

Neural networks for on-the-fly single-shot state classification

Cite as: Appl. Phys. Lett. **119**, 114003 (2021); <https://doi.org/10.1063/5.0065011>

Submitted: 29 July 2021 • Accepted: 01 September 2021 • Published Online: 13 September 2021

 Rohit Navarathna,  Tyler Jones,  Tina Moghaddam, et al.

COLLECTIONS

Paper published as part of the special topic on [Emerging Qubit Systems - Novel Materials, Encodings and Architectures](#)



View Online



Export Citation



CrossMark

ARTICLES YOU MAY BE INTERESTED IN

[Non-adiabatic single-electron pumps in a dopant-free GaAs/AlGaAs 2DEG](#)

Applied Physics Letters **119**, 114001 (2021); <https://doi.org/10.1063/5.0062486>

[Near-field terahertz nanoscopy of coplanar microwave resonators](#)

Applied Physics Letters **119**, 091101 (2021); <https://doi.org/10.1063/5.0061078>

[A quantum engineer's guide to superconducting qubits](#)

Applied Physics Reviews **6**, 021318 (2019); <https://doi.org/10.1063/1.5089550>

 QBLOX



1 qubit

Shorten Setup Time

Auto-Calibration

More Qubits

Fully-integrated

Quantum Control Stacks

Ultrastable DC to 18.5 GHz

Synchronized <<1 ns

Ultralow noise



100s qubits

[visit our website >](#)

Neural networks for on-the-fly single-shot state classification

Cite as: Appl. Phys. Lett. **119**, 114003 (2021); doi: [10.1063/5.0065011](https://doi.org/10.1063/5.0065011)

Submitted: 29 July 2021 · Accepted: 1 September 2021 ·

Published Online: 13 September 2021



View Online



Export Citation



CrossMark

Rohit Navarathna,^{1,2,a)} Tyler Jones,^{1,2,3} Tina Moghaddam,^{1,2} Anatoly Kulikov,^{1,2,4} Rohit Beriwal,^{1,2} Markus Jerger,^{1,2,5} Prasanna Pakkiam,^{1,2} and Arkady Fedorov,^{1,2}

AFFILIATIONS

¹ARC Centre of Excellence for Engineered Quantum Systems, St Lucia, Queensland 4072, Australia

²School of Mathematics and Physics, University of Queensland, St Lucia, Queensland 4072, Australia

³Max Kelsen, Spring Hill, Queensland 4000, Australia

⁴Department of Physics, ETH Zürich, CH-8093 Zürich, Switzerland

⁵JARA-FIT Institute for Quantum Information, Forschungszentrum Jülich, 52425 Jülich, Germany

Note: This paper is part of the APL Special Collection on Emerging Qubit Systems - Novel Materials, Encodings and Architectures.

a) Author to whom correspondence should be addressed: r.navarathna@uq.edu.au

ABSTRACT

Neural networks have proven to be efficient for a number of practical applications ranging from image recognition to identifying phase transitions in quantum physics models. In this paper, we investigate the application of neural networks to state classification in a single-shot quantum measurement. We use dispersive readout of a superconducting transmon circuit to demonstrate an increase in assignment fidelity for both two and three state classifications. More importantly, our method is ready for on-the-fly data processing without overhead or need for large data transfer to a hard drive. In addition, we demonstrate the capacity of neural networks to be trained against experimental imperfections, such as phase drift of a local oscillator in a heterodyne detection scheme.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0065011>

Machine learning (ML) is ubiquitous in modern computer science with a wide range of applications. By virtue of its archetypal problem classes—regression and classification—ML algorithms and neural networks, in particular, have recently found a number of applications in quantum computing, helping researchers to tackle such tasks as optimizing gates and pulse sequences,^{1–4} identifying phase transitions,^{5,6} correcting imperfections of measurement apparatus,^{7–9} classifying states^{10,11} or evolution^{12–14} of a quantum system with little or no *a priori* knowledge, and even optimizing the fabrication process.¹⁵

A proposal to use machine learning to discriminate measurement trajectories was one of the first and most natural applications of ML in the field and has led to improvements in readout assignment fidelity.¹⁶ The technique is now being regularly implemented across the community^{17–19} due to their generalizability and capacity to extract useful features from dense data. A recent advancement uses neural networks to compensate for system dependent errors due to processes, such as crosstalk in multiplexed qubit readout.²⁰ In this work, we also apply neural networks to the readout of a superconducting transmon system. However, our approach works on-the-fly with no data processing overhead and can be trained against experimental parameter drifts.

To deploy our neural-network-based state classification, we use an open source PyTorch library.²¹ Geared toward computer vision and natural language processing, it includes the capability to realize deep neural networks and contains built-in functionality for data processing on a graphics processing unit (GPU). GPU integration enables our pipeline to be fast enough to perform on-the-fly data classification without the need to transfer raw measured signal to a hard drive. Among other advantages, it allows monitoring the readout assignment fidelity in real time.

With the initial training of the neural network taking on the order of minutes, consequent retraining of network weights requires several seconds and allows the readout assignment fidelity to return to the optimal value. More importantly, the convolutional neural network used in the present work may be designed and trained in a way resilient to certain experimental parameter drifts. Specifically, we present a strategy to eliminate the effect of local relative phase drifts induced by generating microwave equipment on the readout assignment fidelity.

In our experiment, we used a primitive of the circuit quantum electrodynamics platform: a transmon coupled to a readout cavity.

After initializing the qubit in its ground state, we used Gaussian pulses of length 20 ns to prepare the transmon in the three basis states: applying no pulse to keep the transmon in its ground state $|g\rangle$, applying one π -pulse at ω_{ge} to prepare the transmon in its first excited state $|e\rangle$, and applying two consecutive π -pulses at ω_{ge} and ω_{ef} respectively, to prepare a transmon in the second excited state $|f\rangle$. These protocols are illustrated in Fig. 1(a).

After the state preparation, we perform measurement by acquiring the probe signal transmitted through the readout resonator. To achieve the single-shot regime, we used a Josephson parametric amplifier (JPA).²² Following amplification through the JPA, the readout signal is further amplified using a High Electron Mobility Transistor (HEMT) amplifier and multiple room temperature amplifiers. The signal is then downconverted to 25 MHz and acquired by a digitizer. After repeating the protocol, acquiring the signal, and classifying the states, we calculate the assignment fidelity, which was used to evaluate the efficacy of different classification methods. We performed the experiments with two different samples (see Table I). The first run (sample A) of experiments was used as a test-bed to compare the quality of various ML methods. The second set of experiments (sample B) demonstrated on-the-fly data processing using a GPU and the methods' stability against phase drifts.

Within our data processing workflow, each acquired waveform undergoes digital downconversion (DDC) by multiplying the acquired signal with $\cos(\omega_{\text{DDC}}t)$ ($\sin[\omega_{\text{DDC}}t]$) where $\omega_{\text{DDC}}/2\pi = 25$ MHz, to obtain the in-phase quadrature $I(t)$ (out-of-phase quadrature $Q(t)$). A Finite Impulse Response (FIR) Filter with a window of 40 samples (20 ns) and a cutoff frequency of 20 MHz is applied to the signal to eliminate the signal image at 50 MHz along with 25 MHz noise (originally DC offset). After obtaining $I(t)$ and $Q(t)$, the signal undergoes further post-processing. This may include time integration, channel correlation, or even being fed through trained PyTorch neural networks.

For the on-the-fly experiments with sample B, we acquired 512 time points per measurement, recorded to the buffer of a 500 MSa/s digitizer Spectrum M4i. After populating the buffer with 2048 time traces, we transferred data to PC memory (RAM) and then to the GPU memory for batch processing. While the data are being

TABLE I. Device parameters. Here, ω_{cav} is the frequency of the readout resonator, ω_{ge} (ω_{ef}) is the frequency of transition between the ground (first excited) state and the first (second) excited state, χ_{ge} and χ_{ef} are the state-dependent dispersive shifts of the resonator frequency, κ is the decay rate of the resonator, and T_1 (T_2) is the relaxation (dephasing) rates of the transmons.

Parameter	Transmon A	Transmon B
$\omega_{\text{cav}}/2\pi$	7.08 GHz	7.63 GHz
$\omega_{ge}/2\pi$	6.27 GHz	5.49 GHz
$\omega_{ef}/2\pi$	5.95 GHz	5.16 GHz
$2\chi_{ge}/2\pi$	8.00 MHz	8.50 MHz
$2\chi_{ef}/2\pi$	5.35 MHz	15.57 MHz
$\kappa/2\pi$	1.31 MHz	1.56 MHz
T_1	11.75 μs	4.07 μs
T_2	3.17 μs	4.29 μs

processed, the digitizer buffer is populated with new acquired waveforms. Due to the large number of cores in the GPU, the data can be processed in parallel, which allowed us to perform real time data acquisition on-the-fly. Although the results of this paper were obtained with a repetition time of 40 μs , our GPU data processing can be run without overhead as fast as 3.2 μs per repetition, obtaining 19×10^6 traces (19×10^{12} samples) in 1 min.

In order to determine the *baseline* readout fidelity, we first employed the conventional classification. Following this method, the heterodyne measurement signal was integrated in time, giving us one complex number I and Q . By repeating the measurements, we can populate histograms for every prepared state on the I-Q plane, as shown in Fig. 1. A measurement response can be classified by selecting the state whose mean response is closest to the mean state response on the I-Q plane. The assignment fidelity can be evaluated as $\mathcal{F}_a = (1/N) \sum_{i=1}^N \mathbb{P}(i|i)$, where $\mathbb{P}(i|i)$ is the probability of obtaining outcome “ i ” given that the system was prepared in j^{th} state. Here, we use $N = 2(3)$ and states $|g\rangle$, $|e\rangle$ ($|g\rangle$, $|e\rangle$, $|f\rangle$) to calculate qubit (qutrit) assignment fidelity.

Alternatively, one can apply a *matched filter* to the heterodyne measurement signal prior to integration in the conventional method. Matched filters are calibrated by taking the means of all acquired signals corresponding to each basis state. An incoming signal is convolved with these filters. The filter, which returns maximum average amplitude, determines the classified basis state.

To identify the best ML algorithm, we first collected data from sample A. The transmon was prepared in the three basis states followed by the 2 μs measurement pulse, resulting in 50 time samples for each trace at 100 MHz. In total, we collected 16 384 traces corresponding to each state for analysis. 90% of these data were used for training, and the rest was used to test and obtain the assignment fidelity.

We evaluated the assignment fidelity of several machine learning methods: the support vector machine, a random forest classifier, and a k-nearest-neighbors algorithm. After this, we deployed neural network architectures, including a single hidden layer (“vanilla”) feedforward neural network, several versions of convolutional neural networks (CNN), and long short term memory networks (LSTM). The assignment fidelities obtained using each of these algorithms are shown in Table II.

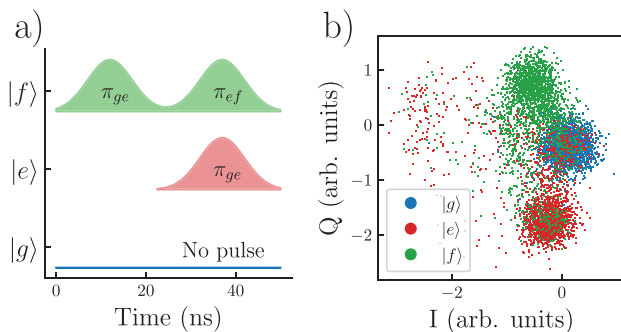


FIG. 1. (a) Transmon control pulses used for preparing three basis states of the transmon. (b) Histogram of integrated cavity responses $I + iQ$. The blue, red, and green points correspond to the transmon being prepared in $|g\rangle$, $|e\rangle$, and $|f\rangle$ states, respectively. The relaxation of the f -state to the excited state and excited state to ground state is visible. Some of the points that are decaying from the f -state to the excited state will be classified as ground state due to their proximity to the ground state cluster.

TABLE II. Assignment fidelities for different machine learning models evaluated on an identical test data set generated with sample A. For all methods, there are significant improvements over the conventional method (time integration of readout signal and setting classification thresholds). CNN is the best model for classifying between three levels. Note that the same data were used to extract both the qubit and qutrit fidelities. Since the readout parameters were optimized for the qutrit case, the CNN model returns higher value for the three state fidelity.

Model	2 state fidelity	3 state fidelity
Conventional	0.841	0.711
Matched filter	0.913	0.747
K nearest neighbors	0.902	0.845
Support vector machine	0.917	0.851
Random forest classifier	0.917	0.874
Vanilla neural network	0.912	0.925
LSTM	0.909	0.904
CNN	0.919	0.928

The k-nearest neighbor algorithm bears the most similarity to the matched filter method. The LSTM algorithm is popular in language processing and was chosen because they are designed to deal with sequences of data and can, therefore, process long time correlations between data points. CNN is most popularly used for pattern recognition and image classification. It is a neural network where the hidden layer is a convolution of the input with a kernel (or filter). We feed the time-domain signal data to the CNN with the I(t) and Q(t) traces as two channel inputs, analogous to the red, blue, and green channels of a color image.

After selecting CNN as the method with the highest assignment fidelity, we apply this model to state classification on-the-fly. The network consists of the following layers:

- (1) 1D convolution: a convolution layer with 2 input channels (corresponding to I and Q), 16 output channels, and a kernel size of 128. The large kernel size filters out higher frequency noise effectively. The initial kernel weights (or filter coefficients) are manually set using the He initialization function,²³ with experimentation demonstrating that CNN performance is sensitive to this weight initialization.
- (2) ReLU activation: maps $f(x) = \max(0, x)$ in order to expedite the learning process.²⁴
- (3) 1D convolution: a convolution layer with 16 input channels (corresponding to the output of the previous convolution layer), 32 output channels, and a kernel size of 5. This expands the previous 16 features to 32, by taking various linear combinations of the prior layer outputs.
- (4) ReLU activation
- (5) Max pooling: the maximums of every three neighboring output values of the previous layer are evaluated, to reduce the size of the data representation and, therefore, computational time in the remainder of the network.
- (6) Flattening: reshapes data to a one-dimensional array.
- (7) Dropout: 50% of the data points are randomly selected to be set to zero (called “neuron deactivation”). This step was initially introduced to prevent the model from being *overtrained*. Overtraining (or overfitting) is known behavior when ML

methods learn “too much” from the training data, showing an artificially high accuracy of classification while training, but failing to maintain performance when tested on new data. This may occur if there are certain features of the signal present in the training dataset due to noise or other conditions relevant to the data-taking process.

- (8) Linear: applies a linear transformation to the incoming data: $y = Ax + b$, where weights (denoted as a matrix A) and biases (denoted as a vector b) are optimized. Here, the size of the output y is half of the size of the input x .
- (9) ReLU activation
- (10) Linear: the size of the output data is designed to be two for the qubit and three for the qutrit classification, corresponding to the possible preparation states.

The Adam optimizer²⁵ and a mean squared error (MSE) loss function were used for optimization. The output was classified using a softmax function. For each training cycle, we record 2048 traces of each state, and pass these through the model. The loss was calculated and gradient descent was undertaken at a learning rate of 10^{-3} . This cycle takes ~ 3 s.

We trained the model on new data at every training cycle, acquired in real-time from the sample. This provides an intrinsic protection from overfitting; since there is a new dataset each time the loss is calculated, the model cannot learn on any spurious signal features localized to a single dataset. The low learning rate assists in helping the model to learn patterns that are common to data across training cycles, thereby increasing model stability. After each update of the model weights, a further 2048 acquisitions were made to test the model. The loss and assignment fidelity on test data were stored for monitoring.

To investigate robustness of the CNN classification model against system parameter drifts, we performed continuous measurements over 24 h and monitored the fidelity values as shown in Fig. 2. First, we evaluated the fidelities obtained from the integrated responses according to the conventional method (“Baseline”). Second, we obtained the fidelity using the mean responses re-calibrated every 2048 repeated measurements (“Cal-Baseline”). Finally, we plot the fidelity obtained by on-the-fly processing with the CNN model.

The model is trained for initial 100 training cycles. The fidelity is then repeatedly tested for ~ 1 – 3.5 h, before training the model again for another 20 training cycles. Figure 3 shows the assignment fidelity during training and retraining. This retraining process is akin to recalibration in the Cal-Baseline measurement. This model performs better than both the Baseline and Cal-Baseline measurements, despite the frequency of recalibration being significantly lower than its Cal-Baseline counterpart. Without retraining, the model performs better than the Baseline throughout the 24 h, but loses its enhanced fidelity after a few hours.

To enforce robustness to input parameter variations, we can train CNN with the dataset containing samples, which are broadly distributed across the domain of that input parameter. This effectively removes the parameter from the possible set of features upon which the model can learn. An example of such a parameter variation can be a global phase drift generated by insufficient instrument synchronization.

To endow the network with global phase robustness, a secondary dataset was created by obtaining 256 traces for each of the three basis states at each of 500 different global phases. Global phase was applied

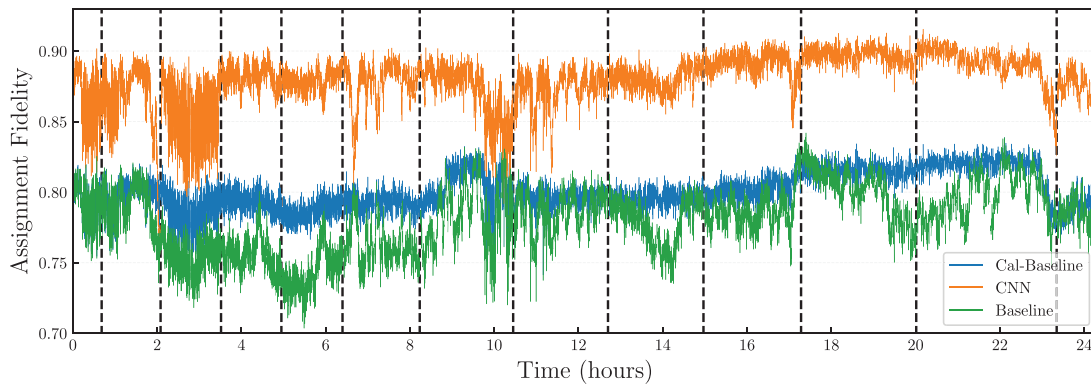


FIG. 2. Fidelity of different classification methods. Each point represents a fidelity evaluation using 2048 traces for each of the three states gathered on the fly, while each plot represents a different training regime. The black dashed lines indicate when the model was retrained. The CNN model consistently does better than the Cal-Baseline measurement, even though it is only trained at discrete intervals.

by imposing an arbitrary wait time at the beginning of each experiment repetition. The readout signals obtained within each experiment are downconverted to 25 MHz using a local oscillator, which is phase-locked to the generator producing the readout signal itself. This phase-locking ensures that if measurements occur at integer multiples of 40 ns within the pulse sequence, phase coherence is conserved. By generating data with uniformly distributed initial wait times between 0 and 40 ns, a predictable global phase was no longer a feature of the training dataset, forcing the model to learn a phase-robust mapping from readout trajectory to qutrit state. We fed these data into a CNN network in the same manner as the standard dataset.

The design of the neural network remains identical for this experiment, aside from using a kernel size of 10 in the first convolutional layer. Assignment fidelities are evaluated at 500 intervals with phase shifts ranging from 0 to 2π over the course of 9 min in Fig. 4, using both the baseline mean integration and CNN processing methods (trained/calibrated using data gathered immediately prior). The CNN

model classifies with fidelities comparable to the baseline state classification fidelities without phase drift, but any non-trivial phase drift will degrade the conventional approach markedly while the CNN maintains accuracy without any re-calibration required.

In summary, we investigated a set of machine learning methods for classification of a transmon with the dispersive measurement. The method classifies the states when two and three levels of the transmon are considered and can be directly extended to more transmon levels. Some modifications accounting for crosstalk could be necessary for applying this method to multi-qubit systems. Convolutional neural network methods demonstrated the highest performance consistent with the results from Ref. 20. CNN methods could also be trained against experimental imperfections, such as the local oscillator phase drift. Improved assignment fidelity and ability to directly train the model against system imperfections are the key advantages of neural networks. The open-source, GPU-friendly, and easily implementable nature of PyTorch makes these neural networks an attractive tool for state classification.

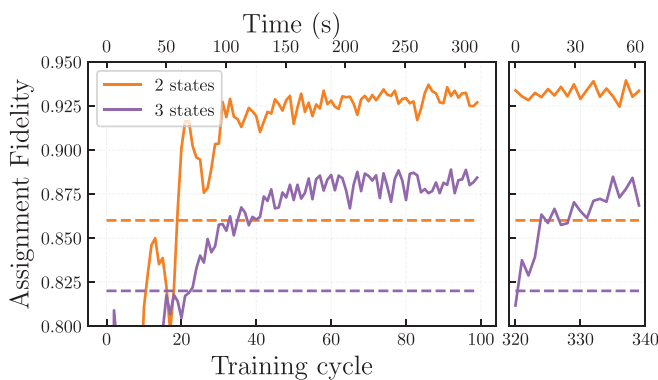


FIG. 3. Training over 100 training cycles and re-training over 20 cycles. The left plot shows the CNN learning to differentiate between the three (or two) states. The model provides better fidelity than the baseline fidelity within 30 training cycles over approximately one minute. The plot on the right shows the model being re-trained after running the experiment for a few hours. Retraining takes only 20 training cycles.

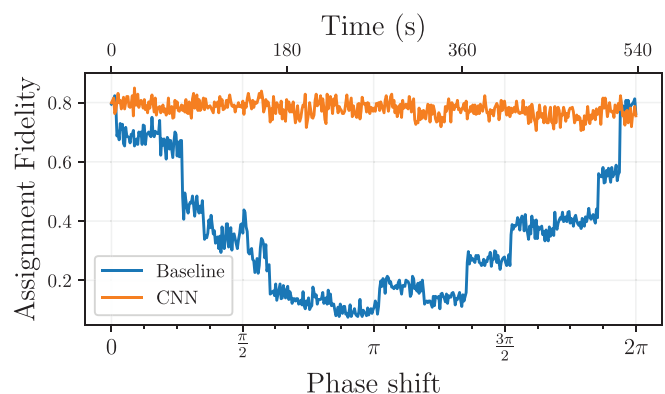


FIG. 4. Assignment fidelities of the baseline mean calibration method and a phase-robust CNN model classifying data with an induced phase drift. Each point represents a fidelity evaluation using 2048 traces for each of the three states. Neither method undergoes any retraining or recalibration over the course of the experiment.

See the [supplementary material](#) for the experiment schematic and an example of readout trajectories with and without decay.

The authors thank Andreas Wallraff, Markus Oppliger, Anton Potočník, and Mintu Mondal for fabricating JPA used in the measurements. The authors were supported by the Australian Research Council Centre of Excellence for Engineered Quantum Systems (EQUS, No. CE170100009) and by Lockheed Martin Corporation via Research Contract No. S19 004.

DATA AVAILABILITY

The data that support the findings of this study and the code used for generating and training the machine learning algorithm are openly available in UQ eSpace at <https://doi.org/10.48610/9d3c2eb>, Ref. 26.

REFERENCES

- ¹E. Zahedinejad, J. Ghosh, and B. C. Sanders, “Designing high-fidelity single-shot three-qubit gates: A machine-learning approach,” *Phys. Rev. Appl.* **6**, 054005 (2016).
- ²M. August and X. Ni, “Using recurrent neural networks to optimize dynamical decoupling for quantum memory,” *Phys. Rev. A* **95**, 012335 (2017).
- ³Y. Ding, Y. Ban, J. D. Martín-Guerrero, E. Solano, J. Casanova, and X. Chen, “Breaking adiabatic quantum control with deep learning,” *Phys. Rev. A* **103**, L040401 (2021).
- ⁴Y. Baum, M. Amico, S. Howell, M. Hush, M. Liuzzi, P. Mundada, T. Merkh, A. R. R. Carvalho, and M. J. Biercuk, “Experimental deep reinforcement learning for error-robust gateset design on a superconducting quantum computer,” *arXiv:2105.01079* (2021).
- ⁵B. S. Rem, N. Käming, M. Tarnowski, L. Asteria, N. Fläschner, C. Becker, K. Sengstock, and C. Weitenberg, “Identifying quantum phase transitions using artificial neural networks on experimental data,” *Nat. Phys.* **15**, 917 (2019).
- ⁶X.-Y. Dong, F. Pollmann, and X.-F. Zhang, “Machine learning of quantum phase transitions,” *Phys. Rev. B* **99**, 121104 (2019).
- ⁷A. M. Palmieri, E. Kovlakov, F. Bianchi, D. Yudin, S. Straupe, J. D. Biamonte, and S. Kulik, “Experimental neural network enhanced quantum tomography,” *npj Quantum Inf.* **6**, 20 (2020).
- ⁸J. P. Zwolak, T. McJunkin, S. S. Kalantre, J. Dodson, E. MacQuarrie, D. Savage, M. Lagally, S. Coppersmith, M. A. Eriksson, and J. M. Taylor, “Autotuning of double-dot devices *in situ* with machine learning,” *Phys. Rev. Appl.* **13**, 034075 (2020).
- ⁹R. Durrer, B. Kratochwil, J. Koski, A. Landig, C. Reichl, W. Wegscheider, T. Ihn, and E. Greplova, “Automated tuning of double quantum dots into specific charge states using neural networks,” *Phys. Rev. Appl.* **13**, 054019 (2020).
- ¹⁰L. Hu, S.-H. Wu, W. Cai, Y. Ma, X. Mu, Y. Xu, H. Wang, Y. Song, D.-L. Deng, C.-L. Zou, and L. Sun, “Quantum generative adversarial learning in a superconducting quantum circuit,” *Sci. Adv.* **5**, eaav2761 (2019).
- ¹¹V. Cimini, M. Barbieri, N. Treps, M. Walschaers, and V. Parigi, “Neural networks for detecting multimode Wigner negativity,” *Phys. Rev. Lett.* **125**, 160504 (2020).
- ¹²M. P. V. Stenberg, O. Köhn, and F. K. Wilhelm, “Characterization of decohering quantum systems: Machine learning approach,” *Phys. Rev. A* **93**, 012122 (2016).
- ¹³E. Flurin, L. S. Martin, S. Hacothen-Gourgy, and I. Siddiqi, “Using a recurrent neural network to reconstruct quantum dynamics of a superconducting qubit from physical observations,” *Phys. Rev. X* **10**, 011006 (2020).
- ¹⁴A. A. Gentile, B. Flynn, S. Knauer, N. Wiebe, S. Paesani, C. E. Granade, J. G. Rarity, R. Santagati, and A. Laing, “Learning models of quantum systems from experiments,” *Nat. Phys.* **17**, 837 (2021).
- ¹⁵A. B. Mei, I. Milosavljevic, A. L. Simpson, V. A. Smetanka, C. P. Feeney, S. M. Seguin, S. D. Ha, W. Ha, and M. D. Reed, “Optimization of quantum-dot qubit fabrication via machine learning,” *Appl. Phys. Lett.* **118**, 204001 (2021).
- ¹⁶E. Magesan, A. D. Gambetta, J. M. Córcoles, and J. M. Chow, “Machine learning for discriminating quantum measurement trajectories and improving readout,” *Phys. Rev. Lett.* **114**, 200501 (2015).
- ¹⁷C. Dickel, J. J. Wesdorp, N. K. Langford, S. Peiter, R. Sagastizabal, A. Bruno, B. Criger, F. Motzoi, and L. DiCarlo, “Chip-to-chip entanglement of transmon qubits using engineered measurement fields,” *Phys. Rev. B* **97**, 064508 (2018).
- ¹⁸S. Kono, K. Koshino, Y. Tabuchi, A. Noguchi, and Y. Nakamura, “Quantum non-demolition detection of an itinerant microwave photon,” *Nat. Phys.* **14**, 546 (2018).
- ¹⁹L. A. Martinez, Y. J. Rosen, and J. L. DuBois, “Improving qubit readout with hidden markov models,” *Phys. Rev. A* **102**, 062426 (2020).
- ²⁰B. Lienhard, A. Vepsäläinen, L. C. G. Govia, C. R. Hoffer, J. Y. Qiu, D. Ristè, M. Ware, D. Kim, R. Winik, A. Melville, B. Niedzielski, J. Yoder, G. J. Ribeill, T. A. Ohki, H. K. Krovi, T. P. Orlando, S. Gustavsson, and W. D. Oliver, “Deep neural network discrimination of multiplexed superconducting qubit states,” *arXiv:2102.12481* (2021).
- ²¹A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019), Vol. 32, pp. 8024–8035.
- ²²C. Eichler, Y. Salathe, J. Mlynek, S. Schmidt, and A. Wallraff, “Quantum-limited amplification and entanglement in coupled nonlinear resonators,” *Phys. Rev. Lett.* **113**, 110502 (2014).
- ²³K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” *arXiv:1502.01852* (2015).
- ²⁴V. Nair and G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines,” in *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML’10* (Omnipress, Madison, Winsconsin, 2010), p. 807–814.
- ²⁵D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv:1412.6980* (2017).
- ²⁶R. Navarathna and T. Jones (2021). “APL paper files,” The University of Queensland, <https://doi.org/10.48610/9d3c2eb>