Corrected: Publisher Correction

ARTICLE OPEN

Efficiently measuring a quantum device using machine learning

D. T. Lennon[1,5], H. Moon[1,5], L. C. Camenzind[2], Liuqi Yu[2], D. M. Zumbühl[2], G. A .D. Briggs[1], M. A. Osborne[3], E. A. Laird[4] and N. Ares[1*]

Scalable quantum technologies such as quantum computers will require very large numbers of quantum devices to be characterised and tuned. As the number of devices on chip increases, this task becomes ever more time-consuming, and will be intractable on a large scale without efficient automation. We present measurements on a quantum dot device performed by a machine learning algorithm in real time. The algorithm selects the most informative measurements to perform next by combining information theory with a probabilistic deep-generative model that can generate full-resolution reconstructions from scattered partial measurements. We demonstrate, for two different current map configurations that the algorithm outperforms standard grid scan techniques, reducing the number of measurements required by up to 4 times and the measurement time by 3.7 times. Our contribution goes beyond the use of machine learning for data search and analysis, and instead demonstrates the use of algorithms to automate measurements. This works lays the foundation for learning-based automated measurement of quantum devices.

npj Quantum Information (2019)5:79 ; https://doi.org/10.1038/s41534-019-0193-4

INTRODUCTION

Semiconductor quantum devices hold great promise for scalable quantum computation. In particular, individual electron spins in quantum dot devices have already shown long spin coherence times with respect to typical gate operation times, high fidelities, all-electrical control, and good prospects for scalability and integration with classical electronics.[1]

A crucial challenge of scaling spin qubits in quantum dots is that electrostatic confinement potentials vary strongly between devices and even in time, due to randomly fluctuating charge traps in the host material. Characterising such devices, which requires measurements of current or conductance at different applied biases and gate voltages, can be very time consuming. It is normally carried out following simple scripts such as grid scans, which are sequential measurements taken from a 2D grid for a pair of voltages. We call a set of voltages that defines the state of a quantum dot a configuration. Measurement of some configurations is more informative for characterising a quantum dot than the other configurations; measuring uncertain signals is more informative than measuring predictable signals. However, grid scans do not prioritise measurement of informative signals, instead just acquiring measurements according to simple rules (e.g. following a raster pattern). Current efforts in the field of automation of quantum dot are focused on tuning,[2–9] a large portion of these relying on grid scanning techniques for measurement. An optimised measurement method that can prioritise and select important configurations is thus key for fast characterisation and automatic tuning. Our method is thus complementary to automating tuning of quantum devices and holds the potential to increase the efficiency of these approaches when combined.

In this paper, we present an algorithm that performs efficient real-time data acquisition for a quantum dot device (Fig. 1a). It starts from a low-resolution uniform grid of measurements, creates a set of full-resolution reconstructions, calculates the predicted information gain (i.e. the acquisition map), selects the most informative measurements to perform next, and repeats this process until the information gain from new measurements is marginal.

In order to select measurements based on information theory, we require a corresponding uncertainty measure (of random variables),[10–12] and hence a probabilistic model of unobserved variables. One typical approach is to use a Gaussian process.[13] Here, we use a conditional variational auto-encoder (CVAE),[14] which is capable of generating high-resolution reconstructions given partial information and is fast enough for real-time decisions. Deep generative models such as adversarial networks (GAN),[15] the variational auto-encoder (VAE)[16] and its extensions, such as CVAE, have shown great success in multi-modal distributions and complex non-stationary patterns of data,[17,18] similar to those of observed in quantum device measurements. These are the main advantages of CVAE over a basic Gaussian process. Also, CVAE is more computationally efficient at generating multiple full-resolution reconstructions. Although progress has been made addressing the limitations of Gaussian processes, deep generative models are overall a better fit to the requirements for efficient quantum device measurements. Deep generative models have been used for: speech synthesis;[19] generating images of digits and human faces;[20,21] transferring image style;[22,23] and inpainting missing regions of images.[24] Recently, VAE models have been used in scientific research to optimise molecular structures.[25–28] In spite of their suitability, these models have not previously been applied to efficient data acquisition. An advantage of deep generative models over simple interpolation techniques, such as nearest-neighbour and bilinear interpolation, is that deep generative models can learn likely patterns from training data and incorporate them into its reconstructions. Our method, as it is data-driven, it is generalizable to different transport regimes, measurement configurations, and more complex device architectures if an appropriate training set is available.

[1]Department of Materials, University of Oxford, Parks Road, Oxford OX1 3PH, UK. [2]Department of Physics, University of Basel, 4056 Basel, Switzerland. [3]Department of Engineering, University of Oxford, Parks Road, Oxford OX1 3PJ, UK. [4]Department of Physics, Lancaster University, Lancaster LA1 4YB, UK. [5]These authors contributed equally: D. T. Lennon, H. Moon. *email: natalia.ares@materials.ox.ac.uk

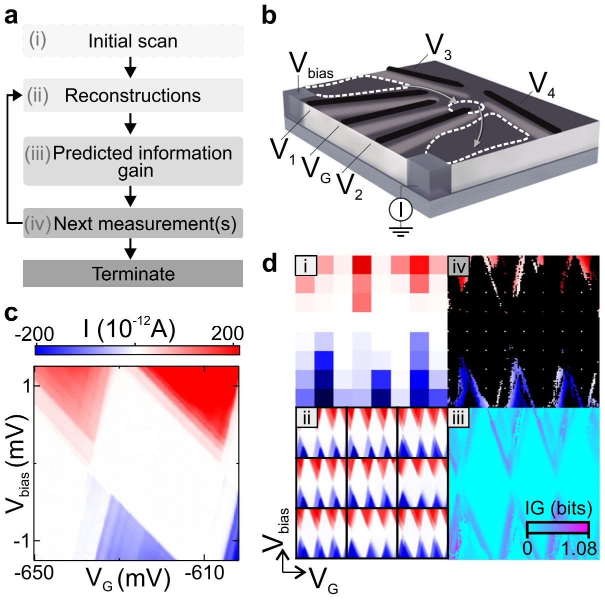

Fig. 1 Overview of the algorithm and the quantum dot device. **a** Schematic of the algorithm's operation. Low-resolution measurements (i) are used to produce reconstructions (ii), which are used to infer the predicted information gain acquisition map (iii). Based on this map, the algorithm chooses the location of the next measurement (iv). The process is repeated until a stopping criterion is met. **b** Schematic of the device. A bias voltage V_{bias} is applied between ohmic contacts to the two-dimensional electron gas. We apply gate voltages labelled V_1 to V_4 and V_{G}. **c** A measured current map as a function of V_{bias} and V_{G}. The Coulomb diamonds are the white regions where electron transport is suppressed, and most of the information necessary to characterise a device is contained just outside these diamonds. **d** Sequential decision algorithm in **a** illustrated with an example of a specific current map. In panel (iv), unmeasured pixels are plotted in black; however, initial measurements (i) are represented so as to fill the entire panel (that is, the sparse grid of measurements is represented as a low-resolution image)

RESULTS

The device

Our device is a laterally defined quantum dot fabricated by patterning Ti/Au gates over a GaAs/AlGaAs heterostructure containing a two-dimensional electron gas (Fig. 1b). In this device, electrons are subject to the confinement potential created electrostatically by gate voltages. Gate voltages V_1 to V_4 tune the tunneling rates while V_{G} mainly shifts the electrical potential inside the quantum dot. The current through the device is determined both by these gate voltages and by the bias voltage V_{bias}. Measurements were performed at 30 mK.

The quantum dot is characterised by acquiring maps of the electrical current as a function of a pair of varied voltages, which we call a current map configuration. We first focus on varying V_{G} and V_{bias} for fixed values of V_1 to V_4. Fig. 1c) shows a typical example. Diamond-shaped regions or 'Coulomb diamonds' indicate Coulomb blockade, where electron tunnelling is suppressed.[29] Most current maps have large areas in which the current is almost constant, and consequently measurements in these regions slow down informative data acquisition. Our algorithm must, therefore, give measurement priority to the informative regions of the map. An overview of an algorithm-assisted measurement of a current map is shown in Fig. 1d.

Training the reconstruction model

The role of the reconstruction model is to characterise likely patterns in a training data set, given by a mixture of measured and simulated current maps. We can utilise these likely patterns to predict the unmeasured signals from partial measurements.

Deep generative models represent this pattern characterisation in a low-dimensional real-valued latent vector $\mathbf{z}$, which can be decoded to produce a full-resolution reconstruction. The latent space representation and the decoding are learned during training. Our CVAE consists of two convolutional neural networks, an encoder and a decoder. The encoder is trained to map full-resolution training examples of current maps Y to the latent space representation $\mathbf{z}$. The encoder also enforces that the distribution $p(\mathbf{z})$ of training examples in latent space is Gaussian.

The decoder is trained to reconstruct Y, from the representation $\mathbf{z}$ combined with an 8×8 subsample of Y. As a result, $\mathbf{z}$ attempts to represent all the information that is missing from the subsampled data. In a plain VAE, the input of a decoder is only $\mathbf{z}$. If a decoder takes additional input except $\mathbf{z}$, then it is called CVAE, and we found that CVAE generates better reconstructions than VAE for the considered measurements. The chosen loss function, which the CVAE tries to minimise, is a measure of the difference between the training data and the corresponding reconstruction. To avoid blurry reconstructions, we define a contextual loss function that incorporates both pixel-by-pixel and higher-order differences like edges, corners, and shapes. Detailed description of these networks and their training can be found in the Supplementary sections Training and loss function, and Network specification.

The model is trained using both simulated and measured current maps. We choose to work with current maps of resolution 128×128. The simulation is based on a constant-interaction model (see Methods). To measure the current maps for training, we set the bias and gate voltage ranges randomly from a uniform distribution. The training data set consists of 25,000 simulations and 25,000 real examples generated by randomly cropping 750 measured current maps. The current maps were subjected to random offsets, rescaling, and added noise to increase the variability of the training set.

Generating reconstructions from partial data

After training, only the trained decoder network is used in the algorithm of Fig. 1a to reconstruct full-resolution current maps from partial data. At each stage, the known partial current map is denoted Y_n, where $n \leq 128^2 = 16,384$ is the number of pixels to be measured. To generate a reconstruction, the decoder takes as input the initial 8×8 grid scan Y_{64}, together with a latent vector $\mathbf{z}$ sampled from the posterior distribution $p(\mathbf{z}|Y_n)$ (see Methods for detail equations and Fig. S1 for the decoder diagram). Note that the posterior density is calculated by the prior density $p(\mathbf{z})$ and a likelihood function, which is comparing reconstructions and the partial data. Multiple posterior samples are drawn from $p(\mathbf{z}|Y_n)$ by the Metropolis–Hastings (MH) method to approximate $p(\mathbf{z}|Y_n)$. From these multiple samples $\mathbf{z}_m$, corresponding reconstructions are then generated, denoted $\hat{Y}_m$. In this paper we set $m = 1, \ldots, 100$. The continuous posterior $p(\mathbf{z}|Y_n)$ is then approximated by a discrete posterior of samples $P_n(m)$, which denotes how probable $\hat{Y}_m$ is. We refer to $P_n(m)$ as the posterior distribution of reconstructions.

Making measurement decisions

With each iteration of the decision algorithm, an acquisition map is computed from the accumulated partial measurements and the resulting reconstructions. This acquisition map assigns to each potential measurement location (i.e. to each pixel location in the current map) an information value for the posterior distribution of reconstructions (Fig. 2). The $(n+1)$th measurement, whose result

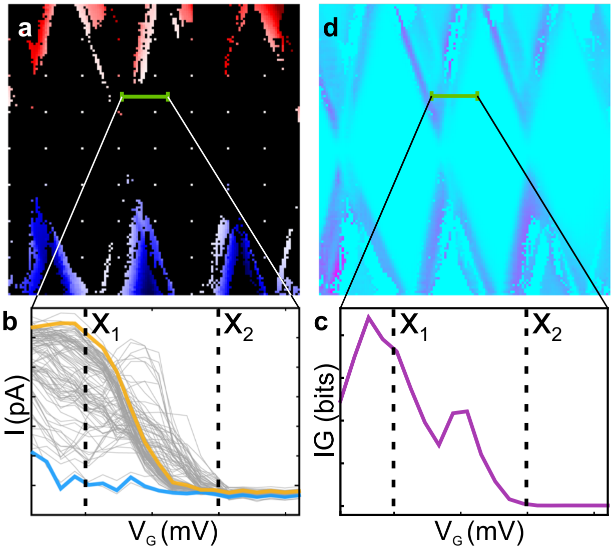

Fig. 2 Computing the acquisition map. **a** Partial current map. To illustrate the first step in the computation of the acquisition map, we consider a trace (green) through an unmeasured region of the map. **b** For the unmeasured trace in **a**, reconstructions provide 100 different predictions. Blue and yellow traces highlight two of these predictions. The objective is to determine the most informative measurement location. At x_2, all predictions are similar, so measuring here will have little impact on the posterior distribution of reconstructions. At x_1, predictions are dissimilar and, therefore, x_1 is a more informative measurement location, with a larger effect on the posterior distribution of reconstructions. **c** Information gain computed for the unmeasured trace in (**a**). **d** Acquisition map of information gain computed from the partial measurements in (**a**), and plotted over the entire current map range

is y_{n+1}, is one pixel taken from the true current map and changes our posterior distribution from $P_n(m)$ to $P_{n+1}(m)$, rendering different reconstructions more or less probable.

The acquisition map is the expected information gain $\mathrm{IG}(x)$ at each potential measurement location x. Our algorithm calculates it by a weighted sum over reconstructions:

$$\mathrm{IG}(x) \equiv \sum_m P_n(m) \times \mathrm{IG}_m(x), \tag{1}$$

where $\mathrm{IG}_m(x)$ is the Kullback–Leibler divergence between the distributions P_n and P_{n+1}, calculated such that y_{n+1} at location x is taken from reconstruction $\hat{Y}_m$. The most informative point is $x^*_{n+1} \equiv \mathrm{argmax}_x \mathrm{IG}(x)$. This criterion is equivalent both to minimising the expected information entropy of the posterior distribution and to Bayesian active learning by disagreement (BALD,[10] see Methods). The difference of the proposed method and BALD is that the proposed method uses random reconstructions of data, which can be multi-modal, whereas BALD assumes that data is normally distributed.

We devised two methods to make decisions based on the acquisition map; a pixel-wise method, and a batch method. The pixel-wise method selects the single best location in the acquisition map. In practice, this is often not optimal in terms of measurement time, because it does not take account of the time needed to ramp the gate voltage between measurement locations (which is limited by details of the measurement electronics and the device settling time). To take account of this limitation, we also devised a batch method, which selects multiple locations from the acquisition map, and then acquires measurements by taking a fast route between them. This reduces the measurement time compared with the pixel-wise method.

Experiments

To test the algorithm, it was used to acquire a series of current maps in real time. First, the device was thermally cycled, to randomise the charge traps and therefore present the algorithm with a configuration not represented in its training data. Gate voltages V_1-V_4 were set to a combination of values, and the algorithm was tasked to measure the corresponding current map using both the batch and the pixel-wise methods. This step was repeated for ten different combinations of bias and gate voltages. Fig. 3 presents data acquired by the algorithm at selected acquisition stages, together with selected reconstructions. As expected, reconstructions become less diverse as more measurements are acquired. The reconstructions do not necessarily replicate the measured current map for large n. This is because reconstructions have a limited variability given by the training data. Decisions are made based on the learned patterns from the training data, which implies that this training data should contain at least general patterns which are to be characterised but does not need to include all possible features in a current map.

As seen, the algorithm gives high priority to regions of the map where the current is rapidly varying, and avoids regions of nearly constant current, such as the interiors of Coulomb diamonds. This strategy is an emergent property of the algorithm and is wise; little information about the device characteristics can be found in low-current gradient regions of the current map. This preference derives from the comparison between reconstructions, which exhibit the greatest disagreement outside Coulomb diamonds. This is also seen in Fig. 4a, which shows two representative measurement sequences using the batch method. The batch method collects grouped measurements while the pixel-wise method distributes measurements more uniformly, given that in this case, the acquisition map is more frequently updated to take account for recently acquired information. Results for other current maps, including for the pixel-wise method, are shown in the Supplementary Figs. 2–6.

We compared the performance of the algorithm with an alternating grid scan method. This type of grid scan starts with 8 × 8 measurements and alternately increases the vertical and the horizontal grid size by 2 (i.e. 16 × 8, 16 × 16, 32 × 16, etc.), without performing the same measurement twice. Over the ten different current maps, the average time for full-resolution data acquisition with the alternating grid scan method is 554 seconds. This time is limited by our bias and gate voltage ramp rate and chosen settling time. The batch method can be implemented with any batch size however for direct comparison with the alternating grid scan we selected increasing batches of 32×2^b, where b is the batch number starting from 1.

Two types of computation are required to make a measurement decision: sampling reconstructions using the MH method, and constructing the acquisition map. One MH sampling iteration takes 63 ms. For experiments, multiple sampling iterations are performed after each batch decision and measurement while acquisition is suspended. Since sampling can be performed simultaneously with measurement acquisition, from now on our measurement times exclude the time for sampling. To compute a single acquisition map takes approximately 50 ms using a NVIDIA GTX 1080 Ti graphics card and Tensorflow[30] implementation. The acquisition map must be computed for every batch or every pixel measurement, except for the initial 8 × 8 grid scan and the final acquisition step (which has no choice of which pixel(s) to measure). To acquire a full resolution current map thus requires 7 computations (350 ms) for the batch method, and 16,319 computations (816 s) for the pixel-wise method. For the batch method, the computation time is negligible compared to the measurement time, but for the pixel-wise method it is a limiting factor in the measurement rate.

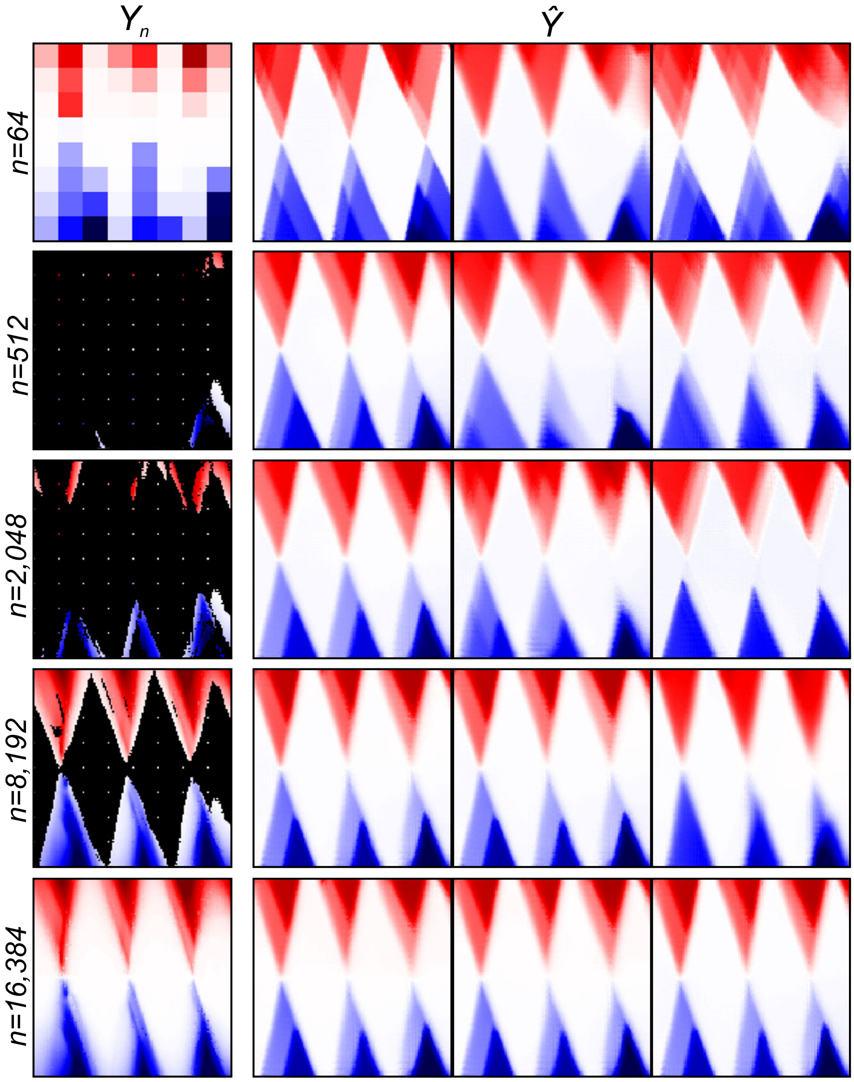

Fig. 3 Updating reconstructions using information from new measurements. In each row, the first column shows the algorithm-assisted measurements, using the batch method, for a given n. The remaining three columns contain example reconstructions given the corresponding n measurements. As n increases, the diversity of the reconstructions is reduced and their accuracy increased. As expected, the uncertainty is almost eliminated in the last row. The residual remaining variance is because slightly different reconstructions are nearly equally consistent with the data

To quantify the algorithm's performance, we have devised a measure based on the observation that the most informative regions of the current map are those where the current varies strongly with V_G and V_{bias}. We therefore define the local gradient of the current map at each location x as

$$v(x) \equiv \| \nabla Y(x) \|_2 = \sqrt{\left(\frac{\partial I(x)}{\partial V_G}\right)^2 + \left(\frac{\partial I(x)}{\partial V_{\text{bias}}}\right)^2}, \tag{2}$$

where $I(x)$ is current measurement at x, and the derivatives are calculated numerically. The error measure $r(n)$ of a partial current map is the fraction of the total gradient that remains uncaptured, i.e.

$$r(n) \equiv 1 - \frac{V(n)}{V(N)} \tag{3}$$

where $V(n) \equiv \sum_{i=1}^{n} v(x_i)$ is the total acquired gradient and x_i is the location of the ith measurement. This error can only be calculated after all measurements have been performed. However, we can utilise the mth reconstruction to generate an estimate $\tilde{r}_m(n)$ in real time by replacing $\|\nabla Y(x)\|_2$ with $\|\nabla \hat{Y}_m(x)\|_2$. The estimates from multiple reconstructions yield a credibility interval for $r(n)$. For an optimal algorithm, the error would be $\bar{r}(n) = 1.0 - \frac{V^*(n)}{V(N)}$, where $V^*(n)$ is the sum of the largest n values of $v(x)$. This would be achieved if each measurement location were chosen knowing the full-resolution current map, and thus the location of the the highest unmeasured current gradient. No decision method can exceed this bound. For the real time estimates of $r(n)$, we have increased the number of reconstructions to 3,000 by adding different noise patterns that are present in typical measured current maps (see Supplementary section Noisy reconstruction). This increase in the variability of the reconstructions is needed to avoid an overconfident estimation of $r(n)$.

Performances for two experiments are shown in Fig. 4c, e. Grid scans reduce $r(n)$ linearly with increasing n. The decision algorithm outperforms a simple grid scan and is nearly optimal. When most of the current gradient is localised, the grid scan is far from optimal and even the decision algorithm has room for improvement. In this case, the performance of the algorithm is determined by how representative the training data is.

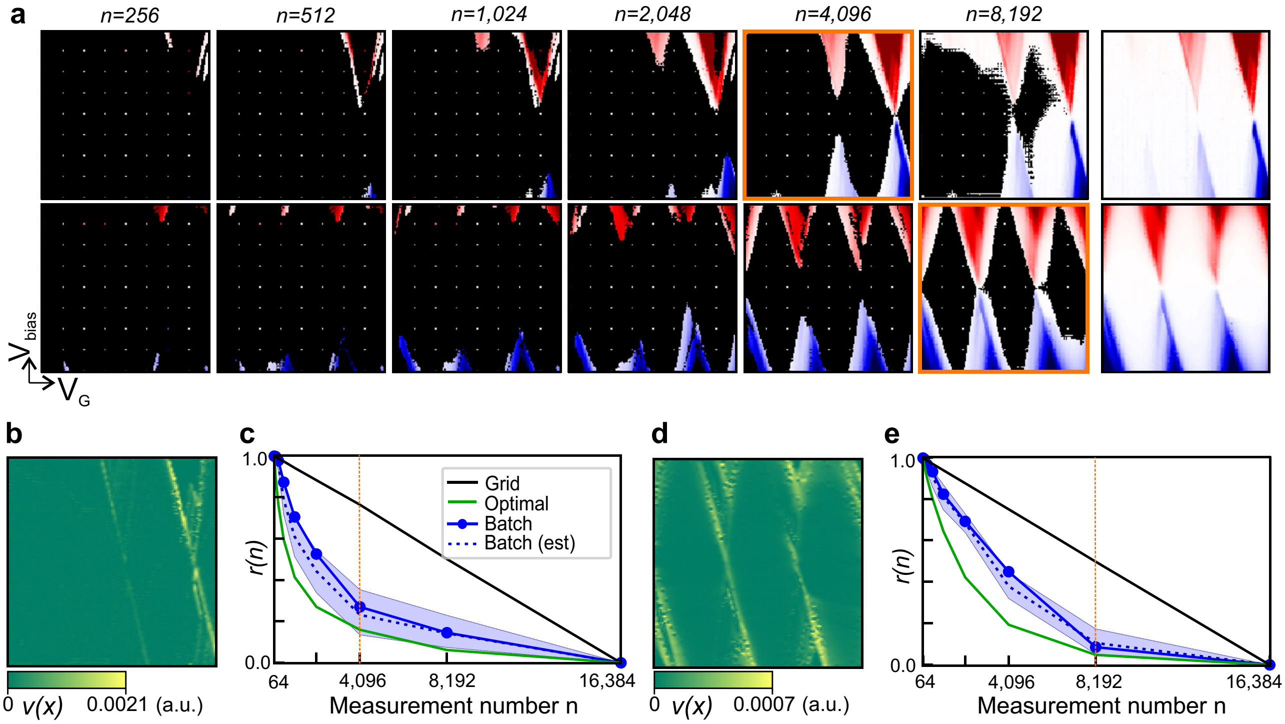

Fig. 4 Measurements of Coulomb diamonds performed by the algorithm. **a** Sequential batch measurement in two different experiments. Each row displays algorithm assisted measurements of the current map as a function of V_{bias} and V_G for different values of n. The last plot in each row is the full-resolution current map. **b**, **d** Current gradient map (defined by Eq. (2)) for each example in (**a**). **c**, **e** Measure of the algorithm's performance $r(n)$, real-time estimate of $r(n)$ across reconstructions (with 90% credible interval shaded), and optimal $r(n)$ for both examples in (**a**). The black line is the value of $r(n)$ corresponding to the alternating grid scan method. The vertical orange line indicates the value of n determined by the stopping criterion. The corresponding current map in **a** is highlighted in orange

Quantitative analysis of all 10 experiments is in Supplementary Figs. 5 and 6.

We propose a simple stopping criterion that uses the estimated reduction of the error $r(n)$ to determine when to stop measuring a given current map, in a scenario where more experiments are waiting to be conducted. For a given current map containing n measured pixels, the error after the next measurement batch is estimated for reconstruction m to be $\tilde{r}_m(n+\Delta)$, where Δ is the size of the batch. Thus the estimated rate at which the error decreases is $\beta_m \equiv |\tilde{r}_m(n+\Delta) - \tilde{r}_m(n)|/\Delta$. In the worst case among the candidate reconstructions, this rate is $\beta \equiv \min_m \beta_m$. However, if the algorithm begins to measure a new map, for which no reconstructions yet exist, the error of that map will decrease at a rate of at least $\alpha \equiv 1/N$; this is the slope achieved by a grid scan and the worst case of the decision algorithm (black lines in Fig. 4c, e). Hence when $\beta < \alpha$, it is beneficial to halt measurement and move onto a new current map that is awaiting measurement. Since α and β are the worst-case estimates for each case, the criterion is conservative. The stopping points by this criterion are shown in Fig. 4c, e, with orange dashed lines. The total average time (measurement time plus decision time) to reach the stopping criterion was 237 s, compared with 554 s to measure the complete current map by grid scan, reducing the time needed by a factor between 1.84 and 3.70 across all 10 test cases. A more sophisticated stopping criterion utilising the number of remaining unmeasured current maps and a total measurement budget is presented in Methods.

Generalising the algorithm

The algorithm described here does not require assumptions about the physics of the acquired data, such as requiring that it show Coulomb diamonds. Provided that training data are available, it should also work for other kinds of measurements. To test this, we applied it to a different current map configuration also encountered in quantum dot tuning. In this case the current flowing through the device is measured as a function of two gate voltages (V_1 and V_2), while keeping other voltages fixed (V_G, V_{bias}, V_3 and V_4). In these current maps, Coulomb blockade leads to large areas where the current scarcely changes, with diagonal features of allowed current. For the training set, we measured 382 current maps with a resolution of 251 × 251 which we randomly cropped to a resolution of 128 × 128 and subjected to simple image augmentation techniques (as for the previous training set).

We tested the performance of the algorithm in this new scenario by taking two different combinations of V_G, V_{bias}, V_3 and V_4 and measuring the corresponding current maps in real time (Fig. 5). The device was thermally cycled after the training set was acquired and also between the acquisition of the two current maps in Fig. 5. The algorithm focuses on measuring regions of high current gradient, the corner edges and, in particular, the Coulomb peaks close to these.

In the top row of Fig. 5a, $n = 4096$ was chosen by the stopping criterion. In the bottom row, the corners edges extended further in the current map and the stopping criterion chose $n = 8192$. This reduced the time needed to measure the current maps by 3.36 and 1.50, respectively, for the two test cases when compared with the alternating grid scans.

DISCUSSION

The proposed measurement algorithm makes real-time informed decisions on which measurements to perform next on a single quantum dot device. Decisions are based on the disagreement of competing reconstructions built from sparse measurements. The algorithm outperforms grid scan in all cases, and in the majority of cases shows nearly optimum performance. The algorithm reduced

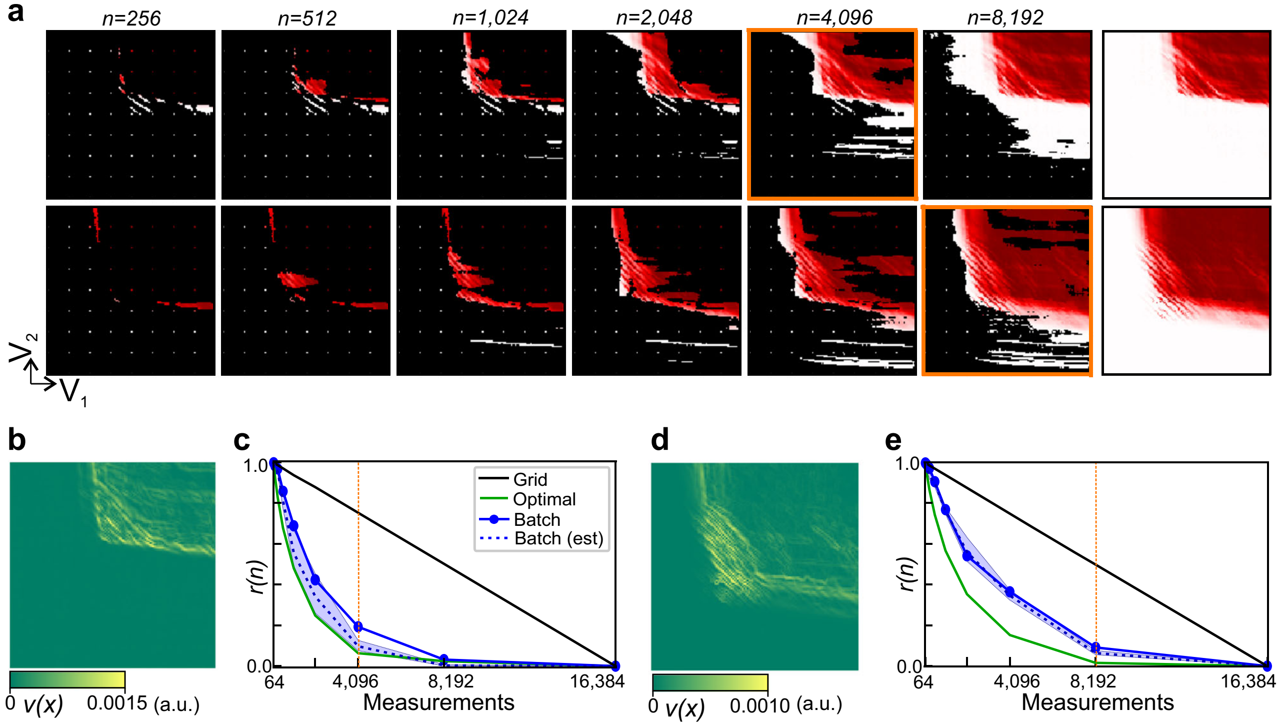

Fig. 5 Measuring a different current map. **a** Sequential batch measurement. Each row displays the algorithm-assisted measurements of a current map as a function of V_1 and V_2 for different values of n. The last plot in each row is the full-resolution current map. **b**, **d** Current gradient map for both examples in (**a**). **c**, **e** Measure of the algorithm's performance $r(n)$, average real-time estimate of $r(n)$ with 90% credible interval, and optimal $r(n)$ for both current maps in (**a**). The black line is the value of $r(n)$ corresponding to the alternating grid scan method. The dashed orange line indicates the value of n determined by the stopping criterion. The corresponding current map in **a** is highlighted in orange. The alternating grid scan took 2267 s and 2333 s to acquire all measurements in the two cases. The batch method took 673 s and 1552 s to reach the stopping criterion

the time required to observe the regions of finite current gradient by factors ranging from 1.5 to 3.7 times. Optimisation of batch sizes or a variable scan resolution might reduce this time further, however, the performance gain is limited by the spread of the information gain over the scan range. This is evidenced in both Fig. 4c, e and Fig. 5c, e, where we show that even an optimal algorithm does not significantly outperform the algorithm.

Our algorithm with no modifications can be re-trained to measure different current maps. It simply requires a diverse data set of training examples from which to learn. The decision algorithm performed well even when trained on a small data set of only 382 current maps (at a resolution of 251 × 251), implying that it is robust to limited training data sets. Our algorithm focused on observing all informative regions present in the current map, making it generalisable to different types of measurements and devices. The acquisition function can still be specifically designed to focus on specific transport features such as Coulomb peaks or Coulomb diamond edges. In additional experiments, we demonstrate how this can be achieved by applying additional transformations to the reconstructions (see Supplementary Section Context-aware decision for stability diagrams).

We believe that our algorithm represents a significant first step in automating what to measure next in quantum devices. For a single quantum dot it provides a means of accelerating what can currently be achieved by human experimenters and other automation methods. When provided with an appropriate training data set our algorithm can be applied to a large variety of experiments. In particular, in any conventional qubit tuning method for which time-consuming grid scans are performed, our algorithm would allow for an improvement in measurement efficiency. It will not be long before this kind of approach enables experiments to be performed, and technology to be developed, that would not be feasible otherwise.

METHODS

Distribution of reconstructions and sampling

Since it is known that deep generative models work well when the data range is from −1 to 1, all measurements are rescaled so that the maximum value of the absolute value of the initial measurement is 1. Let Y be a random vector containing all pixel values. Observation Y_n, where $n \geq 1$, is the set of pairs of location x_j and measurement y_j: $Y_n = \{(x_j, y_j) | j = 1, \ldots, n\}$. Also, a subset of measurements is defined: $Y_{n:n'} = \{(x_j, y_j) | j = n, \ldots, n'\}$. The likelihood of observations given Y is defined by

$$p(Y_n|Y) \propto \exp(-\lambda \Sigma_{(x,y)\in Y_n} |y - Y(x)|), \tag{4}$$

where $Y(x)$ is the pixel value of Y at x, and λ is a free parameter that determines the sensitivity to the distance metric and is set to 1.0 for all experiments in this paper. The posterior probability distribution is defined by Bayes' rule:

$$p(Y|Y_n) \propto p(Y_n|Y)\ p(Y). \tag{5}$$

Likewise, we can find the posterior distribution of $\mathbf{z}$ given measurements instead of Y. Let $\mathbf{z}'$ denote another input of the decoder, which is set to Y_{64} in the experiments. Then the posterior distribution of $\mathbf{z}$ can be expressed with $\mathbf{z}'$ when $n \geq 64$:

$$\begin{aligned} p(\mathbf{z}|Y_n, \mathbf{z}') &\propto p(\mathbf{z}|\mathbf{z}')\ p(Y_n|p(\mathbf{z}, \mathbf{z}') \\ &\propto p(\mathbf{z}) \textstyle\int_Y p(Y_n|Y)\ p(Y|p(\mathbf{z}, \mathbf{z}')\ dY \\ &\propto p(\mathbf{z})\ p(Y_n|Y = \hat{Y}_{\mathbf{z}}), \end{aligned}$$

where $\hat{Y}_{\mathbf{z}}$ is the reconstruction produced by the decoder given $\mathbf{z}$ and $\mathbf{z}'$. Since all inputs of the decoder are given, $p(Y|\mathbf{z}, \mathbf{z}')$ is the Dirac delta function centered at $\hat{Y}_{\mathbf{z}}$. Also, $p(\mathbf{z}|\mathbf{z}') = p(\mathbf{z})$ as $\mathbf{z}$ and $\mathbf{z}'$ are assumed independent. Proposal distribution for MH is set to a multivariate normal distribution having centered mean and a covariance matrix equal to one quarter of the identity matrix. For the experiments in this paper, 400 iterations of MCMC steps are conducted when $n = 32\times 2^b$, where b is any integer larger than or equal to 1. We found that 400 iterations result in good posterior samples. If (x_{n+1}, y_{n+1}) is newly observed, then the posterior can be updated incrementally:

$$\begin{aligned} p(\mathbf{z}|Y_{n+1}, \mathbf{z}') &= \tfrac{p(x_{n+1}, y_{y+1}|\mathbf{z}, \mathbf{z}')}{p(x_{n+1}, y_{n+1}|Y_n, \mathbf{z}')}\ p(\mathbf{z}|Y_n, \mathbf{z}') \\ &= \tfrac{p(x_{n+1}, y_{y+1}|\hat{Y}_{\mathbf{z}})}{p(x_{n+1}, y_{n+1}|Y_n, \mathbf{z}')}\ p(\mathbf{z}|Y_n, \mathbf{z}'), \end{aligned}$$

because each term in (4) can be separated.

Decision algorithm

In this section, we derive a computationally simple form of the information gain and the fact that maximising the information gain is equal to minimising the entropy. Let $p_n(\cdot) = p(\cdot|Y_n, \mathbf{z}')$, and any probabilistic quantity of y_{n+1} has the condition x_{n+1}, but omitted for brevity.

The continuous version of the information gain equation is

$$\begin{aligned} &\mathbb{E}_{y_{n+1}}[\mathbf{KL}(p_n(\mathbf{z}|y_{n+1}) \parallel p_n(\mathbf{z}))] \\ &\quad = \int_{y_{n+1}} p_n(y_{n+1})\mathbf{KL}(p_n(\mathbf{z}|y_{n+1}) \parallel p_n(\mathbf{z}))dy_{n+1} \\ &\quad = \int_{y_{n+1}} p_n(y_{n+1}) \int_{\mathbf{z}} p_n(\mathbf{z}|y_{n+1})\log\tfrac{\mathrm{p_n}(\mathbf{z}|\mathrm{y_{n+1}})}{\mathrm{p_n}(\mathbf{z})}\mathrm{d}\mathbf{z}\mathrm{dy_{n+1}} \\ &\quad = \int_{y_{n+1}} \int_{\mathbf{z}} p_n(\mathbf{z}, y_{n+1})\log\tfrac{\mathrm{p_n}(\mathbf{z},\mathrm{y_{n+1}})}{\mathrm{p_n}(\mathbf{z})\mathrm{p_n}(\mathrm{y_{n+1}})}\mathrm{d}\mathbf{z}\mathrm{dy_{n+1}} \\ &\quad = I(\mathbf{z}|Y_n\ ;\ y_{n+1}|Y_n), \end{aligned} \tag{6}$$

where **KL** is Kullback–Leibler divergence, $I(\cdot;\cdot)$ is mutual information. Since $I(\mathbf{z}|Y_n;\ y_{n+1}|Y_n) = \mathrm{H}(\mathbf{z}|Y_n) - \mathrm{H}(\mathbf{z}|Y_n, y_{n+1})$, maximising the expected **KL** divergence is equivalent to minimising $\mathrm{H}(\mathbf{z}|Y_n, y_{n+1})$, which is the entropy of $\mathbf{z}$ after observing y_{n+1}.

Since this integral is hard to compute, we approximate probability density functions (PDFs) with samples and substitute them into (6). Let n_s denote the number of measurements that are used for sampling reconstructions $\hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_M$ (the samples are converted to $\hat{Y}_1, \ldots, \hat{Y}_M$). Then $p_{n_s}(\mathbf{z}) \approx \frac{1}{M}\sum_m \delta_{\hat{\mathbf{z}}_m}(\mathbf{z})$, or with the sample index m, $P_{n_s}(m) = 1/M$. For any $n \geq n_s$, the probability is updated with the new measurements after n_s: $P_n(m; n_s) = \frac{p(Y_{n_s+1:n}|\hat{Y}_m)}{\Sigma_m p(Y_{n_s+1:n}|\hat{Y}_m)}$, which can be derived from importance sampling. For brevity, the sampling distribution information n_s is omitted for the remaining section. Likewise, $p_n(y_{n+1}) = \int_{\mathbf{z}} p_n(y_{n+1}|\mathbf{z})\ p_n(\mathbf{z}) \approx \sum_m P_n(m)$ $p_n(y_{n+1}|\mathbf{z}_m)$. Lastly, we use the value of $\hat{Y}_m$ at x_{n+1} for a sample of $p_n(y_{n+1}|\mathbf{z}_m)$ for simple and efficient computation. As a result, the information gain is approximated, up to a constant c, by:

$$\mathbb{E}_{y_{n+1}}[\mathbf{KL}(p_n(\mathbf{z}|y_{n+1})\ \parallel\ p_n(\mathbf{z}))] \approx \sum_m P_n(m)\ \mathbf{KL}(P_{n+1}\ \parallel\ P_n) + c.$$

Simulator for training data

To aid the training of the model simulated training data was used to prevent over-fitting. Simulated data produced via a simple implementation of the constant interaction model[29] was used along with basic data augmentation techniques. These techniques were not intended to be physically accurate but instead to produce quickly a diverse set of examples that contain features that mimic real data.

The constant interaction model makes the assumptions that all interactions felt by a confined electrons within the dot can be captured by a simple constant capacitance C_Σ which is given by $C_\Sigma = C_\mathrm{S} + C_\mathrm{D} + C_\mathrm{G}$ where C_S, C_D, and C_G are capacitances to the source, drain and gate respectively. Making this assumption the total energy of the dot $U(N)$ where N is the number of electrons occupying the dot, is $U(N) = \frac{(-|e|(N-N_0) + C_\mathrm{S}V_\mathrm{S} + C_\mathrm{D}V_\mathrm{D} + C_\mathrm{G}V_\mathrm{G})^2}{2C_\Sigma} + \sum_{n=1}^{N} E_n$ where N_0 compensates for the background charge and E_n is a term that represents occupied single electron energy levels that is characterised by the confinement potential.

Using this we derive the electrochemical potential $\mu(N) = U(N) - U(N-1) = \frac{e^2}{C_\Sigma}(N - N_0 - \frac{1}{2}) - \frac{|e|}{C_\Sigma}(V_\mathrm{S}C_\mathrm{S} + V_\mathrm{D}C_\mathrm{D} + V_\mathrm{G}C_\mathrm{G}) + E_n$.

To produce a training example random values are generated for C_S, C_D, and C_G. The energy levels within a randomly generated gate voltage window and source drain bias window are then counted. To aid generalisation to real data we randomly generated energy level transitions (which are also counted) as well as slightly linearly scaled C_Σ, C_S, C_D, and C_G with N. This linear scaling was also randomly generated and results in produced diamonds that vary in size with respect to V_G. Examples of the training data produced by this simulator can be seen in Supplementary Fig. 1.

Stopping criterion

Utility, denoted by u, is the ratio of total measured gradient to the total gradient of a stability diagram: $u(n) = 1.0 - r(n)$. Here, we assume that we have K more stability diagrams to be measured. The location of each diagram is defined by a different voltage range, and $k = 0, \ldots, K$ is the index of the diagrams, where $k = 0$ is the index of the diagram that we are currently measuring.

Let T denote the total measurement budget for the current and remaining stability diagram. In this paper we assume that a unit budget for measuring one pixel is 1.0. The total utility is

$$u_\mathrm{tot} = \sum_{k=0}^{K} u_k(t_k) = u_0(t_0) + u_\mathrm{nxt}(T - t_0),$$

where $u_k(\cdot)$ is the utility from measuring kth diagram, t_k is the planned budget for kth diagram satisfying $\sum_{k=0}^{K} t_k = T$, and $u_\mathrm{nxt}(T - t_0) = \sum_{k=1}^{K} u_k(t_k)$.

Let t denote the already spent budget on the current diagram, $t \leq t_0$. If we stop the measurement then $t_0 = t$, or $t_0 = t + \Delta$ if we decide to continue the measurement, where Δ is a predefined batch size. For the decision, the utilities of two cases are compared: when $t_0 = t$,

$$u_\mathrm{tot} = u_0(t) + u_\mathrm{nxt}(T - t). \tag{7}$$

Otherwise, $t_0 = t + \Delta$ and

$$u_\mathrm{tot} = u_0(t + \Delta) + u_\mathrm{nxt}(T - (t + \Delta)). \tag{8}$$

If (8) < (7), it is better to stop and move to the next voltage range. Rearranging the inequality leads to

$$u_0(t + \Delta) - u_0(t) < u_\mathrm{nxt}(T - t) - u_\mathrm{nxt}(T - (t + \Delta))x. \tag{9}$$

The left-hand-side (lhs) of (9) means the difference of utility if we invest Δ budget more on the current diagram, and the right-hand-side the difference when Δ more budget is used for remaining diagrams. As we discussed in Results section, we can calculate multiple slope estimates β_m for spending Δ to the current diagram: $u_0(t + \Delta) - u_0(t) \approx \beta_m\Delta$.

The right-hand-side (rhs) of (9) can be approximated by $\alpha\Delta$ if $K = \infty$, where $\alpha = 1/16,384$ is the slope of grid scan measuring a new stability diagram. Note that α can be considered as the empirical worst case performance of the decision algorithm measuring a new diagram as it holds for all the experiments we have conducted. If $\Delta = N$, this approximation is the exact quantity for any algorithms as all algorithms satisfy $r(0) = 1.0$ and $r(N) = 0.0$. Since α can be interpreted as the worst case estimate, we also approximate lhs of (9) with the worst case estimate $\beta = min_m\, \beta_m$.

If $K < \infty$, and the remaining budget $T - t$ is more than the budget to measure all of remaining diagrams, there is no utility after all measurements are finished. Hence, the approximation is capped:

$$u_\mathrm{nxt}(T - t) = \alpha \min(T - t, N \times K), \tag{10}$$

where K is the number of remaining diagrams to be measured.

As a result, the stopping criterion when $K = \infty$ is

$$\beta < \alpha. \tag{11}$$

The stopping criterion when $K < \infty$ is

$$\beta < \frac{\alpha(\min(T - t, N \times K) - \min(T - (t + \Delta), N \times K))}{\Delta}. \tag{12}$$

The rhs of (12) is always less than or equal to α, and more total budget T makes it low, which leads to late stopping or no stopping.

CODE AVAILABILITY

A documented implementation of the algorithm in a github repository is available at https://doi.org/10.5281/zenodo.2537934.

DATA AVAILABILITY

The data sets used for the training of the model are available from the corresponding author upon reasonable request.

Received: 11 February 2019; Accepted: 22 August 2019;
Published online: 26 September 2019

REFERENCES

1. Vandersypen, L. M. K. et al. Interfacing spin qubits in quantum dots and donorshot, dense, and coherent. *npj Quantum Info*. **3**, 34 (2017).
2. Baart, T. A., Eendebak, P. T., Reichl, C., Wegscheider, W. & Vandersypen, L. M. K. Computer-automated tuning of semiconductor double quantum dots into the single-electron regime. *Appl. Phys. Lett.* **108**, 213104 (2016).
3. Ito, T. et al. Detection and control of charge states in a quintuple quantum dot. *Sci. Rep*. **6**, 4–6 (2016).
4. VanDiepen, C. J. et al. Automated tuning of inter-dot tunnel coupling in double quantum dots. *Appl. Phys. Lett*. **113**, 033101 (2018).
5. Volk, C. et al. Loading a quantum-dot based "Qubyte" register. *npj Quantum Information*. **5** (2019):29.
6. Stehlik, J. et al. Fast charge sensing of a cavity-coupled double quantum dot using a Josephson parametric amplifier. *Phys. Rev. Appl*. **4**, 1–10 (2015).
7. Kalantre, S. S., et al. Machine Learning techniques for state recognition and autotuning in quantum dots. *npj Quantum Information*. **5** (2019):6.
8. Frees, A. et al. Compressed optimization of device architectures for semiconductor quantum devices. *Phys. Rev. Appl*. **11**, 1 (2019).
9. Teske, J. D. et al. A machine learning approach for automated fine-tuning of semiconductor spin qubits. *Appl. Phys. Lett*. **114**, 1–8 (2019).
10. Houlsby, N., Huszár, F., Ghahramani, Z. & Lengyel, M. Bayesian active learning for classification and preference learning. Preprint at https://arxiv.org/abs/1112.5745 (2011).
11. Ankenman, B., Nelson, B. L. & Staum, J. Stochastic kriging for simulation metamodeling. *Operations Res*. **58**, 371–382 (2010).
12. Sacks, J., Welch, W. J., Mitchell, T. J. & Wynn, H. P. Design and analysis of computer experiments. *Stat. Sci*. **4**, 409–423 (1989).
13. Rasmussen, C. E. & Williams, C. K. I. *Gaussian Processes for Machine Learning*. (The MIT Press, Cambridge, US, 2006).
14. Sohn, K., Lee, H. & Yan X. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems* 28. 3483–3491 (Curran Associates, Montreal, Canada, 2015).
15. Goodfellow, I. et al. Generative adversarial nets. In *Advances in Neural Information Processing Systems* 27. 2672–2680 (Curran Associates, Inc., Montreal, Canada, 2014).
16. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations*. (ICLR, Banff, Canada, 2014).
17. Mescheder, L., Nowozin, S. & Geiger, A. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*. 2391–2400 (PMLR, Sydney, Austrailia, 2017).
18. Srivastava, A., Valkov, L., Russell, C., Gutmann, M. & Sutton, C. VEEGAN: Reducing mode collapse in GANs using implicit variational learning. In *Advances in Neural Information Processing Systems* 30, 3308–3318 (Curran Associates, Inc., Montreal, Canada, 2017).
19. vandenOord, A. et al. WaveNet: A generative model for raw audio Preprint at http://arxiv.org/abs/1609.03499 (2016).
20. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I. & Frey, B. Adversarial autoencoders. Preprint at http://arxiv.org/abs/1511.05644 (2015).
21. Huang, H., Li, Z., He, R., Sun, Z., & Tan, T. IntroVAE: Introspective Variational Autoencoders for Photographic Image Synthesis. In *Advances in Neural Information Processing Systems* 31. 52–63 (Curran Associates, Montreal, Canada, 2018).
22. Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*. 2242–2251 (IEEE, Venice, Italy, 2017).
23. Taigman, Y., Polyak, A. & Wolf, L. Unsupervised cross-domain image generation. In *5th International Conference on Learning Representations* (ICLR, Toulon, France, 2017).
24. Iizuka, S., Simo-Serra, E. & Ishikawa, H. Globally and locally consistent image completion. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)* **36**, 107:1–107:14 (2017).
25. Sanchez-Lengeling, B. & Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **361**, 360–365 (2018).
26. Gómez-Bombarelli, R. et al. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci*. **4**, 268–276 (2018).
27. Kusner, M. J., Paige, B. & Hernández-Lobato, J.M. Grammar variational autoencoder. In *Proceedings of the 34th International Conference on Machine Learning*. 1945–1954 (PMLR, Sydney, Austrailia, 2017).
28. Dai, H., Tian, Y., Dai, B., Skiena, S. & Song, L. Syntax-directed variational autoencoder for structured data. In *6th International Conference on Learning Representations* (ICLR, Vancouver, Canada, 2018).
29. Hanson, R., Kouwenhoven, L. P., Petta, J. R., Tarucha, S. & Vandersypen, L. M. Spins in few-electron quantum dots. *Rev. Mod. Phys*. **79**, 1217–1265 (2007).
30. Abadi, M. et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265–283 (USENIX, Savannah, USA, 2016).

ACKNOWLEDGEMENTS

We acknowledge discussions with J.A. Mol and S.C. Benjamin. This work was supported by the EPSRC National Quantum Technology Hub in Networked Quantum Information Technology (EP/M013243/1), Quantum Technology Capital (EP/N014995/1), Nokia, Lockheed Martin, the Swiss NSF Project 179024, the Swiss Nanoscience Institute, NCCR QSIT and the EU H2020 European Microkelvin Platform EMP under grant No. 824109. This publication was also made possible through support from Templeton World Charity Foundation and John Templeton Foundation. The opinions expressed in this publication are those of the authors and do not necessarily reflect the views of the Templeton Foundations. We acknowledge J. Zimmerman and A.C. Gossard for the growth of the AlGaAs/GaAs heterostructure.

AUTHOR CONTRIBUTIONS

D.T.L. and N.A. and the machine performed the experiment. H.M. developed the algorithm under the supervision of M.A.O. The sample was fabricated by L.C.C., L.Y. and D.M.Z. The project was conceived by G.A.D.B., E.A.L., M.A.O. and N.A. All authors contributed to the manuscript and commented and discussed results.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information is available for this paper at (https://doi.org/10.1038/s41534-019-0193-4).

Correspondence and requests for materials should be addressed to N.A.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.