IOP Publishing Mach. Learn.: Sci. Technol. 3 (2022) 04LT01 https://doi.org/10.1088/2632-2153/ac9036

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
23 June 2022

REVISED
17 August 2022

ACCEPTED FOR PUBLICATION
7 September 2022

PUBLISHED
29 September 2022

LETTER

Data-centric machine learning in quantum information science

Sanjaya Lohani[1,*], Joseph M Lukens[2], Ryan T Glasser[3], Thomas A Searles[1,*] and Brian T Kirby[3,4,*]

[1] Department of Electrical & Computer Engineering, University of Illinois Chicago, Chicago, IL, 60607, United States of America
[2] Quantum Information Science Section, Oak Ridge National Laboratory, Oak Ridge, TN, 37831, United States of America
[3] Tulane University, New Orleans, LA 70118, United States of America
[4] DEVCOM Army Research Laboratory, Adelphi, MD, 20783, United States of America
[*] Authors to whom any correspondence should be addressed.

E-mail: slohan3@uic.edu, tsearles@uic.edu and brian.t.kirby4.civ@army.mil

Keywords: machine learning, quantum noise and quantum operations, quantum tomography

Abstract

We propose a series of data-centric heuristics for improving the performance of machine learning systems when applied to problems in quantum information science. In particular, we consider how systematic engineering of training sets can significantly enhance the accuracy of pre-trained neural networks used for quantum state reconstruction without altering the underlying architecture. We find that it is not always optimal to engineer training sets to exactly match the expected distribution of a target scenario, and instead, performance can be further improved by biasing the training set to be slightly more mixed than the target. This is due to the heterogeneity in the number of free variables required to describe states of different purity, and as a result, overall accuracy of the network improves when training sets of a fixed size focus on states with the least constrained free variables. For further clarity, we also include a 'toy model' demonstration of how spurious correlations can inadvertently enter synthetic data sets used for training, how the performance of systems trained with these correlations can degrade dramatically, and how the inclusion of even relatively few counterexamples can effectively remedy such problems.

1. Introduction

Machine learning (ML) is quickly becoming a standard tool for approaching and analyzing problems in quantum information science (QIS). Recent applications include state classification [1–4], quantum control [5–9], sensing [10–12], parameter estimation for deployed systems [13, 14], turbulence correction [15–19], and state reconstruction [20], among many others [21–25]. Although the motivations for adopting ML in the QIS context vary, they are often related to the ability of ML systems to perform optimization tasks in highly constrained or non-convex situations and the potential improvements in resource scaling compared to standard techniques.

Efforts to improve the performance of ML systems are generally classified as either model-centric or data-centric. Model-centric techniques focus on altering the underlying architecture of an ML system. Examples include increasing the number of hidden layers in a deep neural network, tailoring the structure of a model, modifying the loss function [26] or tweaking the reward function in reinforcement learning. Alternatively, data-centric methods—which leave the system's architecture unchanged—endeavor to improve system performance by using enhanced data [27–29] obtained through, e.g. data set augmentation [30–32], data set distillation [33, 34], label analysis and accuracy improvements [35–38], data set validation [39, 40], domain randomization [41], and combinations of these [42–44].

Given the relative maturity and availability of ML models and systems, and how similarly many state-of-the-art models perform [45], it has been suggested that data-centric techniques represent an undervalued opportunity to boost system performance [46]. This recommendation is especially relevant to domain scientists deploying ML in their particular field of research where model-centric methods may be

outside of their expertise. In other words, applying domain-specific knowledge to improve the quality and accuracy of a data set is likely the most efficient and direct route to performance improvements for those not specializing in ML specifically.

Data-centric ML techniques have found wide applicability in a variety of domains including legal [43], natural language processing [37], image classification [44], and medical prognosis [39]. In the context of QIS, data-centric approaches, namely engineered data sets, have been used to demonstrate prediction advantage in quantum machine learning [47] and improve the accuracy of state reconstruction systems [48–50]. In the latter case, expected statistical and experimental noise were included in training sets of a convolutional neural network (CNN), resulting in overall performance improvements. Beyond the inclusion of artificial errors, it was also recently shown that even very general pieces of prior information in the construction of training sets—such as the expected mean purity—can improve the performance of ML-based state reconstruction systems [51].

This paper develops data-centric heuristics specifically targeting classical ML applications in QIS. As an initial foray, we explore a representative example where spurious correlations—by which we mean correlations present in a dataset that are not representative of the actual underlying system—between purity and entanglement in a training set lead to stark misclassification of separable pure states as entangled. This example provides an intuitive motivation for our first heuristic: engineering training sets to include even a few counterexamples is sufficient to remedy errors due to spurious correlations. We then refine our focus to ML-based quantum state reconstruction of states produced by a cloud-accessed seven-qubit noisy intermediate scale quantum (NISQ) system, demonstrating clear improvements in reconstruction fidelity through the use of engineered training sets. Finally, as two applications of our data-centric heuristics, we demonstrate enhanced reconstruction performance in the high-statistical-noise regime by using intentionally noisy training sets [48–50] and in the case of highly heterogeneous test states with a wide range of purity values. For the latter, we obtain a surprising result: it is not always optimal to train on the exact distribution expected, but rather on a training set biased towards the more mixed end of the distribution—an effect which we explain through the heterogeneity of the free variables in states of differing purity. Overall, our findings suggest exciting opportunities for data-centric approaches in QIS more broadly: heuristics designed to leverage the unique features of quantum state distributions should enable improved training sets—and hence, improved performance—in a variety of QIS applications. The code and data for this study are openly available at [52, 53].

2. Overview of approach

A schematic summarizing our approach for data-centric ML-based quantum state reconstruction is shown in figure 1. The inputs to our system are the ranges of the desired purity and concurrence distributions, but generally any quantifiable property of a distribution can be substituted. In our particular case, we estimate the approximate range of the distribution of states generated from the IBM Quantum Experience (IBMQ) *ibmq_jakarta* processor as in [50], but many other techniques can be used without requiring full state reconstruction [54, 55]. The input purity and concurrence ranges inform the selection of an initial distribution of random quantum states, which is passed through a simultaneous bandpass filter in both purity and concurrence, resulting in our final engineered training set. We then use these engineered training sets to optimize the trainable parameters of an ML model (appendix A), which is shown by a dotted brown arrow 'Training.' Once trained, the resulting neural network infers unknown quantum states probed by measurements on NISQ hardware, as indicated by an 'Inference' arrow. From a data-centric perspective, the core of this vision lies in the synthetic training set; as we will examine through multiple cases below, the design of this set can have a significant impact on performance of the reconstruction procedure.

3. Spurious correlations and lack of variation

As an illustrative example to introduce our data-centric approach to quantum state tomography (QST), we first examine an ML system trained on a set of density matrices containing perfect correlations (one-to-one relationship between two parameters) between purity and entanglement. The specific relationship between purity and entanglement is explained in appendix B. We then employ the trained network as a separability-entanglement classifier on generic states that need not possess the learned correlation. In this sense, how spurious correlations (a relationship between two variables that appears causal but is not) impact reconstruction fidelity and separability-entanglement classification is related to the ML concepts of generalizability [56, 57] and out-of-distribution prediction [58, 59], which consider how well a system will perform on data not included in its training set. We will show that our network indeed learns the correlations between purity and entanglement present in the training set, limiting generalizability and

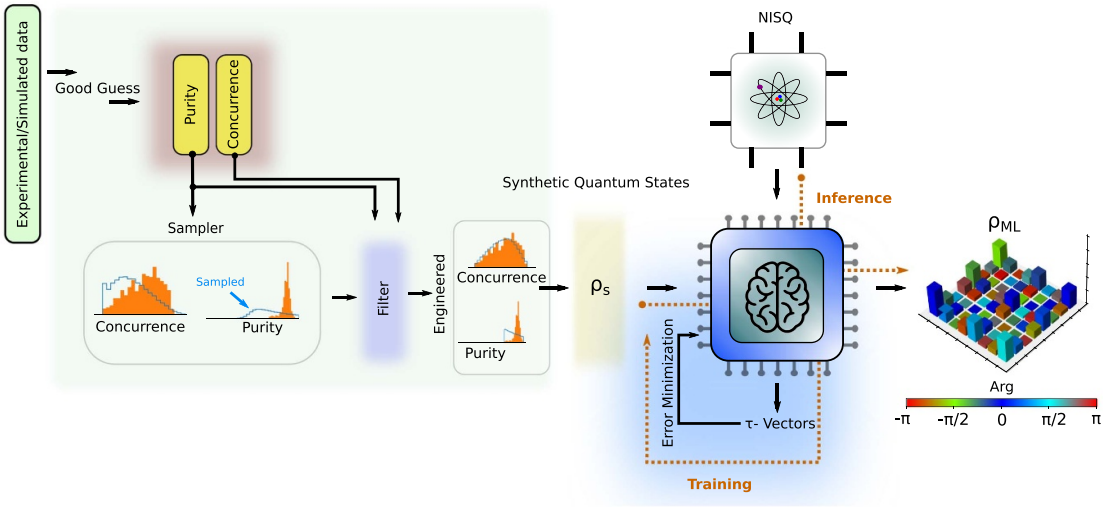


Figure 1. A schematic of a data-centric ML approach in QIS. Inputs are coarse properties of the distributions expected in our test system, e.g. the limits (maximum and minimum) of the purity and concurrence. To obtain the training set, we sample from a suitably chosen distribution of random quantum states, and then apply a bandpass filter to eliminate any samples lying outside of the specified extrema. These engineered training sets are used in optimizing the trainable parameters of an ML model, as indicated by a dotted brown arrow labeled 'Training.' Finally, the resulting neural network is enlisted to infer unknown quantum states probed by measurements on NISQ-era hardware ('Inference' arrow). An example reconstructed three-qubit quantum state is shown at the right, where shading denotes the phase.

resulting in a high error rate when classifying pure states as separable or entangled. Yet we will then demonstrate that including only a modest number of counterexamples in the training set can significantly mitigate this issue—a paradigmatic 'data-centric' improvement.

For training, we consider two-qubit density matrices drawn from the set known as maximally entangled mixed states (MEMS) ρ'_{MEMS} (appendices B and C). These states possess a one-to-one and monotonic relationship between purity P (a measure of how mixed a quantum state is) and concurrence C (a measure of entanglement) such that high C is perfectly correlated with high purity; thus, our network is not exposed any separable states of high purity in training. The steps for evaluating the purity and concurrence are included in appendix F. To understand the effects of these correlations, we perform QST on randomly generated separable states ρ_s (with purity $P > \frac{1}{3}$ to align with the support of MEMS) and classify them as separable or entangled based on the Peres–Horodecki positive partial transpose criterion [60, 61]. Our network has significant error in this scenario and fails to correctly classify approximately 50% of the states from ρ_s, as shown by the first data point in figure 2(a). This result reveals the dramatic impact of spurious correlations on the performance of ML-based QST even for systems where the number of free variables—sixteen in a two-qubit system—is dwarfed by the number of examples in the training set.

We next investigate how these errors can be mitigated through the inclusion of a modest number of counterexamples in the training set. For a total training set size $N_{\text{train}} = 30000$, we include N_s separable mixed states ρ_s (the rest are drawn from ρ'_{MEMS}). As the percentage of separable states in the training set increases, the network state classification accuracy on separable states increases rapidly, reaching 99.25% with only $N_s/N_{\text{train}} = 0.058$. The reconstruction fidelities likewise increase for separable states, as shown by the dotted red line in figure 2(b). Importantly, the improved performance for separable states does not reduce accuracy for entangled state examples; the magenta lines in figures 2(a) and (b) show the entanglement classification accuracy and fidelity of 5000 MEMS-drawn entangled states as well, revealing no observable drop as N_s/N_{train} increases in this scale. Hence, the inclusion of a small number of counterexamples in the training set significantly improves classification performance on out-of-correlation states without reducing overall network performance.

To further illustrate the overall impact of spurious correlation and the effect of the mitigation strategy, we also reconstruct states from across the purity-concurrence plane, considering specifically 5000 samples from the four-dimensional Mai–Alquier (MA) distribution with concentration parameter $\alpha = 0.1$ (appendix D and [51, 62, 63]), chosen for illustrative purpose. Figure 2(c) shows the results for a network trained only on the MEMS class ($N_s/N_{\text{train}} = 0$) and figure 2(d) for a network trained on counterexamples as well ($N_s/N_{\text{train}} = 0.058$). The magenta and blue dots, respectively, represent the MEMS and added separable states (training sets), while the inner dots indicate the MA-distributed test states, plotted at their ground truth purity-concurrence positions, but shaded according to the reconstruction fidelity.

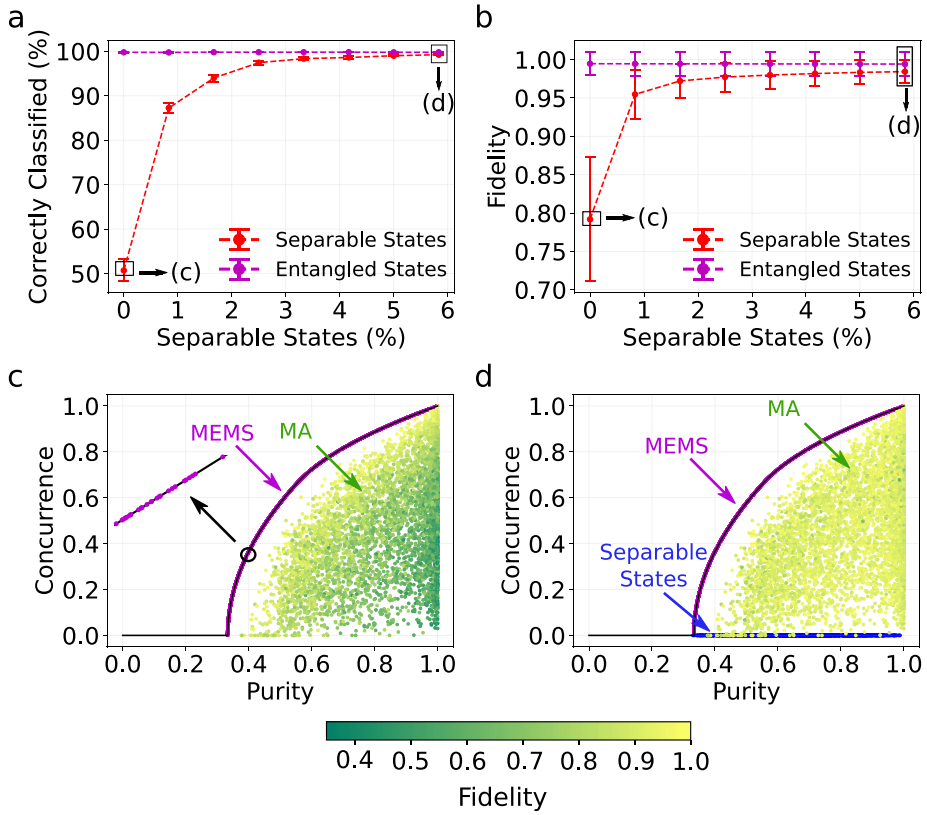


Figure 2. Reducing spurious correlations. (a) Entanglement-separability classification accuracy and (b) network reconstruction fidelity versus the percentage of separable states added to a training set containing entangled states. Reconstruction fidelity for test states from the Mai–Alquier (MA) distribution for a network (c) trained only on maximally entangled mixed states (MEMS) and (d) trained with MEMS plus a small fraction of separable states. MEMS (separable) training states are shown by magenta (blue) dots. A small portion of the MEMS line is magnified and shown in the top-left inset. The MA distribution test sets are shown by the inner dots, with shading indicating their reconstruction fidelity. The rectangular boxes in (a) and (b) represent the pre-trained networks that are used for results in (c) and (d). The error bars represent one standard deviation from the mean.

As expected, in figure 2(c) separable states of high purity (residing toward the lower right hand corner) are reconstructed with low fidelity, since these states possess the most extreme deviation from the correlation found in MEMS. In contrast, reconstruction fidelities increase significantly for the network trained with MEMS and a few examples of separable states (figure 2(d)). Interestingly, although the improvement is most pronounced for separable pure states, the modified training set increases reconstruction fidelity across the entire purity-concurrence plane.

Admittedly, QIS researchers are unlikely to expect generalizability from MEMS training sets, and it is perhaps unsurprising that neural networks trained on them would perform so poorly on other states. Nevertheless, their strong correlations acutely highlight the broad issue of spurious correlations—which in many situations may prove much more difficult to detect—as well as indicate a simple mitigation strategy based on tailored training data. In the following, we apply these general ideas to situations of more practical interest in QIS, exploring a variety of density matrix distributions that all offer full Hilbert space support, and compare their performance in ML-based QST. In these more nuanced cases, we again will find noticeable improvements with engineered training sets, for multiple experimentally relevant contexts.

4. Data-centric state reconstruction with NISQ hardware

Building on the intuitions highlighted in the simple example above, we now tackle the much more complex problem of ML-based QST of NISQ hardware, comparing the performance improvements obtainable with both data-centric and model-centric techniques. Our data-centric methods consist of training our ML-QST system using sets drawn from various standard quantum state distributions (appendix D) and from engineered sets (appendix F). Our model-centric methods consist of increasing the trainable parameters in the network. Ultimately we will find that data- and model-centric approaches work in a complementary fashion and that our greatest data-centric improvement is found using our engineered distribution approach. The test set used for all comparisons in this section are based on two-qubit NISQ-sampled density

matrices with simulated measurement outcomes, a combination chosen to reflect currently available systems while also ensuring knowledge of the ground truth state for accurate benchmarking (appendix E).

A summary of the distributions used to generate training sets for our ML-QST system is included in table 1. Of particular interest are the 'P-distributions' and 'C-distributions' columns which show the NISQ sample distributions in solid green with the training set distributions overlaid. The Hilbert–Schmidt (HS), Bures, and Hilbert–Schmidt–Haar (HS-Haar) distributions have no input parameters and hence the mean purity (P_{mean}), range of purity, and range of concurrence are application-agnostic. Alternatively, the MA, Życzkowski (Z), and engineered distributions all include degrees of freedom that allow for the incorporation of prior information.

The results of the data-centric and model-centric approaches are pictured simultaneously in figure 3(a), with the limiting fidelity (at the largest number of trainable parameters) for each case included in table 1. Each curve corresponds to a different data-centric method, meaning the neural network was trained with states drawn from a different distribution. Increasing along the x-axis corresponds to a 'model-centric' performance improvement, where the training set is fixed but the number of trainable parameters in the network is increased. Each point corresponds to the average reconstruction fidelity for our ML-QST system when reconstructing 500 NISQ-sampled states, using a network trained on 30000 randomly sampled states from the corresponding distribution. All measurements used to train the networks described in figure 3(a) are simulated in the ideal scenario, i.e. the limit of infinite shots, where one 'shot' corresponds to one measurement of every Pauli combination (nine in total for two qubits).

Figure 3(a) shows that the average performance of our system for any training set is improved, at least at first, using model-centric techniques. These model-centric improvements appear to impact each network instantiation in approximately the same way (with a few minor crossovers occurring). Similarly, for a given network size (x-axis position) we find the average reconstruction fidelity can be improved with data-centric methods. In other words, data-centric and model-centric approaches are complementary paths to performance improvement.

Ultimately, figure 3(a) indicates that the engineered training set attains the highest average reconstruction fidelity (bold in table 1). Importantly, only the minimum and maximum values of the NISQ purity and concurrence were used in producing the training sets—not any detailed features of the distribution's shape. The HS–Haar distribution, which is simply a convex sum of HS- and Haar-distributed states, performs close to that of the engineered distribution in the limit of the maximum model-centric improvements. This is especially surprising when considering the table 1 which shows how dramatically the engineered and HS–Haar distributions differ in both purity and concurrence. The impact of these differences evidently shrinks as the number of trainable parameters grows, removing the initially wide separation (at small x-values) in the performance of the neural networks trained on the engineered distribution and HS–Haar.

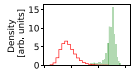
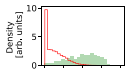
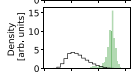
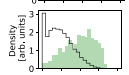
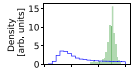
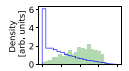
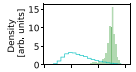
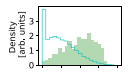
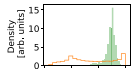
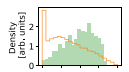
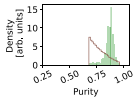
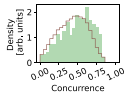
After the engineered and HS-Haar training sets the next highest performing sets are those which can be biased based on mean purity, the MA and Z distributions. Our selection of MA parameters were heuristically chosen based on the targeted minimum purity P_{min}; to better understand the optimality of this selection we also include the reconstruction fidelities with various MA concentration parameters in the appendix F. It is unsurprising that all of these distributions ultimately outperform the Bures and HS training sets, both because they have no parameters with which to incorporate prior information and because they skew significantly more mixed than the average state generated by the NISQ system. Finally, for context, we include in table 1 a comparison with the standard approach of performing quantum state reconstruction using maximum likelihood estimation (MLE) [64]. Ultimately our techniques reach similar reconstruction fidelity as MLE; indeed, the mean reconstruction fidelity for the engineered training set with 5.75×10^6 trainable parameters is equal to the result of MLE (0.986). Crucially, however, aside from the upfront training period, all ML results require significantly fewer computational resources—and time—than MLE to complete [50].

To elucidate how the engineered distribution—which again only takes into account minimum and maximum information—compares to the distribution actually generated by the NISQ device, in figure 3(b) we plot the concurrence of the states sampled as a function of purity from the IBMQ machine (green) and those from the engineered distribution (brown). We see that the engineered set covers the NISQ-sampled set convincingly, albeit weighted more heavily toward mixed states. While this might initially appear to be an inefficiency, we will find below that training set bias toward lower purity actually contributes to the high performance of the engineered set.

5. Applications of data-centric engineering

In this section, we present two additional data-centric techniques for improving the reconstruction fidelity of our system. The first subsection considers situations where statistical noise is present in measurement results and demonstrates that synthetic statistical noise in training sets can significantly improve average

Table 1. Purity and concurrence distributions of explored training sets. P_{mean}, P-distributions, and C-distributions represent the mean purity, purity distributions, and concurrence distributions, respectively, for training sets generated from the density matrix distributions in the left column. The 'P-distributions' and 'C-distributions' columns show the NISQ sample distribution in solid green with the training set distributions overlaid. The average reconstruction fidelities for test states from the NISQ distribution are shown in the right column. Bold are the highest average reconstruction fidelities. The smallest and largest model, respectively, is comprised of 1.3×10^4 and 5.75×10^6 trainable parameters.

						Reconstruction fidelity		
						Neural network		
Distributions	P_{mean}	Range of purity	P-distributions	Range of concurrence	C-distributions	Smallest model	Largest model	MLE
HS	0.471	0.303–0.847		0.000–0.677		0.613 ± 0.083	0.944 ± 0.017	0.986 ± 0.007
Bures	0.562	0.299–0.957		0.000–0.901		0.656 ± 0.113	0.974 ± 0.009	
HS-Haar	0.573	0.288–0.999		0.000–0.977		0.682 ± 0.095	0.983 ± 0.010	
MA (K = 6)	0.581	0.298–0.997		0.000–0.957		0.677 ± 0.105	0.981 ± 0.008	
Z	0.679	0.252–0.999		0.000–0.998		0.749 ± 0.131	0.974 ± 0.012	
Engineered	0.772	0.677–0.960		0.000–0.857		**0.798 ± 0.121**	**0.986 ± 0.006**	

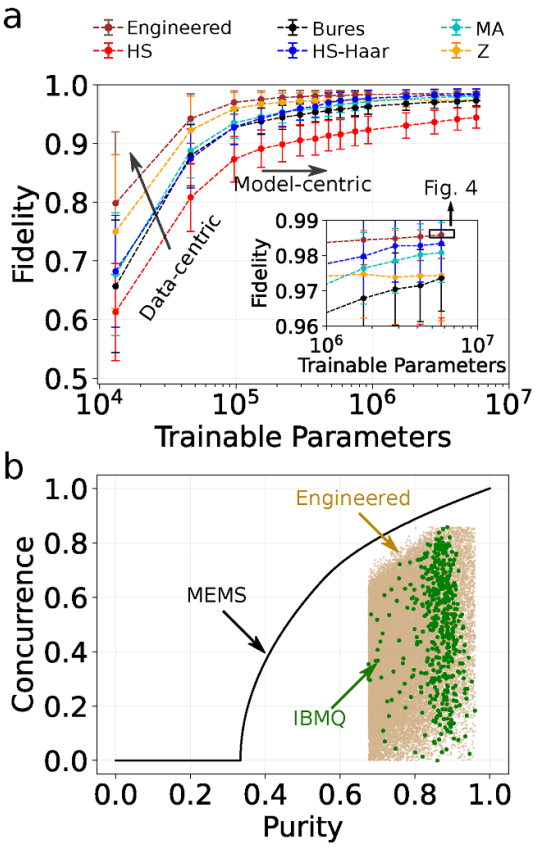


Figure 3. Data-centric and model-centric improvements to ML-QST. (a) Reconstruction fidelity for NISQ-sampled states. The use of various training distributions constitutes a data-centric approach and is shown by an arrow pointing upwards, whereas varying number of trainable parameters is an example of a model-centric approach as indicated by the arrow pointing to the right. The domain from 10^6 to 10^7 parameters is magnified in the inset. The rectangular box in the inset indicates the network architecture used for the results described in figure 4(b). Joint purity-concurrence distributions for the engineered and IBMQ sets, shown by brown and green dots, respectively. The error bars show one standard deviation from the mean. The abbreviations MA, HS, *Z*, and MEMS, respectively, stand for the Mai–Alquier distribution, the Hilbert–Schmidt distribution, the Życzkowski distribution, and maximally entangled mixed states..

reconstruction fidelity. The second subsection describes a surprising result applicable to scenarios where the states composing a test set vary widely in purity. In this case, even given complete access to the distribution of the test set and using that exact distribution to generate the training set, the optimal average reconstruction fidelity is not found by constructing a training set from the same distribution but rather from one slightly more mixed. The two methods in this section can be used independently or in concert with each other and the other heuristics described throughout this paper.

5.1. Low-shot state reconstruction

Experimental data used for state reconstruction will always include statistical noise since measurements can only be repeated a finite number of times. Many practical considerations may further restrict the plausibility of repeating an experiment, such as low count rates, experimental complexity, or the dimension of the underlying quantum system, which results in inefficient scaling of required measurements. The presence of statistical noise in measurement results causes estimated expectation values to differ from their ideal, lowering the reconstruction fidelity. In the context of ML-QST systems, some previous work has demonstrated that incorporating statistical noise comparable to that present in a test set into the training set can improve average reconstruction fidelity of pure states [50]. Here, we extend this fundamentally data-centric technique by applying it both to mixed states generally and to the engineered distribution described in the previous section specifically. In other words, we show that multiple data-centric techniques can be used in a complementary fashion.

For our demonstration we use the same states obtained from NISQ hardware as our test set, but with their measurements simulated at shots ranging from 128 to 8192 (appendix E). Figure 4 plots the reconstruction fidelity as a function of the number of shots used to generate the measurement results in the test set for two different training sets. The red line indicates the reconstruction fidelity of the test set using a network trained on ideal measurement data, meaning we generate measurement probabilities directly from expectation

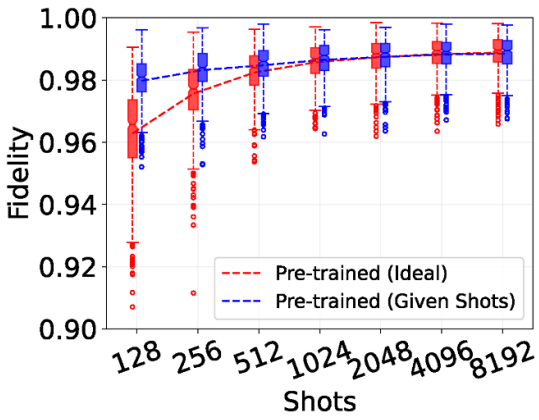


Figure 4. Data-centric approach in the low-shot regime. Reconstructing the NISQ-sampled distribution with simulated measurements performed with shots ranging from 128 to 8192. The red line is the reconstruction fidelity when performed using a network trained on ideal measurements which themselves have no statistical error. The blue line is the reconstruction fidelity when a separate network has been trained for each shot level such that the training set was simulated at the same shot level as the test set. Each box encloses $[Q_1, Q_3]$, with a notch at the median, while the whiskers range from $Q_1 - 1.5(Q_3 - Q_1)$ to $Q_3 + 1.5(Q_3 - Q_1)$, where Q_1 and Q_3 are the first and third quartiles; outliers are plotted as open circles..

values. The blue line is the reconstruction fidelity when the network has been trained on measurement results that have been simulated at the same shot number as the NISQ-sampled test set (x-axis).

Each box encloses $[Q_1, Q_3]$, with a notch at the median, while the whiskers range from $Q_1 - 1.5(Q_3 - Q_1)$ to $Q_3 + 1.5(Q_3 - Q_1)$, where Q_1 and Q_3 are the first and third quartiles; outliers are plotted as open circles. As evident from the divergence of the red and blue lines at low shot numbers, when significant statistical noise is present in a test set it is advantageous to include equivalent statistical noise in the training set. To provide context for the impact of statistical noise at various numbers of shots we note that a significant quantity of measurement results in our data sets, when reconstructed using standard linear inversion, have at least one negative eigenvalue and are hence invalid density matrices. In particular, at 128 shots, over 95% of matrices in our data set would be invalid density matrices if linear inversion were used, and even at 2048 shots, almost 40% of states would be invalid.

5.2. Accounting for heterogeneity in state complexity

An intuitive assumption when generating a training set is that one should aim to match the distribution of the test set as much as possible. Surprisingly, we will find here that it is not always optimal to exactly match the test distribution when the states cover a wide range of purity; instead, a higher reconstruction fidelity is obtained for a given test distribution when we train our network on a slightly more mixed distribution.

To control the relative mean purity of the test and training distributions, we use the MA distribution and fix the test distribution concentration parameter α and simplex size K, where K denotes the number of Haar-random pure states included in the expansion. We then train a network on the same distribution as well as several others with the same concentration parameters but progressively larger K, which increases mixedness in the training distribution [51].

A detailed description for generating heterogeneous test and train states can be found in the appendices E and F. We train a separate network for each K and reconstruct the test states, running each network 10 times and taking the average of all 10 predictions for each test state as the reconstruction fidelity for the given state. The reconstruction fidelities, grouped by the ground truth purity of the test states, are plotted in figure 5(a) for two qubits, (b) for three qubits, and (c) for four qubits. The purity range from 0.3 to 1.0 is divided into 10 bins, and the statistics are evaluated separately in each bin. The vertical and horizontal error bars represent one standard deviation from the binned mean of the reconstruction fidelity and ground truth purity, respectively.

In general, we find that increasing $K > D$ in the training set, which decreases the purity, noticeably enhances reconstruction fidelities for mixed states, while slightly reducing performance for pure states (as shown in the insets of figure 5). Therefore, caution should be taken when choosing the value of K used in the generation of the training set. Nevertheless, on the whole, the improvement for mixed states tends to outweigh any reduction in performance for pure states. We conjecture that this effect can be explained by the difference in the number of terms required to fully describe a state of different purity. For example, a pure

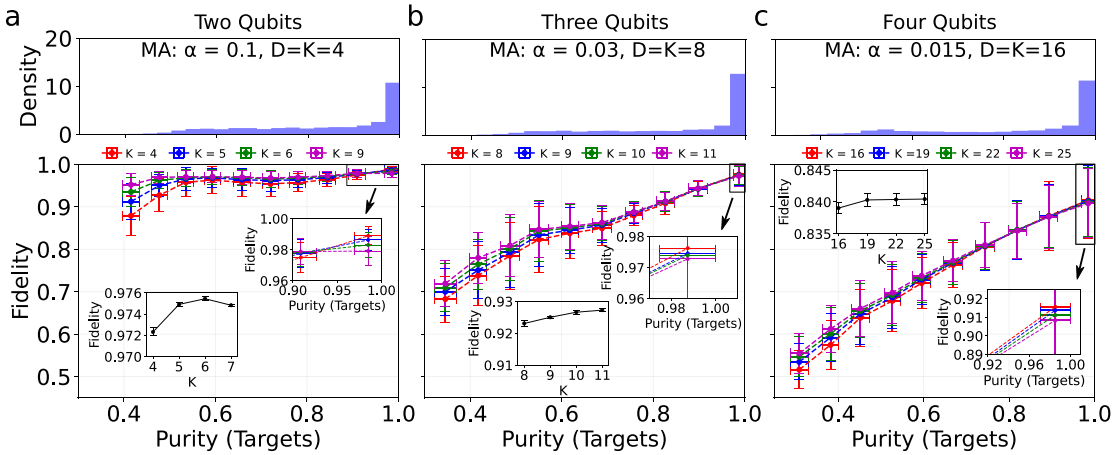


Figure 5. Heterogeneous state complexity. Reconstruction fidelities versus test state purity for (a) two-qubit, (b) three-qubit, and (c) four-qubit quantum systems. The histograms for the purity distributions of the target states are shown at the top. In the lower plots, the red lines indicate reconstruction when the network has been trained using the same distribution (sampled separately) as the test set. Each other curve (blue, green, purple) represents the reconstruction fidelity when the network is instead trained on a more mixed version of the test distribution. The reconstruction fidelities averaged over all test states are shown in the insets, for each K. The vertical and horizontal error bars represent one standard deviation from the binned mean of the reconstruction fidelity and ground truth purity, respectively.

state has fewer free variables than a mixed state of the same dimension, making it more difficult for the network to learn how to reconstruct a mixed state than a pure state. Hence, biasing a training set to be slightly more mixed than the target distribution improves the performance of the network on average.

6. Discussion

Data-centric techniques represent a broad set of valuable and often underutilized strategies for improving the performance of classical ML-based systems used throughout QIS. Unlike model-centric approaches, data-centric methods have the distinct advantage of requiring no alteration to the underlying ML model. Generally speaking, data-centric techniques focus on identifying inadequacies in the construction of data sets, such as false correlations, insufficient variety of examples, and improper scoping. Remedying these deficiencies can significantly improve the performance of ML-based systems, but identifying these errors can require significant domain-specific knowledge. This paper has developed various data-centric heuristics for training set generation that consider prior or domain-specific knowledge to improve system performance, demonstrating the effectiveness of these heuristics with an ML-based quantum state reconstruction system.

Many data-centric heuristics are highly specialized to a particular situation under investigation and broadly include any technique for incorporating prior knowledge, such as the expected average state a system will generate, into the structure of the generated data set. Previous work has considered how to create data sets for ML-based quantum state reconstruction that take into account statistical counting noise, systematic experimental errors, and the expected distribution of states generated by a system [50, 51]. Here, we have added to this list a method for engineering training sets to match distributions of expected experimental scenarios. We compare the effectiveness of our distribution-engineering approach to other standard methods for generating data sets, including those capable of incorporating some amount of prior knowledge such as mean purity.

We describe how spurious correlations can reduce system performance, how it can be challenging to identify these correlations in quantum states given their complexity, and how the inclusion of only a few counterexamples can remedy problems related to these correlations. We show that even for systems as small as two qubits it can be tempting to believe a data set is broadly illustrative of the overall set of possible states. In particular, we consider a training set (MEMS) that includes nearly the full range of possible purity and concurrence values and yet contains a false correlation between the two. We show that, in this example, such a correlation causes our state reconstruction system to misclassify pure separable states as entangled, having only ever seen pure states that are entangled. We then demonstrate that surprisingly few counterexamples need to be added to the training set to remedy this issue. Hence, it is prudent to include several states of every possible classification in any given data set.

More generally, we have also described data-centric heuristics that leverage only broad features of QIS rather than specific prior knowledge about an experiment or scenario. In particular, we find that, given the heterogeneity between the number of free-variables in pure and mixed states, it is not always optimal to endeavor to generate training sets that exactly match the distribution of an experimental scenario in the first place. Instead, when an ML system is to be applied to states covering a wide range of purities, training sets should be biased to be more mixed on average than the expected experimental distribution.

The data-centric heuristics described in this paper focus on situations where training data are synthetically generated, as is often the case in applications of classical ML to QIS-specific problems. The motivation for simulated data sets can be due to convenience, as experimental data may be impractical to obtain, or because the problem itself is theoretical and measured data would only open the possibility of introducing experimental errors. However, the heuristics developed still apply to experimentally obtained training sets, but in those cases can be considered more prescriptive as they suggest the structure of data sets likely to result in the highest-performing ML systems. Due to our focus on synthetic data, we have not included any data-centric methods concerned with data labeling. However, we note that significant work in the ML community has focused on the effects of data labeling and developed a set of data-centric approaches for systematically relabeling or removing mislabeled data to improve overall system performance [47, 65–67]. An interesting problem for future studies would be to consider the application of these label-focused approaches to experimental QIS systems.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/slohani-ai/data-centric-in-qis.

Acknowledgments

Work by S Lohani and T A Searles was supported in part by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Co-design Center for Quantum Advantage (C2QA) under Contract Number DE-SC0012704. A portion of this work was performed at Oak Ridge National Laboratory, operated by UT-Battelle for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. J M L acknowledges funding by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research, through the Early Career Research Program. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. Additionally, we acknowledge use of the IBM Quantum for this work. The views expressed are those of the authors and do not reflect the official policy or position of IBM Quantum. This material is based upon work supported by, or in part by, the Army Research Laboratory and the Army Research Office under Contract/Grant Numbers W911NF-19-2-0087 and W911NF-20-2-0168.

Author contributions

S L and B T K conceived, and all authors contributed to, the design of the research and the analysis of results. S L implemented the neural networks and ran computational experiments. S L, B T K, and J M L drafted the manuscript, with input from all authors. T A S and B T K supervised the work.

Conflict of interest

The authors declare no competing interests.

Appendix A. Neural network details

We implement a custom-designed CNN that takes tomographic measurement values as inputs and reconstructs an estimate of the density matrix as the output, similar to systems described in [48, 50, 51]. Our system has a convolutional layer with a kernel size of (2, 2), stride lengths of 1, ReLU as an activation

function, and filters of size 25. Then we add a max-pooling layer with a pool-size of (2, 2), stride lengths of 2, and a 'valid' padding, followed by a flattening layer. Next, we attach a fully connected dense layer (*dense*_1) using the ReLU activation function. Then, we apply a dropout layer with a 50% dropout rate, followed by another fully connected dense layer (*dense*_2), again, using the ReLU activation function, followed by a dropout layer with the same rate. After this, we attach another fully connected dense layer (*dense*_3) with a linear activation. Note that the number of trainable parameters depends upon the number of neurons at the *dense*_1, *dense*_2, *dense*_3 layers, and the number of qubits.

A.1. Quantum state output

The output of *dense*_3 is a vector τ_{ML} that defines a corresponding density matrix through the Cholesky decomposition. In general, any density matrix can be written as $\rho = \zeta(\tau)\zeta(\tau)^{\dagger}$, where $\zeta(\tau)$ is a lower triangular matrix. The nonzero elements of the matrix $\zeta(\tau)$ can be rearranged into a vector as given by:

$$\zeta(\tau) \longrightarrow (\tau_0, \tau_1, \tau_2, \ldots, \tau_{2^{2d}-1}), \tag{A1}$$

where d is the number of qubits. The first 2^d elements represent the diagonal entries, and the remaining components populate the real and imaginary parts of the off-diagonal entries. As an example, in the two-qubit case $\zeta(\tau)$ is given by:

$$\zeta(\tau) = \begin{bmatrix} \tau_0 & 0 & 0 & 0 \\ \tau_4 + i\tau_5 & \tau_1 & 0 & 0 \\ \tau_{10} + i\tau_{11} & \tau_6 + i\tau_7 & \tau_2 & 0 \\ \tau_{14} + i\tau_{15} & \tau_{12} + i\tau_{13} & \tau_8 + i\tau_9 & \tau_3 \end{bmatrix}. \tag{A2}$$

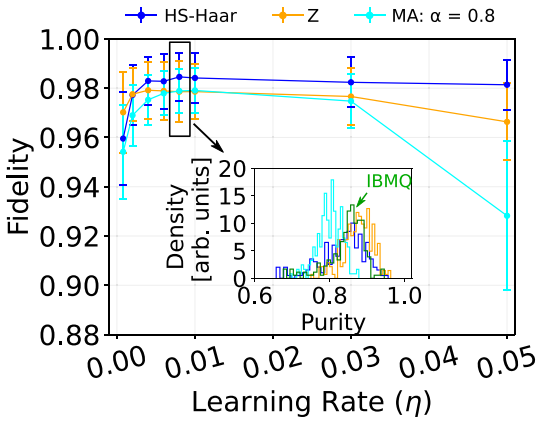
During training, the ground truth target vector τ_g is provided to the network, and the trainable parameters are optimized to minimize the mean squared error (MSE) $\langle ||\tau_{ML} - \tau_g||^2 \rangle$, where the average is taken over the batch of training set. Once trained, the network takes any collection of measurement values as an input and outputs a prediction, τ_{ML}. For validation of the trained network, we utilize measurement data generated from a test set of density matrices ρ_g—which may not match the training set—and compute the density matrix ρ_{ML} corresponding to the network output τ_{ML}. The fidelity $F(\rho_{ML}, \rho_g) = \left[\mathrm{Tr}\sqrt{\sqrt{\rho_g}\rho_{ML}\sqrt{\rho_g}}\right]^2$ is then used to quantify accuracy. Note that our network predicts $\zeta(\tau_{ML})$ rather than the final ρ_{ML} directly, which we recover from $\rho_{ML} = \zeta(\tau_{ML})\zeta(\tau_{ML})^{\dagger}/\mathrm{Tr}[\zeta(\tau_{ML})\zeta(\tau_{ML})^{\dagger}]$. This approach, which is also standard in MLE [68], ensures that our predicted matrices are always physical. Finally, while the approach described here requires prior knowledge of the dimension of the system it will be applied to, recent work has demonstrated an approach for performing any n qubit reconstruction using a reconstruction method designed explicitly for $m \geqslant n$ qubits [69]. While we do not pursue that extension in this manuscript, the techniques described in this paper are compatible with such an approach.

A.2. Learning rate optimization

The learning rate (η) is an important hyperparameter affecting the training of a network [70]. We vary the learning rate for a network from 0.0008 to 0.05 and evaluate the fidelity of the reconstructed two-qubit NISQ-sampled test density matrices. The results for several quantum state training distributions are shown in appendix figure 1. We find that increasing the rate parameter gradually increases the fidelity for all the training cases and peaks around $\eta = 0.008$. Additionally, we show the purity distributions of the reconstructed density matrices when $\eta = 0.008$ in the inset. We find that at $\eta = 0.008$ the predicted purities have good overlap with the target IBMQ distribution and therefore use $\eta = 0.008$ for all results in this paper.

A.3. Synthetically generated training sets

We have opted to use synthetic data in our manuscript not due to any limitation of our approach but rather because our goal here is to evaluate the performance of our constructed distributions. Evaluating our distributions is achieved most straightforwardly when we have direct access to the generation of measurement results and know the ground truth without any error.



Appendix Figure 1. Optimizing learning rate. Fidelity of reconstructed density matrices versus learning rate η. The full purity distributions of the reconstructed states for $\eta = 0.008$ are shown in the inset. The error bars show one standard deviation from the mean fidelity.

Appendix B. Maximally entangled mixed states (MEMS)

To define a restricted subspace within the overall Hilbert space with a strong correlation between purity and entanglement, we can generate training states from local rotations of two-qubit MEMS. The MEMS define a particular class of states which, for a given linear entropy, have the maximum possible concurrence [71, 72]. In general, MEMS can be expressed as (up to local rotations):

$$\rho_{MEMS} = \begin{bmatrix} g(\gamma) & 0 & 0 & \frac{\gamma}{2} \\ 0 & 1-2g(\gamma) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\gamma}{2} & 0 & 0 & g(\gamma) \end{bmatrix}, \tag{B1}$$

where

$$g(\gamma) = \begin{cases} \frac{\gamma}{2} & ; \ \gamma \geqslant \frac{2}{3} \\ \frac{1}{3} & ; \ \gamma < \frac{2}{3} \end{cases}, \tag{B2}$$

and the parameter $\gamma \in [0, 1]$ is equal to the concurrence. The purity of the state in equation (B1) is given by

$$P(\gamma) = 1 - 4g(\gamma) + 6g^2(\gamma) + \frac{\gamma^2}{2}, \tag{B3}$$

which ranges from $\frac{1}{3} \leqslant P(\gamma) \leqslant 1$. We generate an element of our training set ρ'_{MEMS} according to:

$$\rho'_{MEMS} = (U_a(2) \otimes U_b(2))\, \rho_{MEMS}\, (U_a(2) \otimes U_b(2))^\dagger, \tag{B4}$$

where γ is drawn from the Uniform distribution, $\gamma \sim \text{Uniform}(0, 1)$, and $U_i(2)$ is a two-dimensional Haar-random unitary matrix applied to qubit i. When plotting the concurrence of these states as a function of their purity, they form a curve as shown in figures 2(c) and (d). Note that the MEMS span a wide range of possible values of purity and concurrence. We stress this fact about the MEMS to demonstrate how even for two qubits, the simplest of all entangled systems, it can be challenging to detect spurious correlations when only considering general properties of states independent of each other. In this case, the relationship between purity and entanglement [71, 72] is well known, but such relationships may be significantly more difficult to detect in more complex systems.

Appendix C. Supplementing MEMS with separable states

We generate separable states according to:

$$\rho_s = \rho_a \otimes \rho_b, \quad \text{such that} \quad \text{Tr}(\rho_s^2) > \frac{1}{3}, \tag{C1}$$

where ρ_a and ρ_b are random full-rank density matrices sampled from the HS distribution. To improve generalizability of the MEMS training set, we include randomly sampled states from ρ_s such that the total number of states in the training set (states from ρ'_{MEMS} and ρ_s) is:

$$N_{\text{train}} = N' + N_s, \tag{C2}$$

where N_{train} represents the total number of training states, N' is the number of states drawn using equation (B4) and N_s the total number sampled from equation (C1). We fix $N_{\text{train}} = 30000$ and modify the fraction N_s/N_{train}. For the test sets in figure 2, we sample 5000 random states using either equation (B4) (entangled states) or equation (C1) (separable states). Note that the training and test sets are drawn randomly and independently.

We vary N_s from 0 to 1750 with a step size of 250 and train a separate neural network at *dense*_1 = 3050 and *dense*_2 = 1650 up to 400 epochs at a learning rate of 0.008. After this, the pre-trained networks are used to classify the test measurement results as corresponding to either a separable or entangled state. For the results in figure 2, we train the same network architecture 10 times for each case and take the average of all the predictions (from 10 trials) for a given state to minimize the effects of random initialization during training.

Appendix D. Standard distributions of random quantum states

Here we briefly review the salient features of the most common methods for defining distributions of random quantum states. Beyond fundamental motivations [73–75], many efforts to perform state reconstruction and classification using ML-based methods have relied on these distributions to generate training sets [1, 48–51]. These distributions serve as baselines for evaluating ML-based system performance with our data-centric heuristics.

D.1. Hilbert–Schmidt (HS) distribution

Random quantum states distributed according to the HS measure can be induced through the partial trace on Haar-random pure states in higher dimensions [75]. Operationally, ensembles of HS-distributed random quantum states are typically generated by sampling the complex Ginibre ensemble [76], which comprises $D \times D$ complex matrices whose elements are independently drawn from the complex standard normal distribution [75]. Specifically, random quantum states distributed according to the HS measure can be obtained using

$$\rho = \frac{GG^\dagger}{\text{Tr}\,(GG^\dagger)}, \tag{D1}$$

where G is a random matrix from the Ginibre ensemble.

D.2. Bures distribution

Similar to the case of the HS distribution, a random quantum state ρ from the Bures ensemble can be sampled according to

$$\rho = \frac{(\mathbb{1} + U)\, GG^\dagger\, (\mathbb{1} + U^\dagger)}{\text{Tr}\,[(\mathbb{1} + U)\, GG^\dagger\, (\mathbb{1} + U^\dagger)]}, \tag{D2}$$

where G is, again, a random matrix from the Ginibre ensemble and U is a Haar-distributed random unitary from $U(D)$ [77].

D.3. Hilbert–Schmidt–Haar (HS–Haar) distribution

Previous studies have noted that ensembles of random quantum states distributed according to the HS and Bures measures have limited applicability for many NISQ devices due to their low average purities [51]. Therefore we define here a simple technique for biasing an arbitrary input distribution toward a higher average purity. In particular, we consider a convex combination of HS-distributed quantum states (ρ_{HS}) and random Haar-distributed pure states (ρ_H) as given by:

$$\rho = (1 - \delta)\, \rho_{HS} + \delta\, \rho_H, \tag{D3}$$

where δ is chosen uniformly at random from the interval $[0, 1]$.

D.4. Mai–Alquier (MA) distribution

This distribution was originally studied as a prior for Bayesian QST [62, 63, 78–80] and was recently utilized in [51] to generate training sets for ML-based state reconstruction methods. The MA distribution is defined as a mixture of Haar-random pure states with coefficients drawn from the Dirichlet distribution. The probability density function of the Dirichlet distribution for vectors $\mathbf{x} = (x_1, \ldots, x_K)$—where the elements of $\mathbf{x}$ belong to the open $K-1$ simplex ($x_i \geqslant 0$ and $\sum_{i=1}^{K} x_i = 1$)—is:

$$\mathrm{Dir}(\mathbf{x}|\boldsymbol{\alpha}) = \frac{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\prod_{i=1}^{K}x_i^{\alpha_i-1}, \tag{D4}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ with $\alpha_i \geqslant 0$ defines the concentration parameters and $\Gamma(\cdot)$ is the standard gamma function. The concentration parameters provide flexibility to alter the overall features of the distribution. Therefore, an ensemble of D-dimensional mixed states from a convex sum of K Haar-random pure quantum states $|\psi_i\rangle$ is written as:

$$\rho = \sum_{i=1}^{K} x_i |\psi_i\rangle\langle\psi_i|, \tag{D5}$$

where the vector $\mathbf{x}$ is a random variable distributed according to $\mathrm{Dir}(\mathbf{x}|\alpha)$: for simplicity, we specialize to the symmetric case $\boldsymbol{\alpha} = \{\alpha, \ldots, \alpha\}$ only. The expectation value of the purity is:

$$\mathrm{E}_{MA}\left[\mathrm{Tr}(\rho^2)\right] = \frac{D + \alpha(D+K-1)}{D(1+\alpha K)}. \tag{D6}$$

Finally, we note that in [51] strong evidence was presented that the MA distribution reduces to the HS distribution for a specific set of input parameters.

D.5. Życzkowski (Z) distribution

Dirichlet-distributed vectors $\mathbf{x}$ are again employed to generate random density matrices as described by [73], which we refer to as the Z distribution for convenience. This approach relies on the unitary invariance of the eigenvalues of a density matrix and utilizes the Dirichlet vectors $\mathbf{x}$ of length D as the eigenvalues of D-dimensional states. Once the eigenvalues are generated they are placed along the diagonal of a $D \times D$ matrix and a Haar-random unitary from $U(D)$ is applied. Note that the resulting construction is of the same form as equation (D5) for $K = D$, but with all states in the convex sum orthogonal. The expectation value of the purity of Z-distributed states is given by:

$$\mathrm{E}_Z\left[\mathrm{Tr}\left(\rho^2\right)\right] = \sum_{j=1}^{D}\mathrm{E}\left[x_j^2\right] = \frac{1+\beta}{1+D\beta}, \tag{D7}$$

where we have used β as the concentration parameter of the Dirichlet distribution so as not to be confused with the MA expressions (where α is used). Like the MA distribution, the Z distribution can be biased in various ways through manipulation of concentration parameters. A discussion of how the MA and Z distributions compare against experimentally measured distributions can be found in [51]. Finally, we note that the Z distribution is a widely employed method for generating random density matrices including for machine-learning applications such as the state classifier described in [1].

Appendix E. Test distributions

In this section we describe the two main test distributions enlisted for evaluating our data-centric heuristics, one based on experimental states from a NISQ machine and the other selected for its wide variation in purity.

E.1. Test sets from NISQ hardware

To illustrate several of our heuristics on realistic experimental scenarios, we utilize data sets consisting of tomographic measurements performed on random quantum states implemented on *ibmq_jakarta*, one of the IBMQ Falcon processors. We first numerically generate 500 Haar-random two-qubit pure states and initialize these on *ibmq_jakarta*. Then, the states are automatically transpiled from the backend into the required quantum circuits for generation. The depths of the transpiled quantum circuits—i.e. the longest path from input to output—range between 12 and 16 gates. For each state, we perform full state tomography

with a total of 36 measurement projections, corresponding to the four outcomes for all nine two-qubit combinations of the Pauli operators $\{X,Y,Z\}_1 \otimes \{X,Y,Z\}_2$.

We then reconstruct the measured quantum states using maximum likelihood estimation (MLE) according to the algorithm in [64]. Unfortunately, due to random noise on the backend hardware, the estimated states are mixed despite having been programmed as pure [50], leaving uncertainty about the ground truth state that the data represent. In other words, the state we programmed into the IBMQ processor and the state found via MLE reconstruction differ significantly. While this mixing due to noise is expected, it is also unpredictable and presents a practical challenge for comparing MLE-based reconstructions and our ML-based reconstructions as we no longer have a ground truth target state to compare against (what should be the pure state programmed into the processor). Further, if we merely take the MLE reconstruction of the system as the new ground truth while using the experimentally obtained measurement results, this would unfairly assume MLE reconstructs with perfect accuracy. Therefore, to retain the general properties of the distribution generated by *ibmq_jakarta* while permitting the construction of test sets with known ground truth states, we perform additional rounds of tomographic simulations on the MLE-obtained results; these synthetic measurement results comprise the test sets below. We simulate measurement results using the methods described in [50] which further allow us to select the amount of statistical noise (shots) on demand.

The measurement data from the 500-state test set used in figure 3 are generated assuming 1024 shots, meaning every Pauli measurement circuit was executed 1024 times each. The resultant distribution has a purity and concurrence in the range $[0.68, 0.96]$ and $[0, 0.86]$, respectively, which is used to inform the engineered distribution. Note that the test states are completely unknown and hidden from the network during training, and the only information used to inform the construction of training sets is the maximum and minimum of the purity and concurrence.

E.2. Generating heterogeneous states

For our test sets in section 5.2, we draw 5000 random quantum states from the MA distribution with the parameters $(\alpha, K) = (0.1, 4)$ for two qubits, $(\alpha, K) = (0.03, 8)$ for three qubits, and $(\alpha, K) = (0.015, 16)$ for four qubits, simulating ideal Pauli measurements on each. Note that for informationally complete tomography, the number of measurements and number of neurons in *dense*_3, respectively, scales as 6^d and 2^{2d}, where d is the number of qubits. The purity distributions for all cases are shown in the top row of figure 5. At each qubit number, we generate four training sets each with 30 000 random quantum states sampled from MA distributions with the same α as the corresponding test set, but with varying K: $K \in \{4, 5, 6, 7\}$ for two qubits, $K \in \{8, 9, 10, 11\}$ for three qubits, and $K \in \{16, 19, 22, 25\}$ for four qubits.

Appendix F. Engineered training sets

Motivated by the restricted nature of existing techniques, we outline a method for engineering an arbitrary input distribution to conform to certain general characteristics desired in a training set. Several of the standard distributions above allow for biasing based on a single input. This is enough to control, for example, the mean purity [51]. However, a single parameter may not always be enough to meaningfully constrain a distribution for a given use case. For example, even in the two-qubit case considered here, the purity and the entanglement only bound rather than determine each other [71]. The situation becomes even more complex for higher-dimensional systems where several inequivalent classes of entanglement exist [81].

F.1. Method details

The method described in figure 1 consists of repeatedly sampling from a suitably chosen input distribution followed by the application of a simultaneous bandpass filter for both purity and concurrence. In general, the bandpass filter approach can be applied to any measurable property or properties of the sampled states. However, we have chosen purity and concurrence for this demonstration as they are both well-understood properties of two-qubit density matrices. Hence, in many experiments of interest, the approximate maximum and minimum values of the purity and concurrence are easily inferred. In short, the bandpass filter approach can be summarized as first randomly sampling states ρ_π from an arbitrary input distribution Π and then passing them through the simultaneous filter given by:

$$\rho_{\text{eng}} = \rho_\pi \Big[\big(C_{\min} \leqslant C(\rho_\pi) \leqslant C_{\max}\big) \,\&\, \big(P_{\min} \leqslant P(\rho_\pi) \leqslant P_{\max}\big) \Big], \tag{F1}$$

where ρ_{eng} are the filtered (engineered) states, and C and P represent the concurrence and purity, respectively.

Algorithm 1. Engineered distributions.

Input: $P_{\min}$, $P_{\max}$, $C_{\min}$, $C_{\max}$
Output: Engineered States (ρ_{eng})

$$\alpha \leftarrow \frac{D\left[1-P_{\min}\right]}{D\left[DP_{\min}-1\right]-D+1}$$

$\boldsymbol{\alpha} \leftarrow (\alpha, \alpha, \alpha, \alpha, \ldots, \alpha)$ (K terms); $K \geqslant D$

// Draw N samples

$$[\mathbf{x}]_{N,K} \leftarrow \mathrm{Dir}(x|\alpha) = \frac{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\prod_{i=1}^{K}x_i^{\alpha_i-1}$$

// Reshape into a 4D-array

$[x]_{N,K,1,1} \leftarrow \mathbf{x}_{N,K}$
Generate $N \times K$ Haar-random pure-states, $[\rho_{\mathrm{Haar}}]_{N\times K,4,4} \leftarrow |\psi\rangle\langle\psi|$
$[\rho_{\mathrm{Haar}}]_{N,K,4,4} \leftarrow [\rho_{\mathrm{Haar}}]_{N\times K,4,4}$

// Perform the element-wise multiplication, and then, take a sum along the first dimension

$$[\rho_{MA}]_{N,4,4} = \sum_{k=1}^{K}[x]_{N,k,1,1}\cdot[\rho_{\mathrm{Haar}}]_{N,k,4,4}$$

// Collect N Pauli Y-Operators and reshape it to a 3D-array

$[Y]_{N,2,2} \leftarrow Y_N$
$[\tilde{\rho}_{MA}]_{N,4,4} \leftarrow [\rho^{*}_{MA}]_{N,4,4}$
$[R]_{N,4,4} \leftarrow \sqrt{\sqrt{\rho_{MA}}(Y\otimes Y)\tilde{\rho}_{MA}(Y\otimes Y)\sqrt{\rho_{MA}}}$
$[\lambda]_{N,4} \leftarrow eigenvalues([R]_{N,4,4})$

// Sort eigenvalues in increasing order, $\lambda_1 > \lambda_2 > \lambda_3 > \lambda_4$

$$\left[(\lambda_4, \lambda_3, \lambda_2, \lambda_1)\right]_{N,4} \leftarrow sort\left([\lambda]_{N,4}\right)$$

// Evaluate the concurrence (C) and purity (P) of ρ_{MA} as

$[C(\rho_{MA})]_N \leftarrow \max(0, \lambda_1 - \lambda_2 - \lambda_3 - \lambda_4)$
$[P(\rho_{MA})]_N \leftarrow \mathrm{Tr}([\rho^2_{MA}]_{N,4,4})$

// Perform the element-wise logical AND operation

$$\rho_{\mathrm{eng}} \leftarrow \rho_{MA}\left[\left(C_{\min} \leqslant [C(\rho_{MA})]_N \leqslant C_{\max}\right)\&\left(P_{\min} \leqslant [P(\rho_{MA})]_N \leqslant P_{\max}\right)\right]$$

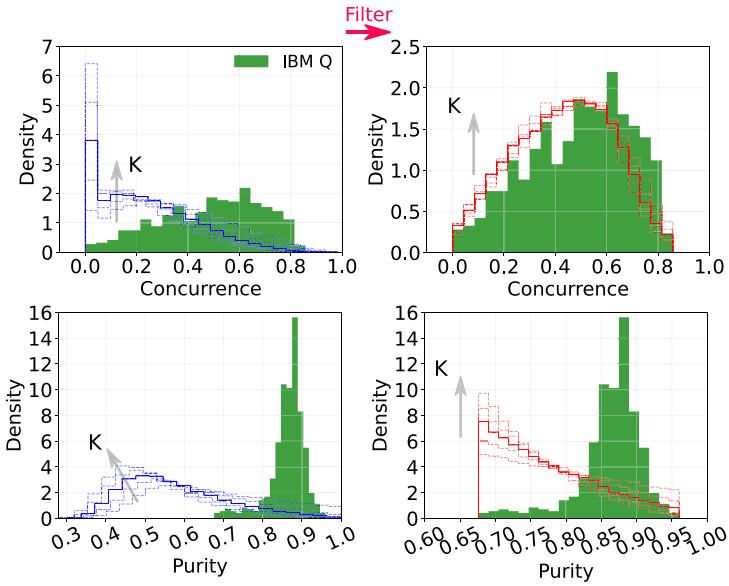
return ρ_{eng}
/*End */

In principle, we can use any input distribution Π in the above approach. Hence, our method for engineering distributions of random states only requires upper and lower bounds on the purity and concurrence, or some other property of the states, as input. However, the MA distribution is particularly convenient as an initial distribution because it allows extra freedom in biasing the resultant distribution of states. We recommend that input distribution Π be chosen according to the following recipe. First, set $K = D$ (the minimum K value capable of producing full rank matrices) and tune α such that the mean of this distribution is equal to the chosen lower bound. Second, based on tuning results below, we then recommend that once α is fixed, the distribution be sampled with $K > D$. The full procedure is outlined in algorithm 1.

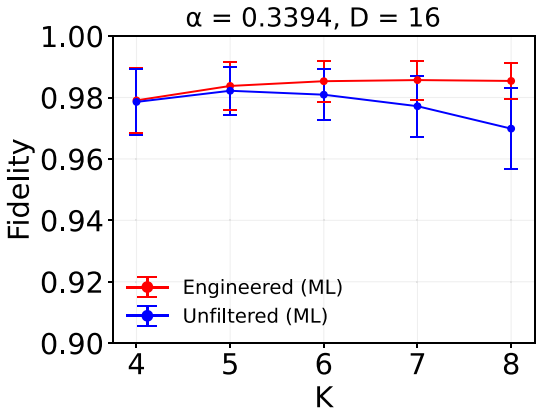
F.2. Tuning *K*

Our engineered training set algorithm relies on specification of the parameter K in the MA distribution. Here we test the impact of tuning this parameter on for the case of $D = 4$ (two qubits) for $K \in \{4, 5, \ldots, 8\}$ and $\alpha = 0.3394$ (chosen according to algorithm 1). The corresponding distributions with respect to the concurrence and purity are shown by unfilled blue histograms in appendix figure 2(left). The solid-line histogram represents the case of $K = 6$, whereas the filled green histogram represents the test distributions obtained from the NISQ hardware. As shown, the increase in the value K is directed upward, increasing the mixedness of the training samples. Without any filter, we find that the increase in reconstruction fidelities with K quickly saturates and then gradually decreases as shown by the blue line in appendix figure 3. The error bars show one standard deviation from the mean.

In order to address the issue, we apply a filter of concurrence and purity to remove unwanted mixed states from the training set (as prescribed in algorithm 1). The concurrence (top) and purity (bottom) histograms for the filtered (engineered) distributions of sampled states are shown by unfilled red histograms in appendix figure 2(right). With a network pre-trained with these engineered states, the reconstruction



Appendix Figure 2. Engineered states. Unfiltered (left) and engineered (right) state distributions from MA with $\alpha = 0.3394$ and $D = 4$, with respect to concurrence (top) and purity (bottom). The value of K consecutively increases in each plot from 4 to 8.



Appendix Figure 3. Engineered states. Reconstruction fidelities versus the value of K.

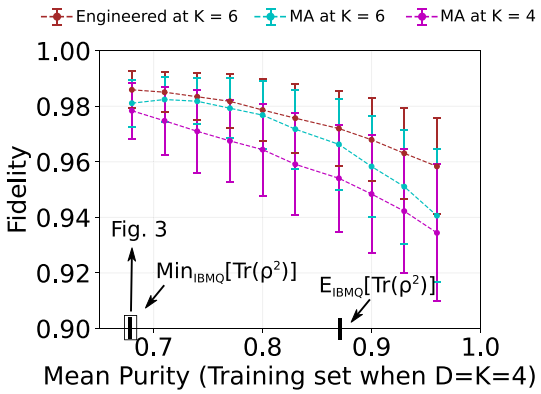
fidelity gradually increases as shown by the red line in appendix figure 3, saturating by around $K = 6$. Based on these findings, we use $K = 6$ for generating the training set used for the 'engineered' results in figure 3.

F.3. Tuning α

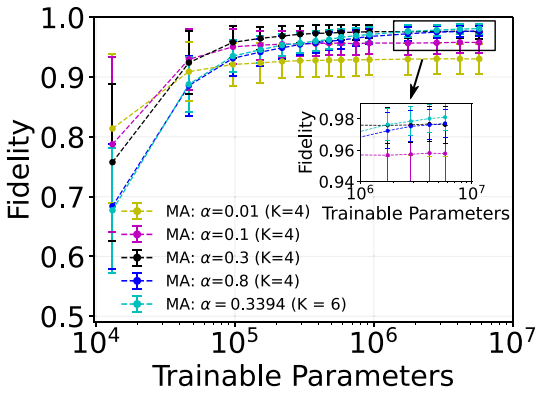
In algorithm 1, we suggest selecting the concentration parameter α of the initial distribution such that its mean purity is equal to $P_{\min}$ of the target distribution. Here we consider the performance of this parameter selection. We generate engineered training sets where the initial concentration parameter α is chosen such that an MA distribution with $K = 4$ will have mean purity given by the x-axis in appendix figure 4. (The mean of the engineered and $K = 6$ training sets will thus differ from this value.) We then use these engineered data sets to reconstruct the same NISQ-sampled data as in figure 3(a). For comparison, we perform the same exercise for MA distributions that have not gone through bandpass filters. We see that the engineered distribution outperforms the base MA distribution in all cases. Further, our heuristic choice of setting the mean of the MA distribution to $P_{\min}$ is near-optimal over the presented range.

F.4. Comparing to other (α, K) combinations

To further consider the performance of our heuristically chosen values for K and α described above, we also redo the tests in figure 3(a) for several other parameter choices in appendix figure 5. We see that our heuristic choice performs slightly worse for models with fewer trainable parameters but ultimately outperforms the different curves as the trainable parameters increase.



Appendix Figure 4. Reconstruction fidelity of NISQ-sampled test set versus mean purity of MA-distributed training states when $K = 4$. The two vertical tick marks along the x-axis emphasized with arrows correspond to the scenarios where the mean purity of the training set matches the minimum and mean purity of the NISQ-sampled states when $D = K = 4$. The mean purity of the states (labeled with subscripts IBMQ) is shown by a vertical line on the x-axis as indicated by an arrow. The error bars represent one standard deviation from the mean.



Appendix Figure 5. Reconstruction fidelity versus trainable parameters for various MA-distributed training sets. The pairs are chosen as $(\alpha, K) \in \{(0.01,4),(0.1,4),(0.3,4),(0.8,4),(0.3394,6)\}$ for training sets. The results shown by a dotted cyan-line is derived from figure 3(a). The error bars show one standard deviation from the mean fidelity.

Appendix Table 1. Number of trainable parameters as a function of number of neurons in fully connected dense layers for two-qubit tomography.

dense_1	50	150	250	350	450	550	650	750	850	950	1050	1550	2050	2550	3050
dense_2	25	75	150	200	250	300	350	400	450	550	650	900	1150	1400	1650
Parameters ($\times 10^6$)	0.013	0.047	0.097	0.152	0.22	0.29	0.38	0.48	0.58	0.75	0.93	1.76	2.85	4.17	5.75

Appendix G. Model-centric approaches

For each training set in figure 3, we also consider the impact of an additional model-centric approach which consists of increasing the number of trainable parameters in the model. As the number of qubits is fixed, the number of neurons in the *dense*_3 layer is also fixed at 16. Therefore, the number of neurons in the *dense*_1 and *dense*_2 layers determine the total number of trainable parameters as shown in appendix table 1. We use the same network architecture for the model with all combinations of sampling distributions.

ORCID iDs

Sanjaya Lohani https://orcid.org/0000-0003-0699-0669
Joseph M Lukens https://orcid.org/0000-0001-9650-4462
Thomas A Searles https://orcid.org/0000-0002-0532-7884
Brian T Kirby https://orcid.org/0000-0002-2698-9887

References

[1] Lu S, Huang S, Li K, Li J, Chen J, Lu D, Ji Z, Shen Y, Zhou D and Zeng B 2018 *Phys. Rev.* A **98** 012315
[2] Harney C, Pirandola S, Ferraro A and Paternostro M 2020 *New J. Phys.* **22** 045001
[3] Ahmed S, Muñoz C S, Nori F and Kockum A F 2021 *Phys. Rev. Res.* **3** 033278
[4] Wu L T, Zhu E Y and Qian L 2021 *CLEO: QELS_Fundamental Science* (Washington, DC: Optical Society of America) p FW3N.1
[5] Niu M Y, Boixo S, Smelyanskiy V N and Neven H 2019 *npj Quantum Inf.* **5** 33
[6] Zhang X-M, Wei Z, Asad R, Yang X-C and Wang X 2019 *npj Quantum Inf.* **5** 85
[7] Porotti R, Tamascelli D, Restelli M and Prati E 2019 *Commun. Phys.* **2** 61
[8] Bukov M, Day A G, Sels D, Weinberg P, Polkovnikov A and Mehta P 2018 *Phys. Rev.* X **8** 031086
[9] Ding Y, Ban Y, Martín-Guerrero J D, Solano E, Casanova J and Chen X 2021 *Phys. Rev.* A **103** L040401
[10] Ban Y, Echanobe J, Ding Y, Puebla R and Casanova J 2021 *Quantum Sci. Technol.* **6** 045012
[11] Xu H, Li J, Liu L, Wang Y, Yuan H and Wang X 2019 *npj Quantum Inf.* **5** 82
[12] Schuff J, Fiderer L J and Braun D 2020 *New J. Phys.* **22** 035001
[13] Wang W and Lo H-K 2019 *Phys. Rev.* A **100** 062334
[14] Ding H-J, Liu J-Y, Zhang C-M and Wang Q 2020 *Quantum Inf. Process.* **19** 60
[15] Lohani S, Knutson E M, O'Donnell M, Huver S D and Glasser R T 2018 *Appl. Opt.* **57** 4180
[16] Lohani S and Glasser R T 2018 *Opt. Lett.* **43** 2611
[17] Knutson E, Lohani S, Danaci O, Huver S D and Glasser R T 2016 *Proc. SPIE* **9970** 997013
[18] Bhusal N, Lohani S, You C, Hong M, Fabre J, Zhao P, Knutson E M, Glasser R T and Magaña-Loaiza O S 2021 *Adv. Quantum Technol.* **4** 2000103
[19] Lohani S, Knutson E M and Glasser R T 2020 *Commun. Phys.* **3** 1
[20] Ahmed S, Muñoz C S, Nori F and Kockum A F 2021 *Phys. Rev. Lett.* **127** 140502
[21] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L and Zdeborová L 2019 *Rev. Mod. Phys.* **91** 045002
[22] Dunjko V and Briegel H J 2018 *Rep. Prog. Phys.* **81** 074001
[23] Bharti K, Haug T, Vedral V and Kwek L-C 2020 *AVS Quantum Sci.* **2** 034101
[24] Lohani S and Glasser R T 2020 *Mach. Learn.: Sci. Technol.* **1** 035006
[25] Tran Q H and Nakajima K 2021 *Phys. Rev. Lett.* **127** 260401
[26] Genois E, Gross J A, Di Paolo A, Stevenson N J, Koolstra G, Hashim A, Siddiqi I and Blais A 2021 *PRX Quantum* **2** 040355
[27] Chu X and Ilyas I F 2019 *Data Cleaning* (New York: ACM/Association for Computing Machinery)
[28] Whang S E, Roh Y, Song H and Lee J-G 2021 (arXiv:2112.06409)
[29] Renggli C, Rimanic L, Gürel N M, Karlaš B, Wu W and Zhang C 2021 (arXiv:2102.07750)
[30] Wei J and Zou K 2019 (arXiv:1901.11196)
[31] Cubuk E D, Zoph B, Mane D, Vasudevan V and Le Q V 2019 *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 113–23
[32] Ratner A J, Ehrenberg H, Hussain Z, Dunnmon J and Ré C 2017 *Advances in Neural Information Processing Systems* vol 30
[33] Wang T, Zhu J-Y, Torralba A and Efros A A 2018 (arXiv:1811.10959)
[34] Motamedi M, Sakharnykh N and Kaldewey T 2021 (arXiv:2110.03613)
[35] Northcutt C, Jiang L and Chuang I 2021 *J. Artif. Intell. Res.* **70** 1373
[36] Pleiss G, Zhang T, Elenberg E R and Weinberger K Q 2020 (arXiv:2001.10528)
[37] Xu L, Liu J, Pan X, Lu X and Hou X 2021 arXiv:2111.08647
[38] Varma P and Ré C 2018 *Proc. VLDB Endowment. Int. Conf. on Very Large Data Bases* vol 12 (NIH Public Access) p 223
[39] Paiva P Y A, Smith-Miles K, Valeriano M G and Lorena A C 2021 (arXiv:2109.14430)
[40] Breck E, Polyzotis N, Roy S, Whang S and Zinkevich M 2019 *Proc. Machine Learning and Systems (MLSys)*
[41] Tremblay J, Prakash A, Acuna D, Brophy M, Jampani V, Anil C, To T, Cameracci E, Boochoon S and Birchfield S 2018 *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops* pp 969–77
[42] Liu Z Y-C, Roychowdhury S, Tarlow S, Nair A, Badhe S and Shah T 2021 arXiv:2111.12548
[43] Westermann H, Šavelka J, Walker V R, Ashley K D and Benyekhlef K 2021 *Legal Knowledge and Information Systems* (Amsterdam: IOS Press) pp 54–57
[44] Lee Y, Kwon O J, Lee H, Kim J, Lee K and Kim K-E 2021 (arXiv:2112.03837)
[45] Lucic M, Kurach K, Michalski M, Gelly S and Bousquet O 2017 (arXiv:1711.10337)
[46] Sambasivan N, Kapania S, Highfill H, Akrong D, Paritosh P and Aroyo L M 2021 *Proc. 2021 CHI Conf. on Human Factors in Computing Systems* pp 1–15
[47] Huang H-Y, Broughton M, Mohseni M, Babbush R, Boixo S, Neven H and McClean J R 2021 *Nat. Commun.* **12** 2631
[48] Lohani S, Kirby B T, Brodsky M, Danaci O and Glasser R T 2020 *Mach. Learn.: Sci. Technol.* **1** 035007
[49] Danaci O, Lohani S, Kirby B T and Glasser R T 2021 *Mach. Learn.: Sci. Technol.* **2** 035014
[50] Lohani S, Searles T A, Kirby B T and Glasser R T 2021 *IEEE Trans. Quantum Eng.* **2** 1
[51] Lohani S, Lukens J M, Jones D E, Searles T A, Glasser R T and Kirby B T 2021 *Phys. Rev. Res.* **3** 043145
[52] Lohani S 2022 Machine Learning for Physical Sciences (MLPHYS) *Computer software* (available at: https://github.com/slohani-ai/machine-learning-for-physical-sciences)
[53] Lohani S 2022 Data-centric Machine Learning in QIS *Computer software* (available at: https://github.com/slohani-ai/data-centric-in-qis)
[54] Huang H-Y, Kueng R and Preskill J 2020 *Nat. Phys.* **16** 1050
[55] Lukens J M, Law K J H and Bennink R S 2021 *npj Quantum Inf.* **7** 113
[56] Kawaguchi K, Kaelbling L P and Bengio Y 2017 (arXiv:1710.05468)
[57] Miller J P, Taori R, Raghunathan A, Sagawa S, Koh P W, Shankar V, Liang P, Carmon Y and Schmidt L 2021 *Int. Conf. on Machine Learning* (PMLR) pp 7721–35
[58] DeVries T and Taylor G W 2018 (arXiv:1802.04865)
[59] Meinke A and Hein M 2019 (arXiv:1909.12180)
[60] Peres A 1996 *Phys. Rev. Lett.* **77** 1413
[61] Horodecki M, Horodecki P and Horodecki R 1996 *Phys. Lett.* A **223** 1
[62] Mai T T and Alquier P 2017 *J. Stat. Plan. Inference* **184** 62
[63] Lukens J M, Law K J H, Jasra A and Lougovski P 2020 *New J. Phys.* **22** 063038

[64] Smolin J A, Gambetta J M and Smith G 2012 *Phys. Rev. Lett.* **108** 070502
[65] Pan I, Mason L R and Matar O K 2022 *Chem. Eng. Sci.* **249** 117271
[66] Ngiam K Y and Khor W 2019 *Lancet Oncol.* **20** e262
[67] Tanaka I, Rajan K and Wolverton C 2018 *MRS Bull.* **43** 659
[68] James D F V, Kwiat P G, Munro W J and White A G 2001 *Phys. Rev.* A **64** 052312
[69] Lohani S, Regmi S, Lukens J M, Glasser R T, Searles T A and Kirby B T 2022 (arXiv:2205.05804)
[70] Bengio Y 2012 *Neural Networks: Tricks of the Trade* (Berlin: Springer) pp 437–78
[71] Munro W J, James D F, White A G and Kwiat P G 2001 *Phys. Rev.* A **64** 030302
[72] Wei T-C, Nemoto K, Goldbart P M, Kwiat P G, Munro W J and Verstraete F 2003 *Phys. Rev.* A **67** 022110
[73] Życzkowski K 1999 *Phys. Rev.* A **60** 3496
[74] Życzkowski K and Sommers H-J 2005 *Phys. Rev.* A **71** 032313
[75] Zyczkowski K and Sommers H-J 2001 *J. Phys. A: Math. Gen.* **34** 7111
[76] Ginibre J 1965 *J. Math. Phys.* **6** 440
[77] Al Osipov V, Sommers H-J and Życzkowski K 2010 *J. Phys. A: Math. Theor.* **43** 055302
[78] Lu H-H, Simmerman E M, Lougovski P, Weiner A M and Lukens J M 2020 *Phys. Rev. Lett.* **125** 120503
[79] Lingaraju N B, Lu H-H, Seshadri S, Leaird D E, Weiner A M and Lukens J M 2021 *Optica* **8** 329
[80] Alshowkan M *et al* 2021 *PRX Quantum* **2** 040304
[81] Gühne O and Tóth G 2009 *Phys. Rep.* **474** 1