New Journal of Physics

The open access journal at the forefront of physics

PAPER

CrossMark

Quantum device fine-tuning using unsupervised embedding learning

N M van Esbroeck[1,2], D T Lennon[1] (iD), H Moon[1] (iD), V Nguyen[1] (iD), F Vigneau[1] (iD),
L C Camenzind[3] (iD), L Yu[3], D M Zumbühl[3] (iD), G A D Briggs[1] (iD), D Sejdinovic[4] (iD) and
N Ares[1,*] (iD)

[1] Department of Materials, University of Oxford, Parks Road, Oxford OX1 3PH, United Kingdom
[2] Department of Applied Physics, Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, The Netherlands
[3] Department of Physics, University of Basel, 4056 Basel, Switzerland
[4] Department of Statistics, University of Oxford, 24-29 St Giles, Oxford OX1 3LB, United Kingdom
[*] Author to whom any correspondence should be addressed.

E-mail: natalia.ares@materials.ox.ac.uk

Abstract

Quantum devices with a large number of gate electrodes allow for precise control of device parameters. This capability is hard to fully exploit due to the complex dependence of these parameters on applied gate voltages. We experimentally demonstrate an algorithm capable of fine-tuning several device parameters at once. The algorithm acquires a measurement and assigns it a score using a variational auto-encoder. Gate voltage settings are set to optimize this score in real-time in an unsupervised fashion. We report fine-tuning times of a double quantum dot device within approximately 40 min.

1. Introduction

Electrostatically defined semiconductor quantum dots are intensively studied for solid-state quantum computation [1–4]. Gate electrodes in these device architectures are designed to separately control electrochemical potentials and tunnel barriers [5, 6]. However, these device parameters vary non-monotonically and not always predictably with applied gate voltages, making device tuning a complex and time consuming task. Fully automated device tuning will be essential for the scalability of semiconductor qubit circuits.

Tuning of electrostatically defined quantum dot devices can be divided into three stages. The first stage is the ultra coarse tuning which consists of setting gate voltages to create the confinement potential for electrons or holes. The second stage, known as coarse tuning, focuses on identifying and navigating different operating regimes of a quantum dot device. The third stage, referred to as fine-tuning, involves optimizing a particular set of charge transitions. Full automation of the first tuning stage has been recently achieved [7]. Automated coarse tuning has been demonstrated using convolutional neural networks to identify the double quantum dot regime [8] and reach arbitrary charge states [9]. Template matching was also used to navigate to the single-electron regime [10]. During this stage, virtual gate electrodes can be used to independently control the electrochemical potential of each quantum dot [11, 12]. Previous work on automated fine-tuning focussed on achieving a target value for the tunnel coupling between two quantum dots by systematically modifying gate voltages [13, 14]. However, these approaches only allow for the optimization of a device parameter easily estimated from the performed measurements and rely on calibration.

Here, we demonstrate an automated approach for simultaneous fine-tuning of multiple device parameters, such as tunnel rates and inter-dot tunnel coupling, without requiring parametrization of the desired measurement features. Our approach is based on a variational auto-encoder (VAE). The VAE

Figure 1. Overview of the quantum dot device and algorithm. (a) Scanning electron microscopy image of a device lithographically identical to the one measured. A bias voltage V_{bias} is applied between two ohmic contacts to drive a current I through the device. Gate voltages V_1 to V_8 define and control the double quantum dot. (b) Current as a function of gate voltages V_3 and V_7, with $V_{bias} = 0.2$ mV. In this stability diagram, at the cross-points of the hexagonal lattice representative of the double quantum dot regime, bias triangles are observed. The zoom in shows a pair of bias triangles before (bottom) and after (top) running our fine tuning algorithm. Bias triangles are much better defined after fine tuning. (c) Schematic overview of the fine-tuning algorithm. A set of gate voltages is fixed by the algorithm. The algorithm then takes a low resolution stability diagram (LRSD), defines a measurement window centered in the observed bias triangles, and takes a high resolution stability diagram (HRSD). This diagram is fed to the VAE, which assigns it a score based on target locations in the latent space identified during training. A new set of gate voltages is fixed according to a decision model and the algorithm restarts. The VAE is trained once (indicated with a dotted red arrow), stage at which target stability diagrams are identified. In every iteration, their location in latent space is used by the VAE.

compresses training data sets to a lower-dimensional space, called the latent or embedding space. In this latent space, locations corresponding to measurements with desirable characteristics (according to human experts) are identified. After training, the algorithm takes a measurement and assigns it a location in latent space. The distance between this location and the target locations is used by the algorithm as a basis to score the measurement. By optimizing this score, device parameters are fine-tuned in real time. We have previously shown that VAEs significantly improve the efficiency of quantum dot measurements [15].VAEs are able to extract very high-level contextual information from data. For example, from pictures of human faces, a VAE model can detect a hairstyle, facial expressions, etc [16]. That high-level contextual information is encoded in the non-linear relationship of several pixels and it is thus extremely difficult to extract it using conventional methods like pixel-wise difference, linear embeddings (e.g. principle component analysis), or other hand-crafted functions. This is the case for the transport features we aim to optimize during fine tuning. We have now, for the first time, used a VAE to fine tune a double quantum dot device by locally optimizing transport features in a completely automated manner. Without requiring any prior knowledge of the device architecture, we are able to fine-tune several device parameters at once.

2. Fine-tuning double quantum dot devices

The device we focus on is a double quantum dot defined in a two-dimensional electron gas (2DEG) [5, 6] at the interface of a GaAs/AlGaAs heterostructure. Ti/Au gate electrodes are patterned on top of the heterostructure (figure 1(a)). DC voltages V_1 to V_8 are applied to these gate electrodes, to define and control the double dot confinement potential. A bias voltage V_{bias} determines the flow of current I through the device. Stability diagrams, displaying I as a function of two gate voltages, allow us to characterize charge transport in the device. A charge stability diagram for our double quantum dot device after coarse tuning is shown in figure figure 1(b). All measurements were performed at approximately 20 mK.

In the double quantum dot regime, charge stability diagrams display pairs of bias triangles, which reveal device parameters such as charging energies, tunnel coupling to the electron reservoirs and interdot tunnel coupling [17]. The shape, sharpness and brightness of these bias triangles are the features guiding humans when fine-tuning device parameters. Humans modify gate voltages until features they identify as favourable, according to their experience, are observed. Fine-tuning a double quantum dot is a very time-consuming task and thus automation is required to scale this technology. Our algorithm modifies gate voltages to achieve various bias triangle characteristics commonly associated with favourable device parameters, as done by humans when tuning these devices.

3. Overview of the algorithm

A schematic of our algorithm is shown in figure 1(c). The device is automatically coarse tuned to the double quantum dot regime [7] and a pair of bias triangles is identified. In each iteration a set of gate voltages N_j,

$\{V_1^j, V_2^j, \ldots, V_8^j\}$ is evaluated. After acquiring an initial LRSD (30 × 30 pixels, 18.4 × 18.4 mV), the bias triangles are centred using Laplacian of Gaussian blob detection. In computer vision, blob detection techniques aim to detect bright regions on dark backgrounds or vice versa [18]. In figure 1(b), the two pairs of bias triangles displayed were centred with this approach. Next, a HRSD of the centered bias triangles is measured (32 × 32 pixels, 17 × 17 mV). The HRSD is then assigned a location in latent space by the VAE. We use a distance metric to score the measurement by comparing this latent space location to the latent space location of a set of targets chosen by a human expert during VAE training. Based on the score value S_i, the algorithm sets the gate voltage configuration for the next stability diagram measurement N_{j+1}. This process is iterated until the bias triangles measured are as close as possible in latent space to the target measurements.

4. VAE implementation

The VAE is a convenient approach to evaluate a combination of features observed in an image, in this case a stability diagram that is costly to measure (approximately 1 minute), with the aim of creating a score metric to guide the choice of gate voltages. The VAE ensures that only the most important features in the stability diagram are encoded in the latent space.

The VAE consists of an encoder and decoder, which are probability distributions whose distribution parameters are computed by neural networks [19]. The encoder $q_\phi(z|x)$ maps input data x to a low-dimensional latent vector z which is real-valued. The decoder $p_\theta(x|z)$ maps a latent vector to a reconstruction $\hat{x}$. The parameters of the encoder and the decoder neural networks are ϕ and θ, respectively. The VAE is a generative model; it seeks to preserve the maximum amount of information during the encoding process so that input data can be reconstructed with minimal error during the decoding process. During a training phase, ϕ and θ are iteratively updated to minimize a loss function. The loss function is given by a reconstruction error $\mathcal{L}_{rec}$, which penalizes the networks for producing reconstructions that are dissimilar from the input data, and a regularization term $\mathcal{L}_{reg}$, which enforces input data with similar characteristics to be encoded in close proximity in latent space. The reconstruction error and the regularization term have weights α and β, respectively. Mathematical definitions of $\mathcal{L}_{rec}$ and $\mathcal{L}_{reg}$ are provided in the supplementary material (https://stacks.iop.org/NJP/00/000000/mmedia).

We implement Factor-VAE [20], an adaption of VAE that seeks to generate a latent space in which each dimension corresponds to a unique characteristic of the input data. The Factor-VAE framework assumes that there are underlying independent factors associated with the data. If fully disentangled, each of those factors can be identified with a dimension in latent space. By using a Factor-VAE, we aim to generate a latent space in which each dimension is associated with a single bias triangle characteristic, such as size or brightness. In this way, the distance in latent space to a target location results in a good metric to score acquired measurements.

The loss function of Factor-VAE includes a total correlation term which encourages the distribution of embeddings $q_\phi(z)$ to be disentangled. It is given by:

$$\mathcal{L}_{\text{Factor}-\text{VAE}} = \mathcal{L}_{rec} + \mathcal{L}_{reg} + \gamma D_{\text{KL}}\left(q_\phi(z) \,\middle|\middle|\, \prod_j q_\phi(z_j)\right) \tag{1}$$

where the total correlation term is given by $D_{\text{KL}}\left(q_\phi(z) \,||\, \prod_j q_\phi(z_j)\right)$, i.e. the Kullback–Leibler divergence between the distribution of embeddings $q_\phi(z)$ and the product of the distribution of embedding components $\prod_j q_\phi(z_j)$, with the index j corresponding to the jth latent space dimension. Note that the distribution of embeddings is given by $q_\phi(z)$, whereas $q_\phi(z_j)$ describes the distribution of a single embedding dimension z_j. The total correlation loss term has a weight γ. Since this term is intractable, it is estimated using a discriminator $D(z)$. The discriminator is trained to classify between non-factorial and factorial samples, i.e. that its input is a sample from $q_\phi(z)$ rather than from $\prod_j q_\phi(z_j)$. The loss function of the discriminator can be found in the supplementary material.

The training set for the Factor-VAE was collected from a device which differs considerably in material, architecture and transport regime from the device used to demonstrate the performance of the algorithm, evidencing its generality. The VAE was trained using 2253 sets of bias triangles, measured on a double quantum dot defined in a Ge/Si core−shell nanowire [21, 22]. In order to increase the robustness of the VAE, simple data augmentation techniques were applied. Data augmentation included rotation, mirroring, Gaussian noise and random contrast, resulting in a total training set of 8732 stability diagrams of pixel resolution 32 × 32. The dimension of the latent space was set to 10, as in reference [20], given the similar structure of input data. We tried multiple combinations of weights α, β and γ to achieve the optimal VAE

Figure 2. Schematic overview of the VAE. The VAE consists of an encoder and decoder. The encoder $q_\phi(z|x)$ compresses input stability diagrams to a lower-dimensional latent space. The decoder is denoted by $p_\theta(x|z)$ and maps vectors in latent space z to the distribution of input data. In this way, the input vector x, for which each element is the brightness of one pixel in the stability diagram, is transformed into a reconstruction vector $\hat{x}$. In order to visualise the ten-dimensional latent space, t-distributed Stochastic Neighbour Embedding is applied for dimensionality reduction [23]. The resultant two-dimensional latent space is described by a vector w. Each dot represents the embedding of an input stability diagram. The embedding location of one of the target inputs is highlighted in red. It is expected that embeddings which are close to each other in latent space are generated by input data with similar characteristics. Test example 1, with similar characteristics to the target, can be found in close proximity to the target, whereas test example 2 is further away in latent space. The target and example stability diagrams are plotted as a function of V_3 and V_7 and use a color scale running from red, the highest current measured, to blue, the lowest current.

performance, which was found empirically for $\alpha = 34$, $\beta = 1$ and $\gamma = 1$. The full architectures of the VAE and discriminator neural networks are presented in the supplementary material.

5. Score metric

The score metric used by the algorithm is given by the distance between the latent space representation z of an input stability diagram and the latent space representation $\{\tilde{z}\}$ of a set of target inputs. Note that the loss function is used during training, while the score metric is used to evaluate each new measurement. A measurement acquired by the algorithm is assigned a low (high) VAE score if its representation in latent space is near to (far from) the targets in latent space. Embeddings that are close together in latent space have similar z, implying that the original inputs can be generated using similar underlying variables. As a result, bias triangles that are assigned a low score possess similar characteristics to the target bias triangles. The target bias triangles are chosen from the unaugmented training set by a human expert who recognizes in these triangles the characteristics indicative of favourable quantum dot parameters. The targets are augmented using the same augmentation techniques as described in section 4. Augmentation of 30 selected targets resulted in a total target set of size 360.

In figure 2 the latent space of the test set of the trained VAE is shown, and the embedding locations of example target and training inputs are indicated. A full plot of the latent space of the trained VAE with original input stability diagrams is shown in the supplementary material.

To write the expression for the score S_i, where i denotes the ith input measurement, we use the latent vector z^i produced by the encoder for this measurement. The output of the encoder is assumed to follow a multivariate Gaussian with diagonal covariance structure: $q_\phi(z|x) = \mathcal{N}(z; \mu, \text{diag}(\sigma^2))$, where the mean μ and variance σ^2 are outputs of the encoding network. Considering two independent normal distributions in latent space, the expectation value of the squared distance between the distributions is given by:

$$d(z^1, z^2) = \mathbb{E}\left(\|z^1 - z^2\|^2\right) = \|\mu_{z^1} - \mu_{z^2}\|^2 + \sigma_{z^1}^2 + \sigma_{z^2}^2. \tag{2}$$

Figure 3. Overview of the decision model. In this figure, the node N_j represents the best scored gate voltage configuration obtained after a number of algorithm iterations. The branched arrows represent the different gate voltage adjustment options, which are changes of $\pm\Delta V$ in every gate electrode to be tuned. In this case, we consider 5 gate voltage branches since we focus on 3 gate electrodes for fine-tuning. (a) Score S_i corresponding to a new configuration N_{j+1} is lower than at N_j, so the gate voltage change is accepted. For N_{j+1}, a new random gate voltage branch is selected and explored. (b) Score S_i corresponding to a new configuration N_{j+1} is higher than at N_j, so the gate voltage change is rejected. For N_j, one of the remaining gate voltage branches is randomly selected and explored. (c) If all possible gate voltage configurations are rejected the algorithm returns to the closest previously accepted gate voltage node that has unexplored branches. At this configuration, a gate voltage branch is randomly selected and explored.

For each input measurement embedded in latent space z^i, (2) is used to determine the distance in latent space to target input $\tilde{z}^j$. The final score S_i consists of the average of the distance to its k nearest targets:

$$S_i = \frac{1}{k}\sum_{\tilde{z}^j \in A_k}^{k} d\left(\tilde{z}^j, z^i\right) \tag{3}$$

where A_k is the set of k targets closest to z^i in latent space. In this way, optimal tuning corresponds to a low score. We found that for $k = 3$ the score metric produced a robust ranking of the training inputs in terms of their similarity to the targets.

6. Decision model

The decision model adjusts the gate voltage settings, so that this new gate voltage configuration can be measured and scored. The decision model is illustrated in figure 3. In each iteration, one gate electrode is selected at random, and the voltage applied to this electrode is modified by a fixed amount $\pm\Delta V$. Therefore, the algorithm chooses between a number of gate voltage branches equal to twice the number of gate electrodes to be tuned minus one, as branches that lead to the reversal of the latest accepted gate voltage change are excluded. We chose $\Delta V = 2$ mV based on human experience in tuning similar devices. After centering and acquiring a high resolution measurement of the resulting bias triangles, the value of S_i determines the algorithm's decision. If S_i is lower than the previously best (lowest) scored bias triangles, the gate voltage change is accepted, leading to a new gate voltage configuration N_{j+1}. Conversely, if S_i is higher, the gate voltage change is rejected and the gate voltage setting returns to its previous configuration. In this case, the rejected gate voltage change will be an excluded branch in the random selection corresponding to the next iteration. It is possible that all gate voltage branches become depleted, in which case the decision model returns to the previously accepted gate voltage configuration with unexplored branches.

Figure 4. Experimental demonstration of the algorithm. The first column shows four different pairs of bias triangles before the algorithm was run. Each row displays each of these bias triangles at selected iterations of the algorithm. For all these iterations, the applied gate voltage change led to a decrease in score S_i. All measurements were performed with $V_{bias} = 0.2$ mV. The stability diagrams were measured as a function of barrier gates V_3 and V_7, while gate voltages V_1, V_2 and V_8 were tuned by the algorithm.

Figure 5. Score S_i and gate voltage space trajectories during fine-tuning. (a) Score S_i as a function of the number of iterations of our algorithm. Solid lines 1 to 4 correspond to the fine-tuning runs presented in figure 4. The dashed line represents a different run of the algorithm for case 2. (b) Gate voltage space trajectories for the fine-tuning runs in figure 4. The starting (final) gate voltage configuration is denoted by a circle (cross).

7. Experimental demonstration

We test the algorithm for different bias triangles measured on our device. Stability diagrams are measured as a function of barrier gate voltages V_3 and V_7, which are adjusted during centering of the bias triangles. The gate voltages tuned by the algorithm are V_1, V_2 and V_8. For simplicity, we chose to keep gate voltages V_4, V_5 and V_6 fixed. We checked their effect on the bias triangles was weak. All measured stability diagrams are min−max normalized with respect to the minimum and maximum current values measured during coarse tuning to the double quantum dot regime, in this case achieved with the algorithm in reference [7]. In figure 4, the fine-tuning of four different pairs of bias triangles is shown.

In cases 1 to 3, the initial bias triangles lack a well-defined shape, indicative of small inter-dot tunnel coupling [17]. Furthermore, pronounced co-tunnelling lines, which are denotative of second-order transport processes, are observed. As the fine-tuning progresses, the bias triangles separate from each other and acquire a sharper triangular shape. Also, co-tunnelling currents are reduced. In the fourth case, the initial stability diagram shows very faint bias triangles. The algorithm proves capable of increasing the current flowing through the double quantum dot while preserving most of the other bias triangle characteristics. More bias triangle fine-tuning examples achieved by our algorithm can be found in the supplementary material.

Figure 5(a) shows S_i as a function of the number of iterations of our algorithm for cases 1 to 4. Most of the fine-tuning takes place during the first ten iterations, after which the score does not change significantly. In all cases, the algorithm completes the fine-tuning within 26 iterations, corresponding to a total tuning time of 36 min. This time is limited by the measurement time, which could be drastically reduced by radio-frequency reflectometry techniques [24–31].

In figure 5(b) we plot the trajectories in gate voltage space corresponding to each fine-tuning run of the algorithm. The average distance in gate voltage space between the initial gate voltage configurations is greater than for the final gate voltage configurations. This suggests that there exists a region in gate voltage space for which the bias triangles exhibit the most favourable transport characteristics, regardless of their values of V_3 and V_7. Additional data can be found in the supplementary material.

8. Conclusion

We experimentally demonstrate an algorithm for the fine-tuning of bias triangles in gate-defined quantum dots. The algorithm scores real-time measurements by computing distances in the embedding space of a VAE. We show that this score can be used to locally optimize double quantum dot parameters in a completely automated manner. Our results demonstrate that, even with a simple decision model implementation, the score metric proves capable of tuning multiple device parameters at once with no prior knowledge of the device architecture.

The robustness and efficiency of the decision model for proposing new voltage configurations and minimizing the score could potentially be improved by using reinforcement learning or Bayesian optimization, appropriate for cases such as this fine-tuning problem for which data acquisition is costly. Also, while we utilized the Euclidean distance between two Gaussian distributions for computing scores, recent work argues that the decoder induces a Riemannian metric in the latent space [32]. This would imply that shortest paths in latent space do not correspond to straight lines. Therefore, it might prove insightful to implement a Riemannian metric to measure latent space distances. Finally, the influence of selecting targets with different characteristics, such as different excited state energies, could be investigated in the future.

While all measurements presented are performed on a gate-defined GaAs double quantum dot, the VAE was trained on data obtained from a Ge/Si core–shell nanowire device, showing the algorithm is readily applicable to different types of devices. Moreover, our algorithm can be adapted to include any number of additional gate electrodes, paving the way for the tuning of quantum dot arrays.

ORCID iDs

D T Lennon ⓘ https://orcid.org/0000-0001-5831-633X
H Moon ⓘ https://orcid.org/0000-0003-3529-6991

V Nguyen ⓘ https://orcid.org/0000-0002-0294-4561
F Vigneau ⓘ https://orcid.org/0000-0002-7557-4493
L C Camenzind ⓘ https://orcid.org/0000-0002-2278-1915
D M Zumbühl ⓘ https://orcid.org/0000-0001-5831-633X
G A D Briggs ⓘ https://orcid.org/0000-0003-1950-2097
D Sejdinovic ⓘ https://orcid.org/0000-0001-5547-9213
N Ares ⓘ https://orcid.org/0000-0003-2588-6322

References

[1]   Loss D and DiVincenzo D P 1998 *Phys. Rev.* A **57** 120–6
[2]   Kloeffel C and Loss D 2013 *Annu. Rev. Condens. Matter Phys.* **4** 51–81
[3]   Hanson R, Kouwenhoven L P, Petta J R, Tarucha S and Vandersypen L M 2007 *Rev. Mod. Phys.* **79** 1217–65
[4]   Vandersypen L M K, Bluhm H, Clarke J S, Dzurak A S, Ishihara R, Morello A, Reilly D J, Schreiber L R and Veldhorst M 2017 *npj Quant. Inf.* **3** 34
[5]   Camenzind L, Yu L, Stano P, Zimmerman J, Gossard A, Loss D and Zumbühl D 2018 *Nat. Commun.* **9** 3454
[6]   Camenzind L C, Yu L, Stano P, Zimmerman J D, Gossard A C, Loss D and Zumbühl D M 2019 *Phys. Rev. Lett.* **122** 207701
[7]   Moon H *et al* 2020 *Nat. Commun.* **11** 4161
[8]   Zwolak J P *et al* 2019 arXiv:1909.08030
[9]   Durrer R, Kratochwil B, Koski J V, Landig A J, Reichl C, Wegscheider W, Ihn T and Greplova E 2019 arXiv:1912.02777
[10] Baart T A, Eendebak P T, Reichl C, Wegscheider W and Vandersypen L M 2016 *Appl. Phys. Lett.* **108** 213104
[11] Mills A R, Zajac D M, Gullans M J, Schupp F J, Hazard T M and Petta J R 2019 *Nat. Commun.* **10** 1063
[12] Volk C *et al* 2019 *npj Quant. Inf.* **5** 29
[13] Van Diepen C J, Eendebak P T, Buijtendorp B T, Mukhopadhyay U, Fujita T, Reichl C, Wegscheider W and Vandersypen L M 2018 *Appl. Phys. Lett.* **113** 033101
[14] Teske J D, Humpohl S S, Otten R, Bethke P, Cerfontaine P, Dedden J, Ludwig A, Wieck A D and Bluhm H 2019 *Appl. Phys. Lett.* **114** 133102
[15] Lennon D T, Moon H, Camenzind L C, Yu L, Zumbühl D M, Briggs G A D, Osborne M A, Laird E A and Ares N 2019 *npj Quantum Information* **5** 79
[16] Larsen A B L, Sønderby S K, Larochelle H and Winther O 2016 Autoencoding beyond pixels using a learned similarity metric *Proc. of Machine Learning Research* vol 48 pp 1558–66
[17] van der Wiel W G, De Franceschi S, Elzerman J M, Fujisawa T, Tarucha S and Kouwenhoven L P 2002 *Rev. Mod. Phys.* **75** 1–22
[18] Lindeberg T 1993 *Int. J. Comput. Vis.* **11** 283–318
[19] Kingma D P and Welling M 2014 *ICLR 2014*
[20] Kim H and Mnih A 2018 *35th Int. Conf. on Machine Learning, ICML 2018* vol 6 pp 4153–71
[21] Camenzind L C 2019 Spins and orbits in semiconductor quantum dots *Ph.D Thesis* University of Basel
[22] Froning F N M, Rehmann M K, Ridderbos J, Brauns M, Zwanenburg F A, Li A, Bakkers E P A M, Zumbühl D M and Braakman F R 2018 *Appl. Phys. Lett.* **113** 073102
[23] van der Maaten L and Hinton G 2008 *J. Mach. Learn. Res.* **9** 2579–605
[24] Petersson K D, Smith C G, Anderson D, Atkinson P, Jones G A C and Ritchie D A 2010 *Nano Lett.* **10** 2789–93
[25] Ares N *et al* 2016 *Phys. Rev. Appl.* **5** 034011
[26] Schupp F J *et al* 2018 arXiv:1810.05767
[27] Barthel C, Reilly D J, Marcus C M, Hanson M P and Gossard A C 2009 *Phys. Rev. Lett.* **103** 160503
[28] Pakkiam P, Timofeev A V, House M G, Hogg M R, Kobayashi T, Koch M, Rogge S and Simmons M Y 2018 *Phys. Rev.* X **8** 041032
[29] West A *et al* 2019 *Nat. Nanotechnol.* **14** 437–41
[30] Urdampilleta M *et al* 2019 *Nat. Nanotechnol.* **14** 737–41
[31] Zheng G, Samkharadze N, Noordam M L, Kalhor N, Brousse D, Sammak A, Scappucci G and Vandersypen L M K 2019 *Nat. Nanotechnol.* **14** 742–6
[32] Arvanitidis G, Hansen L K and Hauberg S 2018 Latent space oddity: on the curvature of deep generative models *Int. Conf. on Learning Representations*