Supplementary materials

Supplementary Text

Supplementary Figures S1 to S8

Supplementary Tables S1 to S5

Supplementary Movie M1 and M2

Supplementary References

Supplementary Text

Training and loss function The purpose of this section is to define the losses used for the training of the CVAE. The training is performed by minimising user-defined loss terms through changing the decoder and encoder parameters θ and ϕ using a gradient decent based method. The two loss terms that are minimised to train the encoder and decoder networks are the difference loss and the latent loss.

The difference loss consists of two difference metrics. The first is a sum of the pixel-wise difference between the reconstruction and the training example. The second is a contextual difference which is similar in concept to GAN; the contextual loss is taken from another convolutional neural network called the discriminator. The discriminator is trained in tandem with the encoder and decoder and is trained to distinguish between reconstructions and training examples. The input to the discriminator is a training example Y or reconstruction $\hat{Y}$ and the output is a value between 0 and 1, representing the probability the input is a training example or a reconstruction. As the discriminator is trained to distinguish between training examples and reconstructions, it learns to decode contextual features that distinguish reconstructions from training examples. We then calculate the difference between intermediate layer representations of the training example and intermediate layer representations of its reconstruction. If we ignore

the contextual loss, the decoder produces only blurry reconstructions.

The latent loss is applied only to the encoder and forces the set of encoded training examples $\{z\}$ to be normally distributed with mean of the zero vector and the covariance of a diagonal matrix. This can be achieved by minimising the Kullback-Leibler (KL) divergence between the output distribution of the encoder and the target zero-mean distribution.

Network specification The specification of the convolutional neural networks used in this paper is described in Table S1∼S3. Exponential linear unit is applied after each layer except the final layer of the encoder, decoder, and the discriminator. Batch normalisation is applied after all convolution layers except as separately described. The first and second number in parentheses of the layer names indicate kernel size and stride.

Noisy reconstruction For the estimation $\tilde{r}_m(n)$, a single reconstruction $\hat{Y}_m$ is augmented to 30 noisy reconstructions:

$$\hat{Y}_{m,j,\text{SNR}}(x) = \hat{Y}_m(x) + \alpha_{m,j,\text{SNR}} \times \mathcal{E}_j(x) \quad \text{for all} \quad x \text{ in } X,$$

where $\mathcal{E}_j = \{(x, \epsilon_x^j) | x \in X\}$ is a noise profile consisting of pairs of location and noise, X is a set of all voltage pairs in a 2D domain, and $\alpha_{m,j,\text{SNR}}$ is a multiplier that makes the signal-to-noise ratio SNR, where the signal is $\hat{Y}_m$ and the noise is $\mathcal{E}_j$. We measured 10 noise profiles at non-conducting voltage ranges, but very close to Coulomb diamonds, and j is the index of the profile. SNR is chosen from $\{20^2, 40^2, 80^2\}$, which leads to a high noise, medium noise, and low noise.

Context-aware decision for stability diagrams By converting reconstructions to some context maps, we can make a decision related with the context map. We have developed a segmentation method, that produces a segmentation map which has a value is 1 if the location is inside

a diamond or 0 otherwise. This segmentation method is based on another deep neural network called a U-net [1]. Training data for the segmentation network are pairs of current map and segmentation map, which are generated by the same simulator used for the reconstruction network. Fig. S7 shows the segmentation result of a trained network for 10 real stability diagrams.

By producing segmentation maps of reconstructions, their segmentation disagreement can be calculated. This produces large disagreement along the edges of reconstructions resulting in measurements that focus on diamond edges as show in Fig. S8A. Noise is also added to the outside of diamond segmented maps. This supplies further disagreement between segmentation maps which prioritises measurement outside of the diamond after edges are measured.

The success measure $e(n)$ in Fig. S8 C and E is calculated by applying the segmentation model to the fully measured current map and then applying a Sobel filter to the resulting segmented map; this produces an edge map. The error and optimal performance are then calculated as the ratio of this remaining quantity in the same way as was done for $r(n)$ except substituting the edge maps for transconductance maps.

Supplementary Figures

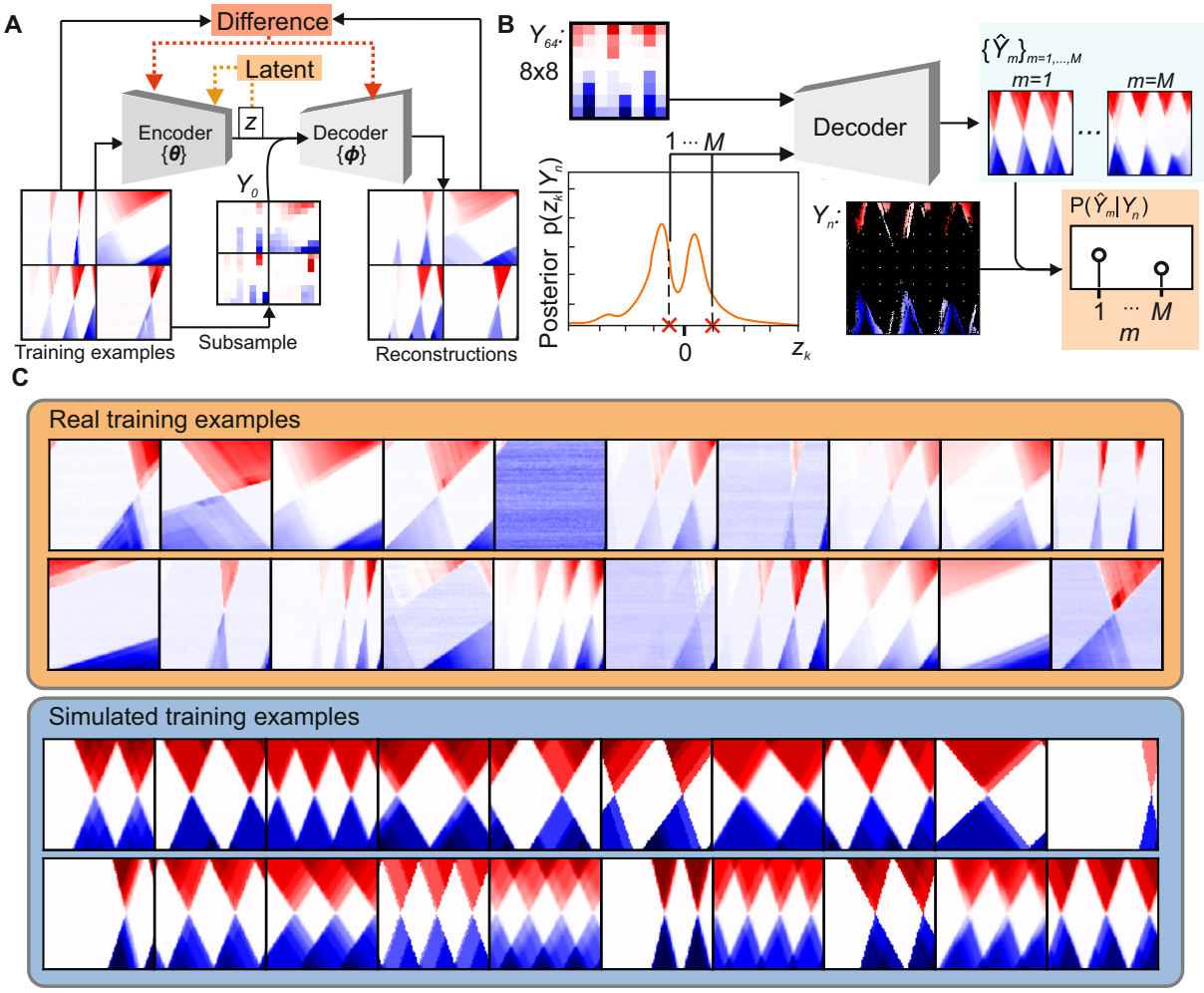

Figure S1: **A**, Training procedure. Training examples are converted to latent vectors by the encoder. Latent vectors and 8×8 sub-sampled training examples are transformed into reconstructions by the decoder. The difference (red box) between original examples and reconstructions is used to optimise the encoder/decoder parameters θ and ϕ. The distribution of training examples in the latent space is enforced during training by latent loss (orange box). **B**, Generation of reconstructions. After 8×8 initial measurements, latent vectors are sampled from the posterior distribution of z and transformed by the decoder to generate multiple reconstructions $\hat{Y}_1, \ldots, \hat{Y}_M$. Posterior probability for reconstructions $P(\hat{Y}_m|Y_n)$ is calculated with respect to acquired partial measurements Y_n. **C**, Real and simulated training examples.

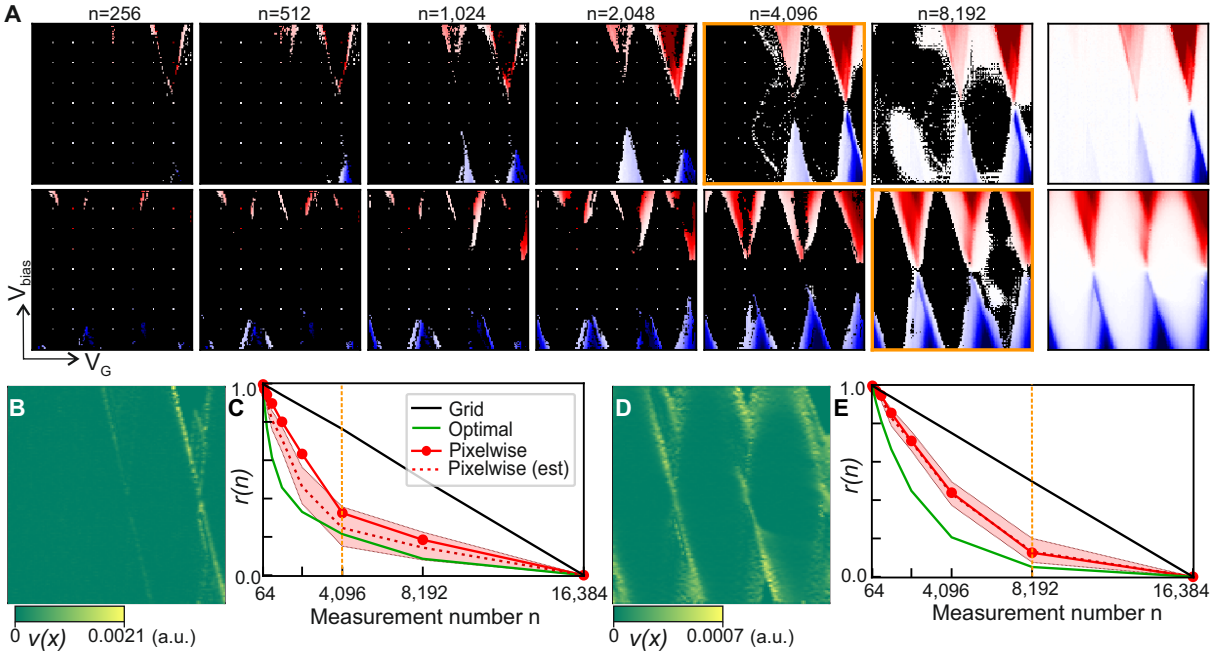

Figure S2: **Pixel-wise measurements of Coulomb diamonds performed by the algorithm.** **A**, Sequential pixel-wise measurement in two different experiments. Each row displays algorithm assisted measurements of the current map as a function of V_{bias} and V_{G} for different values of n. The last plot in each row is the full-resolution current map. **B**, **D**, Current gradient map for each example in **A**. **C**, **E**, Measure of the algorithm's performance $r(n)$, average real-time estimate of $r(n)$ across reconstructions with 90% credible interval, 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 dashed orange line indicates the value of n determined by the stopping criterion. The corresponding current map in **A** is highlighted in orange.

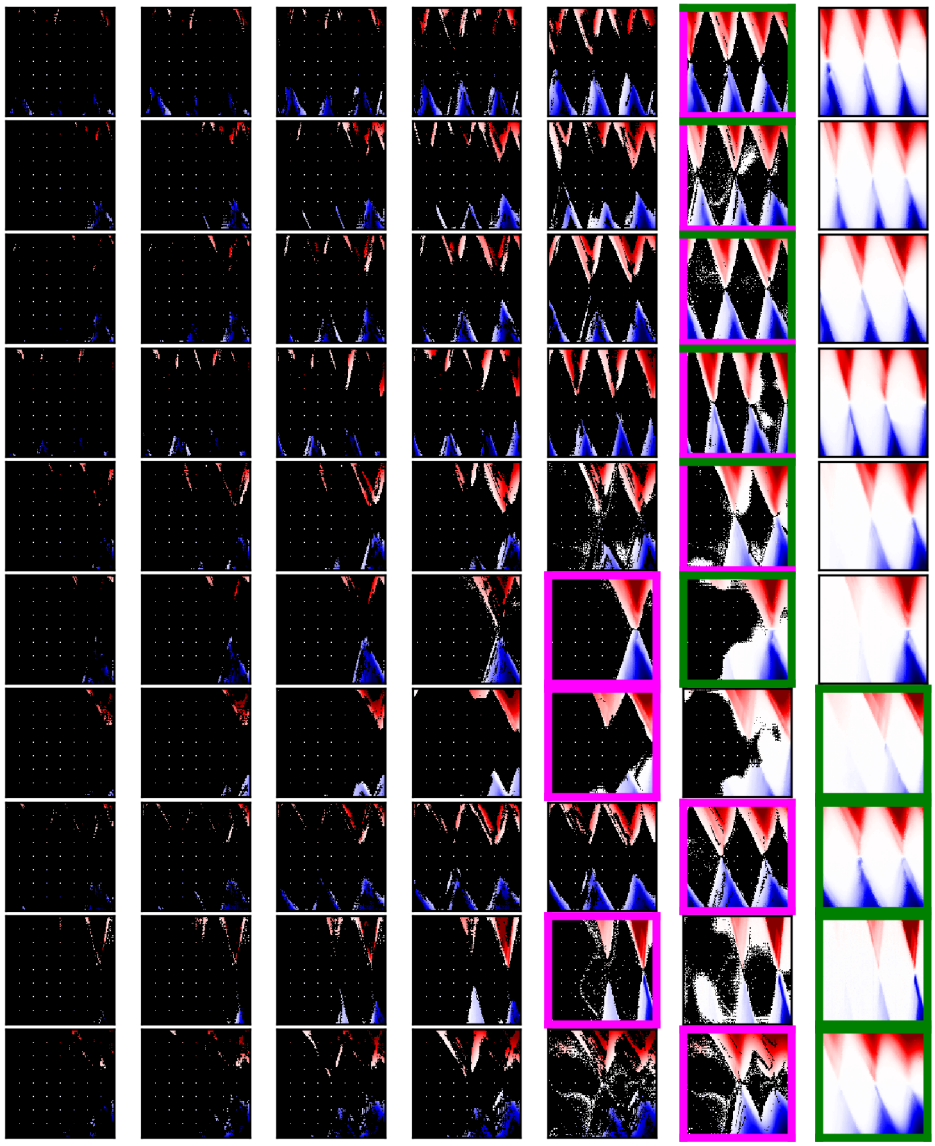

Figure S3: Each row shows intermediate steps of point-wise decision for given voltage ranges. Magenta box indicates the default stopping criterion, and green box indicates when we have allocated a measurement budget of 70% for full measurement of all 10 examples.

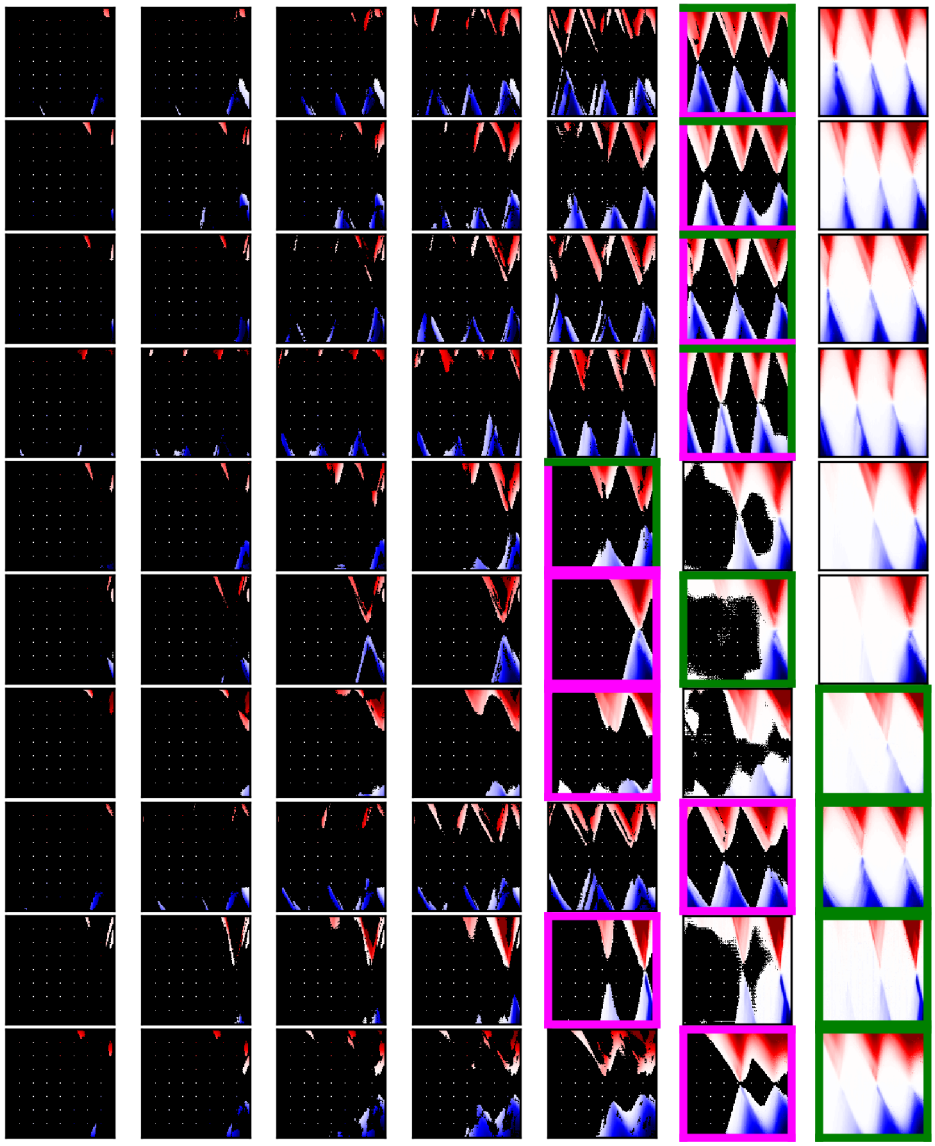

Figure S4: Each row shows intermediate steps of batch decision for given voltage ranges. Magenta box indicates the default stopping criterion, and green box indicates when we have 70% budget for full measurement of all 10 examples.

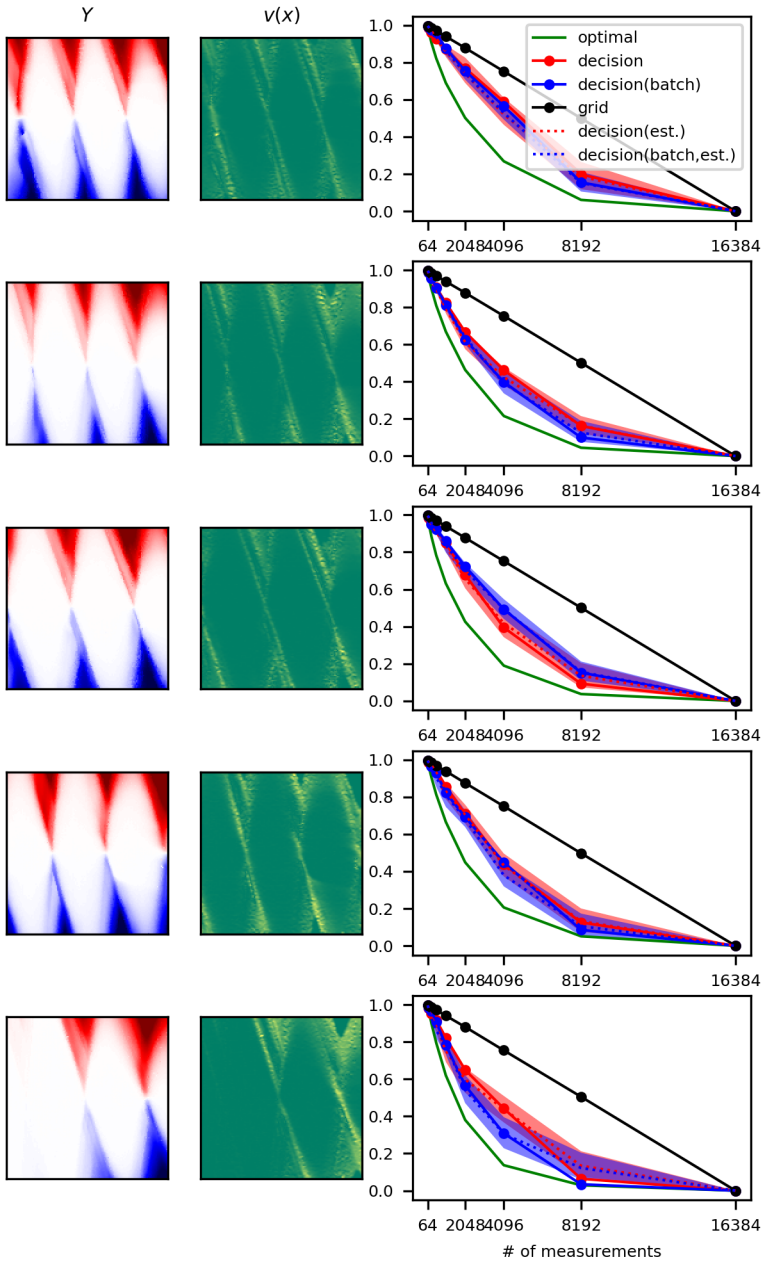

Figure S5: Quantitative analysis for experiment number 1∼5.

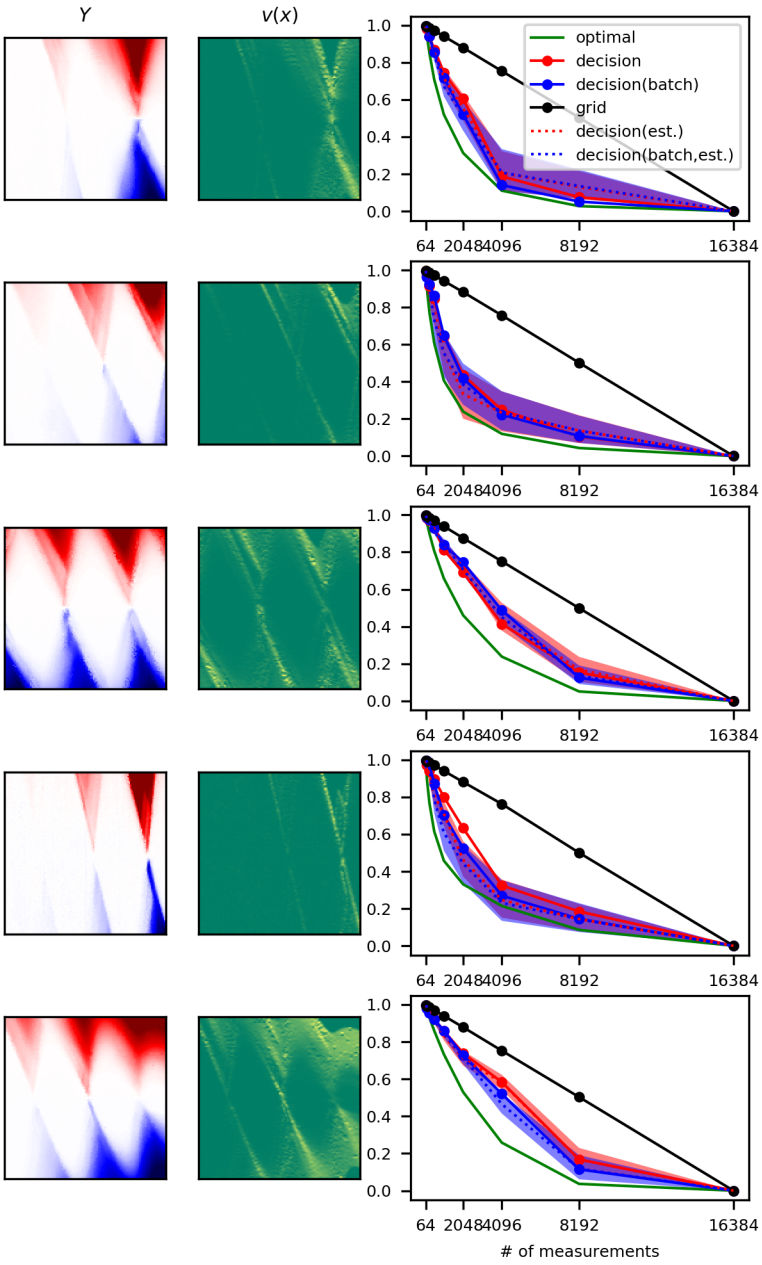

Figure S6: Quantitative analysis for experiment number 6∼10

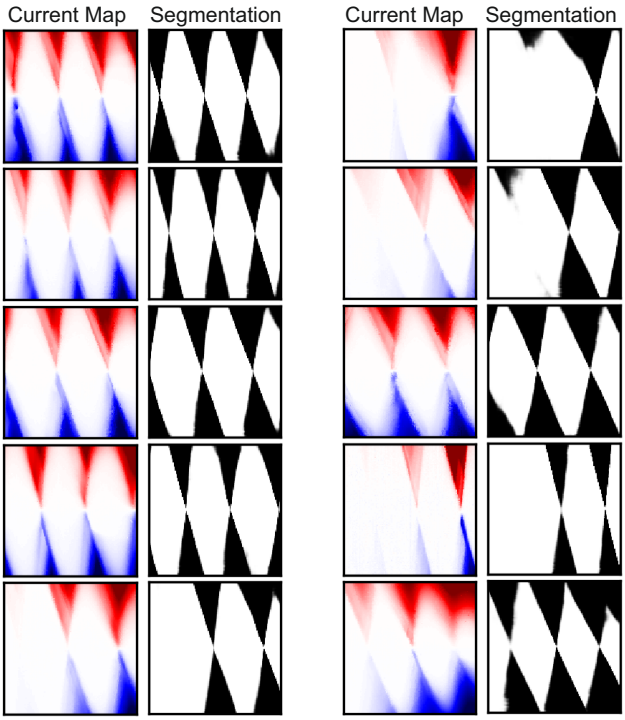

Figure S7: Full resolution current map and segmented result from the segmentation network.

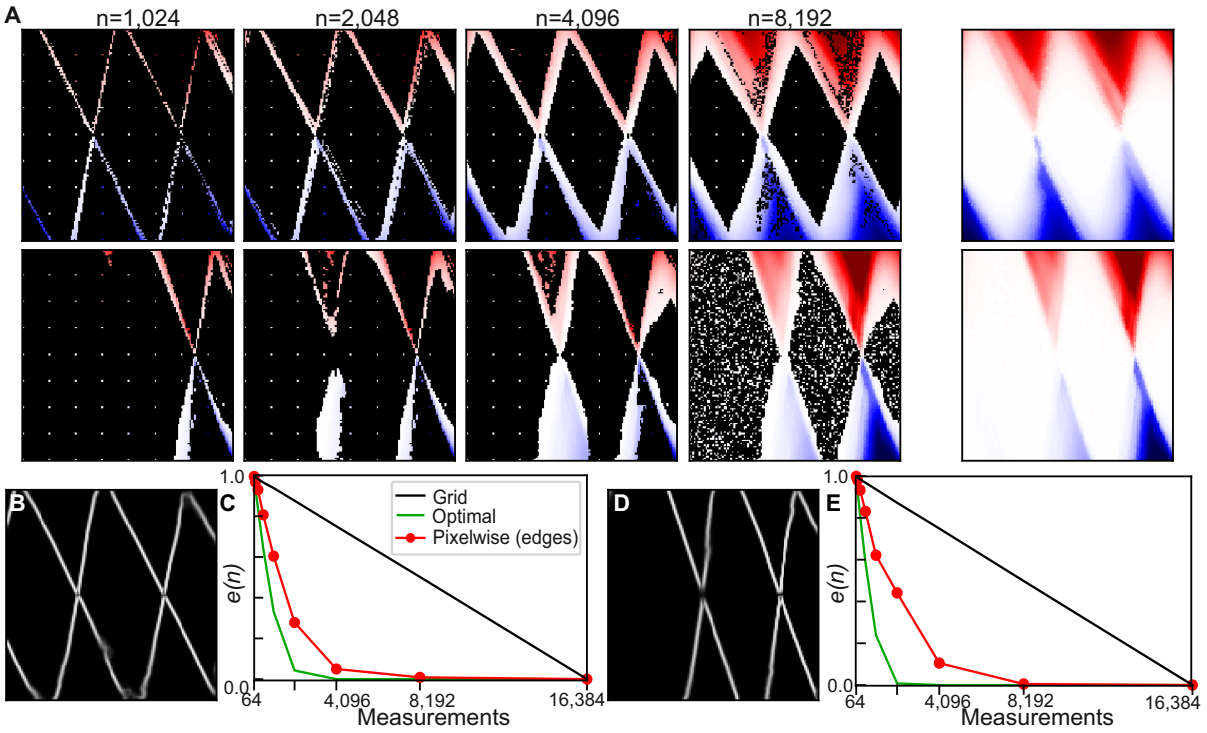

Figure S8: **Measurements of a current map performed by the context-aware implementation of the algorithm. A**, Sequential pixel-wise measurements. Each row displays the algorithm assisted measurements of a current map as a function of V_{bias} and V_{G} for different values of n. The last plot in each row is the full-resolution current map. **B**, **D**, Sobel filter applied to the segmentation map of the full current map for both examples in **A**. **C**, **E**, Measure of the algorithm's performance $e(n)$ and optimal $e(n)$ for both current maps in **A**. The black line is the value of $e(n)$ corresponding to the alternating grid scan method.

Supplementary Tables

Layer name	output size
Initial	128x128x1
Conv(5,2)	64x64x64
Max pooling(3,2)	32x32x64
Conv(3,1)	32x32x128
Conv(3,2)	16x16x128
Conv(3,1)	16x16x128
Conv(3,2)	8x8x128
Conv(3,1)	8x8x128
Conv(3,2)	4x4x128
Fully connected	200

Table S1: Specification of the encoder.

Layer name	output size
Initial	1x1x(100+64)
Conv'(3,2)	2x2x1,024
Conv(3,1)	2x2x1,024
Conv'(3,2)	4x4x512
Conv(3,1)	4x4x512
Conv'(3,2)	8x8x256
Conv(3,1)	8x8x256
Conv'(3,2)	16x16x128
Conv(3,1)	16x16x128
Conv'(3,2)	32x32x64
Conv(3,1)	32x32x64
Conv'(3,2)	64x64x64
Conv(3,1)	64x64x64
Conv'(3,2)	128x128x32
Conv(3,1)	128x128x32
Conv(1,1,tanh)	128x128x1

Table S2: Specification of the decoder.

Layer name	output size	remark
Initial	128x128x1	
Conv(5,2)	64x64x64	context loss
Conv(3,1)	64x64x128	No BN
Conv(3,2)	32x32x128	context loss
Conv(3,1)	32x32x128	No BN
Conv(3,2)	16x16x128	context loss
Conv(3,1)	16x16x128	No BN
Conv(3,2)	8x8x128	context loss
Conv(3,1)	8x8x128	No BN
Conv(3,2)	4x4x128	context loss
Global average pooling	1x1x128	context loss
Fully connected	2	

Table S3: Specification of the discriminator. In remarks, *NO BN* indicates that batch normalisation is not applied to the layer, and *context loss* indicates that the layer is used to calculate the context loss.

case index	64	128	256	512	1,024	2,048	4,096	8,192	16,384
1	10.50	13.13	18.32	28.07	47.27	84.51	156.98	294.91	561.21
2	10.14	12.71	17.46	26.78	44.95	81.32	152.31	289.03	554.98
3	10.49	13.08	17.77	26.66	45.97	82.82	53.86	291.15	557.51
4	9.40	12.16	17.09	27.34	46.69	82.99	152.64	289.00	555.27
5	10.52	13.12	18.10	26.99	45.66	81.28	151.10	287.97	553.07
6	10.14	12.73	17.21	27.07	45.17	80.55	149.18	285.82	550.95
7	15.71	18.34	22.77	31.73	50.61	87.45	159.77	296.92	563.10
8	14.96	17.85	22.53	32.66	52.08	89.56	161.62	300.80	566.80
9	10.14	12.82	17.57	26.89	45.08	80.56	149.62	286.04	551.51
10	15.31	17.88	22.71	31.71	51.10	87.94	159.44	296.00	561.27

Table S4: Measurement time for the batch method

case index	64	128	256	512	1,024	2,048	4,096	8,192	16,384
1	10.51	13.53	18.52	28.27	46.09	81.51	149.73	285.62	552.69
2	10.12	13.13	18.09	27.73	45.50	80.93	149.20	285.43	552.76
3	10.50	13.26	18.19	27.78	45.66	80.98	149.22	282.37	546.65
4	9.38	12.16	17.09	26.75	44.49	80.07	148.31	284.37	551.89
5	10.51	13.28	18.21	27.86	45.69	81.02	149.20	285.02	552.15
6	10.15	12.93	17.87	27.51	45.41	80.80	148.97	285.47	552.62
7	15.70	18.64	24.02	34.52	52.28	87.49	155.75	292.05	559.88
8	14.97	17.89	23.25	33.73	51.51	86.83	155.02	288.72	553.71
9	10.13	12.89	17.87	27.57	45.42	80.97	148.89	285.08	553.30
10	15.34	18.28	23.64	34.15	52.05	87.50	155.68	291.94	559.65

Table S5: Measurement time for grid scanning

Movies

Movie M1: Animation of selected partial current maps corresponding to a real time measurement of current as a function of bias and gate voltage, with corresponding acquisition map, using the batch method.

Movie M2: Animation of selected partial current maps, corresponding to a real time measurement of current as a function of two gate voltages, with corresponding acquisition map, using the batch method.

References

[1] Ronneberger, O., Fischer, P. & Brox, T. U-net: convolutional networks for biomedical image segmentation. In *Proc. MICCAI*, 234–241 (2015).