carried out by the convolution and pooling layers. In this example, the 8×8 input layer is reduced to a 6×6 output layer after going through a convolution layer consisting of a single 3×3 filter array of trainable parameters that takes stride 1. This output layer becomes the input layer with respect to either the max-pooling or average-pooling layer that each consists of a single 2×2 filter array of stride 2. The final output layer (rounded off for illustration purposes) is therefore a 4×4 numerical array.

For predicting s_{cvx} and $\mathcal{F}$, both ICCNet and FidNet employ a sequence of two-dimensional array manipulating layers. Two important types of layers responsible for these operations are the convolution layer, which are two-dimensional filters that carry out multiplicative convolutions with the layer input numerical array, and the pooling layer that generally down-samples a layer input array into a smaller output array with a simple numerical-summarizing computation. To each convolution layer, an activation function is applied to further introduce nonlinear characteristics for predicting general network output functions.

The convolutional ICCNet and FidNet take on a similar architecture, which consists of convolution, max-pooling and average-pooling layers. Each convolution layer consists of n_{f} filters, where each filter is a 3×3 array window that slides vertically and across layer input arrays with stride 1 in both directions. We design the sequence of convolution layers to have an exponentially increasing n_{f} with the network depth. A max-pooling layer is a type of pooling layer that picks the maximum number from the layer input within a selected window. Similarly, the average-pooling layer computes average values over the selected window. These pooling layers are generally responsible for shrinking the layer input array to a smaller layer output array. The actions of all types of layers are summarized in Fig. 2. We insert the default rectified



(a) (b)

FIG. 3. A juxtaposition of (a) a 33×33 pixelated ICCNet input-data image, which encodes a four-qubit POVM containing $K = 4$ bases and corresponding probabilities, and (b) a down-sampled photograph of a stuffed toy of the same resolution.

linear unit ("relu") activation function after every convolution layer, which is defined as $f_{\text{relu}}(x) = \max(0, x)$. At the end of ICCNet and FidNet, the respective output values are computed with the sigmoid activation function given by $f_{\text{sigmoid}}(x) = 1/(1 + \mathrm{e}^{-x})$.

Overfitting can be an issue in machine learning, in which case the neural networks are prone to fitting training datasets much better than unseen ones. It is therefore essential to regulate network training by keeping the problem of overfitting in check so that the resulting trained models have high predictive power. This problem often arises when the neural network is deep. The addition of dropout layers has been proven to be an effective method for combating overfitting [79–81], which randomly exclude trainable parameters. More recently, it has been demonstrated that neural-network training can be further enhanced by adding batch normalization layers. This was supported not only by the initial observation that the distribution of layer input values are stabilized with batch normalization [82], but also by the even more relevant finding that the gradient landscape of the network loss function (the figure of merit quantifying the difference between the actual output value and that computed by the network) seen by the optimization routine that trains the network becomes smoother [83], making training much more stable.

All trainable parameters in the relevant neural layers of ICCNet and FidNet are optimized using a variant of stochastic gradient descent known as NAdam [84], where the network gradients are computed in batches of the training data. To prepare ICCNet training input datasets, for von Neumann measurements of a fixed number (K) of bases considered in Secs. IV A and IV B, the initial network input $\boldsymbol{X}$ is an $m \times K(d^2 + d)$ matrix that contains m training datasets, each recording the K measured bases and corresponding relative frequencies $\{\nu_{jk}\}_{j=0}^{d-1}{}_{k=1}^{K}$ ($\sum_j \nu_{jk} = 1$). To encode the measurement bases, we regard all bases as some unitary rotation $U_k |j\rangle \langle j| U_k^\dagger$ of the standard computational basis $\{|j\rangle\}_{j=0}^{d-1}$, where $U_1 = 1$. These unitary operators are then logarithmized in order to obtain their Hermitian exponents $H_k = -\mathrm{i} \log U_k$ ($H_1 = 0$), from which the diagonals and upper triangular real and imaginary matrix elements are extracted. Each row of $\boldsymbol{X}$ is thus a flattened $K(d^2 + d)$-dimensional row of real numerical values formatted properly to encode U_1, U_2, …, U_K, ν_{01}, …, $\nu_{d-1\,1}$, ν_{02}, …, $\nu_{d-1\,2}$, …, $\nu_{0\,K}$, …, $\nu_{d-1\,K}$ in this order. This input matrix is then processed into a $\lceil\sqrt{K(d^2+d)}\rceil \times \lceil\sqrt{K(d^2+d)}\rceil$ grayscale image to be fed to ICCNet (see Fig. 3). Similarly, for a fixed set of L projective measurements discussed in Sec. IV C, analogous arguments lead to the necessary $\lceil\sqrt{L(d^2+1)}\rceil \times \lceil\sqrt{L(d^2+1)}\rceil$ input grayscale images. For each dimension, the randomly generated full-rank state Z needed to solve (2.1) is fixed during the training and testing stages.

On the other hand, training the FidNet requires input information about not only the measured bases (or projectors) and their corresponding data, but also the additional m target states to be included as inputs, one for each dataset. The correct dimensions of the input grayscale images are $\lceil\sqrt{(K+1)d^2 + Kd}\rceil \times \lceil\sqrt{(K+1)d^2 + Kd}\rceil$ or $\lceil\sqrt{(L+1)d^2 + L}\rceil \times \lceil\sqrt{(L+1)d^2 + L}\rceil$ respectively for basis and projective measurements. We note that to predict fidelities for simulated test datasets of $d = 16$, 32 and 64 as shown in Fig. 7 and 8, FidNet training is done with target states defined by the true states that generated the simulated training datasets. For all experimental results in Fig. 9, training is carried out simultaneously with the target states derived from the corresponding true states and those that deviate from them in order to account for systematic errors more effectively and improve average prediction accuracy. The list of hyperparameters that define the architectures of ICCNet and FidNet, as well as the technical analyses of network input-data generation and network training are given in Appendices B and C.

III. EXPERIMENTS

A. Spatial-mode photonic systems

Apart from evaluating simulation test datasets, we also run the trained ICCNet and FidNet models to benchmark real experimental datasets. In the first group of experiments, we showcase the accuracy of ICCNet and FidNet predictions on experimental data acquired from an attenuated laser source prepared in quantum states projected onto Hermite-Gaussian spatial-mode bases of various dimensions d. With this group of experiments, for the sake of variety, we shall consider measurement bases that are obtained from adaptive compressive tomography (ACT). These are eigenbases of the state that minimizes the von Neumann entropy subject to the same SDP constraints in (2.1). It has been demonstrated that successive measurements of such eigenbases result in a fast convergence of s_{CVX} [51, 52]. An explicit protocol to construct these bases is given in Appendix A.

The Hilbert space of photonic spatial degrees of freedom is typically discretized using an appropriate basis of transverse modes. To produce high-dimensional quantum states we attenuate an 808-nm diode laser, filter the resulting radiation with a single-mode optical fiber and then adjust the spatial structure of the light field with a spatial light modulator (SLM, see Fig. 4). The holographic approach [85] allows us

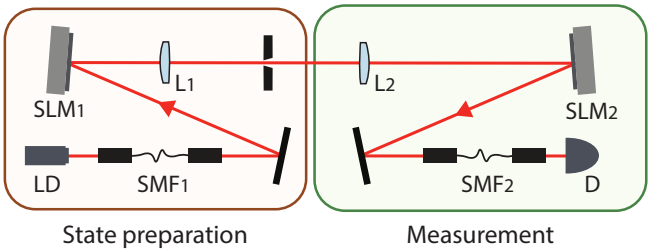
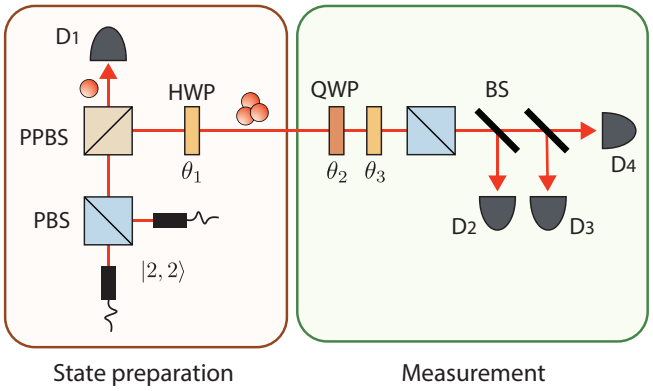


FIG. 4. Experimental scheme to generate and characterize spatial photon states. Attenuated radiation of laser diode (LD) is spatially filtered by a single-mode optical fiber (SMF1) and directed on the first spatial light modulator (SLM1). Hologram displayed on the SLM1 transforms the fundamental fiber mode into the desired superposition of Hermite-Gaussian beams defining the particular quantum state of photons. The iris placed in the middle of the telescope with unit magnification (lenses L1 and L2) is used to clean the structured beam from the undiffracted light by selection of the first order of diffraction at the far-field plane of the SLM1. The second light modulator (SLM2) followed by a single-mode optical fiber (SMF2) and a single photon counter (D) plays a role of spatial detector, which realize a projective measurement by the right choice of a hologram on the second SLM display.



(θ_2, θ_3)	$(22.0°, 17.7°)$, $(169.9°, 73.3°)$, $(45.0°, -31.5°)$, $(-135.0°, -67.5°)$, $(-65.0°, -32.5°)$, $(0.0°, 45.0°)$	$(169.9°, 96.6°)$, $(22.0°, 4.3°)$, $(45.0°, -13.5°)$, $(68.0°, 40.7°)$, $(100.1°, 38.4°)$,	$(-25.0°, -12.5°)$, $(135.0°, -85.5°)$, $(135.0°, -49.5°)$, $(100.1°, 61.7°)$, $(68.0°, 27.3°)$,

FIG. 5. Experimental scheme to generate and characterize three-photon states. Two horizontally polarized photons and two vertically polarized photons, produced by the double-pair emission of non-collinear spontaneous parametric down-conversion process, are spatially combined with a PBS, thereby producing the four-photon state $|2,2\rangle\langle 2,2|$. After detecting a single photon at detector D_1, the reflected three-photon system from a PPBS are prepared in a particular quantum state, determined by the HWP angle θ_1. For state characterization, four-fold coincidence counts at detectors D_1, D_2, D_3, and D_4 are acquired for all 16 rank-one projectors pictorialized in Fig. 6. These measurement projectors are determined by the HWP and QWP angles of θ_2 and θ_3 in the table with a PBS and BS.

to transform the incident light into arbitrary transverse modes by controlling the phase pattern on the SLM's display.

We work with Hermite-Gaussian (HG) modes $\mathrm{HG}_{nm}(x,y)$, which are the solutions to the Helmholtz equation in Cartesian coordinates (x,y) and form a complete orthonormal basis. By bounding the sum of beam orders $n+m$, we restrict the dimension of the generated quantum systems. Since holograms displayed on the SLM make use of a blazed grating, in order to select the first diffraction order, we place an iris in the middle of the telescope, where different diffraction orders are well separated. Using a second SLM, a single-mode optical fiber, followed by a single photon counting module, we realize a well-known technique of projective measurements in the spatial-mode space [86]. These allows us to also conveniently implement general ACT basis measurements in arbitrary dimensions.

B. Multiphoton systems

In the second group of experiments, we switch to a different flavor of informational completeness by discussing two-mode photon-number states. In particular, we look at quantum states of up to three photons occupying two optical modes. Such three-photon states were of interest in the study of high-order quantum polarization properties beyond the Stokes vectors [87]. The resulting Hilbert space is effectively 4-dimensional and spanned by the basis $\{|n_{\mathrm{H}}, n_{\mathrm{V}}\rangle\}_{n_{\mathrm{H}}+n_{\mathrm{V}}=3} = \{|0,3\rangle, |1,2\rangle, |2,1\rangle, |3,0\rangle\}$. Here n_{H} and n_{V} denote the number of photons in the horizontal and vertical polarization modes, respectively.

To perform tomography on the multiphoton quantum states, expectation values of a set of 16 rank-one projectors are measured. In principle, any set of 16 linearly independent projectors are suitable for a complete characterization of arbitrary 4-dimensional states *without* ICC. For these experiments, we define each projector Π_j by a ket $b_j^{\dagger 3}|0,0\rangle/\sqrt{6}$, where $b_j^\dagger$ and the other unobserved counterpart $c_j^\dagger$ are photonic creation operators derived from an SU(2) unitary operator $\widetilde{U}_j$ according to the transformation

$$\begin{pmatrix} b_j^\dagger \\ c_j^\dagger \end{pmatrix} = \widetilde{U}_j \begin{pmatrix} a_{\mathrm{H}}^\dagger \\ a_{\mathrm{V}}^\dagger \end{pmatrix}, \tag{3.1}$$

and $a_{\mathrm{H}}^\dagger$ and $a_{\mathrm{V}}^\dagger$ are the creation operators of the horizontal and vertical polarization modes [88]. Clearly, $\sum_j \Pi_j \neq 1$ this time, as the projectors are independently measured.

Figure 5 depicts the experimental setup to generate and characterize three-photon states. Four photons are produced through double pair emission of non-collinear spontaneous parametric down-conversion (SPDC) process. The initial state is prepared in $|2,2\rangle$ by combining two horizontally polarized photons and two vertically polarized photons with a polarizing beam splitter (PBS). To ensure that the photons are indistinguishable in the frequency domain, interference filters of 3 nm bandwidth centered at 780 nm are placed before sending the photons into the PBS. The four photons are then reduced into

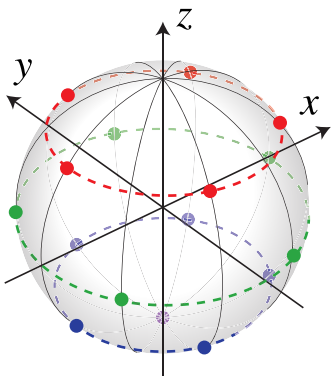


FIG. 6. Reduced visualization of the 16 two-photon measurement projectors on the single-qubit Bloch sphere. The projectors of three-photon states are defined as $b_j^{\dagger 3}\,|0,0\rangle/\sqrt{6}$ in accordance with Eq. (3.1). The projection states are chosen to equally distribute the corresponding single-photon component pure states $b_j^\dagger\,|0,0\rangle$ on the equatorial great circle and two small circles on the Bloch sphere, together with the south pole.

three photons by detecting a photon at D_1 and the reflected three photons from a partially-polarizing beam splitter (PPBS) are in the state of $|1,2\rangle\langle 1,2|$. The PPBS perfectly reflects vertically polarized photons and reflects 1/3 of horizontally polarized photons. The half-wave plate (HWP) setting of $\theta_1 = 0°$ leaves the state unchanged, whereas the setting of $\theta_1 = 45°$ transforms the state into $|2,1\rangle\langle 2,1|$. In addition, the mixed state $(|1,2\rangle\langle 1,2| + |2,1\rangle\langle 2,1|)/2$ is obtained by incoherently adding the relevant pure states through post-processing. These three-photon states are used to demonstrate the performances of ICCNet and FidNet in Fig. 9(b).

Experimentally [87], the three-photon states were characterized by acquiring the four-fold coincidence counts at D_1, D_2, D_3, and D_4 for 16 rank-one projectors after passing through a PBS and beam splitters (BS). The SU(2) unitary operators $\widetilde{U}_j$ that define the projectors $\Pi_j = b_j^{\dagger 3}\,|0,0\rangle\,\frac{1}{6}\,\langle 0,0|\,b_j^3$ according to rule (3.1) are determined by the quarter-wave plate (QWP) and HWP angles of θ_2 and θ_3 inasmuch as $\widetilde{U}_j = H(\theta_3)Q(\theta_2)$, where the matrix representations for the wave plates are given by

$$Q(\theta) \mathrel{\widehat{=}} \frac{1}{\sqrt{2}}\begin{pmatrix} 1-\mathrm{i}\cos 2\theta & -\mathrm{i}\sin 2\theta \\ -\mathrm{i}\sin 2\theta & 1+\mathrm{i}\cos 2\theta \end{pmatrix},$$
$$H(\theta) \mathrel{\widehat{=}} \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{pmatrix}. \tag{3.2}$$

In our experiments, we consider SU(2) rotations that fairly distribute the single-photon component $b_j^\dagger|0,0\rangle$ on three Bloch-spherical circles parallel to the equatorial plane [87, 89] as shown in Fig. 6. The measurement angles that realize these projectors are given in Fig. 5.

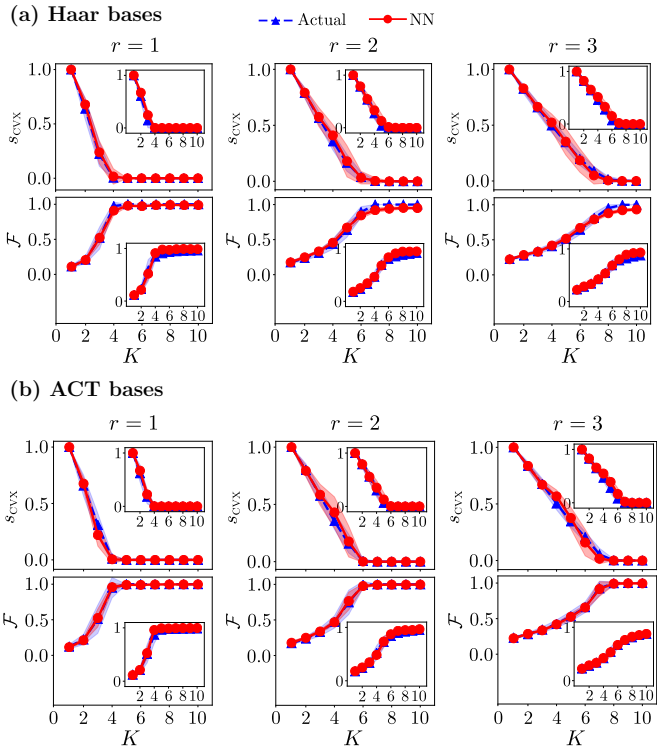


FIG. 7. Performance of ICCNet and FidNet in the prediction of s_{CVX} and $\mathcal{F}$ for different number (K) of measurement bases generated by (a) random unitaries sampled from the Haar unitary group and (b) adaptive unitaries from ACT, accompanied by 1-σ error bars derived from 50 simulated test experiments for each rank r that are not seen by the neural networks. The main plots correspond to perfect measurement data, whereas the insets show results under statistical noise with $N = 1000$ sampling copies per basis. Both the actual computed values and neural-network (NN) predictions are evidently in extremely good agreement.

IV. RESULTS

A. Simulations

We first present performance graphs of ICCNet and FidNet in Fig. 7 based on two sets of simulations on four-qubit states ($d = 16$) using random measurement bases generated with the Haar measure for the unitary group (see Appendix A), and bases found using ACT. In each set of simulations, for both cases where statistical noise is either absent or present, we collect simulation data of various number (K) of bases (s_{CVX} is normalized to 1 at $K = 1$ by default), each case recording measurements of 5000 randomly-generated quantum states of uniformly distributed rank $1 \le r \le 3$. The explicit CNN architecture employed is specified in Sec. II B. The accurate fit between the actual computed values and those predicted by ICCNet and FidNet suggests that faithful neural network predictions of both the degree of informational completeness and fidelity are a definite possibility in both noiseless and statistically noisy environments. Sample codes for network training and evaluation with four-qubit simulation

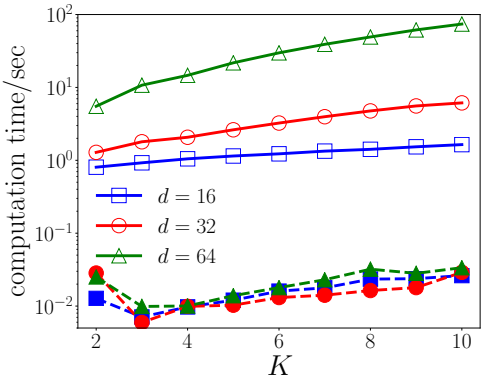


FIG. 8. Comparison of the average ICC computation time by carrying out the grayed subroutine (physical-probabilities extraction and SDP-based ICC) in Fig. 1 (unfilled markers) and a trained ICCNet model (solid markers) over many simulated experimental runs and states of various ranks. For $d = 32$ and 64, a set of 1000 datasets ($N = 1000$) each is used to acquire average computation times that are sufficiently representative (the s_{CVX} and $\mathcal{F}$ graphs are separately presented in the Appendix). These timing are obtained through CUDA 10.2 interfaced with the GPU-enabled TensorFlow 1.9 package on Python 3.5.3, with the Keras 2.1.6 frontend running on a twelve-core Intel(R) Xeon(R) CPU E5-2620 v3 at 2.40 GHz and an Nvidia GTX 1080 TI GPU of native settings. A trained FidNet model, on average, performs fidelity benchmarking in times that are roughly the same orders of magnitude.

datasets are available online [90].

In separate simulations on four- ($d = 16$), five- ($d = 32$) and six-qubit ($d = 64$) systems with random Haar measurement bases, numerical evidence presented in Fig. 8 shows that the computation times in s_{CVX} neural-network predictions can be significantly reduced by about four orders of magnitude relative to ordinary SDP calculations, and this difference grows wider with larger dimensions. The corresponding ICCNet and FidNet performance graphs similar to Fig. 7 are given in the Appendix.

B. Experimental performance with spatial-mode photonic states

For each value of d, we experimentally generated random pure states and construct their respective ACT measurement bases in order to evaluate the performance of ICCNet and FidNet, which were previously trained with 10000 simulation datasets of random quantum states of uniformly distributed rank $1 \leq r \leq 3$ and different K values. These simulated training datasets are modeled with statistical noise arising from a multinomial distribution defined by $N = 5000$ sampling copies per basis, which is close to the experimental average.

Owing to experimental noise, the resulting spatial-mode quantum states are, as a matter of fact, nearly pure but sufficiently low-rank. Figure 9(a) confirms that ICC and fidelity benchmarking with simulation-trained neural network models are accurate even with real experimental test data. One can observe the relative network-prediction stability of s_{CVX} in contrast with that of $\mathcal{F}$. This coincides with the expectation that while the fidelity is strongly affected by statistical noise and other imperfections such as systematic errors, the degree of informational completeness is more intimately related to the quantum measurements and rank of the quantum state, such that noise only introduces perturbations on the functional behavior of s_{CVX}. Regardless, Fig. 9(a) shows that all predictions made by the simulation-trained ICCNet and FidNet models remain roughly within the error margins of actual computed values.

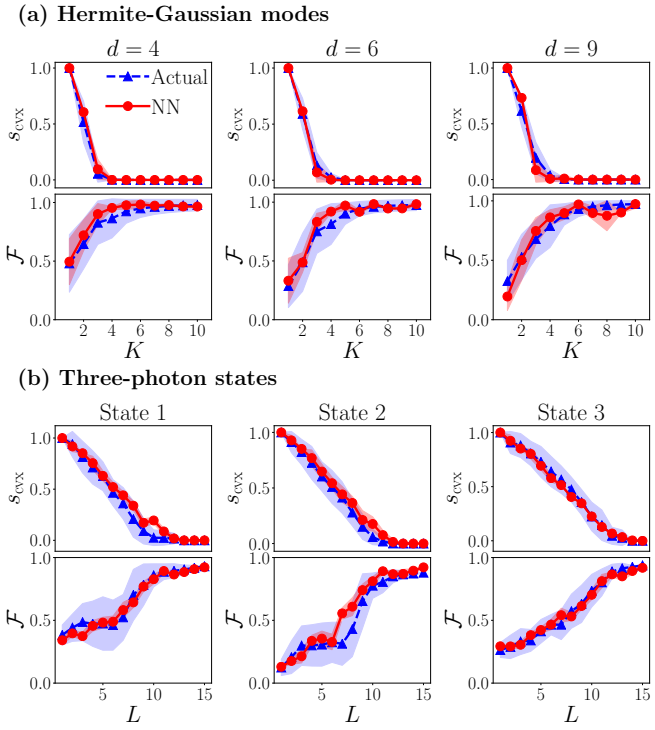


FIG. 9. (a) The neural-network predictions of s_{CVX} and $\mathcal{F}$ for spatial-mode photonic states of dimensions $d = 4$, 6 and 9. All graphs and 1-σ error bars of each dimension d are obtained from 15 experimental test states used to evaluate the networks. The average fidelity mapped out by FidNet lies closely with the actual computed curve. (b) Performances on three-photon systems are given for states $|1,2\rangle\langle 1,2|$, $|2,1\rangle\langle 2,1|$, and the rank-two $(|1,2\rangle\langle 1,2| + |2,1\rangle\langle 2,1|)/2$ in this order. All graphs and 1-σ error bars are obtained from 20 experimental test runs per quantum state. Despite the large error bars of the actual values owing to noise and experimental imperfections, the average fidelity curve is correctly identified by FidNet.

C. Experimental performance with multiphoton states

For every fixed number (L) of projectors chosen from the complete set of 16 defined in Sec. III B, simulation datasets of 10000 random $d = 4$ quantum states of uniformly distributed r are fed into both ICCNet and FidNet for training. These datasets are obtained from randomized sequences of the 16 projectors described above. Statistical noise is introduced into the simulation with multinomial distributions defined by

$N = 500$ per projective measurement. To test the trained models and acquire prediction results depicted in Fig. 9(b), we make use of three different sets of 20 experimental runs outside the training datasets, each set corresponding to a different quantum state.

D. Noise training and reduction

Experimental noise due to imperfections and systematic errors are always present in any real dataset. Fluctuating deviations of neural-network predicted values from actual ones as observed in Fig. 9 arise from the lack of such experimental noise in all simulated training datasets, apart from purely statistical fluctuations, used to train ICCNet and FidNet.

When more knowledge about the noisy environment is acquired, data simulation from such knowledge may be carried out to improve the network predictions under such an environment. Here, we show that when some samples of experimental data that are sufficiently representative of the overall noise behavior can be spared for training, it is possible to train ICCNet and FidNet with both statistically-noisy simulated datasets and bootstrapped experimental datasets in order to learn the experimental noise effects approximately well and improve network predictions.

Bootstrapping entails using a given experimental dataset to generate numerous mock datasets using Monte Carlo procedures. More specifically, in the multinomial setting, the column $\boldsymbol{\nu}_k$ of relative frequencies for the kth basis possess a Gaussian distribution of mean $\boldsymbol{p}_k$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{p}}^{(k)} = [\mathrm{diag}(\boldsymbol{p}_k) - \boldsymbol{p}_k \boldsymbol{p}_k{}^{\mathrm{T}}]/N$ for sufficiently large N owing to the central limit theorem, where $\mathrm{diag}(\,\cdot\,)$ forms a diagonal matrix whose diagonals are defined by the argument. A direct substitution of $\boldsymbol{\nu}_k$ for $\boldsymbol{p}_k$ leads to the following simple rule for bootstrapping experimental ACT datasets from Hermite-Gaussian mode photonic system: $\boldsymbol{\nu}'_k = \mathcal{N}_{\geq 0}\{\boldsymbol{\nu}_k + \boldsymbol{w}_k\}$, where $\boldsymbol{w}_k$ is a column of random variables collectively distributed according to the Gaussian distribution of zero mean and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\nu}}^{(k)}$, where $\boldsymbol{\Sigma}_{\boldsymbol{\nu}}^{(k)}$ is to be evaluated with the measurement relative frequencies of the particular kth basis and N is set to 5000, which is the estimated number of copies per ACT basis considered in Sec. IV B. The operation $\mathcal{N}_{\geq 0}$ is a composition of absolute value of the argument followed by its sum normalization over $0 \leq j \leq d-1$ for the kth ACT basis. Finally, the states that produce the bases relative frequencies used in the bootstrapping procedure are different from the test states used to evaluate the network predictions.

Owing to a limited set of three-photon states, we adopt a different method to bootstrap experimental datasets acquired from these states. Since these datasets are obtained from measuring independent projectors, we randomly permute these projectors and their corresponding relative (unnormalized) frequencies in order generate new measurement sequences as mock datasets. The 16 projectors offer us a total of 16! permutations for each state, allowing us to conveniently generate an abundance of bootstrapped training datasets that are clearly different from those used for testing. By a similar token to the spatial-mode photonic systems, each relative frequency ν_l here is a binomial random variable normalized by the number of copies N used to measure the lth projector. Therefore, bootstrapping these relative frequencies may be carried out by additive Gaussian random variables inasmuch as $\nu'_l = \nu_l + w_l\sqrt{\nu_l(1-\nu_l)/N}$, where w_l is a standard Gaussian random variable of zero mean and unit variance, and $N = 500$ is fixed as the estimated number of copies used to obtain the measured relative frequency for each projector, consistent with Fig. 9(b).

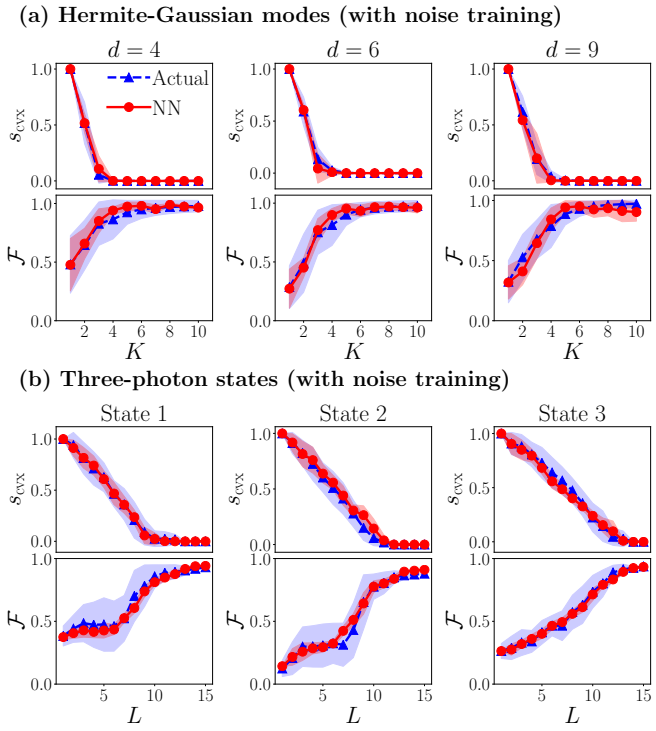


FIG. 10. The bootstrapped performance of ICCNet and FidNet in predicting s_{cvx} and $\mathcal{F}$ for the same test datasets that are used in Fig. 9, where fluctuating features are generally smoothened with bootstrapped noise training.

Figure 10 shows the enhanced prediction performances of ICCNet and FidNet. To generate this figure, a total of 5000 simulated and 5000 bootstrapped datasets are employed ($m = 10000$) for each group of experiments to train the networks for every value of K and L. These new plots indicate that slightly fluctuating neural-network prediction curves on noisy experimental data can be smoothened when bootstrapped information about the noisy environment is incorporated into the training.

V. CONCLUDING REMARKS

We took advantage of the universality of convolutional networks to train two neural networks that can very efficiently certify a low-measurement-cost quantum-state characterization scheme. These networks can respectively benchmark the quantum completeness of a given set of measurement out-

comes and corresponding data for reconstructing an unknown quantum state, as well as the resulting fidelity without explicitly carrying out the state reconstruction. Our machine-learning-assisted scheme allows experimentalists to rapidly assess the sufficiency of measurement resources for an unambiguous characterization of arbitrary quantum states and achieve accelerated real-time verification without having to perform any optimization routine during the experiment. This becomes essential for many practical quantum tasks that do require fast execution times to avoid noise accumulation and drifts.

An arguably interesting problem would be to minimize the time required to acquire the trained neural networks. This includes both the training time and the generation of adequate training datasets. While the former can now be easily parallelized with graphics processing units, the latter involves two rounds of semidefinite programming per dataset as discussed in Sec. II A, the acceleration of which is still a subject of ongoing research [91].

On the other hand, while classical algorithms for these procedures have worst-case polynomial time complexities in the dimension of the Hilbert space, it is known [92] that quantum algorithms can execute semidefinite programs with polylogarithmic time complexities in the dimension. This immediately reveals the possibility of completely transforming the neural networks employed here, or part thereof, into their quantum counterparts (fused with the training-data processing procedures that use quantum semidefinite programming) that could assimilate into a much larger set of networks for a grander purpose. Practical feasibility in implementing such extended quantum neural networks still remains to be seen.

Note.—Nearing the submission of our work, we discovered another very recent preprint reference [93] that purely discusses the estimation of the fidelity with fully-connected networks. Apart from the clear distinction in architectures, we remark that while the networks in this reference were specifically trained for Pauli measurements, our CNN-based FidNet is compatible with generalized measurement inputs that can be readily used in parallel with ICCNet or any other quantum task that relies on arbitrary measurements. The objectives of both works are hence very different. The next key distinction is network training, which in this reference is based on categorical training that splits the continuous fidelity range into small intervals. As mentioned in the preprint itself, network training can be slow when the intervals are too small. In our current work, FidNet directly computes the fidelity values without such output splitting, and hence training efficiency is not sacrificed for prediction accuracy.

ACKNOWLEDGMENTS

Y.S.T., S.S. and H.J. acknowledge support by the National Research Foundation of Korea (Grant Nos. 2019R1A6A1A10073437, 2019M3E4A1080074, 2020R1A2C1008609, and 2020K2A9A1A06102946). G.L. acknowledges support by the Center of Excellence ≪Center of Photonics≫ funded by the Ministry of Science and Higher Education of the Russian Federation, contract No. 075-15-2020-906. Y.K. and Y.-H.K. acknowledge support by the National Research Foundation of Korea (Grant No. 2019R1A2C3004812) and the ITRC support program (IITP-2020-0-01606). L.L.S.S. acknowledges support from European Union's Horizon 2020 research and innovation program (ApresSF and STORMYTUNE) and the Ministerio de Ciencia e Innovación (PGC2018-099183-B-I00). The MSU team acknowledges support from the Russian Foundation for Basic Research (RFBR Project No. 19-32-80043 and RFBR Project No. 19-52-80034) and support under the Russian National Technological Initiative via MSU Quantum Technology Centre.

** All tables and figures for the appendix are found behind the bibliography section.*

Appendix A: Procedures for generating random measurements

We state the recipes for generating two types of measurement bases used to generate all training datasets discussed in Sec. IV. We start with the set of von Neumann bases derived from rotations with random unitary operators U_k^{Haar} distributed according to the Haar measure. Such a set of unitary operators was shown to be derivable from a modified version of the QR decomposition [94]:

Constructing a random Haar basis

Starting from a reference basis $\{|0\rangle, |1\rangle, \ldots, |d-1\rangle\}$:

1. Generate a random $d \times d$ matrix $\boldsymbol{A}$ with entries i.i.d. standard Gaussian distribution.
2. Compute the two matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ by searching for the QR decomposition $\boldsymbol{A} = \boldsymbol{QR}$.
3. Define $\boldsymbol{R}_{\mathrm{diag}} = \mathrm{Diag}\{\boldsymbol{R}\}$ (sets all off-diagonal elements to zero)
4. Define $\boldsymbol{L} = \boldsymbol{R}_{\mathrm{diag}} \oslash |\boldsymbol{R}_{\mathrm{diag}}|$ ($\oslash$ refers to the Hadamard division).
5. Define the new basis $|u_l\rangle \mathrel{\widehat{=}} \boldsymbol{U}_{\mathrm{Haar}} |l\rangle$ for $0 \leq l \leq d-1$.

The next kind of measurement is an adaptive set of von Neumann bases that are sequentially inferred directly from previous measurement data. They were meant for establishing an ACT scheme that is highly compressive [51, 52]. The logic behind their construction is that low-rank true states are relatively closer to rank-deficient state estimators, and minimum-entropy estimators obtained from non-IC bases set give a very compressive sequence of eigenbases to quickly reach informational completeness. We state the construction of such a set of K bases $\{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_K\}$ below, with the first basis $\mathcal{B}_1 = \{|l\rangle \langle l|\}_{l=0}^{d-1}$ being the standard computational basis:

Constructing an ACT basis
Beginning with $k = 1$ and a random computational basis $\mathcal{B}_1$:

1. Measure $\mathcal{B}_k$ and collect the relative frequency data $\sum_{j'=0}^{d-1} \nu_{j'k} = 1$.
2. From $\{\nu_{0k'}, \ldots, \nu_{d-1\ k'}\}_{k'=1}^{k}$, obtain kd physical probabilities.
3. Perform ICC with the physical probabilities and compute $s_{\text{CVX},k}$:
   - **If** $s_{\text{CVX},k} < \epsilon$, terminate this ACT scheme and take $\rho_{\text{max}} \approx \rho_{\text{min}}$ as the estimator and report $s_{\text{CVX},k}$.
   - **Else** Proceed.
4. Choose an estimator $\widehat{\rho}_k$ that minimizes the von Neumann entropy $S(\widehat{\rho}_k) = -\text{tr}\{\widehat{\rho}_k \log \widehat{\rho}_k\}$ subject to the positivity and data constraints.
5. Define $\mathcal{B}_{k+1}$ to be the eigenbasis of $\widehat{\rho}_k$.
6. Set $k = k + 1$ and repeat.

Appendix B: Hyperparameters of ICCNet and FidNet

All hyperparameters used in both ICCNet and FidNet are manually optimized so that the validation loss is minimized (see Appendix C). Figure 11 succinctly consolidates all important operational hyperparameter settings adopted for training ICCNet in various dimensions. Networks (a) and (b) have identical architecture with different dropout rates. For larger dimensions such as $d = 64$, we find that the use of deeper CNN networks can yield better training results. In these cases, *residual blocks* implemented in network (c) help to avoid the so-called vanishing-gradient problem [64]. Each residual block consists of repeated convolutional blocks ($\mathbf{BLK}_{\text{s}}(t)$) that maintain the array dimensions, sandwiched by a skip connection that adds the input of these repeated convolutional blocks to their output. Another notable difference between Fig. 11(c) and Figs. 11(a) and (b) is the inclusion of an average-pooling layer and one fully-connected layer, which are standard components of residual-based networks. Figure 12 shows the corresponding hyperparameters for FidNet.

Appendix C: Explicit network training procedures

1. Input data preparation

The initial input data matrix $\boldsymbol{X}$ for training the ICCNet comprises the m datasets of K measurement bases, or L projectors, and their corresponding relative frequencies. These data are reshaped into either a $m \times \lceil\sqrt{K(d^2+d)}\rceil \times \lceil\sqrt{K(d^2+d)}\rceil$ or $m \times \lceil\sqrt{L(d^2+1)}\rceil \times \lceil\sqrt{L(d^2+1)}\rceil$ three-dimensional matrix $\widetilde{\boldsymbol{X}}$ to be processed by the convolution networks.

The input data matrix for training the FidNet requires the additional m target states to be assigned to the respective datasets. For $d = 16$, 32 and 64, FidNet is trained by supplying the true states as the ("right") target states. This is sufficient as only statistical fluctuation exist in the simulation data obtained from finite copies. To benchmark fidelities for real experimental data, it is important that FidNet also recognizes inputs with systematic errors. We numerically show that training FidNet with both the "right" and "wrong" target states, the latter referring to targets differing from the true states for the same datasets, can improve the benchmarking accuracy on average. Combining these datasets give the resulting reshaped three-dimensional matrix that is either of size $2m \times \lceil\sqrt{(K+1)d^2+Kd}\rceil \times \lceil\sqrt{(K+1)d^2+Kd}\rceil$ or $2m \times \lceil\sqrt{(L+1)d^2+L}\rceil \times \lceil\sqrt{(L+1)d^2+L}\rceil$. As a demonstration, we take "wrong" target state ρ_{wrong} to be a randomly generated operator from the true state $\rho \mathrel{\widehat{=}} \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\dagger$ diagonalized with the unitary matrix $\boldsymbol{U}$ and diagonal matrix $\boldsymbol{\Lambda}$ according to the following prescription:

For a d-dimensional column $\boldsymbol{v}$ of uniformly-distributed entries, each in the range [0,1], define $\rho_{\text{wrong}} \mathrel{\widehat{=}} \boldsymbol{U}'\boldsymbol{\Lambda}'\boldsymbol{U}'^\dagger$ using the matrices

$$\begin{aligned}
\boldsymbol{\Lambda}' &= \text{Diag}((1-\lambda_1)\,\text{diag}(\boldsymbol{\Lambda}) + \lambda_1 \boldsymbol{w})\ , \\
\boldsymbol{U}' &= \boldsymbol{V}\boldsymbol{U}\ , \\
\boldsymbol{w} &= \mathcal{N}\{-\log(\boldsymbol{v}) \odot \text{diag}(\boldsymbol{\Lambda})\}\ , \\
\boldsymbol{V} &= \text{wHaar}(\lambda_2)\,,
\end{aligned} \tag{C1}$$

where λ_1 and λ_2 are uniformly distributed in $[\lambda_{\text{min}},\lambda_{\text{max}}]$, $\text{diag}(\cdot)$ and $\text{Diag}(\cdot)$ are respectively diagonal-element extracting and diagonal-matrix transforming operations, $\mathcal{N}\{\cdot\}$ normalizes a column by its element-wise sum, $\odot$ denotes the Hadamard product, and $\text{wHaar}(\lambda)$ refers to the weighted Haar unitary function that outputs a random unitary according to the assigned weight λ. By definition, the special case $\text{wHaar}(0) = \mathbf{1}$ holds, and the more general function is given by

Weighted Haar unitary (wHaar) of weight λ

1. Generate a random $d \times d$ matrix $\boldsymbol{A}$ with entries i.i.d. standard Gaussian distribution.
2. Define $\boldsymbol{A}' = \lambda \boldsymbol{A} + (1-\lambda)\mathbf{1}$.
3. Compute $\boldsymbol{Q}$ and $\boldsymbol{R}$ from the QR decomposition $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$.
4. Define $\boldsymbol{R}_{\text{diag}} = \text{Diag}\{\boldsymbol{R}\}$ and $\boldsymbol{L} = \boldsymbol{R}_{\text{diag}} \oslash |\boldsymbol{R}_{\text{diag}}|$ ($\oslash$ refers to the Hadamard division).
5. Define $\boldsymbol{V} = \boldsymbol{Q}\boldsymbol{L}$.

It is clear that ρ_{wrong} so defined has exactly the same rank as ρ, and at times can be close to ρ. A practical justification for these sort of target states is that very typically in experiments, although the target states are not exactly ρ due to various noisy imperfections, the actual true states are, nevertheless, very often almost as rank-deficient as the intended target states, with a rapidly decaying eigenvalue spectrum. Figures 14 and 15 show that fidelity benchmarking is typically optimal when training is performed with both the "right" and "wrong" target states simultaneously. For these experimental data, we find that $\lambda_1 = 0.8$ and $\lambda_2 = 1$ gives rather accurate fidelity benchmarking. For more general noisy situations, we may need to introduce more structured noise models in generating the simulated training datasets.

2. Training validation

The specifications of every input data matrix are tabulated in Tab. I. Generally speaking, the action of training these convolutional networks is equivalent to carrying out an optimization routine to minimize the "distance", quantified by a so-called loss function, between the predicted output $\boldsymbol{y}_{\text{pred}}$ and the original training output $\boldsymbol{y}$. In all training procedures, the momentum-based gradient-descent algorithm NAdam [84] is employed for the minimization. The batch size is chosen to strike a compromise between training convergence and gradient-computation accuracy. For ICCNet, we consider the mean absolute-error (MAE) as the loss function for training. For FidNet, the mean squared-error (MSE) loss is used.

It is important to track the training activities so that overtraining or overfitting does not happen. To do this, a very common way is to first split the complete data ($\widetilde{\boldsymbol{X}}$,$\boldsymbol{y}$) into data for training ($\widetilde{\boldsymbol{X}}_{\text{train}}$,$\boldsymbol{y}_{\text{train}}$), validation ($\widetilde{\boldsymbol{X}}_{\text{val}}$,$\boldsymbol{y}_{\text{val}}$) and testing ($\widetilde{\boldsymbol{X}}_{\text{test}}$,$\boldsymbol{y}_{\text{test}}$). During training, at each epoch (gradient-descent iterative step), as the loss function between $\boldsymbol{y}_{\text{pred}}$ and $\boldsymbol{y}_{\text{train}}$ is reduced, the resulting validation loss, that is loss between $\boldsymbol{y}_{\text{pred}}$ and $\boldsymbol{y}_{\text{val}}$, is also reported. Training is successful when both training and validation losses decay simultaneously with the number of epochs. As an additional precaution, we confirm both the training and validation progress by performing one final prediction with $\boldsymbol{y}_{\text{test}}$ to verifying that the test loss is also comparatively small. In our context, the split ratio between training, validation and test datasets is set to 0.8:0.1:0.1.

For all the training datasets reflected in Tab. I, each row of $\boldsymbol{X}$ consists of information about the POVM and relative frequencies for the case of ICCNet (and an additional target state for FidNet) that originate from a randomly generated quantum state of rank $r \in [1, 3]$. The distribution of continuous entries in the output $\boldsymbol{y}$ can also affect training efficiency. In the case of s_{CVX} output for ICCNet, there can coexist two groups of values, one group containing values that are substantially far away from zero and the other containing those that are almost zero (IC). We find that a reversible mapping that maps each output value $y \equiv s_{\text{CVX}}$ to $y' = -\log_{10}(y)/10$, with the conditional definition $y < 10^{-10} \to y = 10^{-10}$ to ensure that $y' \leq 1$, can improve training efficiencies. All ICCNet architectures shown in Fig. 11 and ICCNet training graphs in Fig. 16 refer to these logarithmized outputs. One can understand this logarithmic training as a switch of training focus to order of magnitude estimation for s_{CVX}, which is an alternatively relevant outcome since all one really needs to know is whether a given measurement is IC or not.

3. Training results analyses

After every network training, only the model weights corresponding to the lowest validation loss are saved for later predictions. As sample illustrations, we explicitly show the progress of training and validation losses for $d = 16$ in Fig. 16. For this dimension, the input datasets of all four data types listed in Tab. I are stacked for training ICCNet and FidNet at one go. In order to verify the test accuracies offered by the trained neural-network models, we also supply Figs. 17 and 18 for $K = 4$ assorted von Neumann bases. For completeness, we also furnish numerical performance indicators for ICCNet and FidNet in Tabs. II and III to supplement Figs. 7, 9 and 10 in the main text.

As far as the analyses of the training results are concerned, the aforementioned figures and tables are sufficient to verify the training qualities of ICCNet and FidNet. Going by a different route, one may additionally fall back on other more conventional tools to analyze the neural-network-predicted s_{CVX} values. In theory, the measurements are IC when s_{CVX} is zero. In practice, however, we assign a small threshold that distinguishes datasets that are IC from those that are not. In Fig. 19, we present the so-called *confusion matrix* that simplistically quantifies how well the datasets are correctly grouped into the "IC" and "non-IC" classes. The threshold is set at 10^{-3}. Clearly, the ideal prediction result is such that the off-diagonal elements of the confusion matrix are all zero. Such a perfect binary classification does not exist in realistic machine learning applications. Instead, upon labeling the IC cases as the "positives", there would be predictions that are false positives (fp) (as opposed to the true positives tp) or false negatives (fn) (as opposed to the true negatives tn), giving rise to nonzero, but small, off-diagonal values. We define the concepts of *precision* $\text{prec} = \text{tp}/(\text{tp} + \text{fp})$ and *recall* $\text{rec} = \text{tp}/(\text{tp} + \text{fn})$, and a predictive ICCNet should generally output values that have high precision and recall. More specifically, there exists the so-called F1 score, or more appropriately the Sørensen–Dice coefficient [95, 96], $F_1 = 2\,\text{tp}/(2\,\text{tp} + \text{fp} + \text{fn})$ defined as the harmonic mean of pr and rec that speculates such a binary prediction power.

Rather than fixing a particular threshold, it is more objective to scan a range of threshold values and parametrically plot the so-called precision-recall (PR) curves as shown in Fig. 20. Then a natural figure of merit to gauge the binary classification power would be the "area-under-curve" (AUC) measure for these curves, since a unit area entails the largest possible coverage of prec and rec. This figure of merit also possesses one crucial advantage. If we remember that all training datasets are obtained from random states of uniformly distributed ranks $r \in [1, 3]$, it is then easy to see that the $K = 4$ datasets, for instance, have much fewer positive cases as com-

pared to negative cases. Such a class imbalance biases the binary classification analysis, and occurs ubiquitously in our context since informational completeness is rank-sensitive, and thus highly dependent on the state ranks used to generate the training datasets. The AUC for the PR curve is consequently lowered mainly because of this bias rather than a weak binary-classification capability. In such cases, perhaps a better option would be to investigate the AUC of the so-called receiver operating characteristic (ROC) curve [97]. This plots the true positive rate (tpr = tp/total positives) against the false positive rate (fpr = fp/total negatives). For such imbalanced cases, while the PR curve takes a larger K value to recuperate its area, the AUC of the ROC curve remains high for all tested K values (see Fig. 20). A loosely, yet intuitive understanding is that the ROC curve accounts for both tp and fp values evenly, whereas the PR curve focuses only the tp values, which form the minority class in a class imbalance situation.

Despite the above observations, just like any other single-number criterion that is popularly adopted in statistics and machine learning owing to its computation simplicity, these figures of merit are *ad hoc* by nature. Therefore, care must be taken in interpreting these measures.

[1] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, "An adaptive variational algorithm for exact molecular simulations on a quantum computer," Nat. Commun. **10**, 3007 (2019).

[2] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell et al., "Quantum supremacy using a programmable superconducting processor," Nature **574**, 505–510 (2019).

[3] L. Hu, Y. Ma, W. Cai, X. Mu, Y. Xu, W. Wang, Y. Wu, H. Wang, Y. P. Song, C.-L. Zou, S. M. Girvin, L.-M. Duan, and L. Sun, "Quantum error correction and universal gate set operation on a binomial bosonic logical qubit," Nat. Phys. **15**, 503–508 (2019).

[4] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," Nature **567**, 209–212 (2019).

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

[6] B. T. Gard, L. Zhu, G. S. Barron, N. J. Mayhall, S. E. Economou, and E. Barnes, "Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm," npj Quantum Inf. **6**, 10 (2020).

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

[8] A. Holmes and A. Y. Matsuura, "Efficient quantum circuits for accurate state preparation of smooth, differentiable functions," in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (2020) pp. 169–179.

[9] V. M. Schäfer, C. J. Ballance, K. Thirumalai, L. J. Stephenson, T. G. Ballance, A. M. Steane, and D. M. Lucas, "Fast quantum logic gates with trapped-ion qubits," Nature **555**, 75 (2018).

[10] X.-F. Shi, "Accurate quantum logic gates by spin echo in rydberg atoms," Phys. Rev. Applied **10**, 034006 (2018).

[11] T. Ono, R. Okamoto, M. Tanida, H. F. Hofmann, and S. Takeuchi, "Implementation of a quantum controlled-swap gate with photonic circuits," Sci. Rep. **7**, 45353 (2017).

[12] R. B. Patel, J. Ho, F. Ferreyrol, T. C. Ralph, and G. J. Pryde, "A quantum Fredkin gate," Sci. Adv. **2**, e1501531 (2016).

[13] J. Fiurášek, "Linear optical Fredkin gate based on partial-swap gate," Phys. Rev. A **78**, 032317 (2008).

[14] W. K. Wootters and B. D. Fields, "Optimal state-determination by mutually unbiased measurements," Ann. Phys. **191**, 363–381 (1989).

[15] M. A. Nielsen, "Quantum computation by measurement and quantum memory," Phys. Lett. A **308**, 96 – 100 (2003).

[16] R. Raussendorf and H. J. Briegel, "A one-way quantum computer," Phys. Rev. Lett. **86**, 5188–5191 (2001).

[17] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. V. den Nest, "Measurement-based quantum computation," Nat. Phys. **5**, 19 – 26 (2009).

[18] T. Durt, B.-G. Englert, I. Bengtsson, and K. Życzkowski, "On mutually unbiased bases," Int. J. Quantum Inf. **08**, 535–640 (2010).

[19] A. J. Scott, "Tight informationally complete quantum measurements," J. Phys. A: Math. and Gen. **39**, 13507–13530 (2006).

[20] H. Zhu and B.-G. Englert, "Quantum state tomography with fully symmetric measurements and product measurements," Phys. Rev. A **84**, 022327 (2011).

[21] H. Zhu, "Quantum state estimation with informationally overcomplete measurements," Phys. Rev. A **90**, 012115 (2014).

[22] I. Chuang and M. Nielsen, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).

[23] M. G. A. Paris and J. Řeháček, eds., *Quantum State Estimation*, Lect. Not. Phys., Vol. 649 (Springer, Berlin, 2004).

[24] Y. S. Teo, *Introduction to Quantum-State Estimation* (World Scientific Publishing Co., Singapore, 2015).

[25] J. L. O'Brien, G. J. Pryde, A. Gilchrist, D. F. V. James, N. K. Langford, T. C. Ralph, and A. G. White, "Quantum process tomography of a controlled-not gate," Phys. Rev. Lett. **93**, 080502 (2004).

[26] J. F. Poyatos, J. I. Cirac, and P. Zoller, "Complete characterization of a quantum process: The two-bit quantum gate," Phys. Rev. Lett. **78**, 390 (1997).

[27] Y. S. Teo, B.-G. Englert, J. Řeháček, and Z. Hradil, "Adaptive schemes for incomplete quantum process tomography," Phys. Rev. A **84**, 062125 (2011).

[28] A. Luis and L. L. Sánchez-Soto, "Complete characterization of arbitrary quantum measurement processes," Phys. Rev. Lett. **83**, 3573–3576 (1999).

[29] J. Fiurášek, "Maximum-likelihood estimation of quantum measurement," Phys. Rev. A **64**, 024102 (2001).

[30] G. M. D'Ariano, L. Maccone, and P. L. Presti, "Quantum calibration of measurement instrumentation," Phys. Rev. Lett. **93**, 250407 (2004).

[31] Y. Chen, M. Farahzad, S. Yoo, and T.-C. Wei, "Detector tomography on ibm quantum computers and mitigation of an imperfect measurement," Phys. Rev. A **100**, 052315 (2019).

[32] L. Zhang, A. Datta, H. B. Coldenstrodt-Ronge, X.-M. Jin, J. Eisert, M. B. Plenio, and I. A. Walmsley, "Recursive quantum detector tomography," New J. Phys. **14**, 115005 (2012).

[33] M. Altorio, M. G. Genoni, F. Somma, and M. Barbieri, "Metrology with unknown detectors," Phys. Rev. Lett. **116**, 100802 (2016).

[34] Y. Kim, Y.-S. Kim, S.-Y. Lee, S.-W. Han, S. Moon, Y.-H. Kim,

and Y.-W. Cho, "Direct quantum process tomography via measuring sequential weak values of incompatible observables," Nat. Commun. **9**, 192 (2018).

[35] A. Gaikwad, D. Rehal, A. Singh, Arvind, and K. Dorai, "Experimental demonstration of selective quantum process tomography on an nmr quantum information processor," Phys. Rev. A **97**, 022311 (2018).

[36] A. Bendersky and J. P. Paz, "Selective and efficient quantum state tomography and its application to quantum process tomography," Phys. Rev. A **87**, 012122 (2013).

[37] C. T. Schmiegelow, A. Bendersky, M. A. Larotonda, and J. P. Paz, "Selective and efficient quantum process tomography without ancilla," Phys. Rev. Lett. **107**, 100502 (2011).

[38] A. Bendersky, F. Pastawski, and J. P. Paz, "Selective and efficient quantum process tomography," Phys. Rev. A **80**, 032116 (2009).

[39] A. Bendersky, F. Pastawski, and J. P. Paz, "Selective and efficient estimation of parameters for quantum process tomography," Phys. Rev. Lett. **100**, 190403 (2008).

[40] T. Proctor, K. Rudinger, K. Young, M. Sarovar, and R. Blume-Kohout, "What randomized benchmarking actually measures," Phys. Rev. Lett. **119**, 130502 (2017).

[41] J. Helsen, X. Xue, L. M. K. Vandersypen, and S. Wehner, "A new class of efficient randomized benchmarking protocols," npj Quantum Inf. **5**, 71 (2019).

[42] Y. Lu, J. Y. Sim, J. Suzuki, B.-G. Englert, and H. K. Ng, "Direct estimation of minimum gate fidelity," Phys. Rev. A **102**, 022410 (2020).

[43] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert, "Quantum state tomography via compressed sensing," Phys. Rev. Lett. **105**, 150401 (2010).

[44] A. Kalev, R. L. Kosut, and I. H. Deutsch, "Quantum tomography protocols with positivity are compressed sensing protocols," npj Quantum Inf. **1**, 15018 (2015).

[45] C. H. Baldwin, I. H. Deutsch, and A. Kalev, "Strictly-complete measurements for bounded-rank quantum-state tomography," Phys. Rev. A **93**, 052105 (2016).

[46] A. Steffens, C. A. Riofrío, W. McCutcheon, I. Roth, B. A. Bell, A. McMillan, M. S. Tame, J. G. Rarity, and J. Eisert, "Experimentally exploring compressed sensing quantum tomography," Quantum Sci. Technol. **2**, 025005 (2017).

[47] C. A. Riofrío, D. Gross, S. T. Flammia, T. Monz, D. Nigg, R. Blatt, and J. Eisert, "Experimental quantum compressed sensing for a seven-qubit system," Nat. Commun. **8**, 15305 (2017).

[48] C. H. Baldwin, A. Kalev, and I. H. Deutsch, "Quantum process tomography of unitary and near-unitary maps," Phys. Rev. A **90**, 012110 (2014).

[49] A. V. Rodionov, A. Veitia, R. Barends, J. Kelly, D. Sank, J. Wenner, J. M. Martinis, R. L. Kosut, and A. N. Korotkov, "Compressed sensing quantum process tomography for superconducting quantum gates," Phys. Rev. B **90**, 144504 (2014).

[50] A. Shabani, R. L. Kosut, M. Mohseni, H. Rabitz, M. A. Broome, M. P. Almeida, A. Fedrizzi, and A. G. White, "Efficient measurement of quantum dynamics via compressive sensing," Phys. Rev. Lett. **106**, 100401 (2011).

[51] D. Ahn, Y. S. Teo, H. Jeong, F. Bouchard, F. Hufnagel, E. Karimi, D. Koutný, J. Řeháček, Z. Hradil, G. Leuchs, and L. L. Sánchez-Soto, "Adaptive compressive tomography with no *a priori* information," Phys. Rev. Lett. **122**, 100404 (2019).

[52] D. Ahn, Y. S. Teo, H. Jeong, D. Koutný, J. Řeháček, Z. Hradil, G. Leuchs, and L. L. Sánchez-Soto, "Adaptive compressive tomography: A numerical study," Phys. Rev. A **100**, 012346 (2019).

[53] Y. S. Teo, G. I. Struchalin, E. V. Kovlakov, D. Ahn, H. Jeong, S. S. Straupe, S. P. Kulik, G. Leuchs, and L. L. Sánchez-Soto, "Objective compressive quantum process tomography," Phys. Rev. A **101**, 022334 (2020).

[54] Y. Kim, Y. S. Teo, D. Ahn, D.-G. Im, Y.-W. Cho, G. Leuchs, L. L. Sánchez-Soto, H. Jeong, and Y.-H. Kim, "Universal compressive characterization of quantum dynamics," Phys. Rev. Lett. **124**, 210401 (2020).

[55] I. Gianani, Y. Teo, V. Cimini, H. Jeong, G. Leuchs, M. Barbieri, and L. Sánchez-Soto, "Compressively certifying quantum measurements," PRX Quantum **1**, 020307 (2020).

[56] L. Vandenberghe and S. Boyd, "Semidefinite programming," SIAM Rev. **38**, 49–95 (1996).

[57] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, "Neural-network quantum state tomography," Nature Physics **14**, 447–450 (2018).

[58] A. M. Palmieri, E. Kovlakov, F. Bianchi, D. Yudin, S. Straupe, J. D. Biamonte, and S. Kulik, "Experimental neural network enhanced quantum tomography," npj Quantum Information **6**, 20 (2020).

[59] M. Neugebauer, L. Fischer, A. Jäger, S. Czischek, S. Jochim, M. Weidemüller, and M. Gärttner, "Neural-network quantum state tomography in a two-qubit experiment," Phys. Rev. A **102**, 042604 (2020).

[60] S. Lohani, B. T. Kirby, M. Brodsky, O. Danaci, and R. T. Glasser, "Machine learning assisted quantum state estimation," Machine Learning: Science and Technology **1**, 035007 (2020).

[61] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE **86**, 2278 (1998).

[62] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, Vol. 25, edited by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Curran Associates, Inc., 2012) pp. 1097–1105.

[63] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision (IJCV) **115**, 211–252 (2015).

[64] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016) pp. 770–778.

[65] D.-X. Zhou, "Universality of deep convolutional neural networks," Appl. Comput. Harmon. A **48**, 787 – 794 (2020).

[66] Y. Ming, C.-T. Lin, S. D. Bartlett, and W.-W. Zhang, "Quantum topology identification with deep neural networks and quantum walks," npj Comput. Mater. **5**, 88 (2019).

[67] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," Nat. Phys. **15**, 1273–1278 (2019).

[68] A. A. Melnikov, L. E. Fedichkin, and A. Alodjants, "Predicting quantum advantage by quantum walk with convolutional neural networks," New J. Phys. **21**, 125002 (2019).

[69] Y.-H. Tsai, M.-Z. Yu, Y.-H. Hsu, and M.-C. Chung, "Deep learning of topological phase transitions from entanglement aspects," Phys. Rev. B **102**, 054512 (2020).

[70] J. Řeháček, Z. Hradil, E. Knill, and A. I. Lvovsky, "Diluted maximum-likelihood algorithm for quantum tomography," Phys. Rev. A **75**, 042108 (2007).

[71] Y. S. Teo, H. Zhu, B.-G. Englert, J. Řeháček, and Z. Hradil, "Quantum-state reconstruction by maximizing likelihood and entropy," Phys. Rev. Lett. **107**, 020404 (2011).

[72] J. Shang, Z. Zhang, and H. K. Ng, "Superfast maximum-likelihood reconstruction for quantum tomography," Phys. Rev.

A **95**, 062336 (2017).
[73] T. Kariya and H. Kurata, *Generalized Least Squares* (John Wiley & Sons, New Jersey, 2004).
[74] A. C. Rencher and W. F. Christensen, *Methods of Multivariate Analysis* (John Wiley & Sons, New Jersey, 2012).
[75] G. C. Knee, E. Bolduc, J. Leach, and E. M. Gauger, "Quantum process tomography via completely positive and trace-preserving projection," Phys. Rev. A **98**, 062336 (2018).
[76] J. P. Boyle and R. L. Dykstra, "A method for finding projections onto the intersection of convex sets in hilbert spaces," in *Advances in Order Restricted Statistical Inference*, edited by R. Dykstra, T. Robertson, and F. T. Wright (Springer New York, New York, NY, 1986) pp. 28–47.
[77] D. E. Rumelhart and J. L. McClelland, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations* (1987) pp. 318–362.
[78] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer, New York, 2009).
[79] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," (2012), arXiv:1207.0580 [cs.NE].
[80] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," Journal of Machine Learning Research **15**, 1929–1958 (2014).
[81] D. Warde-Farley, I. J. Goodfellow, A. C. Courville, and Y. Bengio, "An empirical analysis of dropout in piecewise linear networks," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, edited by Y. Bengio and Y. LeCun (2014).
[82] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15 (JMLR.org, 2015) p. 448–456.
[83] S. Santurkar, D. Tsipras, A. Ilyas, and A. Mądry, "How does batch normalization help optimization?" in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18 (Curran Associates Inc., Red Hook, NY, USA, 2018) p. 2488–2498.
[84] T. Dozat, "Incorporating nesterov momentum into adam," in *ICLR Workshop* (2016) p. 2013–2016.
[85] E. Bolduc, N. Bent, E. Santamato, E. Karimi, and R. W. Boyd, "Exact solution to simultaneous intensity and phase encryption with a single phase-only hologram," Opt. Lett. **38**, 3546–3549 (2013).
[86] A. Mair, A. Vaziri, G. Weihs, and A. Zeilinger, "Entanglement of the orbital angular momentum states of photons," Nature **412**, 313 (2001).
[87] Y. Kim, G. Björk, and Y.-H. Kim, "Experimental characterization of quantum polarization of three-photon states," Phys. Rev. A **96**, 033840 (2017).
[88] U. Schilling, J. von Zanthier, and G. S. Agarwal, "Measuring arbitrary-order coherences: Tomography of single-mode multiphoton polarization-entangled states," Phys. Rev. A **81**, 013826 (2010).
[89] Y. Israel, I. Afek, S. Rosen, O. Ambar, and Y. Silberberg, "Experimental tomography of noon states with large photon numbers," Phys. Rev. A **85**, 022115 (2012).
[90] Codes written in MATLAB and Python are uploaded onto Github at `https://github.com/ACAD-repo/ICCNet-FidNet`.
[91] A. Majumdar, G. Hall, and A. A. Ahmadi, "Recent scalability improvements for semidefinite programming with applications in machine learning, control, and robotics," Annual Review of Control, Robotics, and Autonomous Systems **3**, 331–360 (2020).
[92] F. G. S. L. Brandão, A. Kalev, T. Li, C. Y.-Y. Lin, K. M. Svore, and X. Wu, "Quantum SDP Solvers: Large Speed-Ups, Optimality, and Applications to Quantum Learning," in *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 132, edited by C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2019) pp. 27:1–27:14.
[93] X. Zhang, M. Luo, Z. Wen, Q. Feng, S. Pang, W. Luo, and X. Zhou, "Direct fidelity estimation of quantum states using machine learning," (2021), arXiv:2102.02369 [quant-ph].
[94] F. Mezzadri, "How to generate random matrices from the classical compact groups," Notices of the AMS **54**, 592 (2007).
[95] T. Sørensen, "Experimental tomography of noon states with large photon numbersa method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons," Kongelige Danske Videnskabernes Selskab. **5**, 1 (1948).
[96] L. R. Dice, "Measures of the amount of ecologic association between species," Ecology **26**, 297–302 (1945).
[97] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," ACM Comput. Surv. **49** (2016), 10.1145/2907070.

APPENDIX TABLES AND FIGURES

d	data type	N	m	bs/loss (ICCNet)	bs/loss (FidNet)
4	random projectors	500	10000	256/MAE	1024/MSE
4	ACT bases	5000	10000	256/MAE	1024/MSE
6	ACT bases	5000	10000	256/MAE	1024/MSE
9	ACT bases	5000	10000	256/MAE	1024/MSE
16	Haar bases	1000	5000	256/MAE	1024/MSE
16	Haar bases	Inf	5000	256/MAE	1024/MSE
16	ACT bases	1000	5000	256/MAE	1024/MSE
16	ACT bases	Inf	5000	256/MAE	1024/MSE
32	Haar bases	1000	1000	64/MAE	512/MSE
64	Haar bases	1000	1000	64/MAE	512/MSE

TABLE I. A table of the simulated training data types, number of copies N per basis or projector, number of training datasets m for each value of K or L, and the batch size (bs) and loss function (loss) used for training the respective Nets. For the first data type of $d = 4$, the random projectors are chosen from the fixed set defined by Eq. (3.2) and the 16 measurement angles listed in Fig. 11 in the main text.

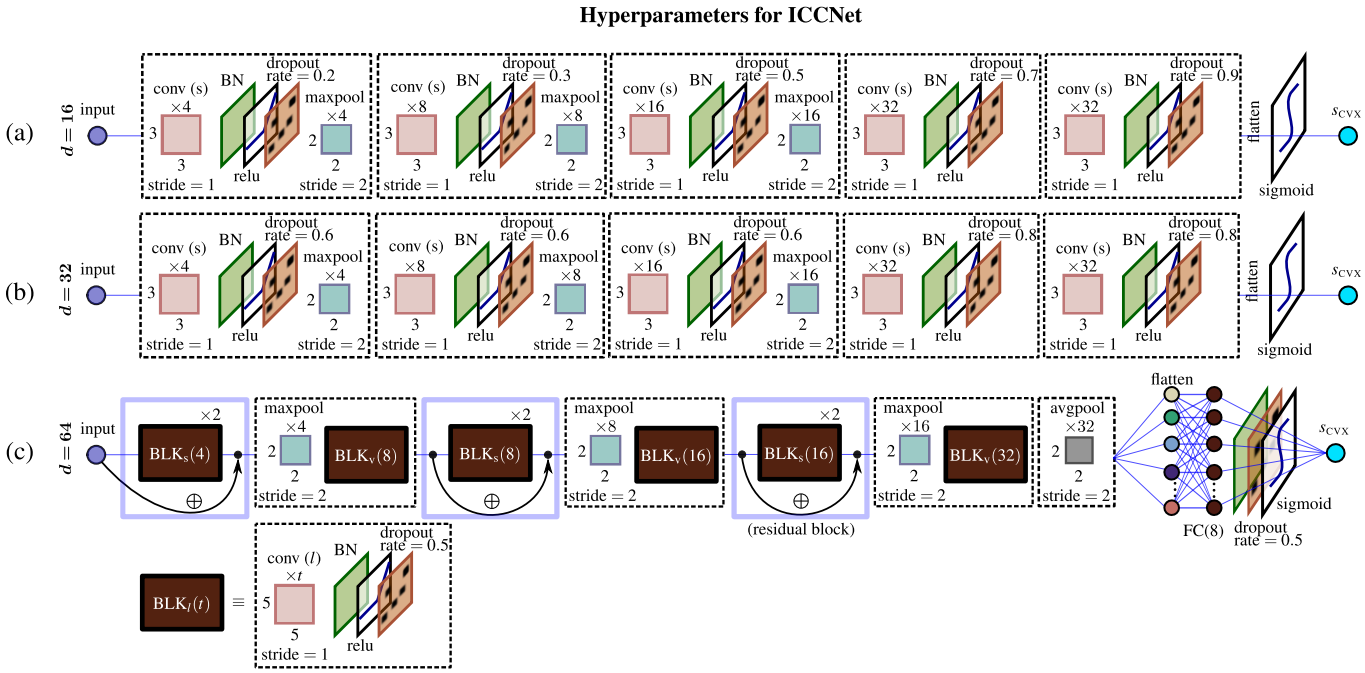


FIG. 11. A list of hyperparameters for ICCNet (filter number, kernel or filter dimensions, stride length and dropout rate) used during training for various Hilbert-space dimensions. The notations conv (s) and conv (v) signifies whether zeros are padded to the output (same) in order to maintain the same array dimensions after going through the convolution layer or not (valid). For $d = 4$, 6 and 9, network (a) is adopted with the first two dropout rates set to zero. Additionally, for $d = 4$, we change the max-pooling stride length to 1. Network (c) consists of a deeper CNN layer with additional elements like an average-pooling layer and a fully-connected layer with 8 artificial neurons.

		$(d=16)$	$K=2$	3	4	5	6	7	8	9	10	$K=2$	3	4	5	6	7	8	9	10		
Haar bases (N =Inf)	$r=1$	**act. avg.**	0.20	0.80	4.26	6.42	6.93	7.41	7.53	7.83	7.84	0.32	1.55	4.57	6.10	5.87	5.83	5.85	5.87	5.82	$d=4$	Hermite-Gaussian modes
		pred. avg.	0.17	0.65	3.98	5.64	6.66	7.00	8.09	8.44	8.54	0.22	1.75	5.21	6.18	6.63	6.65	6.70	6.61	6.65		
		avg. mae	0.05	0.23	1.35	0.89	0.64	0.61	0.62	0.68	0.76	0.15	1.00	1.49	0.54	1.03	0.86	0.88	0.81	0.88		
	$r=2$	**act. avg.**	0.10	0.25	0.46	0.84	2.2	5.18	6.43	6.95	7.33	0.24	1.09	3.51	6.63	6.75	6.95	6.79	6.72	6.86	$d=6$	
		pred. avg.	0.10	0.25	0.40	1.24	2.82	4.5	5.37	7.07	7.57	0.21	1.27	4.51	5.30	5.56	6.34	6.33	6.50	6.50		
		avg. mae	0.02	0.05	0.10	0.64	1.42	1.23	1.22	0.69	0.66	0.08	0.46	1.79	2.42	1.39	1.02	0.93	0.87	0.86		
	$r=3$	**act. avg.**	0.08	0.18	0.31	0.47	0.72	1.12	2.67	5.31	6.34	0.22	0.87	2.75	5.95	6.63	6.09	6.38	6.20	6.28	$d=9$	
		pred. avg.	0.08	0.18	0.29	0.51	0.86	2.36	3.97	5.04	5.81	0.14	1.81	5.23	5.65	6.27	6.87	7.68	5.69	6.14		
		avg. mae	0.02	0.02	0.06	0.15	0.26	1.35	1.57	0.95	1.11	0.10	1.17	3.03	2.24	1.04	1.35	1.48	1.02	1.13		
Haar bases ($N=5000$)	$r=1$	**act. avg.**	0.23	1.76	4.14	4.65	5.44	6.12	7.50	8.10	8.39	0.32	1.55	4.57	6.10	5.87	5.83	5.85	5.87	5.82	$d=4$	Hermite-Gaussian modes
		pred. avg.	0.17	0.64	3.98	5.58	6.72	7.02	8.09	8.47	8.56	0.31	1.39	5.45	6.44	6.78	6.76	6.74	6.67	6.70		
		avg. mae	0.07	1.17	0.46	1.07	1.48	1.42	0.76	0.59	0.48	0.15	1.02	1.61	0.53	0.95	0.96	0.92	0.87	0.94		
	$r=2$	**act. avg.**	0.11	0.28	0.72	3.32	4.35	4.53	4.84	6.13	7.12	0.24	1.09	3.51	6.63	6.75	6.95	6.79	6.72	6.86	$d=6$	
		pred. avg.	0.10	0.27	0.68	1.58	3.28	4.86	5.73	7.14	7.63	0.22	1.61	5.27	6.40	5.75	6.37	6.48	6.53	6.47		
		avg. mae	0.02	0.07	0.48	1.86	1.35	0.62	1.01	1.29	1.17	0.10	0.83	2.47	2.16	1.41	1.16	0.95	0.88	0.92		
	$r=3$	**act. avg.**	0.09	0.20	0.35	0.65	2.51	4.33	4.49	4.71	5.08	0.22	0.87	2.75	5.95	6.63	6.09	6.38	6.20	6.28	$d=9$	
		pred. avg.	0.08	0.19	0.29	0.55	1.09	2.86	4.16	5.30	5.82	0.32	1.30	5.41	5.46	6.42	6.70	7.42	6.27	6.23		
		avg. mae	0.02	0.03	0.07	0.26	1.56	1.48	0.50	0.78	0.99	0.14	0.91	2.68	2.19	1.27	1.16	1.52	1.07	1.05		
ACT bases (N =Inf)	$r=1$	**act. avg.**	0.19	0.54	6.74	8.25	8.1	8.14	8.12	8.26	8.30											
		pred. avg.	0.17	0.69	6.64	6.56	7.29	7.87	8.18	8.19	8.32											
		avg. mae	0.05	0.17	1.12	1.72	0.86	0.69	0.66	0.61	0.73											
	$r=2$	**act. avg.**	0.10	0.24	0.46	0.95	6.12	7.04	7.39	7.62	7.73											
		pred. avg.	0.10	0.24	0.42	1.13	5.78	6.55	6.91	7.53	7.64											
		avg. mae	0.02	0.05	0.16	0.49	0.99	0.81	0.75	0.37	0.35											
	$r=3$	**act. avg.**	0.08	0.17	0.30	0.47	0.69	1.96	5.62	6.27	6.76											
		pred. avg.	0.08	0.17	0.25	0.45	1.19	3.38	5.19	6.60	6.70											
		avg. mae	0.01	0.02	0.06	0.11	0.59	1.62	0.99	0.87	0.60											
ACT bases ($N=5000$)	$r=1$	**act. avg.**	0.25	1.36	4.50	4.98	6.34	6.73	7.65	7.8	8.23											
		pred. avg.	0.18	0.68	5.02	6.31	7.19	7.75	8.14	8.15	8.30											
		avg. mae	0.10	0.78	0.53	1.38	1.05	1.23	0.62	0.53	0.49											
	$r=2$	**act. avg.**	0.10	0.26	0.68	2.65	4.32	4.52	4.69	5.40	5.88											
		pred. avg.	0.10	0.26	0.67	1.48	4.72	4.94	4.77	5.64	5.62											
		avg. mae	0.02	0.07	0.39	1.46	0.77	0.49	0.24	0.74	0.90											
	$r=3$	**act. avg.**	0.08	0.19	0.33	0.68	2.43	4.18	4.33	4.47	4.66											
		pred. avg.	0.08	0.18	0.26	0.43	1.04	3.13	4.52	4.44	4.55											
		avg. mae	0.02	0.03	0.08	0.30	1.62	1.15	0.28	0.13	0.18											

(noise-trained)

		$(d=4)$	$L=2$	3	4	5	6	7	8	9	10	11	12	13	14	15
Three-photon states	state 1	**act. avg.**	0.03	0.10	0.16	0.21	0.35	0.80	1.93	3.09	4.27	4.37	7.05	8.28	8.71	8.51
		pred. avg.	0.04	0.07	0.12	0.20	0.29	0.36	0.47	0.78	0.72	1.28	3.20	6.70	7.25	5.02
		avg. mae	0.03	0.07	0.08	0.07	0.12	0.50	1.53	2.44	3.56	3.17	4.08	1.93	1.52	3.49
	state 2	**act. avg.**	0.04	0.09	0.15	0.23	0.31	0.41	0.73	1.74	2.72	3.84	7.05	7.28	7.88	8.53
		pred. avg.	0.03	0.07	0.12	0.19	0.27	0.36	0.45	0.71	0.88	1.51	4.70	6.30	7.31	6.55
		avg. mae	0.02	0.05	0.06	0.07	0.10	0.11	0.34	1.09	1.84	2.34	2.81	2.24	0.61	1.98
	state 3	**act. avg.**	0.05	0.06	0.10	0.15	0.21	0.26	0.34	0.47	0.87	1.50	4.41	5.56	5.53	5.42
		pred. avg.	0.04	0.07	0.10	0.16	0.24	0.29	0.39	0.46	0.65	0.90	1.17	1.48	2.59	5.01
		avg. mae	0.02	0.05	0.04	0.06	0.08	0.09	0.11	0.11	0.35	0.85	3.41	4.27	2.95	0.42
(noise-trained) Three-photon states	state 1	**act. avg.**	0.03	0.10	0.16	0.21	0.35	0.80	1.93	3.09	4.27	4.37	7.05	8.28	8.71	8.51
		pred. avg.	0.04	0.09	0.13	0.22	0.35	0.46	0.65	2.30	3.64	4.53	7.91	8.23	8.66	8.51
		avg. mae	0.03	0.05	0.08	0.08	0.13	0.48	1.45	1.78	1.58	0.97	1.36	0.61	0.34	0.28
	state 2	**act. avg.**	0.04	0.09	0.15	0.23	0.31	0.41	0.73	1.74	2.72	3.84	7.05	7.28	7.88	8.53
		pred. avg.	0.04	0.09	0.13	0.20	0.26	0.37	0.52	0.61	1.13	2.59	6.61	6.92	8.15	8.50
		avg. mae	0.01	0.03	0.05	0.06	0.11	0.13	0.28	1.15	1.75	1.69	1.92	2.23	1.13	0.11
	state 3	**act. avg.**	0.05	0.06	0.10	0.15	0.21	0.26	0.34	0.47	0.87	1.50	4.41	5.56	5.53	5.42
		pred. avg.	0.05	0.07	0.10	0.17	0.26	0.32	0.40	0.49	0.63	0.84	1.79	4.38	5.08	5.39
		avg. mae	0.01	0.02	0.04	0.07	0.07	0.09	0.11	0.12	0.39	0.82	3.79	3.11	0.46	0.07

TABLE II. Table of the actual averages, predicted ones and average MAE of $-\log_{10} s_{\mathrm{cvx}}$ that are obtained from all trained ICCNet models.

(noise-trained)

		$(d=16)$	K 1	2	3	4	5	6	7	8	9	10
Haar bases (N =Inf)	$r=1$	**act. avg.**	0.11	0.21	0.52	0.98	1.00	1.00	1.00	1.00	1.00	1.00
		pred. avg.	0.11	0.21	0.52	0.91	0.98	0.97	0.99	0.99	0.99	0.99
		avg. mae	0.01	0.03	0.10	0.08	0.02	0.03	0.01	0.01	0.01	0.01
	$r=2$	**act. avg.**	0.17	0.24	0.31	0.43	0.64	0.89	1.00	1.00	1.00	1.00
		pred. avg.	0.18	0.25	0.33	0.46	0.67	0.85	0.92	0.94	0.95	0.95
		avg. mae	0.01	0.02	0.04	0.05	0.08	0.08	0.08	0.06	0.05	0.05
	$r=3$	**act. avg.**	0.22	0.27	0.33	0.41	0.51	0.64	0.81	0.95	1.00	1.00
		pred. avg.	0.22	0.28	0.33	0.42	0.52	0.67	0.79	0.88	0.92	0.93
		avg. mae	0.01	0.02	0.02	0.03	0.04	0.06	0.06	0.08	0.08	0.07
Haar bases ($N=5000$)	$r=1$	**act. avg.**	0.12	0.23	0.58	0.84	0.91	0.93	0.95	0.95	0.96	0.96
		pred. avg.	0.11	0.21	0.52	0.91	0.98	0.97	0.99	0.99	0.99	0.99
		avg. mae	0.01	0.04	0.15	0.08	0.07	0.04	0.04	0.04	0.03	0.03
	$r=2$	**act. avg.**	0.18	0.24	0.33	0.47	0.69	0.81	0.85	0.88	0.90	0.92
		pred. avg.	0.18	0.25	0.34	0.47	0.68	0.84	0.91	0.95	0.95	0.95
		avg. mae	0.01	0.03	0.03	0.06	0.08	0.05	0.06	0.06	0.04	0.03
	$r=3$	**act. avg.**	0.23	0.28	0.34	0.42	0.54	0.69	0.78	0.82	0.85	0.87
		pred. avg.	0.22	0.29	0.33	0.42	0.53	0.69	0.80	0.89	0.92	0.93
		avg. mae	0.01	0.02	0.02	0.03	0.05	0.08	0.05	0.07	0.07	0.06
ACT bases (N =Inf)	$r=1$	**act. avg.**	0.12	0.22	0.51	0.95	1.00	1.00	1.00	1.00	1.00	1.00
		pred. avg.	0.11	0.21	0.52	0.96	0.99	0.99	1.00	1.00	1.00	1.00
		avg. mae	0.01	0.04	0.12	0.05	0.01	0.01	0.00	0.00	0.00	0.00
	$r=2$	**act. avg.**	0.17	0.24	0.33	0.47	0.75	0.99	1.00	1.00	1.00	1.00
		pred. avg.	0.18	0.25	0.34	0.47	0.77	0.98	0.99	1.00	1.00	1.00
		avg. mae	0.01	0.02	0.04	0.06	0.06	0.02	0.01	0.00	0.00	0.00
	$r=3$	**act. avg.**	0.23	0.28	0.34	0.43	0.53	0.66	0.93	1.00	1.00	1.00
		pred. avg.	0.22	0.29	0.34	0.42	0.53	0.66	0.92	0.99	1.00	1.00
		avg. mae	0.01	0.02	0.02	0.03	0.04	0.05	0.02	0.01	0.00	0.00
ACT bases ($N=5000$)	$r=1$	**act. avg.**	0.12	0.21	0.51	0.87	0.96	0.98	0.98	0.98	0.99	0.99
		pred. avg.	0.11	0.21	0.53	0.96	0.99	0.99	0.99	1.00	1.00	1.00
		avg. mae	0.01	0.03	0.12	0.09	0.03	0.01	0.02	0.01	0.01	0.01
	$r=2$	**act. avg.**	0.18	0.24	0.34	0.52	0.72	0.84	0.89	0.92	0.94	0.95
		pred. avg.	0.18	0.25	0.35	0.48	0.73	0.87	0.94	0.95	0.95	0.97
		avg. mae	0.01	0.03	0.05	0.09	0.06	0.05	0.05	0.03	0.02	0.03
	$r=3$	**act. avg.**	0.23	0.28	0.34	0.42	0.54	0.67	0.76	0.81	0.85	0.88
		pred. avg.	0.22	0.29	0.34	0.42	0.53	0.67	0.76	0.82	0.85	0.88
		avg. mae	0.01	0.02	0.03	0.04	0.07	0.05	0.03	0.04	0.04	0.03

(noise-trained)

		K 1	2	3	4	5	6	7	8	9	10
Hermite-Gaussian modes $d=4$	**act. avg.**	0.48	0.65	0.83	0.86	0.92	0.95	0.96	0.97	0.98	0.98
	pred. avg.	0.50	0.72	0.90	0.95	0.98	0.98	0.97	0.98	0.97	0.97
	avg. mae	0.07	0.16	0.11	0.11	0.05	0.03	0.03	0.03	0.04	0.05
Hermite-Gaussian modes $d=6$	**act. avg.**	0.29	0.49	0.75	0.81	0.90	0.94	0.96	0.97	0.97	0.98
	pred. avg.	0.33	0.49	0.83	0.92	0.97	0.92	0.98	0.95	0.95	0.98
	avg. mae	0.08	0.17	0.14	0.15	0.07	0.06	0.03	0.05	0.05	0.02
Hermite-Gaussian modes $d=9$	**act. avg.**	0.32	0.53	0.68	0.79	0.89	0.93	0.95	0.96	0.97	0.97
	pred. avg.	0.20	0.50	0.75	0.86	0.90	0.97	0.90	0.87	0.90	0.97
	avg. mae	0.14	0.14	0.16	0.14	0.08	0.05	0.07	0.11	0.07	0.02
Hermite-Gaussian modes $d=4$	**act. avg.**	0.48	0.65	0.83	0.86	0.92	0.95	0.96	0.97	0.98	0.98
	pred. avg.	0.49	0.64	0.83	0.94	0.97	0.97	0.98	0.98	0.97	0.97
	avg. mae	0.05	0.17	0.14	0.10	0.05	0.05	0.03	0.03	0.04	0.04
Hermite-Gaussian modes $d=6$	**act. avg.**	0.29	0.49	0.75	0.81	0.90	0.94	0.96	0.97	0.97	0.98
	pred. avg.	0.27	0.45	0.75	0.90	0.94	0.94	0.97	0.96	0.96	0.96
	avg. mae	0.05	0.19	0.13	0.11	0.05	0.04	0.02	0.03	0.04	0.03
Hermite-Gaussian modes $d=9$	**act. avg.**	0.32	0.53	0.68	0.79	0.89	0.93	0.95	0.96	0.97	0.97
	pred. avg.	0.33	0.38	0.65	0.90	0.94	0.93	0.93	0.91	0.88	0.93
	avg. mae	0.04	0.20	0.14	0.12	0.05	0.03	0.04	0.07	0.09	0.05

(noise-trained)

		$(d=4)$	L 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Three-photon states state 1		**act. avg.**	0.38	0.44	0.49	0.47	0.47	0.46	0.52	0.70	0.78	0.86	0.89	0.89	0.91	0.92	0.93
		pred. avg.	0.34	0.40	0.37	0.46	0.48	0.49	0.58	0.64	0.77	0.83	0.89	0.86	0.88	0.91	0.92
		avg. mae	0.07	0.09	0.15	0.13	0.18	0.17	0.18	0.18	0.12	0.08	0.05	0.05	0.04	0.02	0.01
Three-photon states state 2		**act. avg.**	0.12	0.21	0.30	0.30	0.31	0.32	0.31	0.43	0.65	0.78	0.81	0.85	0.87	0.87	0.88
		pred. avg.	0.13	0.18	0.21	0.34	0.35	0.33	0.56	0.61	0.74	0.81	0.89	0.87	0.87	0.90	0.92
		avg. mae	0.05	0.11	0.13	0.12	0.16	0.15	0.27	0.26	0.17	0.10	0.09	0.04	0.03	0.03	0.04
Three-photon states state 3		**act. avg.**	0.26	0.29	0.33	0.34	0.41	0.47	0.47	0.57	0.63	0.74	0.80	0.89	0.91	0.93	0.94
		pred. avg.	0.29	0.29	0.31	0.38	0.42	0.47	0.54	0.53	0.61	0.70	0.81	0.87	0.85	0.89	0.92
		avg. mae	0.05	0.08	0.10	0.08	0.06	0.08	0.14	0.13	0.12	0.11	0.12	0.06	0.07	0.03	0.02
(noise-trained) Three-photon states state 1		**act. avg.**	0.38	0.44	0.49	0.47	0.47	0.46	0.52	0.70	0.78	0.86	0.89	0.89	0.91	0.92	0.93
		pred. avg.	0.37	0.41	0.43	0.42	0.43	0.43	0.53	0.61	0.74	0.81	0.85	0.88	0.92	0.94	0.94
		avg. mae	0.06	0.08	0.12	0.14	0.17	0.16	0.15	0.19	0.13	0.09	0.06	0.05	0.03	0.02	0.01
Three-photon states state 2		**act. avg.**	0.12	0.21	0.30	0.30	0.31	0.32	0.31	0.43	0.65	0.78	0.81	0.85	0.87	0.87	0.88
		pred. avg.	0.14	0.22	0.26	0.28	0.29	0.32	0.42	0.51	0.64	0.78	0.80	0.84	0.90	0.90	0.91
		avg. mae	0.06	0.08	0.11	0.12	0.14	0.14	0.19	0.20	0.16	0.09	0.08	0.05	0.03	0.03	0.03
Three-photon states state 3		**act. avg.**	0.26	0.29	0.33	0.34	0.41	0.47	0.47	0.57	0.63	0.74	0.80	0.89	0.91	0.93	0.94
		pred. avg.	0.26	0.28	0.32	0.36	0.40	0.46	0.49	0.56	0.61	0.71	0.79	0.83	0.89	0.92	0.93
		avg. mae	0.04	0.07	0.08	0.08	0.08	0.06	0.10	0.13	0.14	0.11	0.10	0.07	0.04	0.01	0.01

TABLE III. Table of the actual averages, predicted ones and average MAE of $\mathcal{F}$ that are obtained from all trained FidNet models.

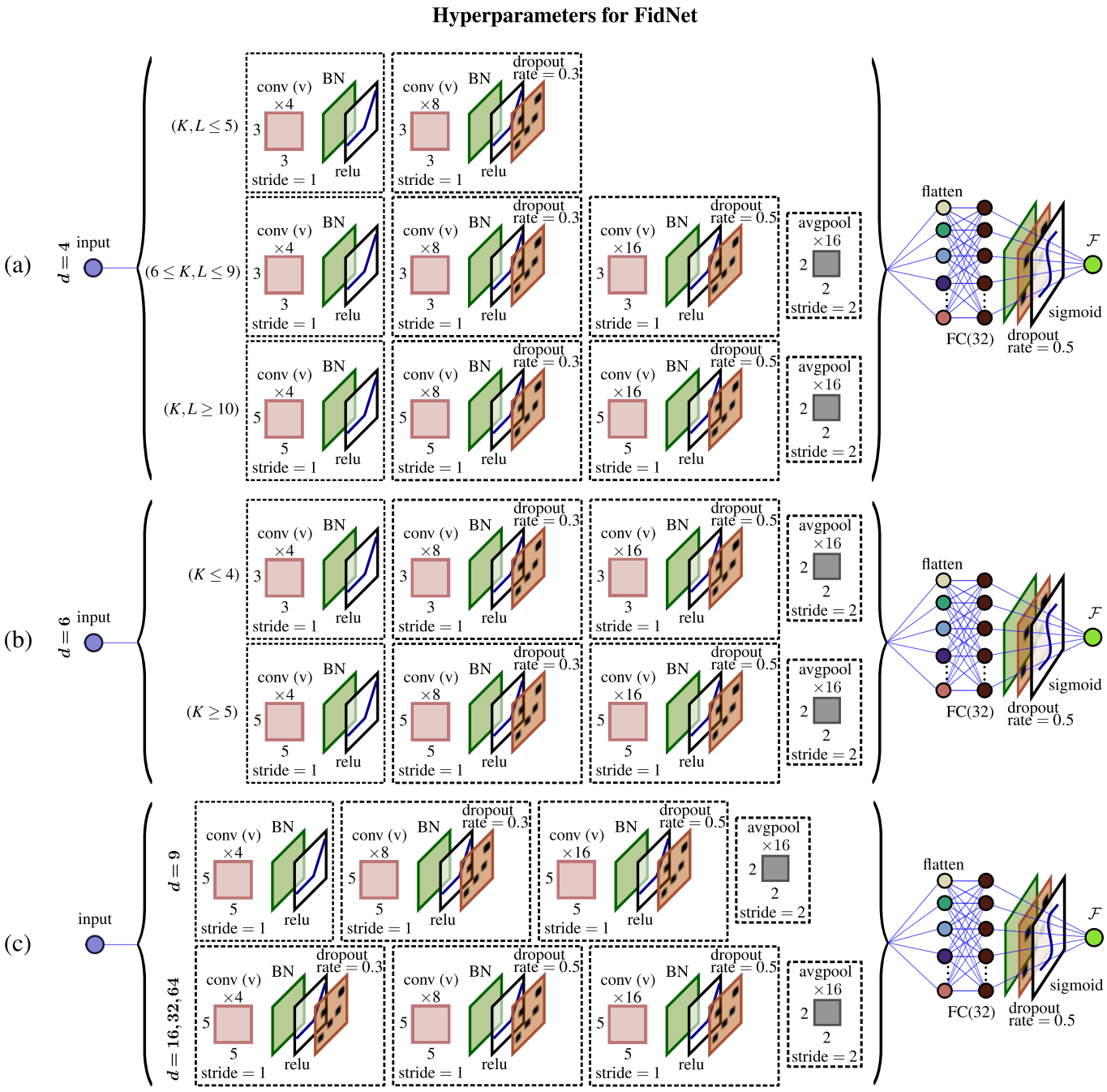


FIG. 12. The hyperparameter settings used to train FidNet and plot the figures in the main article. A single average-pooling layer is sufficient in all the FidNet architectures.

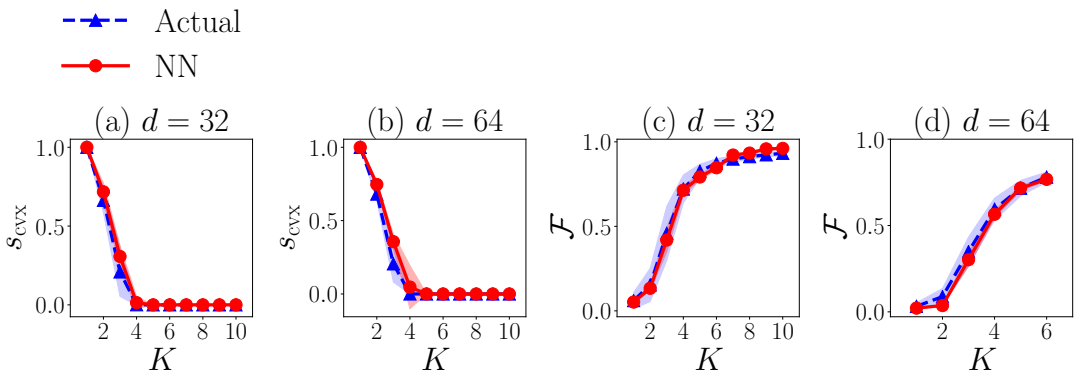


FIG. 13. Performances of ICCNet and FidNet for pure quantum systems of dimensions (a,c) $d = 32$ and (b,d) $d = 64$ obtained for the sake of generating Fig. 8 in the main text. Statistical noise from $N = 1000$ copies per basis has been considered (refer to Tab. I). In this case, the FidNet is trained to recognize fidelities with the "right" target states for simplicity, which is valid as no systematic errors are present here. FidNet training stops at $K = 6$ for $d = 64$ due to limited GPU resources.

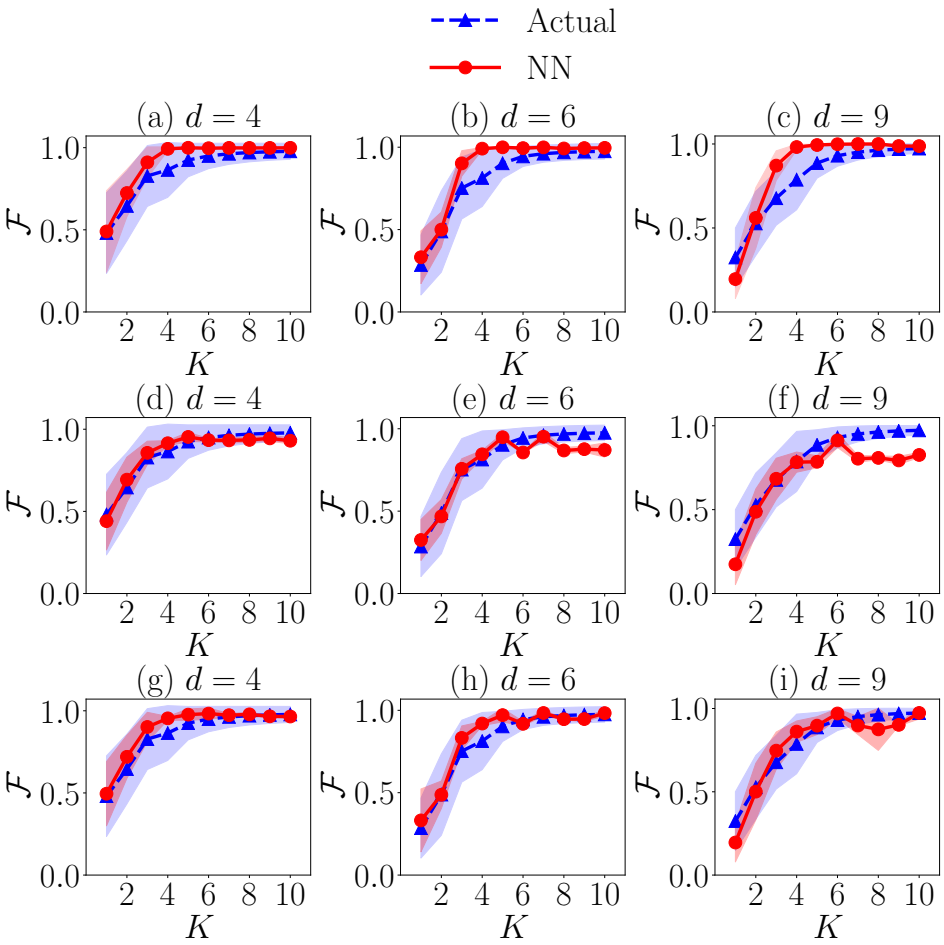


FIG. 14. Performances of FidNet for benchmarking spatial-mode photonic datasets (a,b,c) with the "right" target states, (d,e,f) the "wrong" target states, and (g,h,i) both types of target states. Average-fidelity benchmarking accuracies are higher when FidNet is trained with both "right" and "wrong" target states.

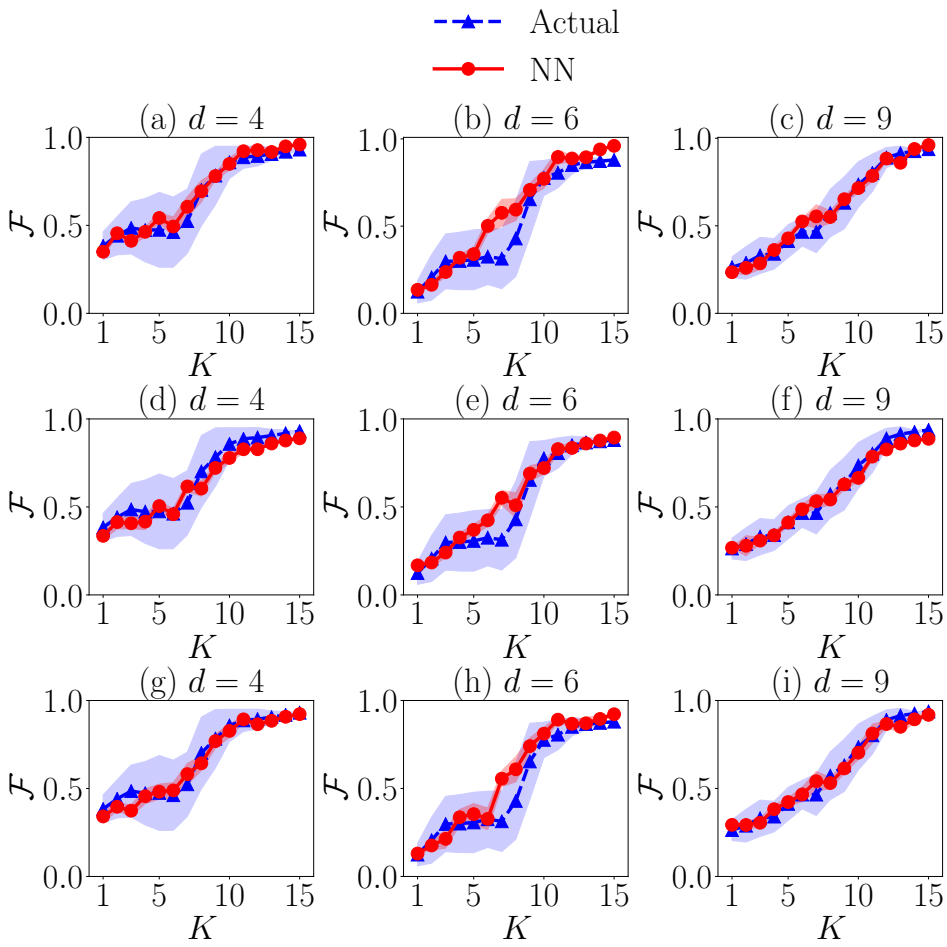


FIG. 15. Performances of FidNet for benchmarking three-photon datasets, where all specifications follow those of Fig. 14. For these three tested states, training with all the different types of target states give comparable accuracies.

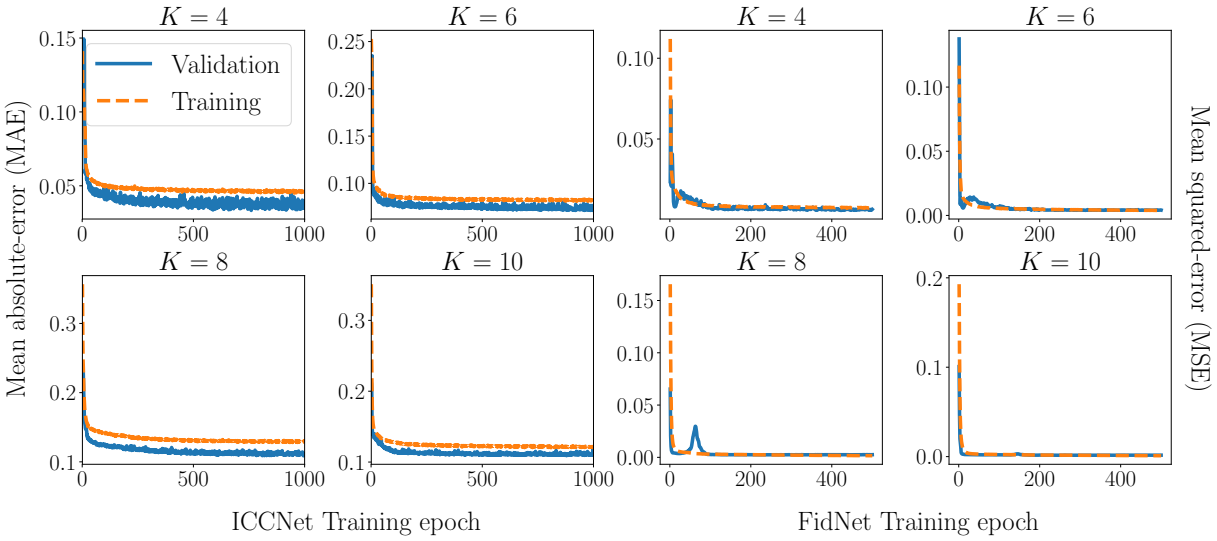


FIG. 16. The progress of both ICCNet and FidNet training and validation loss values with the number of training epochs (iterative steps) for $d = 16$. All four data types (see Tab. I) are stacked and trained using a common architecture for each Net. The loss values are computed with outputs $\boldsymbol{y}_{\text{train}}$ and $\boldsymbol{y}_{\text{val}}$. The trained models corresponding to the lowest validation loss are saved.

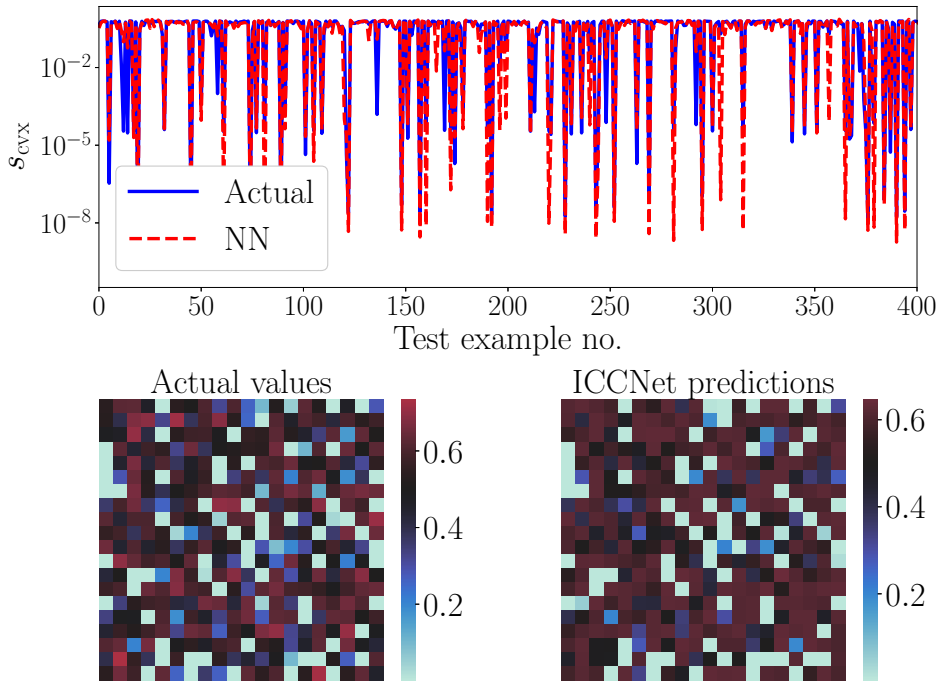


FIG. 17. A sample of 400 s_{cvx} test values of assorted datasets (a mixture of all the $d = 16$ data types listed in Tab. I) unseen during ICCNet training for $K = 4$. The 20×20 heat maps that respectively represent these 400 actual and their corresponding ICCNet predicted (NN) values serve to facilitate a more convenient visual comparison.

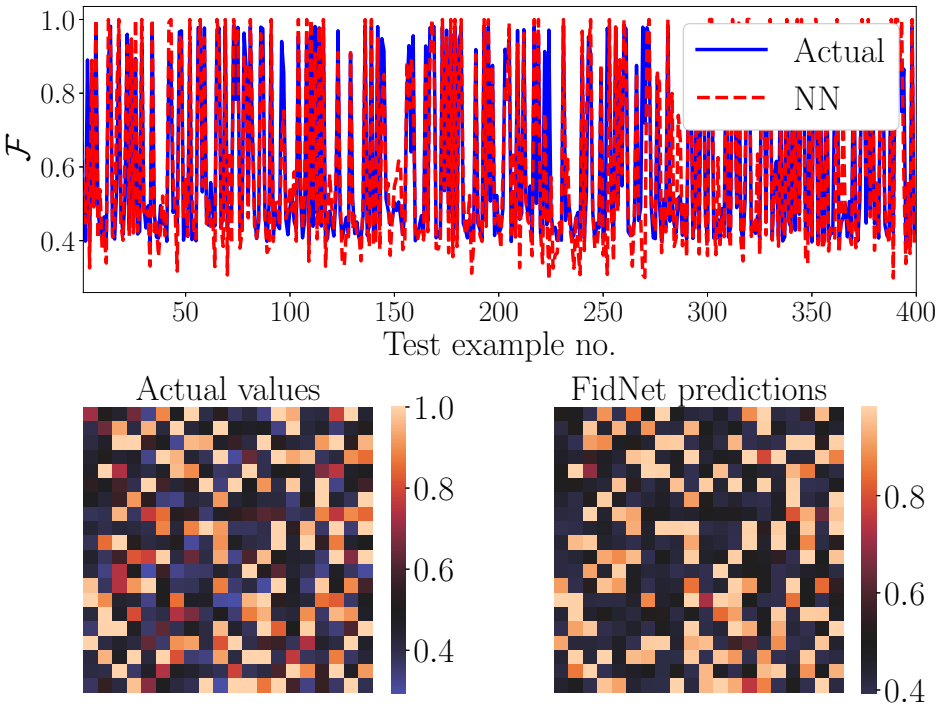


FIG. 18. A sample of 400 fidelity ($\mathcal{F}$) test values of assorted datasets (a mixture of all the $d = 16$ data types listed in Tab. I) unseen during FidNet training for $K = 4$. The 20×20 heat maps that respectively represent these 400 actual and their corresponding FidNet predicted (NN) values serve to facilitate a more convenient visual comparison.

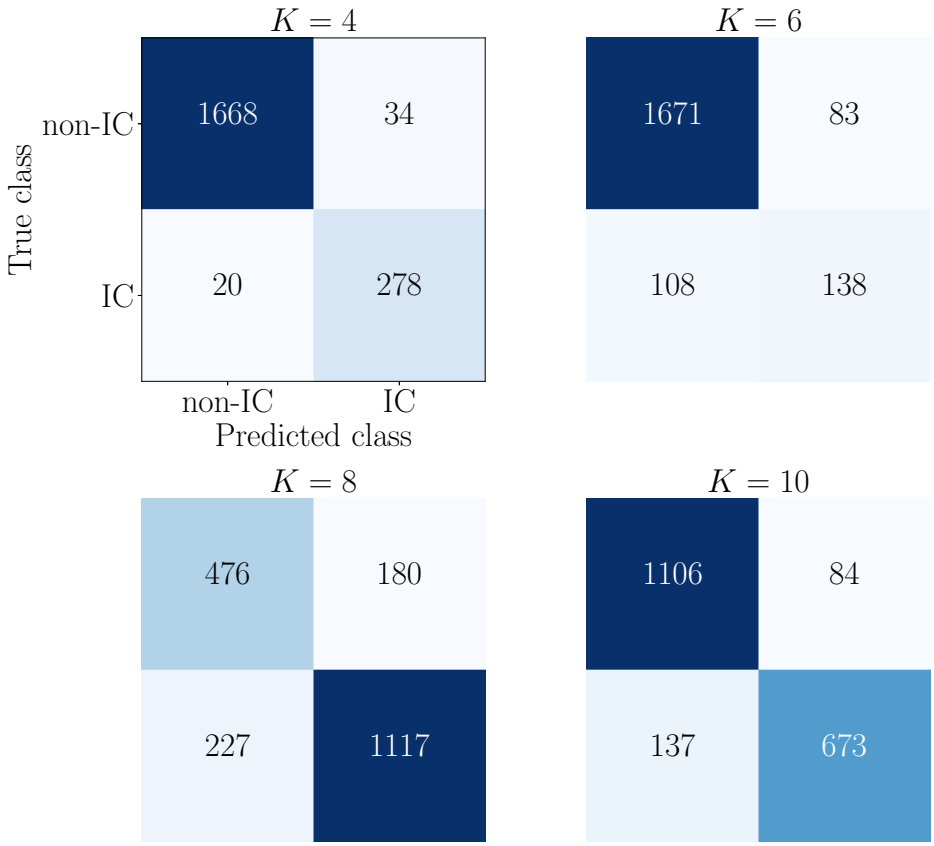


FIG. 19. The confusion matrix for $d = 16$ and various K values that classifies a total of 2000 test examples previously unseen by ICCNet. The larger the diagonals, the better the ICCNet prediction quality, as they represent successful predictions of correct classes of values. The respective F1 scores, in ascending order of K shown here, are 0.911, 0.591, 0.846 and 0.859.

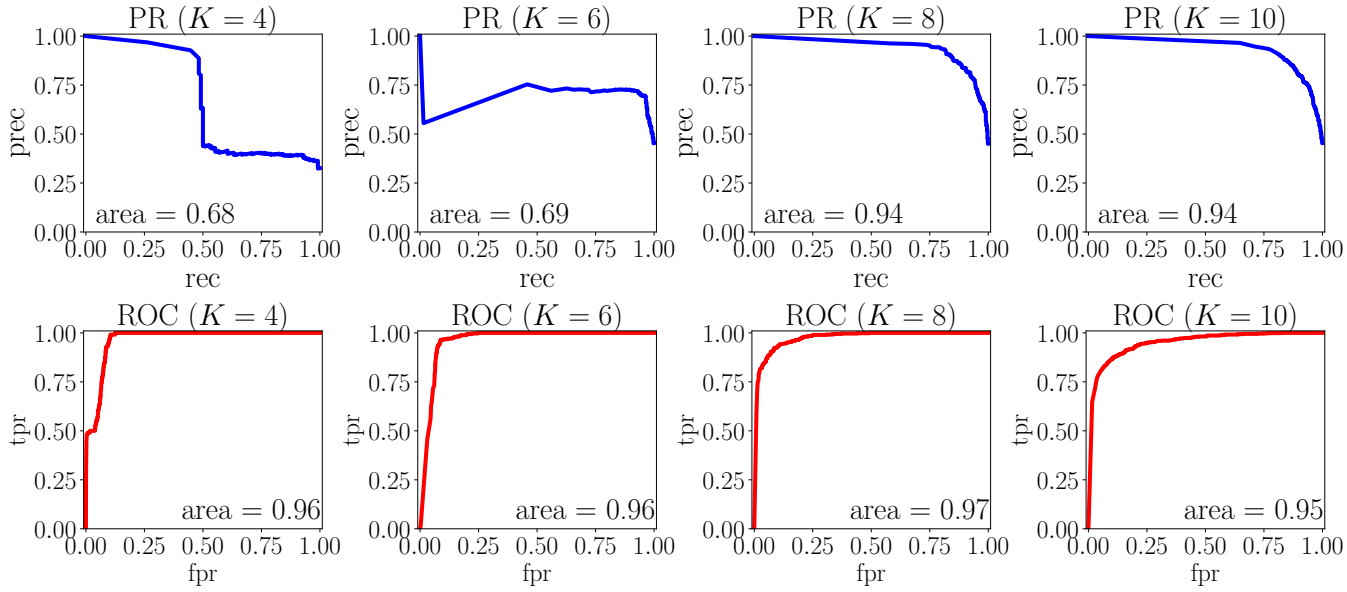


FIG. 20. Area under the PR and ROC curves.