Supplementary Material

A. Latent space

A representation of the latent space of the trained VAE with the bias triangles corresponding to each embedding is shown in figure S1.

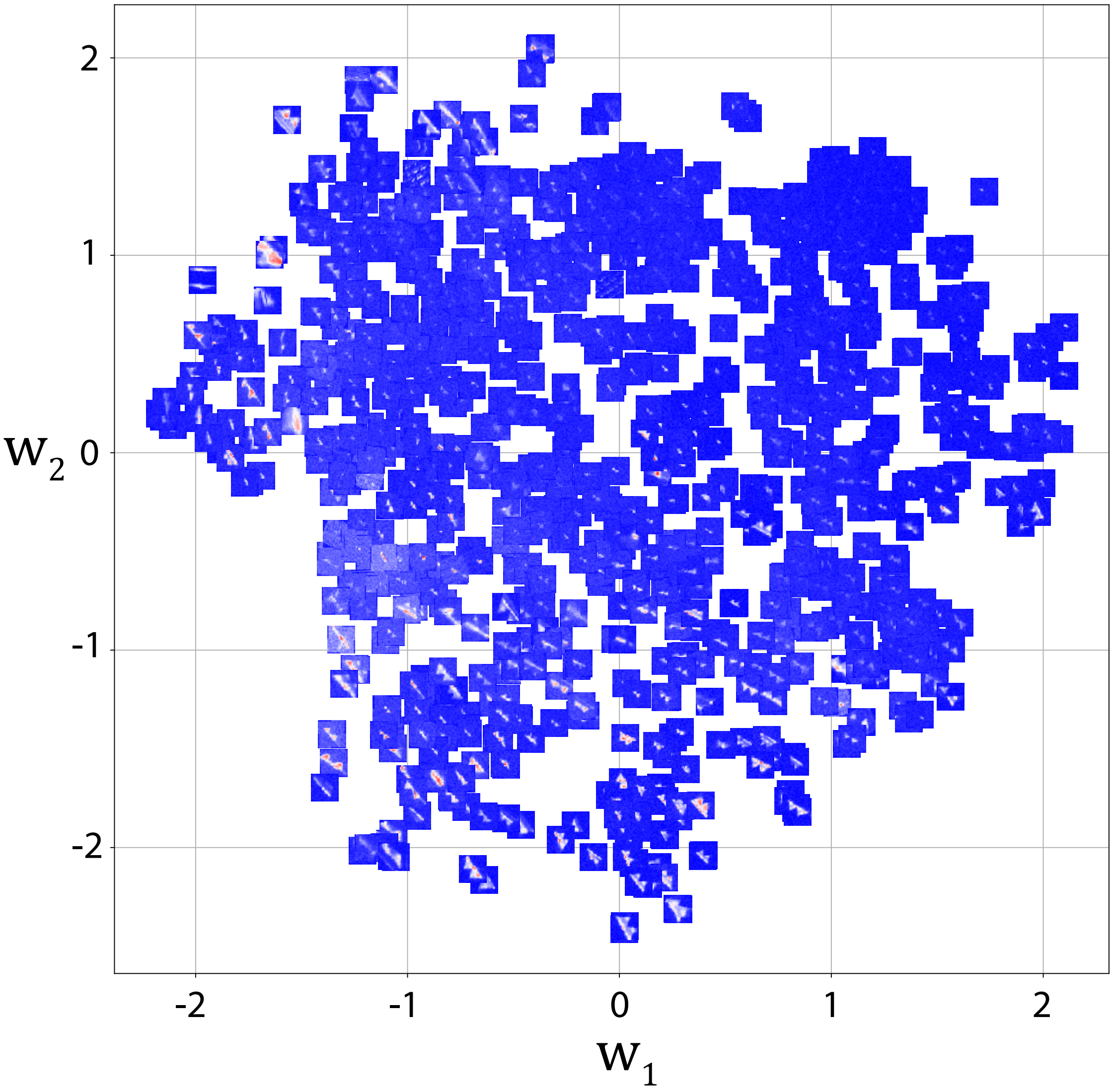


Figure S1: Latent space of the trained VAE. In order to visualise the ten-dimensional latent embeddings, t-SNE [1] is applied for dimensionality reduction. The new two-dimensional latent space is described by a vector $\boldsymbol{w}$. The original training inputs are plotted at the embedding locations.

B. Optimisation

The optimisation of different pairs of bias triangles is shown in figure S2 (cases S1 to S8) and figure S3 (cases S9 and S10). The stability diagrams correspond to gate voltage configurations for which a decrease in score S_i is observed. Figure S4 presents the VAE score S_i as a function of the number of iterations of the optimisation algorithm for the optimisation cases in figure S2 and figure S3. The total gate voltage changes during fine-tuning are presented in Table S1.

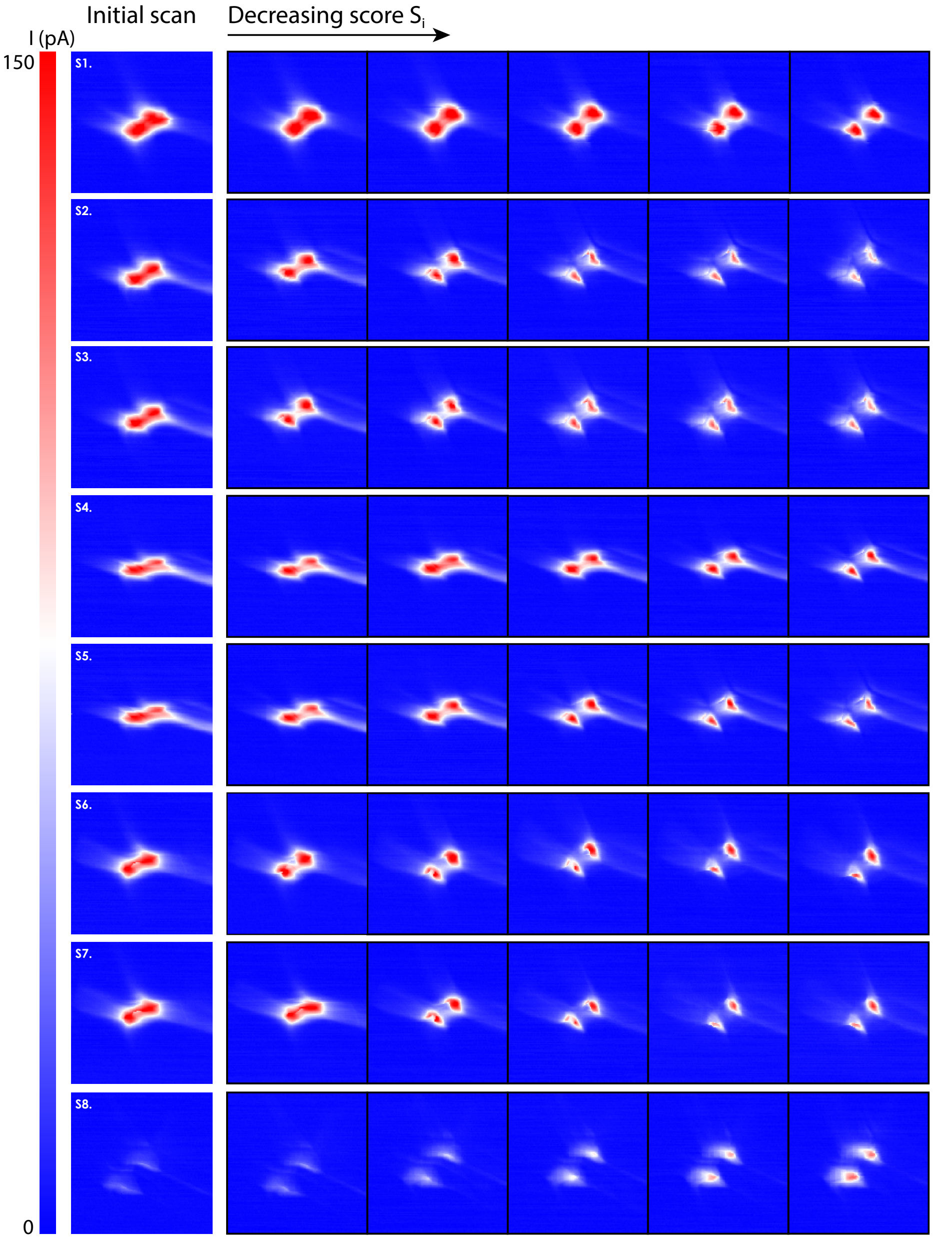


Figure S2: Stability diagrams of bias triangles at different iterations of the optimisation algorithm. The stability diagrams correspond to iterations for which the gate voltage configuration led to a decrease in score S_i. Only a selection of the bias triangle measurements at accepted gate voltage configurations are plotted.

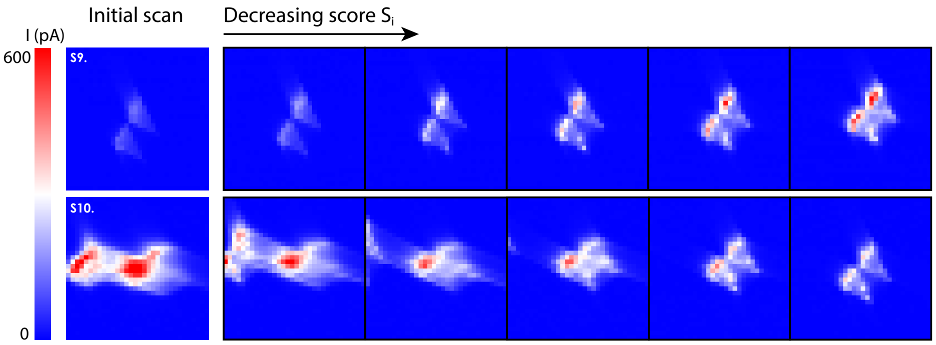


Figure S3: Stability diagrams of bias triangles at different iterations of the optimisation algorithm. The stability diagrams correspond to iterations for which the gate voltage configuration led to a decrease in score S_i. Only a selection of the bias triangle measurements at accepted gate voltage configurations are plotted.

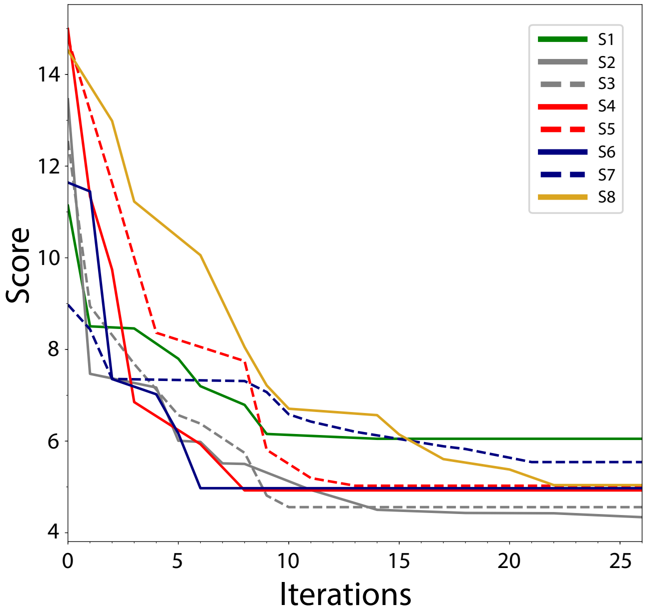


Figure S4: Score S_i as a function of the number of iterations of the optimisation algorithm. The indexed lines correspond to the optimisation cases presented in figure S2. Dashed lines of the same colour represent different runs of the optimisation algorithm for given cases.

Table S1: Total gate voltage change (mV) during fine-tuning. Gates voltages V_1, V_2 and V_8 were optimised, whereas gates V_3 and V_7 were used to center the bias triangles.

Case	ΔV_1	ΔV_2	ΔV_3	ΔV_7	ΔV_8
1	−8.0	0	6.29	5.92	−6.0
2	−6.0	−8.0	6.57	0	6.0
3	−4.0	−8.0	5.26	1.97	0
4	6.0	4.0	−3.94	−7.89	10.0
S1	−10.0	0	5.26	4.60	0
S2	−4.0	−10.0	5.92	1.31	0
S3	0	−8.0	2.63	0	−2.0
S4	−6.0	−2.0	3.94	1.97	2.0
S5	−4.0	−6.0	4.60	0.66	2.0
S6	0	−8.0	3.29	−1.31	2.0
S7	−2.0	−8.0	3.94	0.66	0
S8	14.0	0	−7.23	−9.86	4.0

C. Targets

The target bias triangle pairs used by the fine-tuning algorithm are shown in figure S5. The targets are selected from the unaugmented training set from a Ge/Si nanowire device.

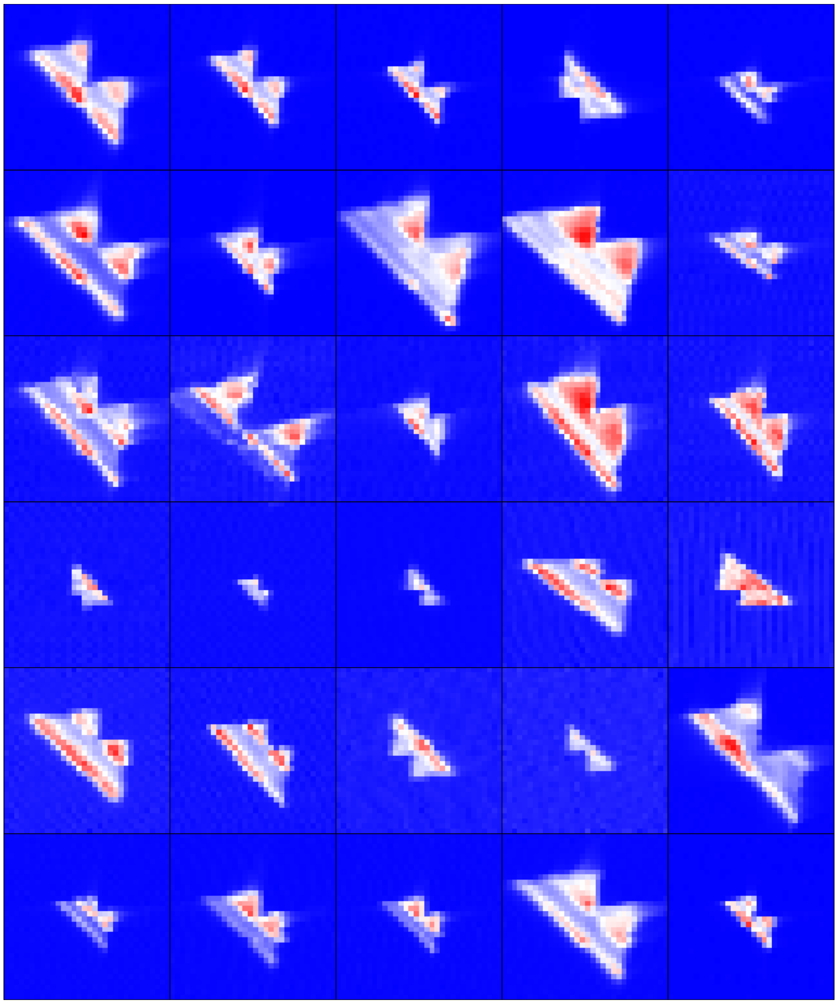
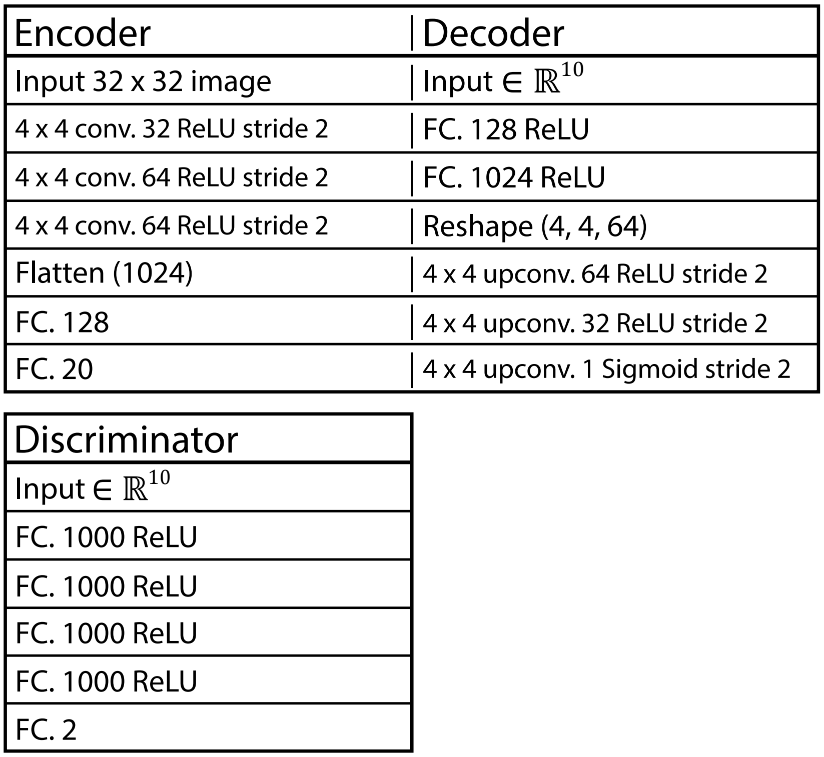


Figure S5: Stability diagrams of the target bias triangle pairs from a Ge/Si nanowire device. Stability diagrams are plotted as a function of barrier gates V_3 and V_7. The colour scale runs from red, the highest current measured, to blue, the lowest current.

D. VAE architecture

The encoder and decoder part of the VAE are embodied in convolutional neural networks. The discriminator consists of a fully connected neural network. The architecture of both the VAE and discriminator neural networks is shown in Fig. S6.



Encoder	Decoder
Input 32 x 32 image	Input ∈ $\mathbb{R}^{10}$
4 x 4 conv. 32 ReLU stride 2	FC. 128 ReLU
4 x 4 conv. 64 ReLU stride 2	FC. 1024 ReLU
4 x 4 conv. 64 ReLU stride 2	Reshape (4, 4, 64)
Flatten (1024)	4 x 4 upconv. 64 ReLU stride 2
FC. 128	4 x 4 upconv. 32 ReLU stride 2
FC. 20	4 x 4 upconv. 1 Sigmoid stride 2

Discriminator
Input ∈ $\mathbb{R}^{10}$
FC. 1000 ReLU
FC. 1000 ReLU
FC. 1000 ReLU
FC. 1000 ReLU
FC. 2

Figure S6: Neural network architectures of the VAE and discriminator.

E. Augmentation

Augmentation is used to expand the size of the training data set by creating modified versions of measurements in the original data set. The variation of images improves the robustness of the model and reduces its sensitivity to device specific parameters and noise.

Mirroring of the stability diagrams was performed around the V_3-V_7 axis and applied to the entire data set. This created an extra mirrored copy of all bias triangle stability diagrams. The orientation of the bias triangles depends on the applied bias voltage. Reversing the bias voltage leads to mirroring of the bias triangle pair around the V_3-V_7 axis. The unaugmented training data set consisted of stability diagrams of bias triangles measured at positive and negative bias voltage. Therefore, in order to increase the variability of the data set, a mirrored version of the stability diagrams was required in the training data set. Thanks to the mirroring augmentation, the algorithm can be operated in an experimental setting at both positive and negative biases.

Rotation was applied about a random angle within -21 to +21 degrees. An original copy of each stability diagram was stored in the training data set. The rotation of bias triangles corresponds to different couplings of the quantum dots to the gate electrodes.

The strength of the Gaussian noise augmentation was 0.25. A random array with values between 0 and 1 was multiplied with 0.25 and added to the stability diagram array multiplied with a factor 0.75. The strength of the Gaussian noise might vary between devices due to their quality in terms of charge traps and defects in the material.

Lastly, random contrast in the range 0.8 to 1.2 was applied. This implied multiplying the entire stability diagram array by a random scalar between 0.8 and 1.2. This contrast would depend on device specific parameters such as tunnel barriers.

F. VAE loss functions

The loss function of a VAE is given by a reconstruction error $\mathcal{L}_{rec}$ and regularisation error $\mathcal{L}_{reg}$. The mathematical definitions of $\mathcal{L}_{rec}$ and $\mathcal{L}_{reg}$ are:

$$\mathcal{L}_{rec} = -\alpha \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \left[\log \left(p_\theta \left(\boldsymbol{x}|\boldsymbol{z}\right)\right)\right] \tag{1}$$

$$\mathcal{L}_{reg} = \beta D_{KL} \left(q_\phi \left(\boldsymbol{z}|\boldsymbol{x}\right) || p\left(\boldsymbol{z}\right)\right) \tag{2}$$

where α and β are weight constants for the reconstruction error and the regularisation term, respectively.

The loss function of the discriminator is given by:

$$\mathcal{L}_{Disc} = \log \left(D_\psi \left(\boldsymbol{z}\right)\right) - \log \left(1 - D_\psi \left(\boldsymbol{z}'\right)\right) \tag{3}$$

where $\boldsymbol{z} \sim q_\phi\left(\boldsymbol{z}\right)$, $z' \sim \prod_j q_\phi\left(z_j\right)$, and D_ψ is the discriminator neural network parametrised by ψ. The discriminator outputs the probability between 0 and 1 that its input is a sample from $q_\phi\left(\boldsymbol{z}\right)$ rather than from $\prod_j q_\phi\left(z_j\right)$. In order to sample from $q_\phi\left(\boldsymbol{z}\right)$, a random batch of data x is selected and then sampled from $q_\phi\left(\boldsymbol{z}|\boldsymbol{x}\right)$. Sampling from $\prod_j q_\phi\left(z_j\right)$ is done by randomly permuting the sampled batch from $q_\phi\left(\boldsymbol{z}\right)$ across the batch for each latent dimension. The discriminator is trained in tandem with the encoder and decoder.

References

[1] van der Maaten L and Hinton G 2008 Journal of Machine Learning Research **9** 2579–2605