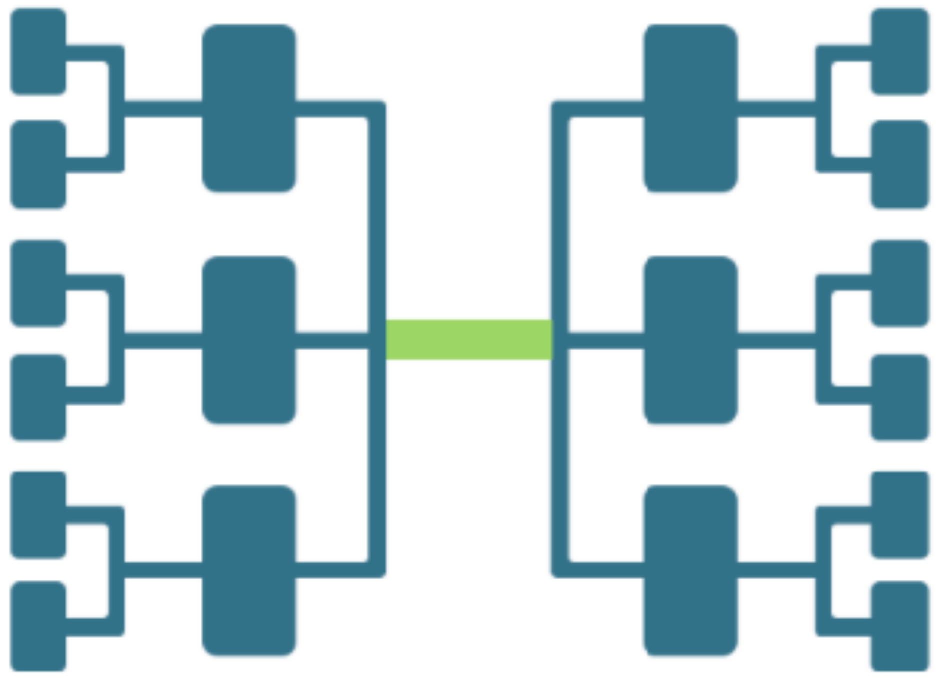
Machine Learning in Quantum Physics and Chemistry, Warsaw 2021

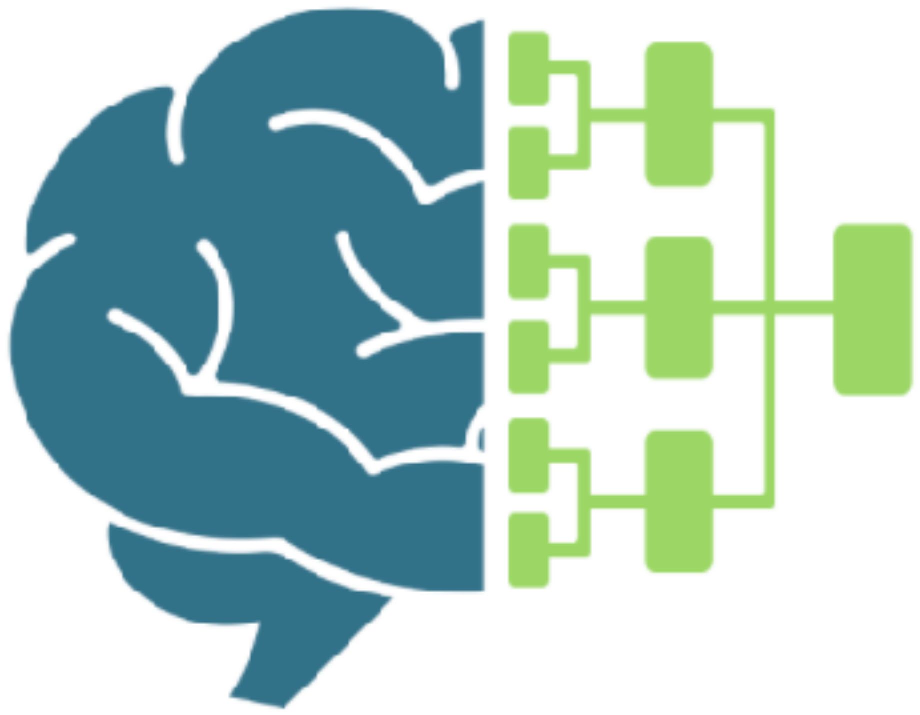
Eliška Greplová, TU Delft

So what is machine learning?

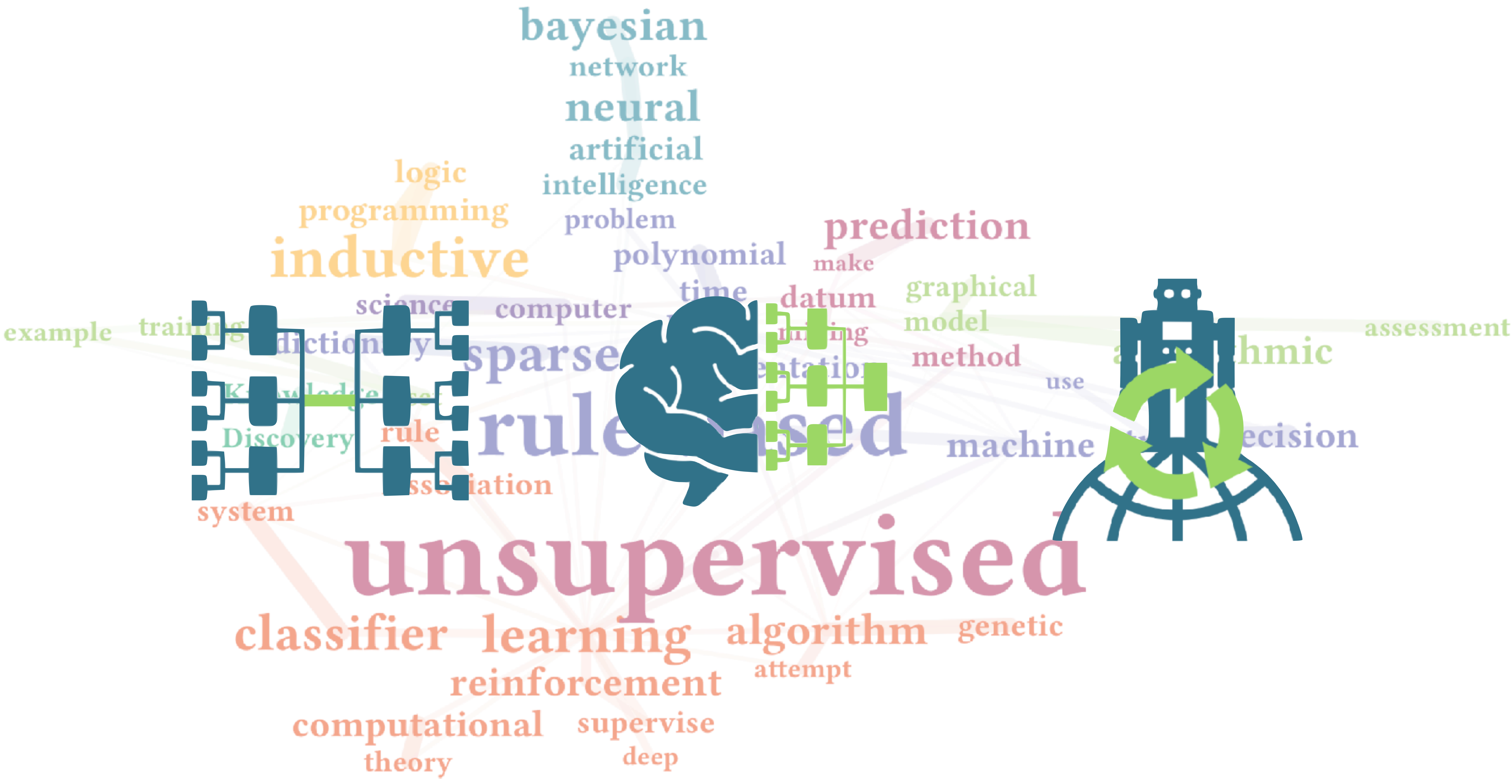
Supervised



Unsupervised



Reinforcement



arXiv: 1708.03569

ML algorithms *learn* general rules from data

How is this useful in quantum physics?
Do we have"big data" in quantum matter and quantum tech?

Quantum Big Data

Wave-function

$$\psi$$

QMAI | TU Delft

Wave-function

2 SPINS = SIZE 4

100 SPINS = SIZE 10^30

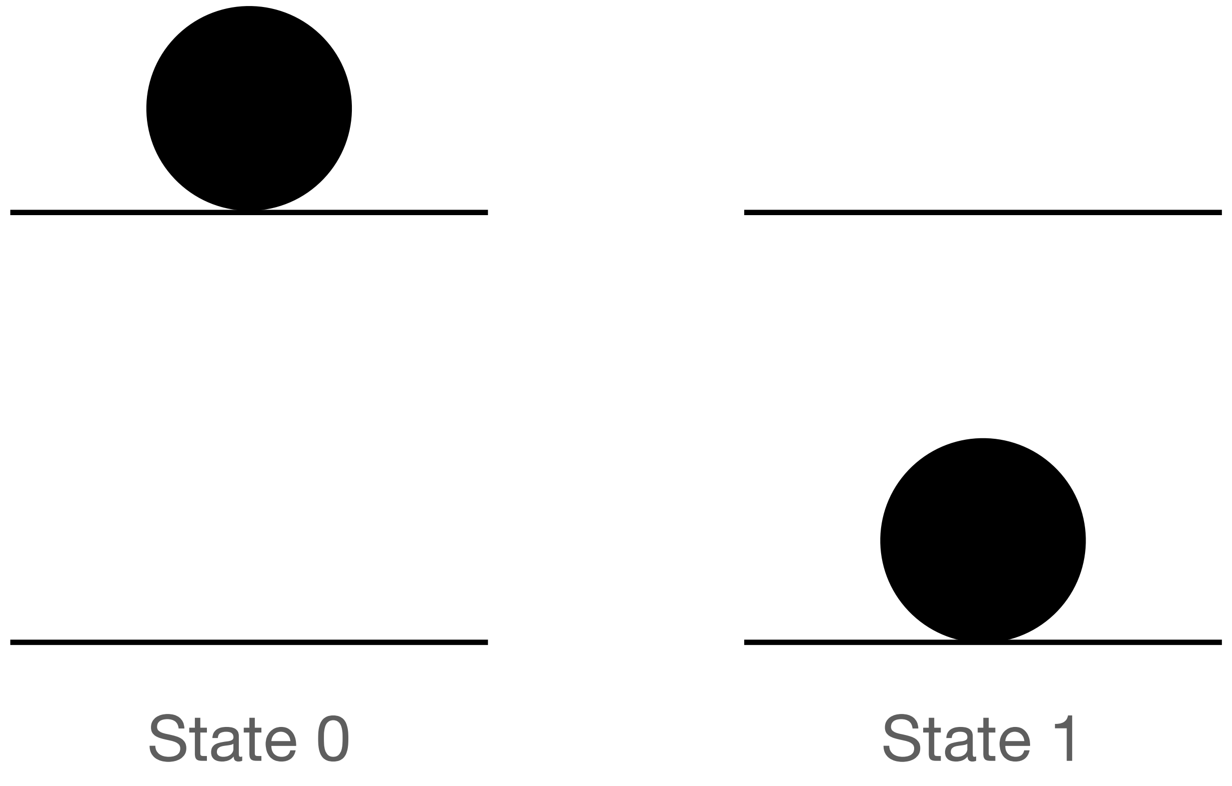
ψ

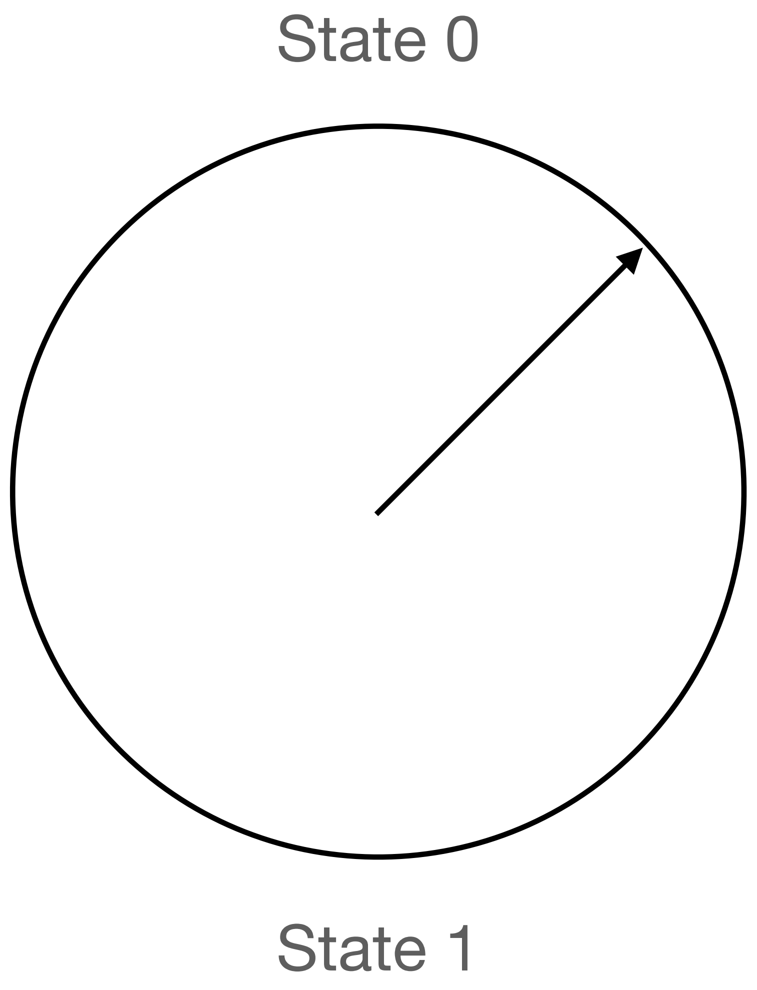
10 SPINS = SIZE 1024

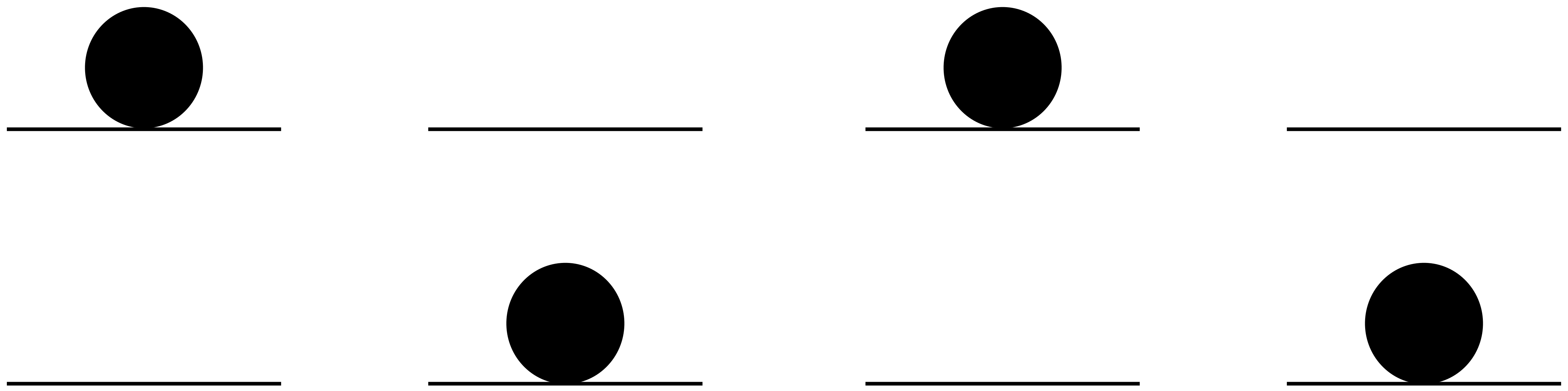
1000 SPINS = SIZE 10^300

50 SPINS = SIZE 10^15

Wave-function

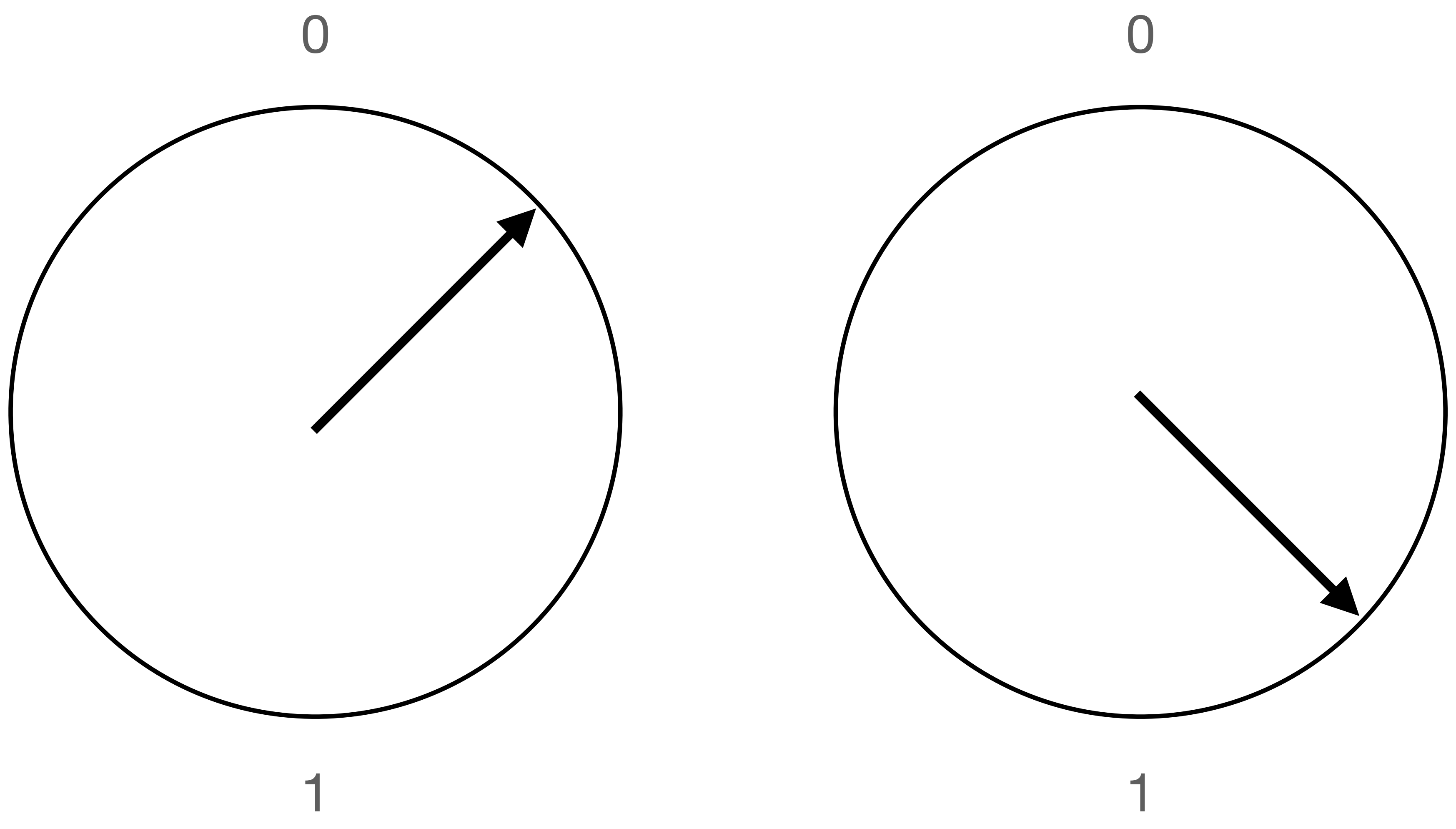






spin #1: state 0 or 1,
spin #2: state 0 or 1,

N spins ~ N numbers



a |00>+b |11>+c |10>+d |01>
N spins ~ 2^N numbers

QMAI | TU Delft

age of the universe: 10^22 seconds 🤣

Wave-function

2 SPINS = SIZE 4

100 SPINS = SIZE 10^30

ψ

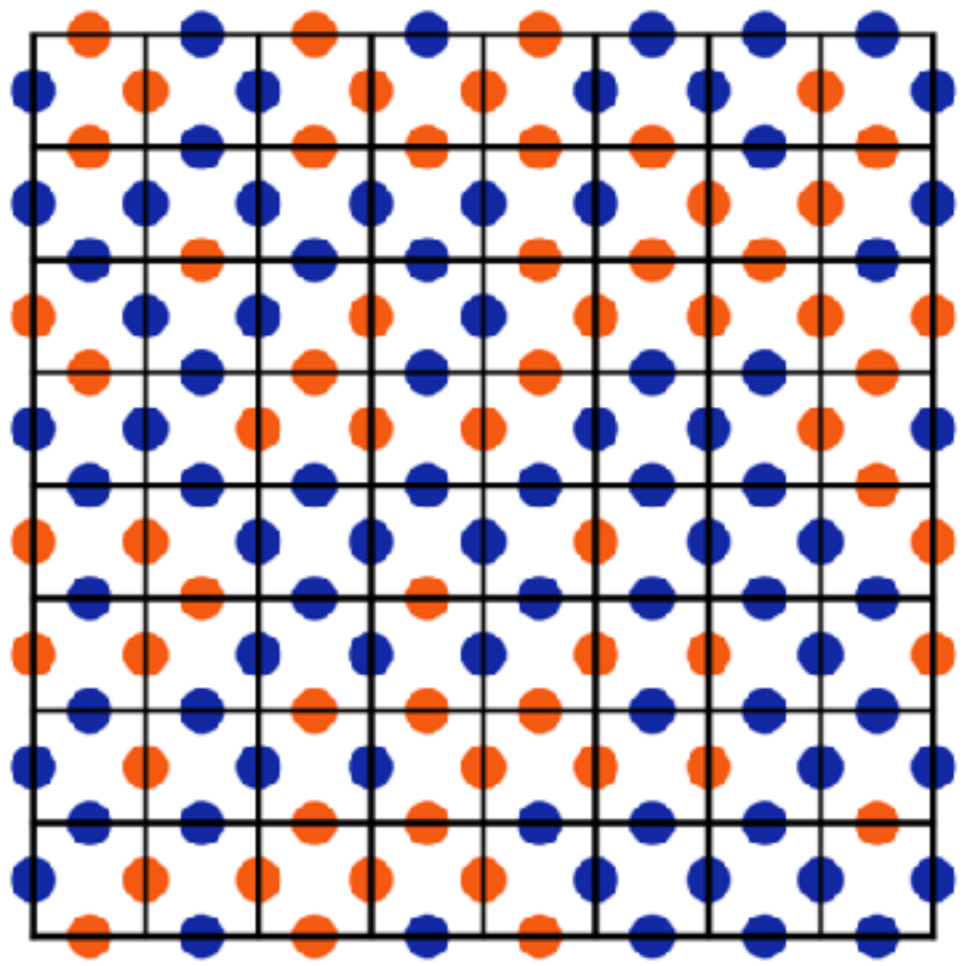
10 SPINS = SIZE 1024

1000 SPINS = SIZE 10^300

50 SPINS = SIZE 10^15

QMAI | TU Delft

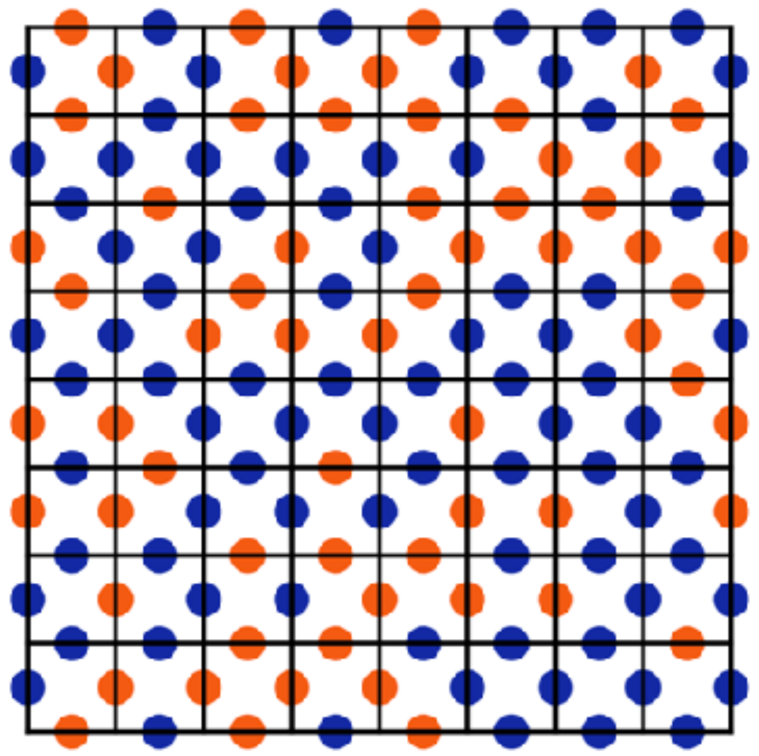
Experimental measurements



QMAI | TU Delft

EXPERIMENT

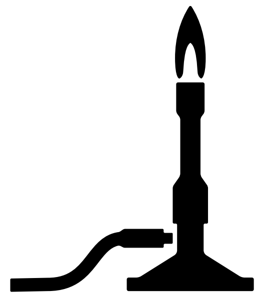
THEORY



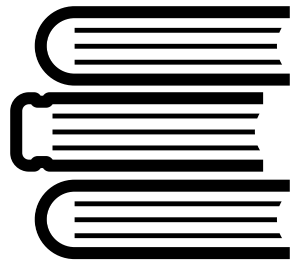
measurement

H, ψ

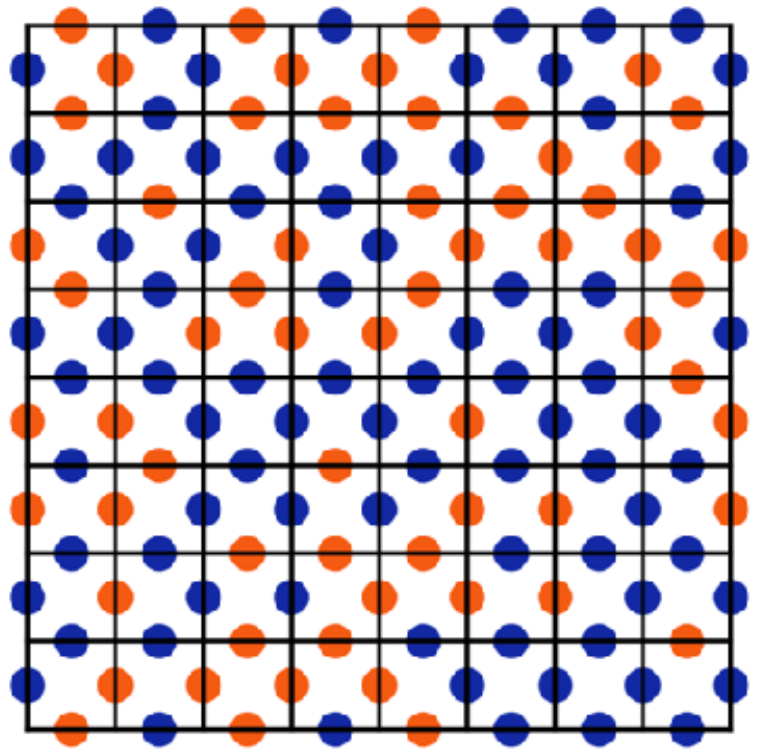
wave function
Hamiltonian



EXPERIMENT



THEORY



measurement

$$H, \psi$$

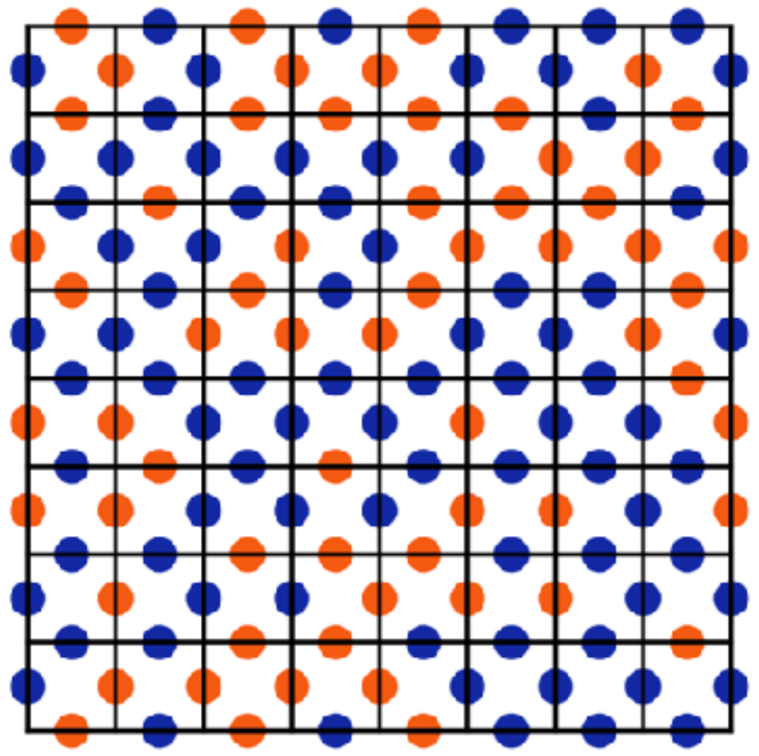
wave function
Hamiltonian

Since wave-functions are so complex - it might take A LOT of measurement data to gather enough information about them.

Q: Do we have 'big data' in quantum physics?

A: Of course!

1. Wave-functions are data-intensive objects that do not scale well
2. (Large) scale quantum experiments = large scale data



H, ψ

Data driven approaches

Simple toy models

condensed matter
quantum technology
quantum chemistry
quantum experiments

Machine Learning & Quantum Physics

Discover new physics from data

Efficiently Approximate Quantum States

Automated Control of Quantum Devices

Benchmark quantum computers

Design quantum experiments and novel materials

Discover and optimise quantum algorithms

Machine Learning & Quantum Physics

Discover new physics from data

Efficiently Approximate Quantum States

Automated Control of Quantum Devices

Benchmark quantum computers

Design quantum experiments and novel materials

Discover and optimise quantum algorithms

QMAI | TU Delft

Machine Learning & Quantum Physics

Discover new physics from data

Efficiently Approximate Quantum States

Automated Control of Quantum Devices

Benchmark quantum computers

Design quantum experiments and novel materials

Discover and optimise quantum algorithms

QMAI | TU Delft

Machine Learning & Quantum Physics

Discover new physics from data

Efficiently Approximate Quantum States

Automated Control of Quantum Devices

Benchmark quantum computers

Design quantum experiments and novel materials

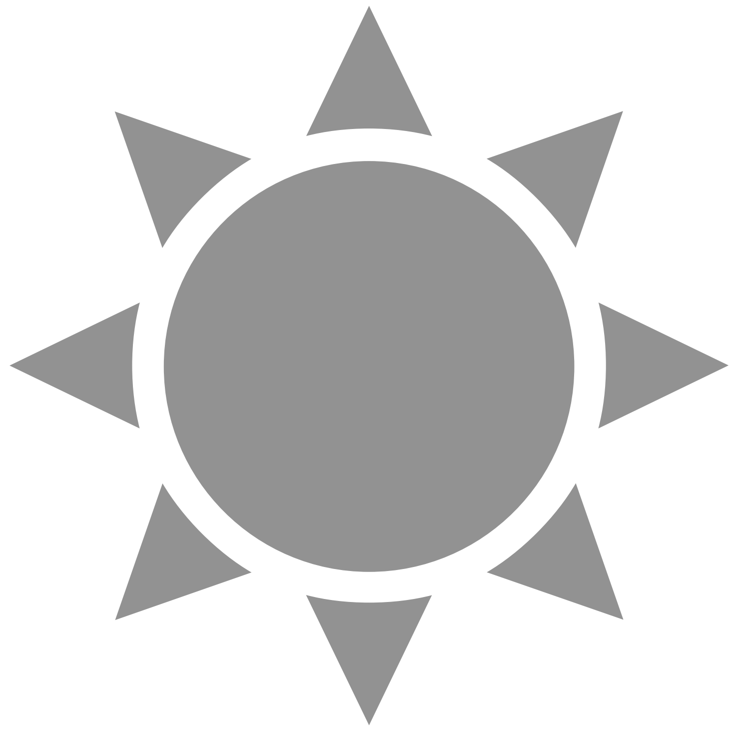
Discover and optimise quantum algorithms

Machine Learning & Quantum Physics

Discover new physics from data

Efficiently Approximate Quantum States

Automated Control of Quantum Devices



Benchmark quantum computers

Design quantum experiments and novel materials

Discover and optimise quantum algorithms

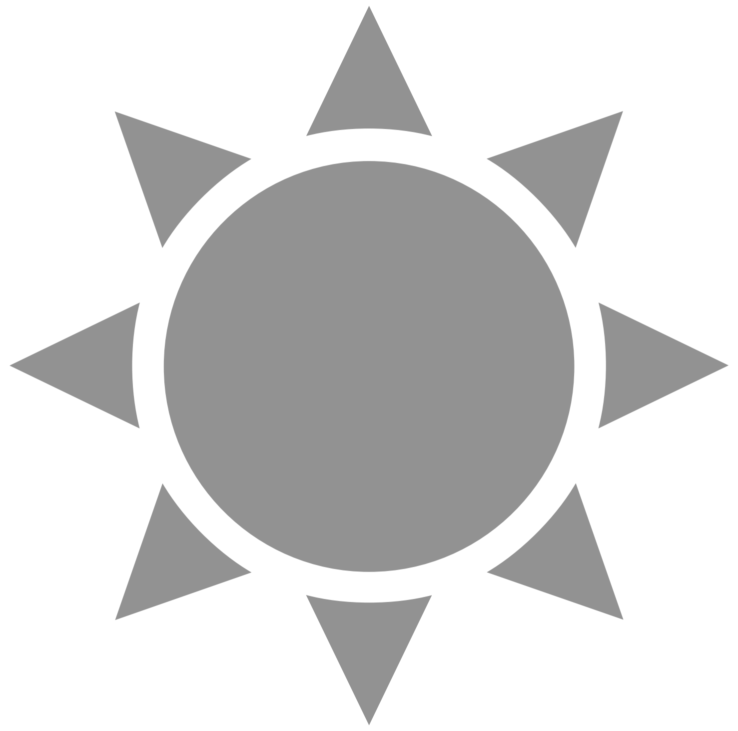
QMAI | TU Delft

Machine Learning & Quantum Physics

Discover new physics from data

Efficiently Approximate Quantum States

Automated Control of Quantum Devices



Benchmark quantum computers

Design quantum experiments and novel materials

Discover and optimise quantum algorithms

Machine Learning & Quantum Physics

Discover new physics from data

Efficiently Approximate Quantum States

Automated Control of Quantum Devices

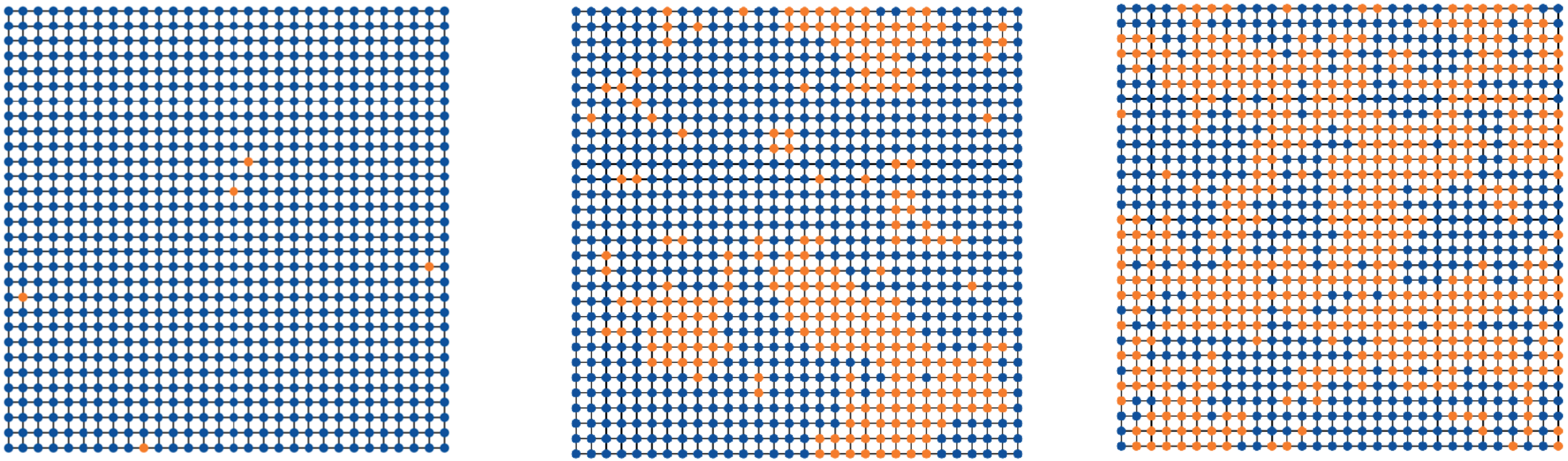
Benchmark quantum computers

Design quantum experiments and novel materials

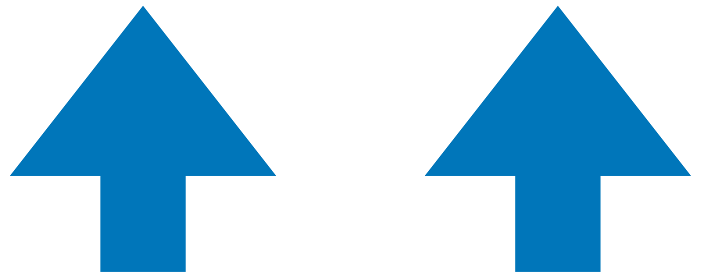
Discover and optimise quantum algorithms

QMAI | TU Delft

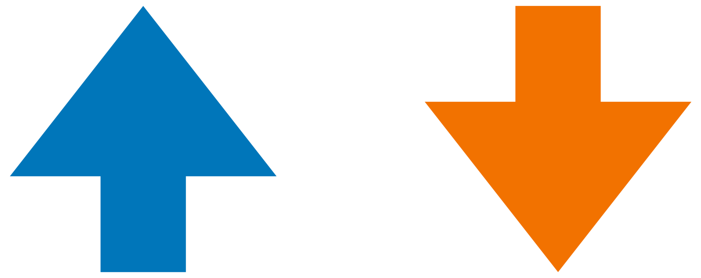
Ising model



$$H = -J \sum_{<i,j>} \sigma_i \sigma_j$$



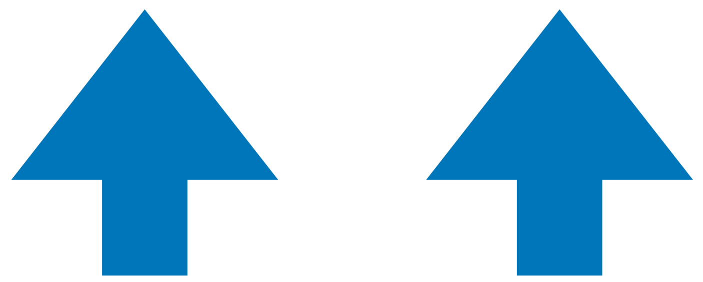
$\sigma_i \sigma_j$ = 1 energy = - J

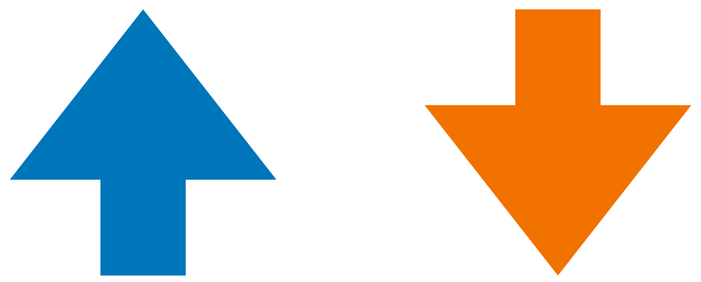


$\sigma_i \sigma_j$ = -1 energy = + J

$$p(\sigma) = \frac{e^{-\beta H(\sigma)}}{Z}$$

= probability of configuration σ at the temperature $T = 1/\beta$

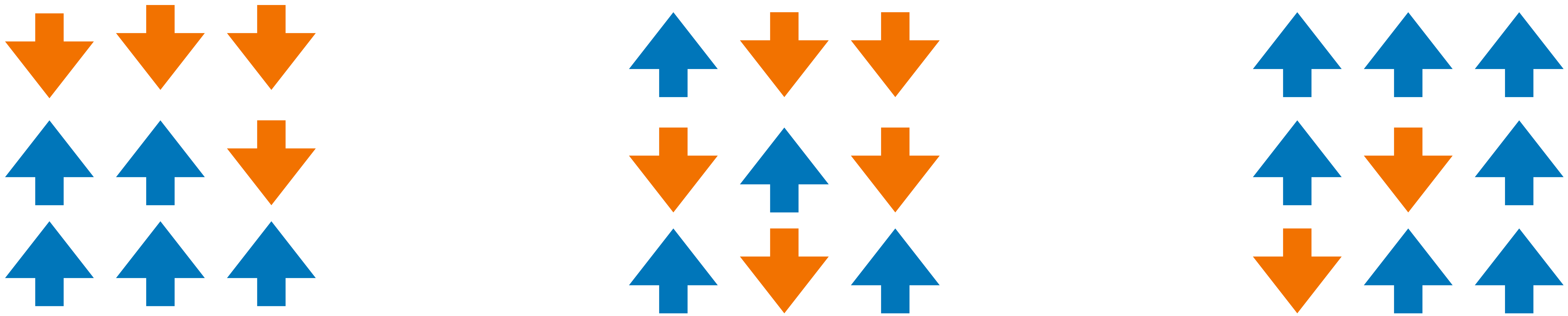




adding temperature makes the difference smaller

Ising QUIZ

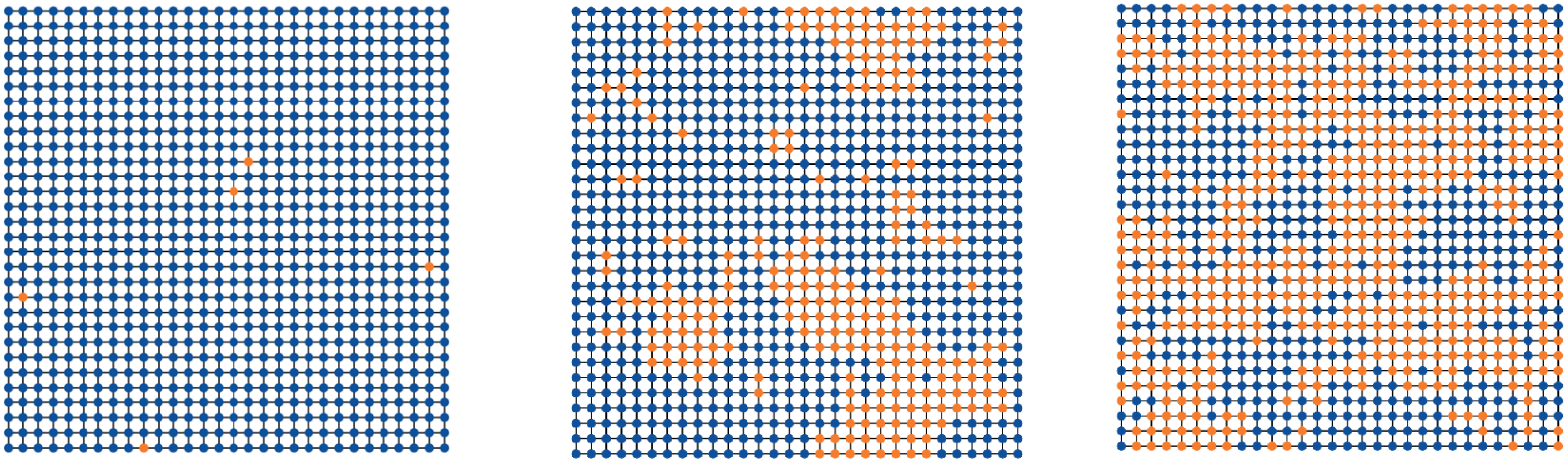
Which of these has the lowest energy?



A B C

$$H = -J \sum_{<i,j>} \sigma_i \sigma_j$$

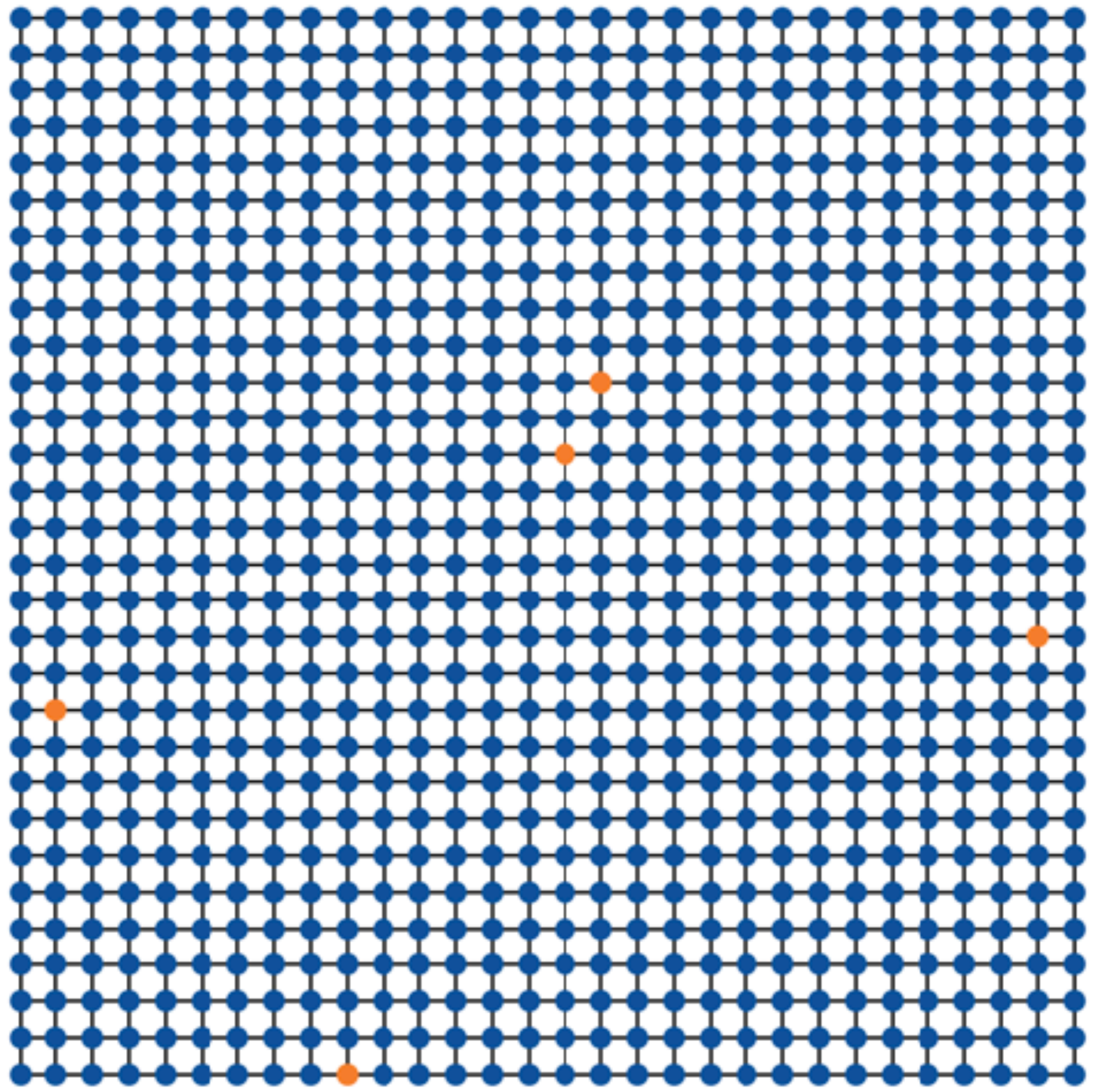
Ising model



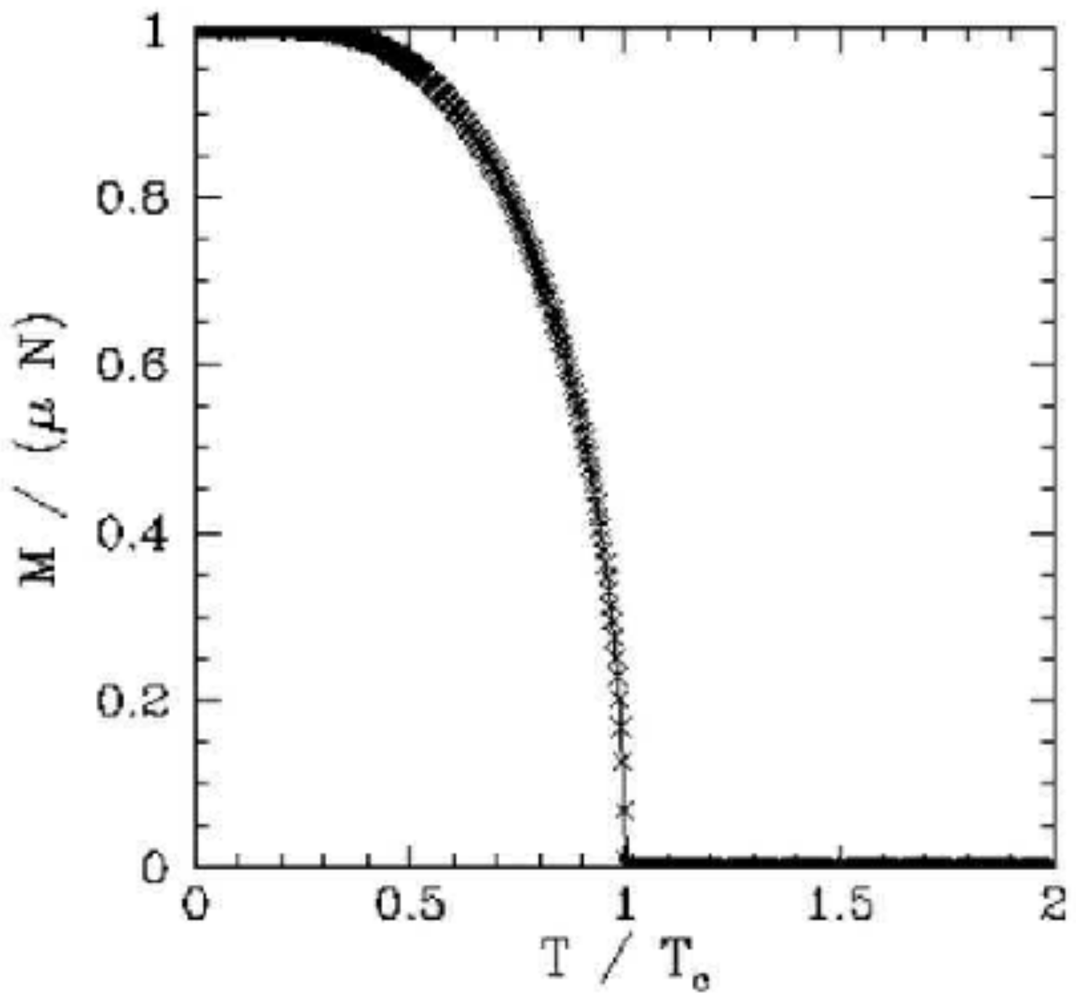
T=1.66 T=2.32 T=3.37

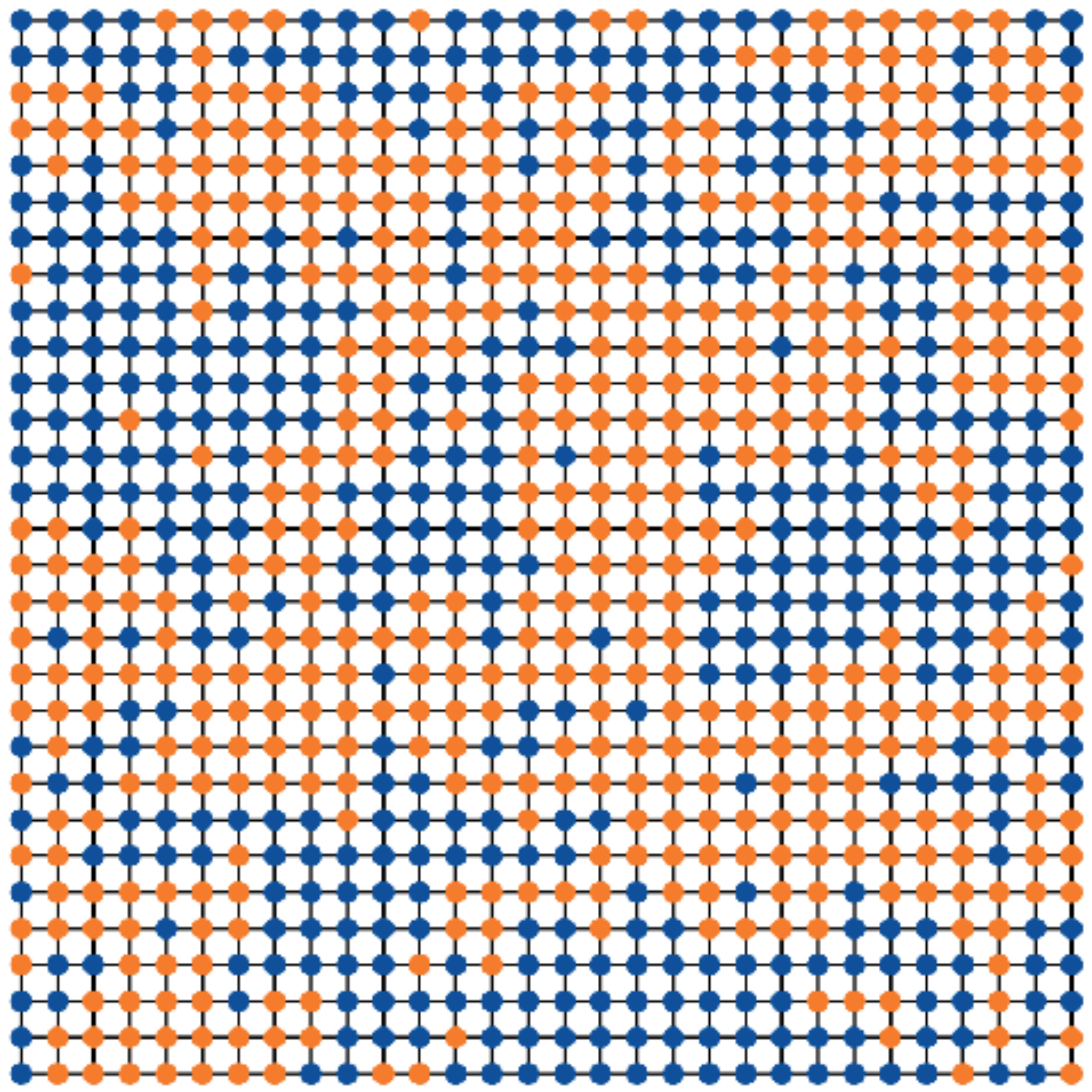
temperature

Phase Transition



$$T_c = \frac{2J}{k \ln(1+\sqrt{2})}$$

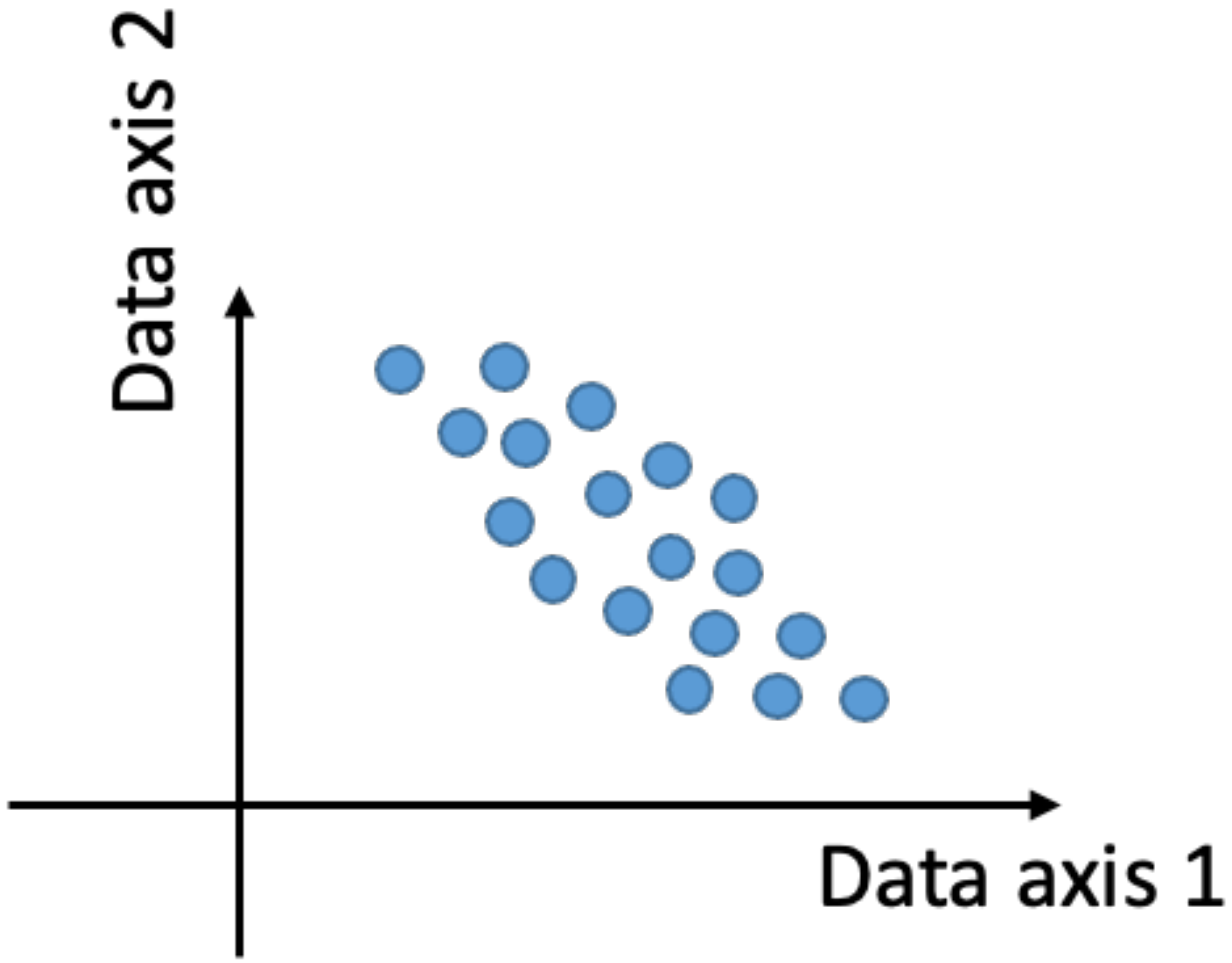




QMAI | TU Delft

Principal component analysis

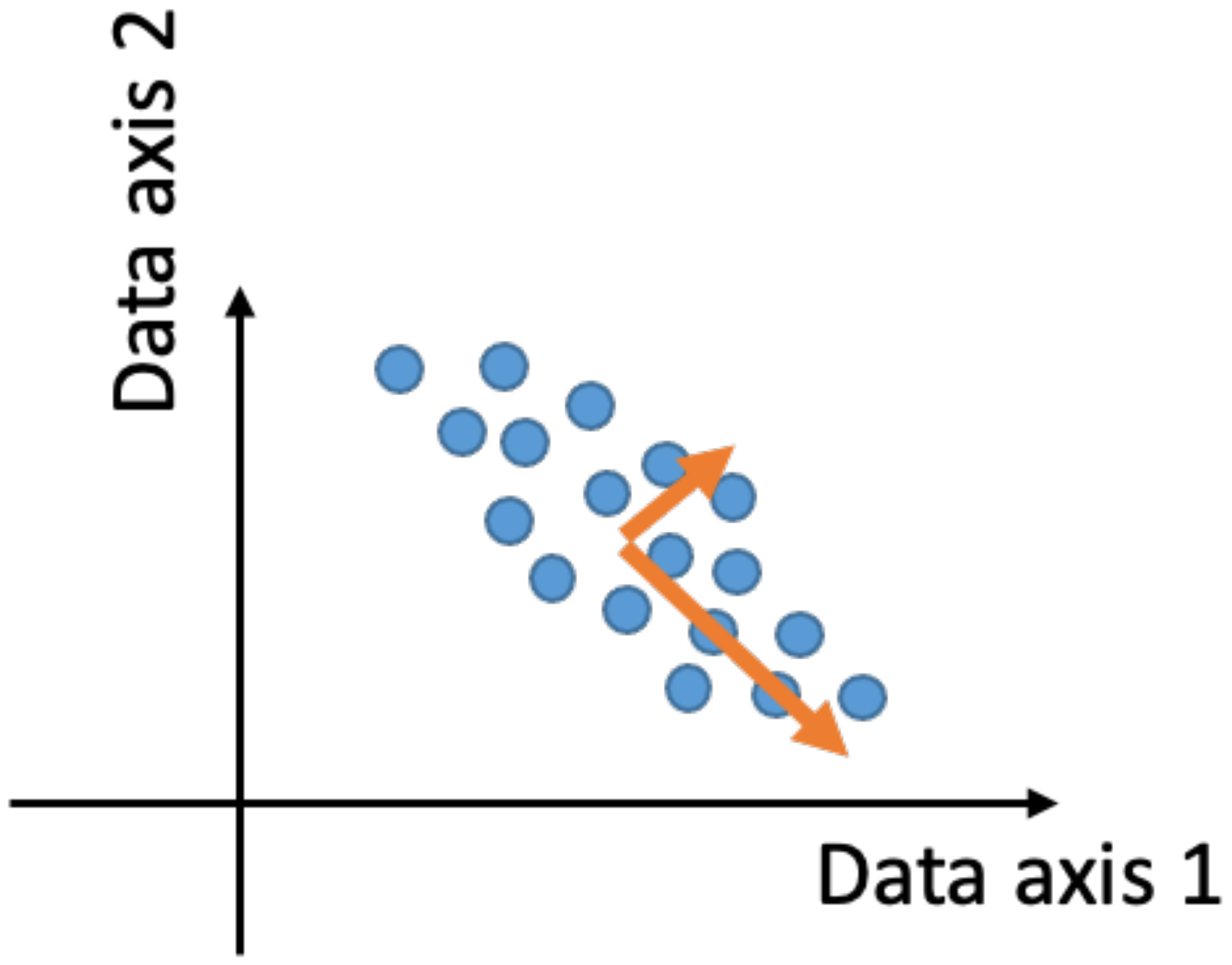
Data:
N points in k dimensions



Basic Idea of PCA

Data:
k points in N dimensions

PCA:
1. make p vectors that best follows the data (minimal square distance)



QMAI | TU Delft

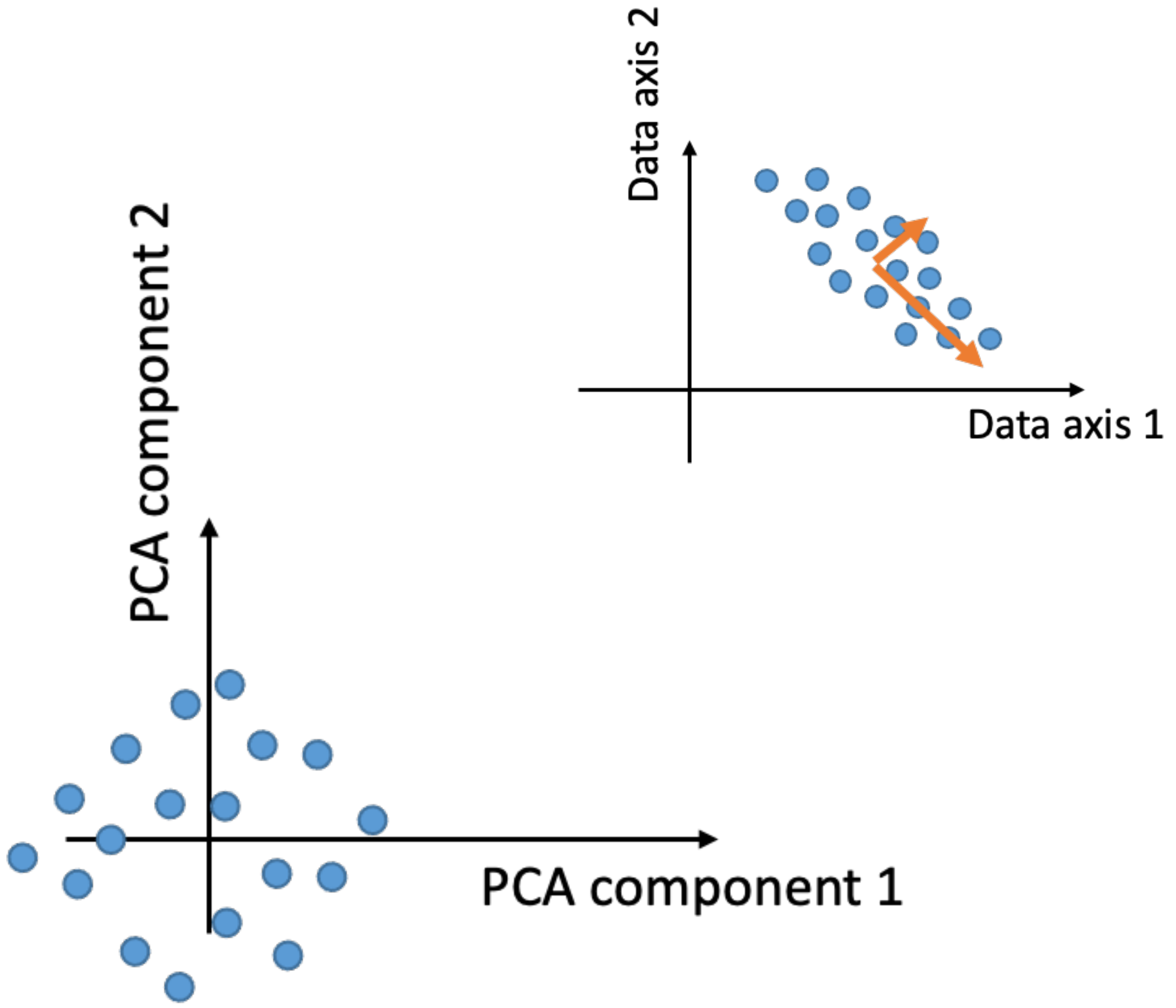
Basic Idea of PCA

Data:
k points in N dimensions

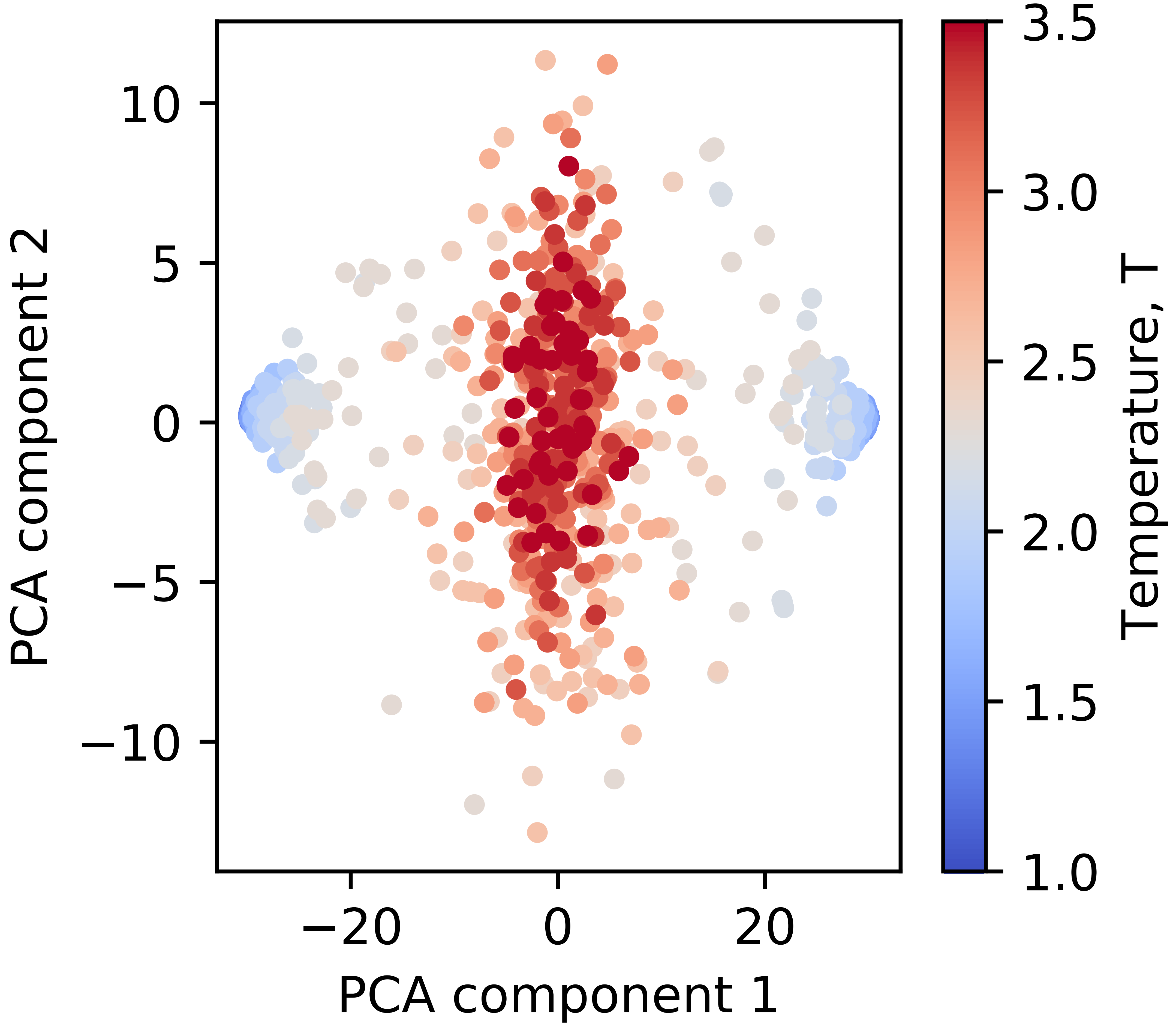
PCA:

1. make p vectors that best follows the data (minimal square distance)

2. project the data into PCA vectors



Simple clustering algorithm



Lei Wang, PHYSICAL REVIEW B 94, 195105 (2016)

PCA: The Math

Step 1: Construct matrix X such that each column is a data instance (N x k)

Step 2: Make each row zero mean

Step 3: Find vector w_1 that maximizes:

$$\frac{(w_1^T X^T X w_1)}{(w_1^T w_1)}$$

PCA: The Math

Step 1: Construct matrix X such that each column is a data instance (N x k)

Step 2: Make each row zero mean

Project the data

Step 3: Find vector w_1 that maximizes:

$$\frac{(w_1^T X^T X w_1)}{(w_1^T w_1)}$$

QMAI | TU Delft

PCA: The Math

Step 1: Construct matrix X such that each column is a data instance (N x k)

Step 2: Make each row zero mean

Step 3: Find vector w_1 that maximizes:

Take the mean

Project the data

$$\frac{(w_1^T X^T X w_1)}{(w_1^T w_1)}$$

PCA: The Math

QMAI | TU Delft

Step 1: Construct matrix X such that each column is a data instance (N x k)

Step 2: Make each row zero mean

Step 3: Find vector w_1 that maximizes:

Take the mean

Project the data

$$\frac{(w_1^T X^T X w_1)}{(w_1^T w_1)}$$

Normalize

PCA: The Math

Step 1: Construct matrix X such that each column is a data instance (N x k)

Step 2: Make each row zero mean

Step 3: Find vector w_1 that maximizes: $$\frac{(w_1^T X^T X w_1)}{(w_1^T w_1)}$$

Step 4: Find the next one:

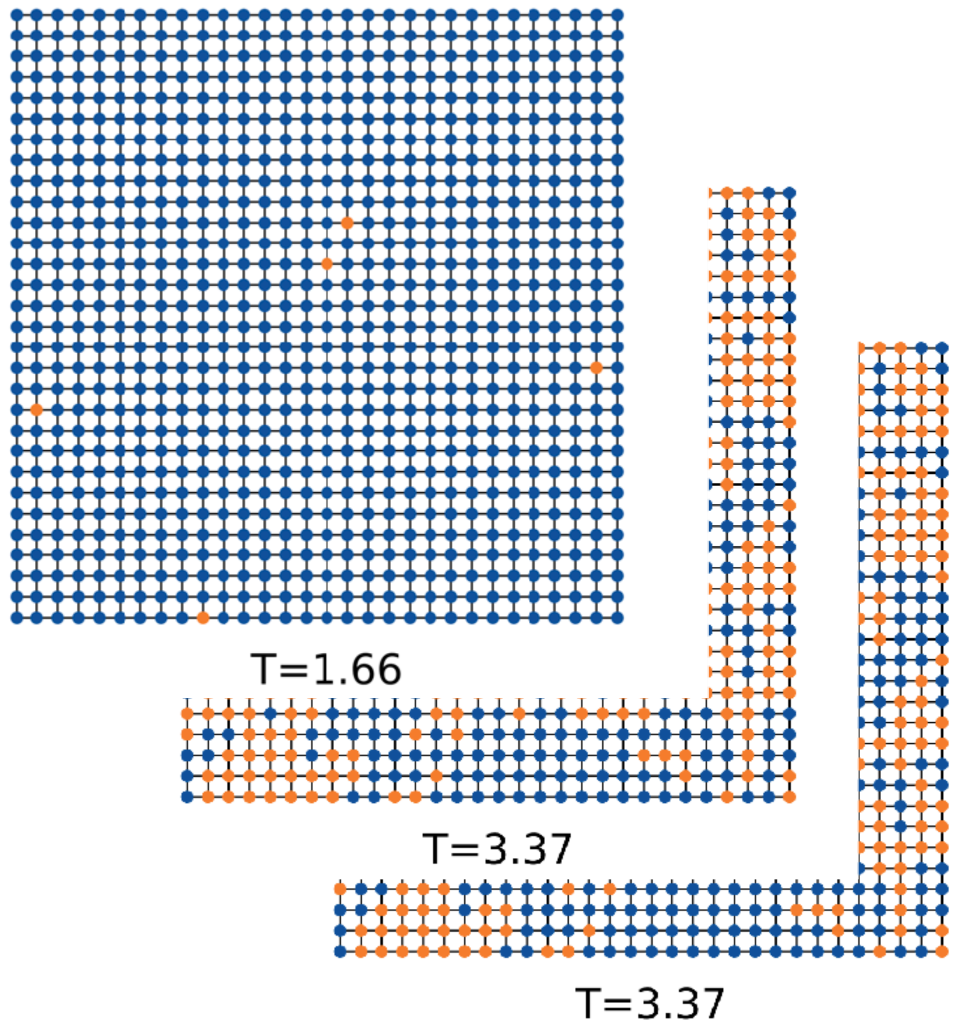
Substract the previous components: $$X_p = X - \sum_{s=1}^{p-1} X w_s w_s^T$$

Find w_p that maximises: $$\frac{w_p^T X_p^T X_p w_p}{w_p^T w_p}$$

QMAI | TU Delft

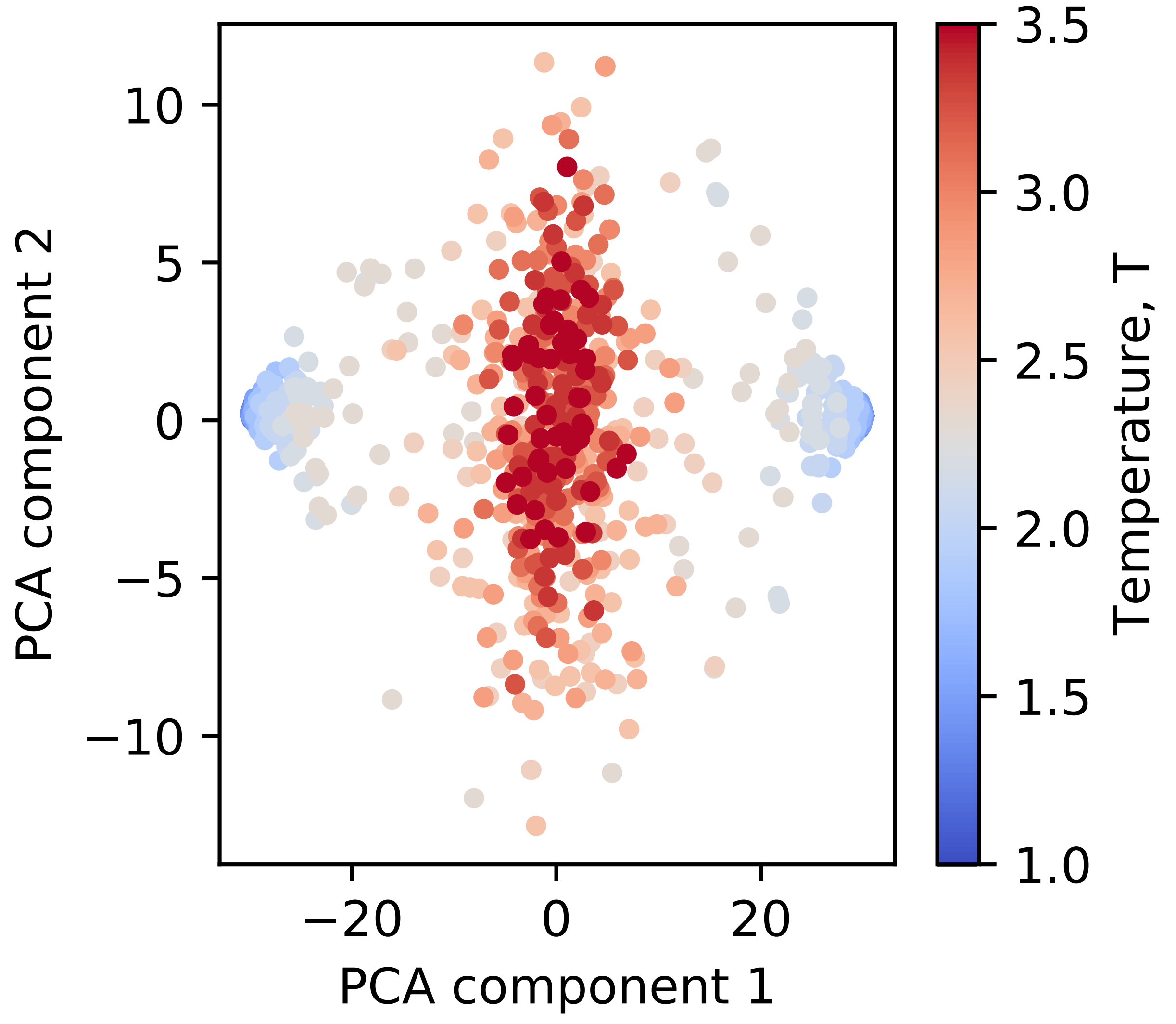
Exercise notebook 1

You will do PCA on this data:

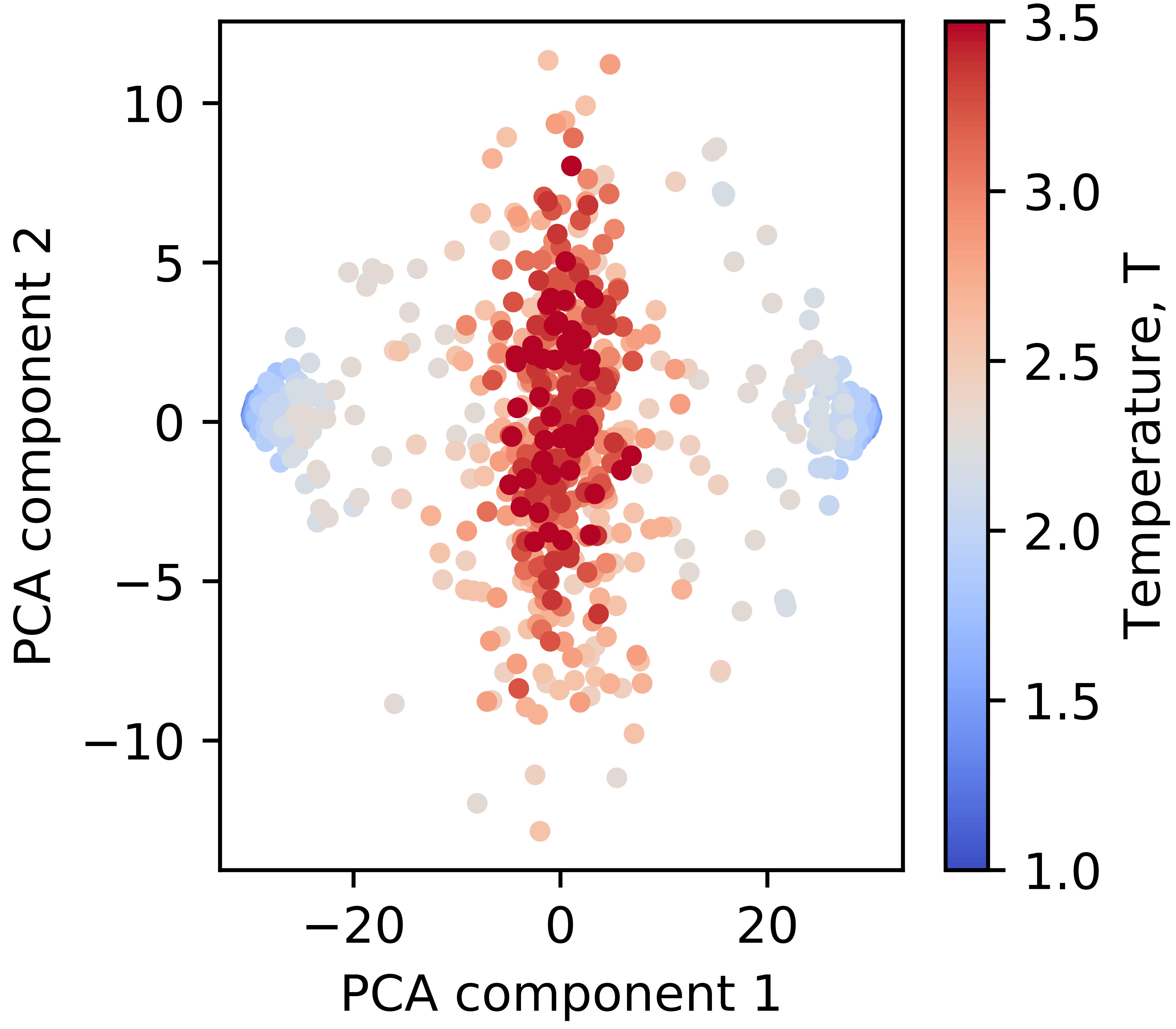


You will:

- flatten each 2D configuration into the vector
- create matrix X from these columns
- use singular values to find TWO PCA components



Data driven vs. toy model driven



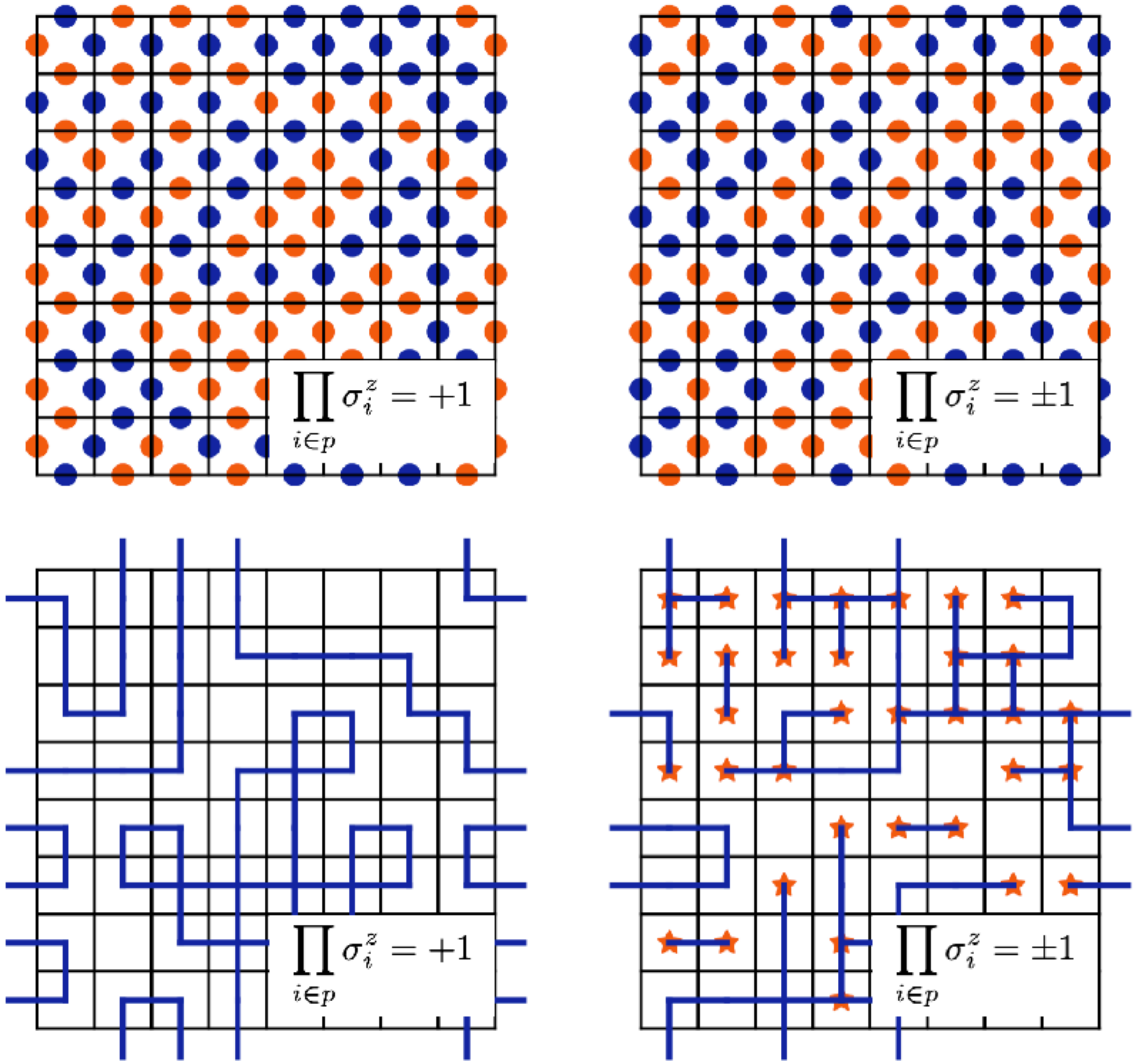
$$H = -J \sum_{<i,j>} \sigma_i \sigma_j$$

$$p(\sigma) = \frac{e^{-\beta H(\sigma)}}{Z}$$

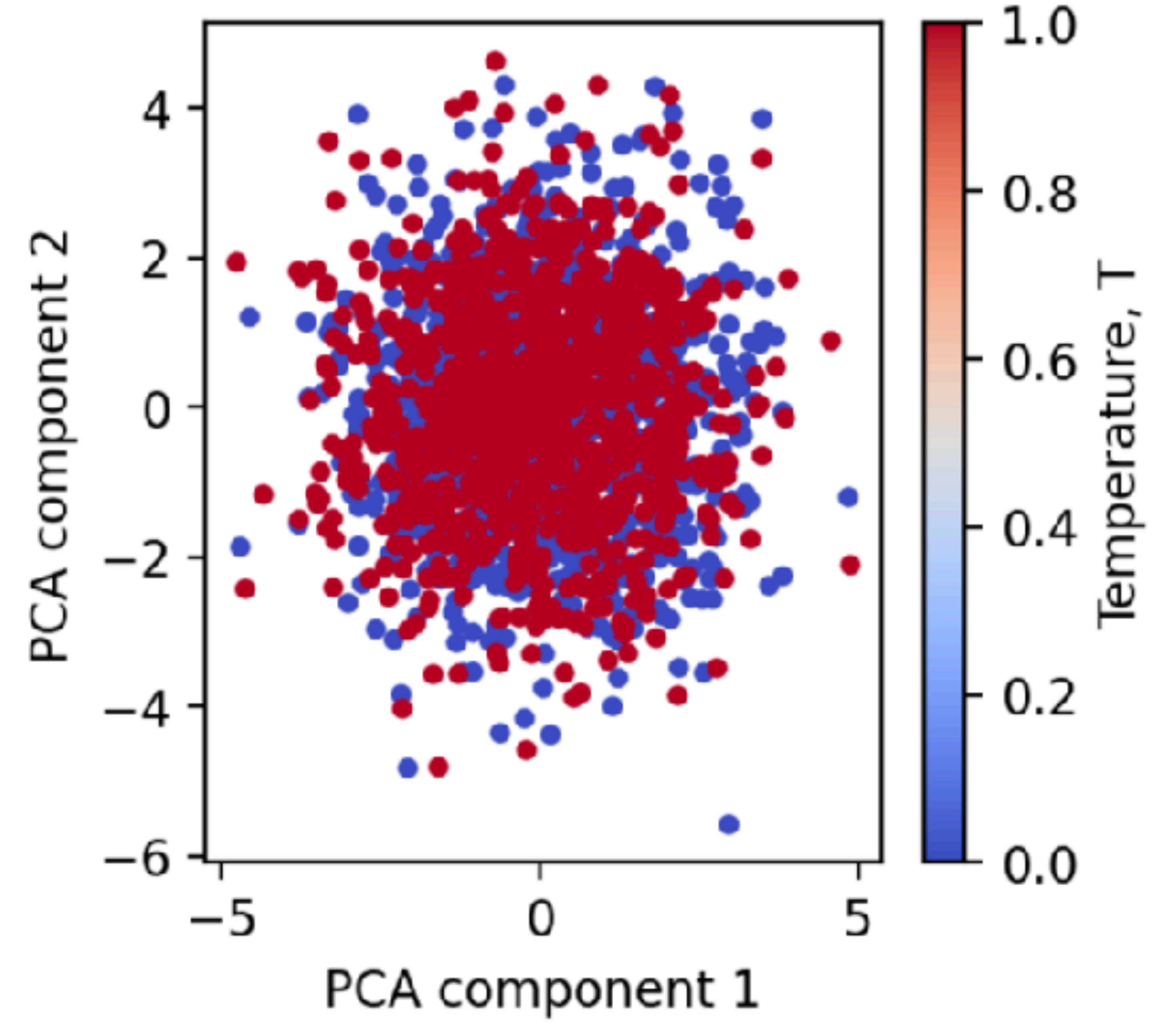
$$T_c = \frac{2J}{k \ln(1+\sqrt{2})}$$ ~ 2.27

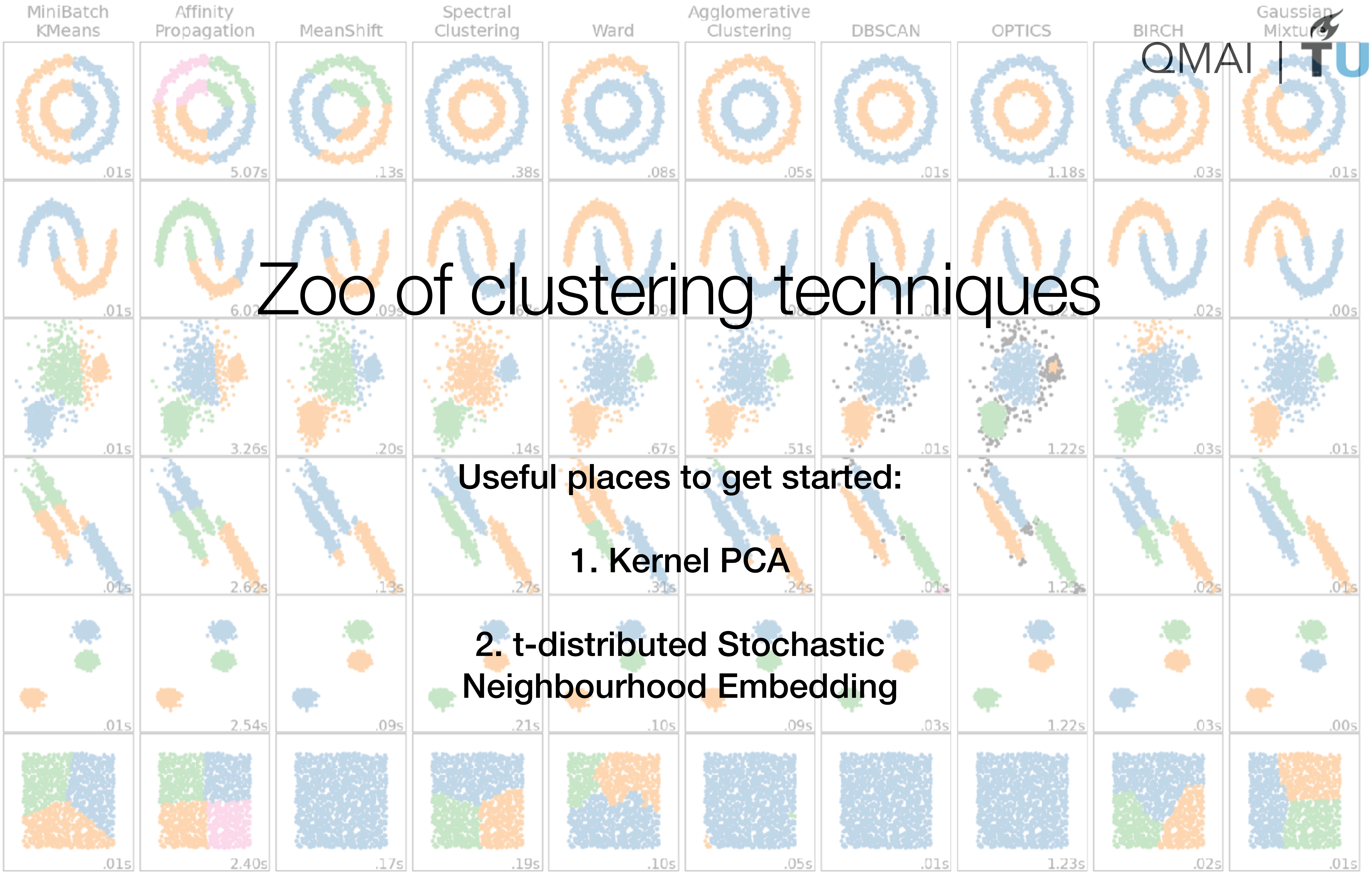
Lei Wang, PHYSICAL REVIEW B 94, 195105 (2016)

Also in Exercise notebook 1



$$H_{IGT} = -J\sum_p \prod_{i\in p}\sigma_i^z$$





QMAI | TU Delft

Zoo of clustering techniques

Useful places to get started:

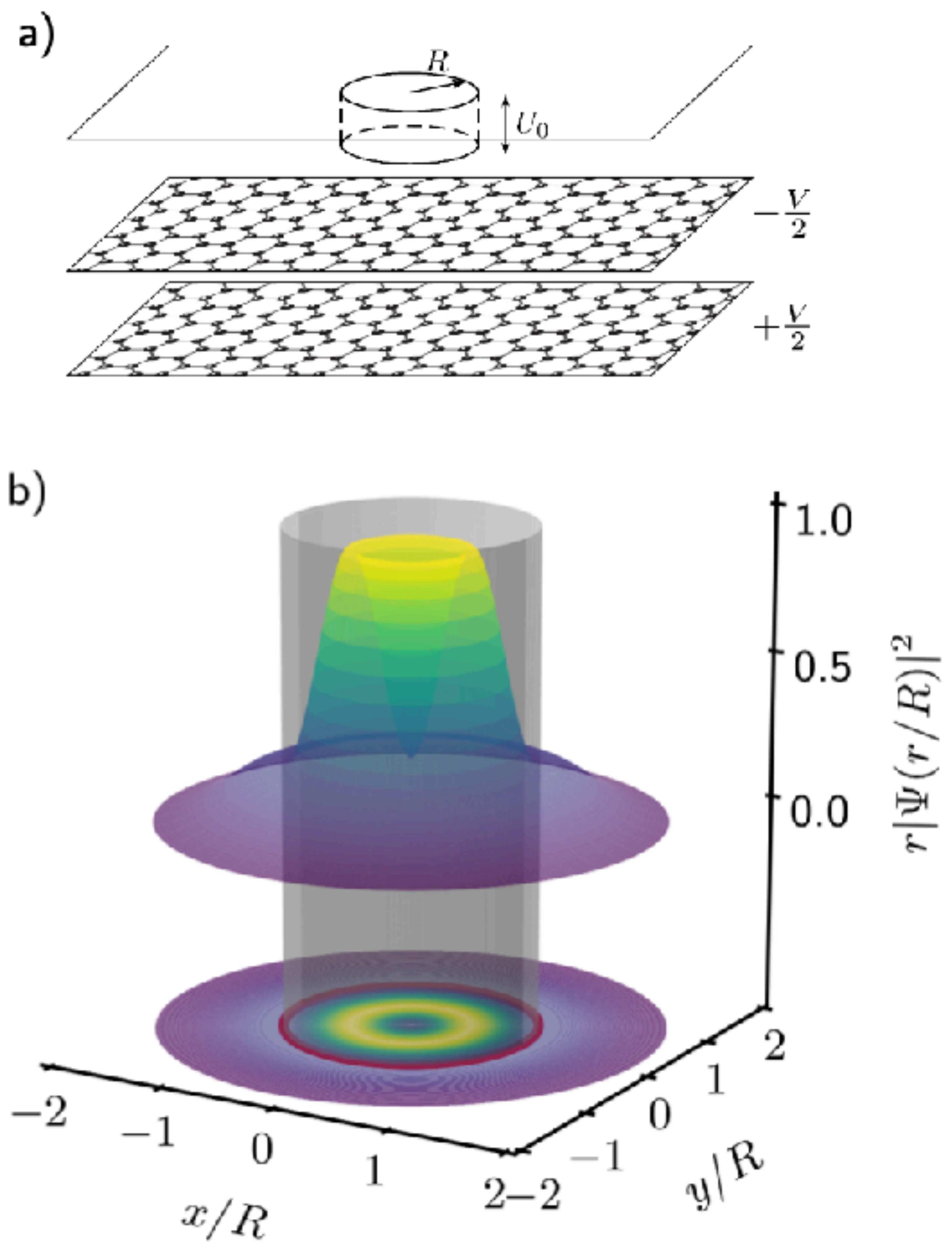
1. **Kernel PCA**

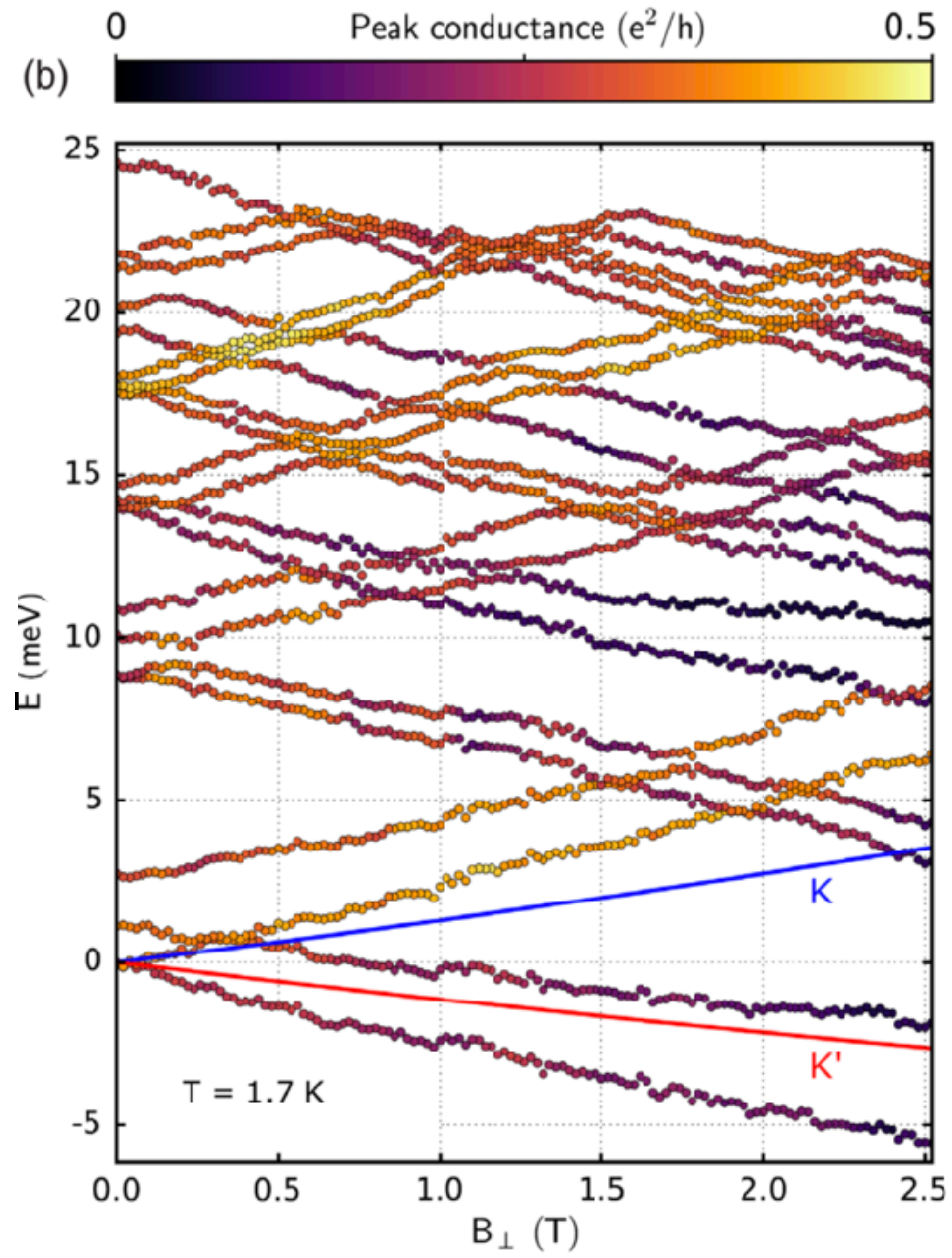
2. **t-distributed Stochastic Neighbourhood Embedding**

Scikit Learn

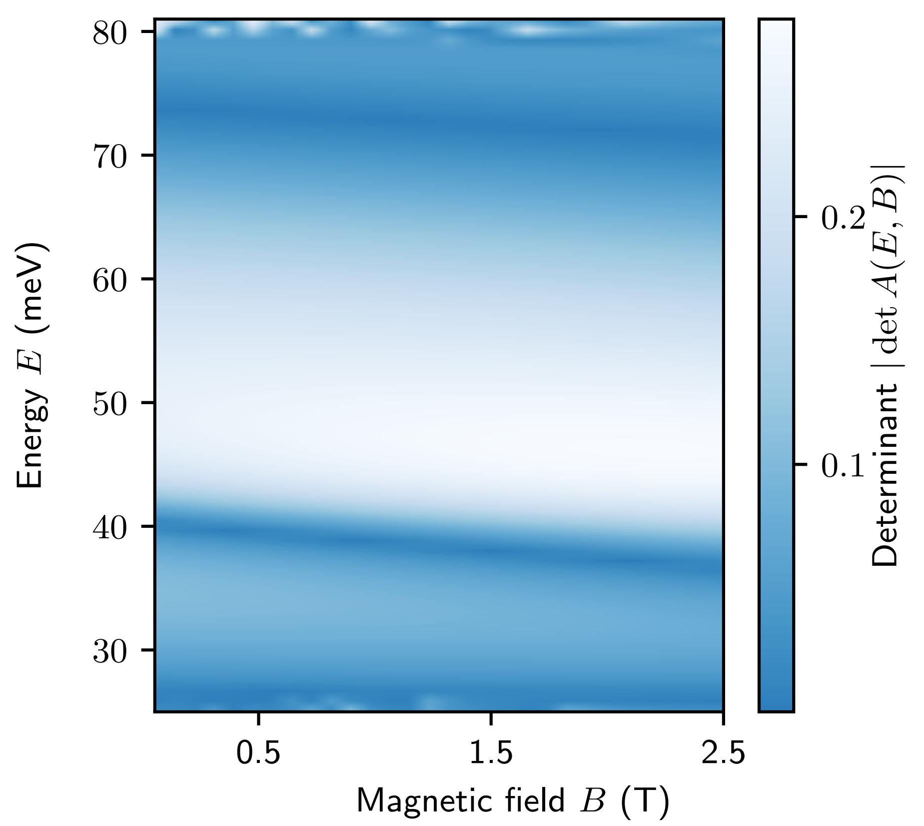
QMAI | TUDelft

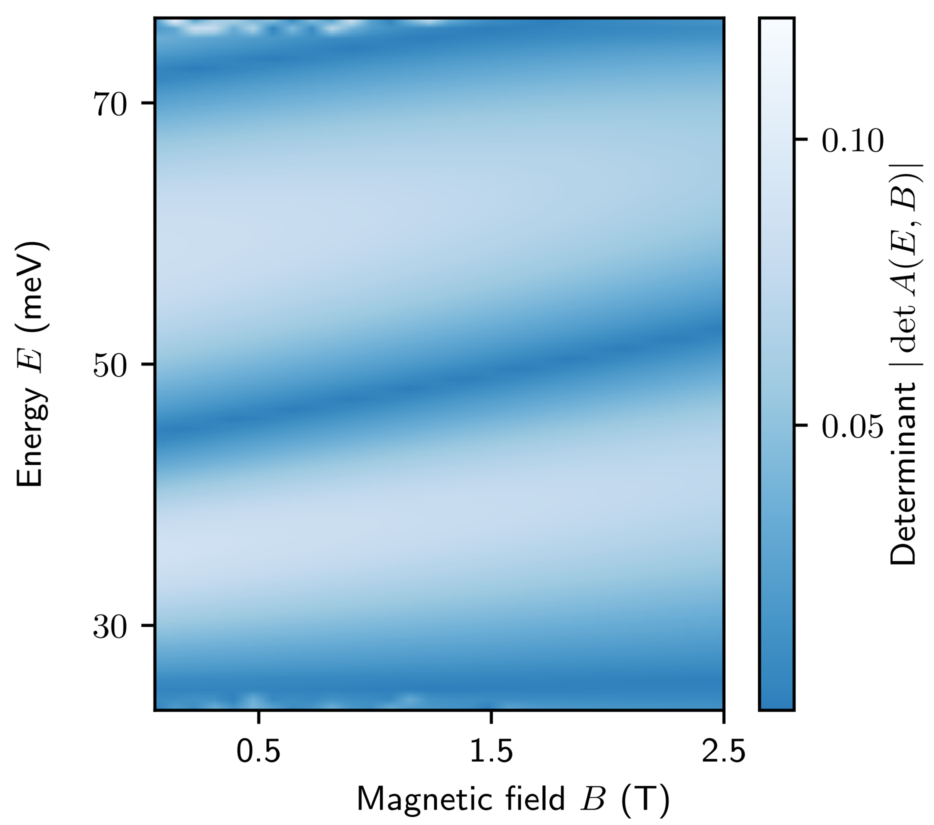
Clustering in QMAI research

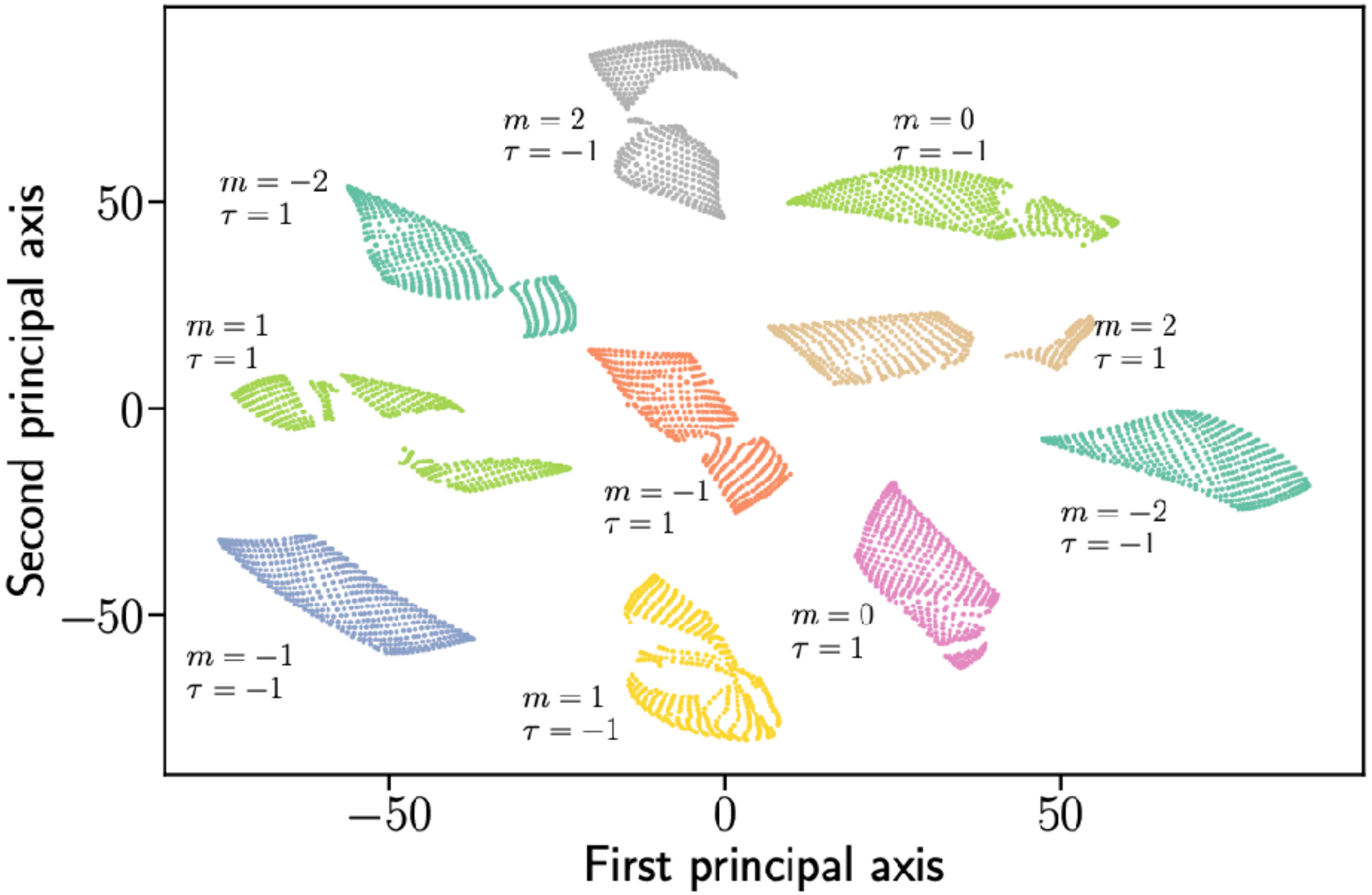




Phys. Rev. X **8**, 031023 (2018)

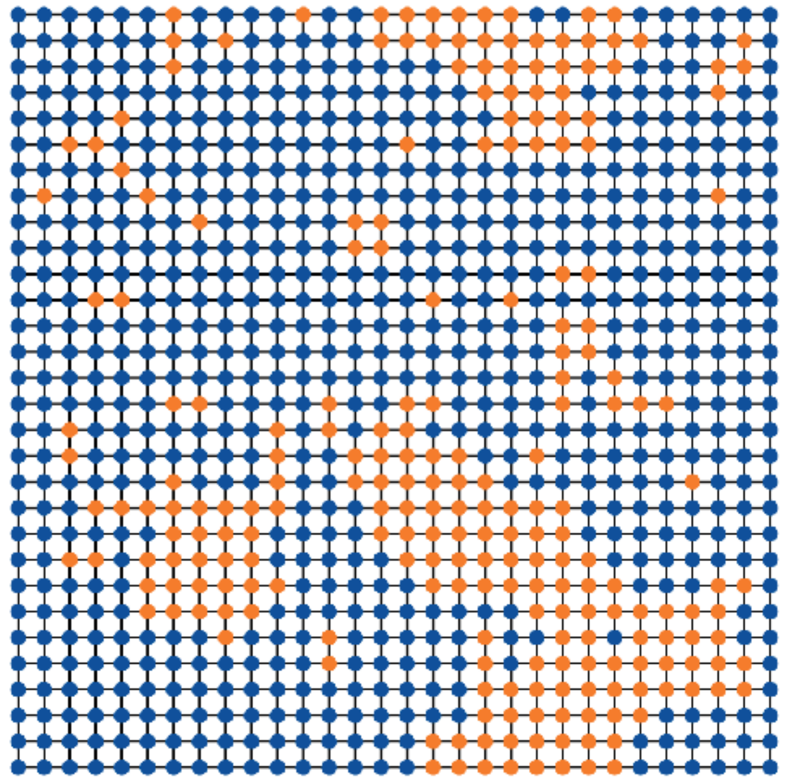

Energy E (meV)
Magnetic field B (T)
Determinant $|\det A(E,B)|$
80
70
60
50
40
30
0.5
1.5
2.5
0.2
0.1


Energy E (meV)
Magnetic field B (T)
Determinant $|\det A(E,B)|$
70
50
30
0.5
1.5
2.5
0.10
0.05

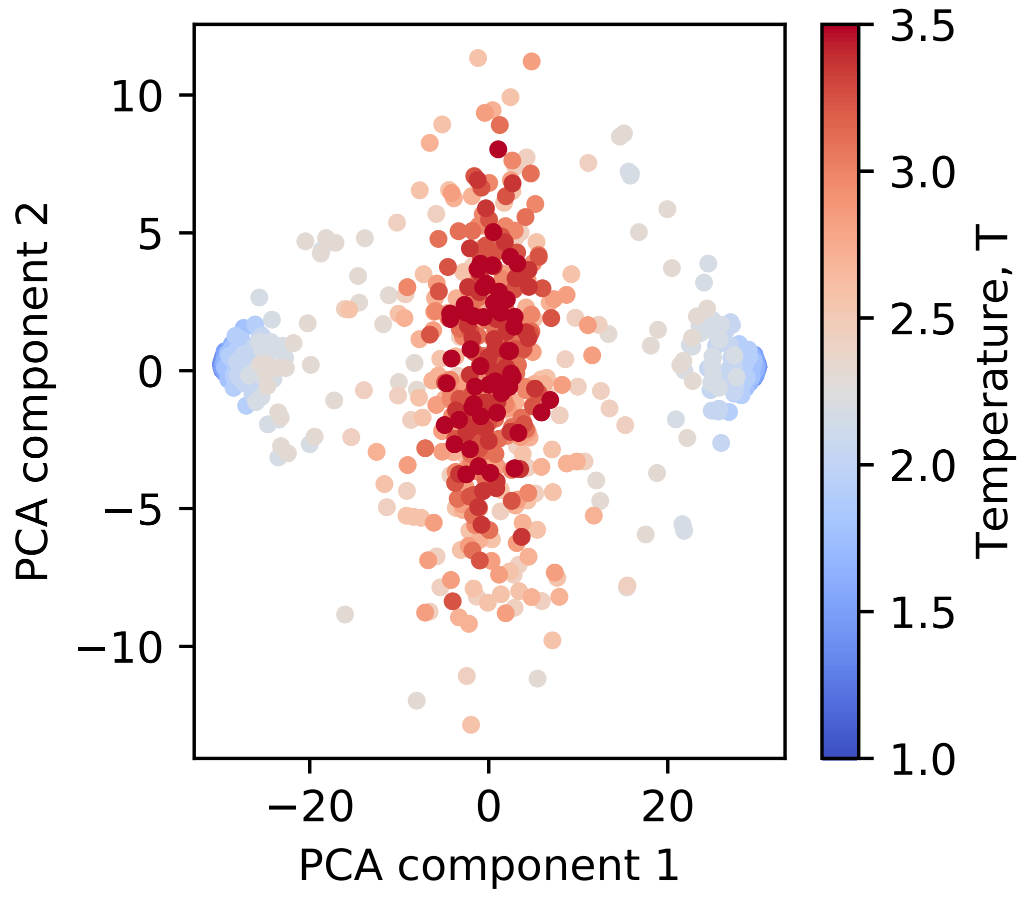

Second principal axis
First principal axis
50
0
−50
$m=2$ $\tau=-1$
$m=0$ $\tau=-1$
$m=-2$ $\tau=1$
$m=1$ $\tau=1$
$m=2$ $\tau=1$
$m=-1$ $\tau=1$
$m=-2$ $\tau=-1$
$m=0$ $\tau=1$
$m=-1$ $\tau=-1$
$m=1$ $\tau=-1$

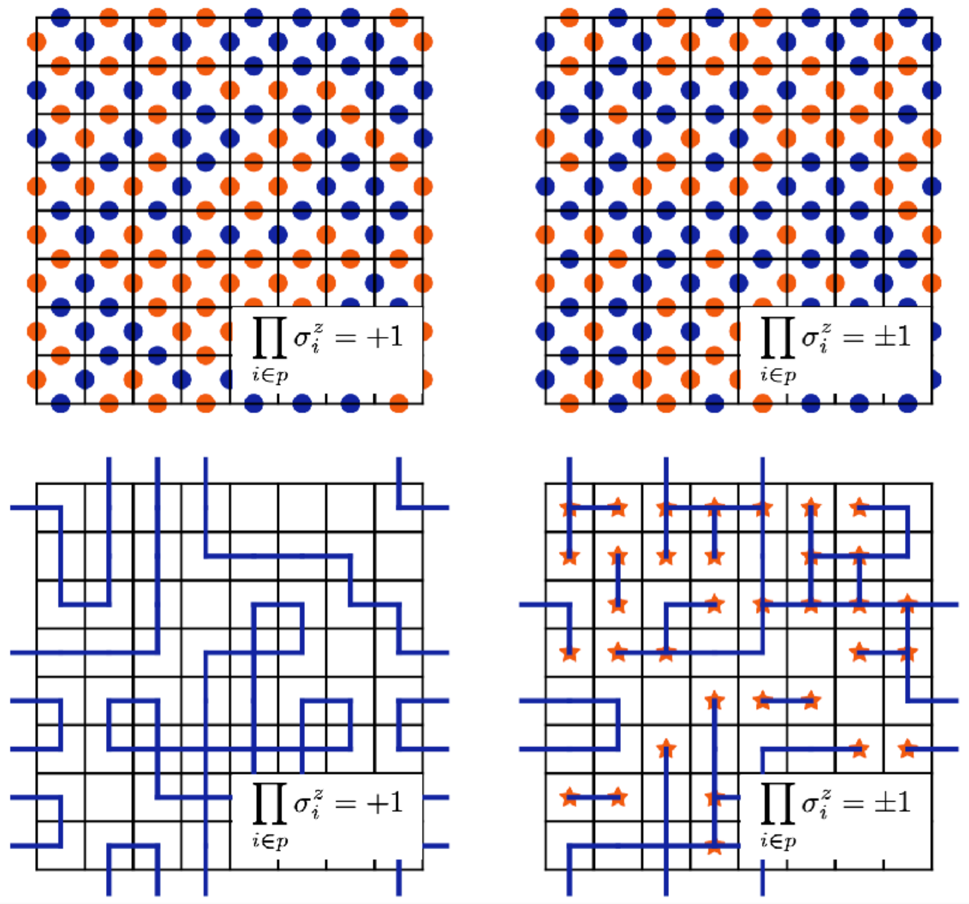
BREAK 15 mins

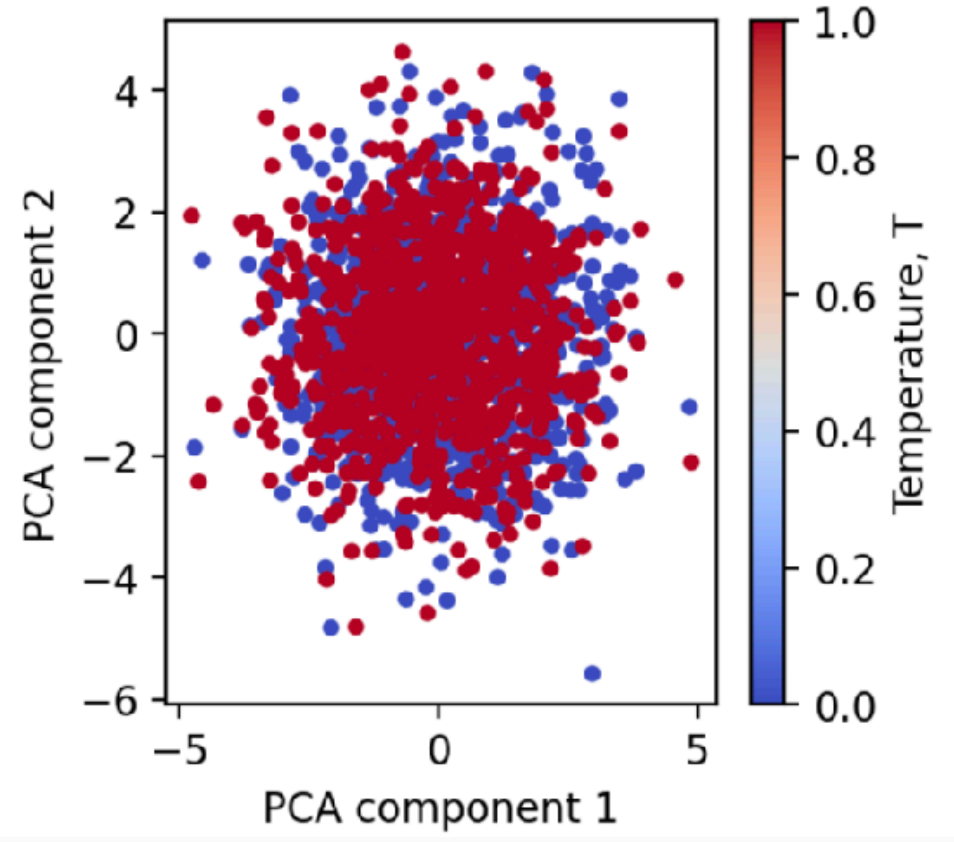
Before the break..



T=2.32

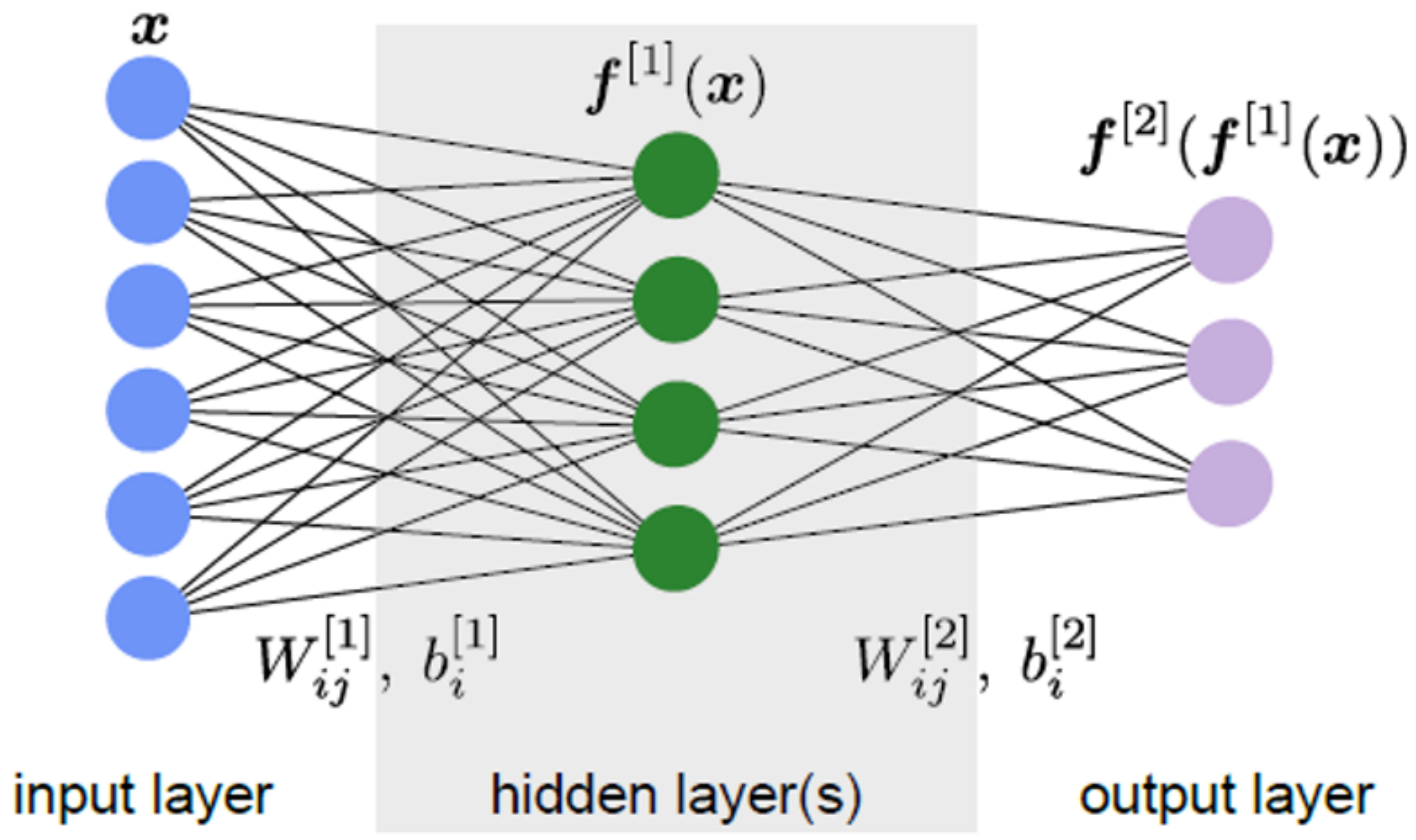






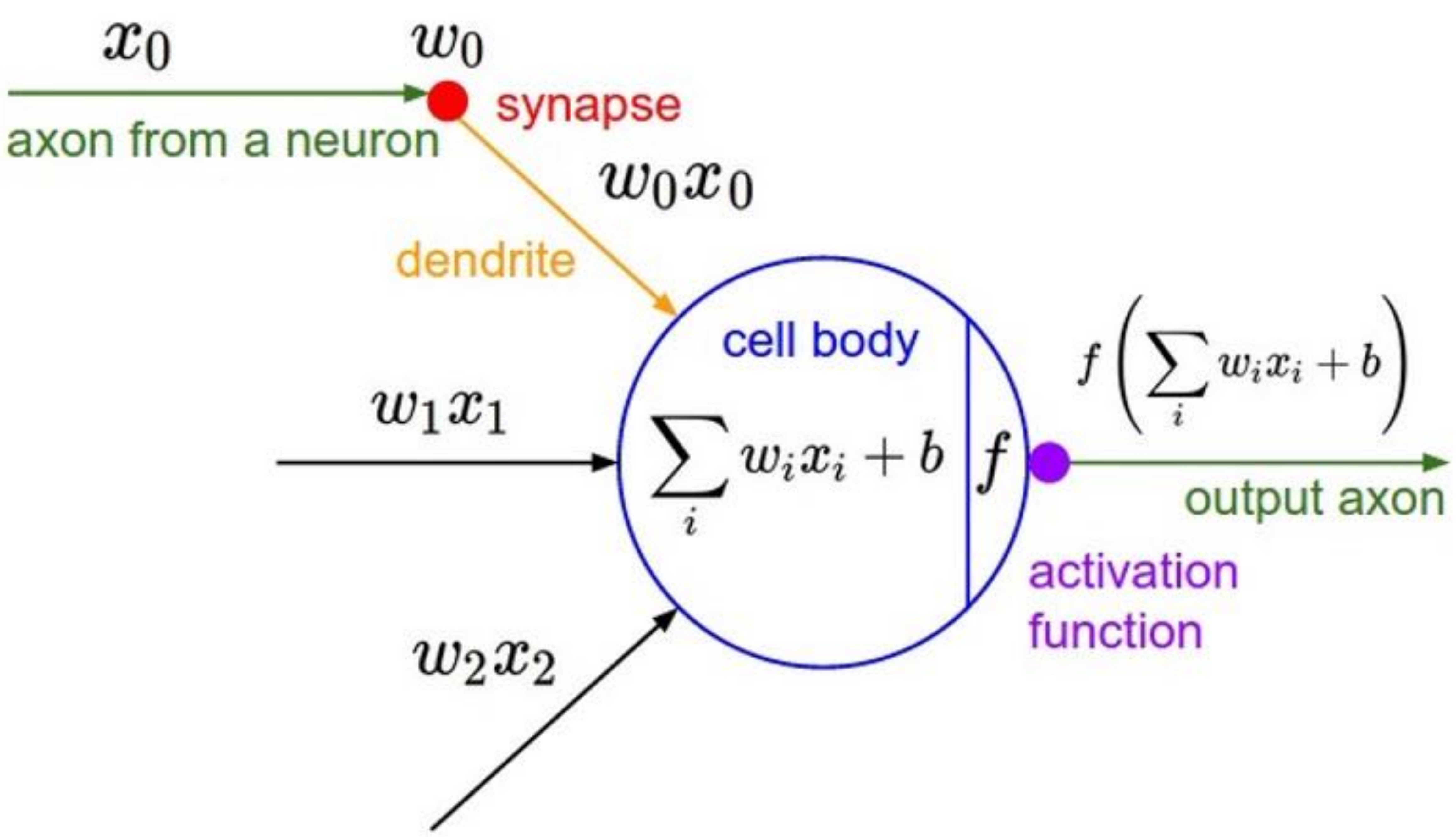
QMAI | TU Delft

Neural Networks 101: Supervised learning



QMAI | TU Delft

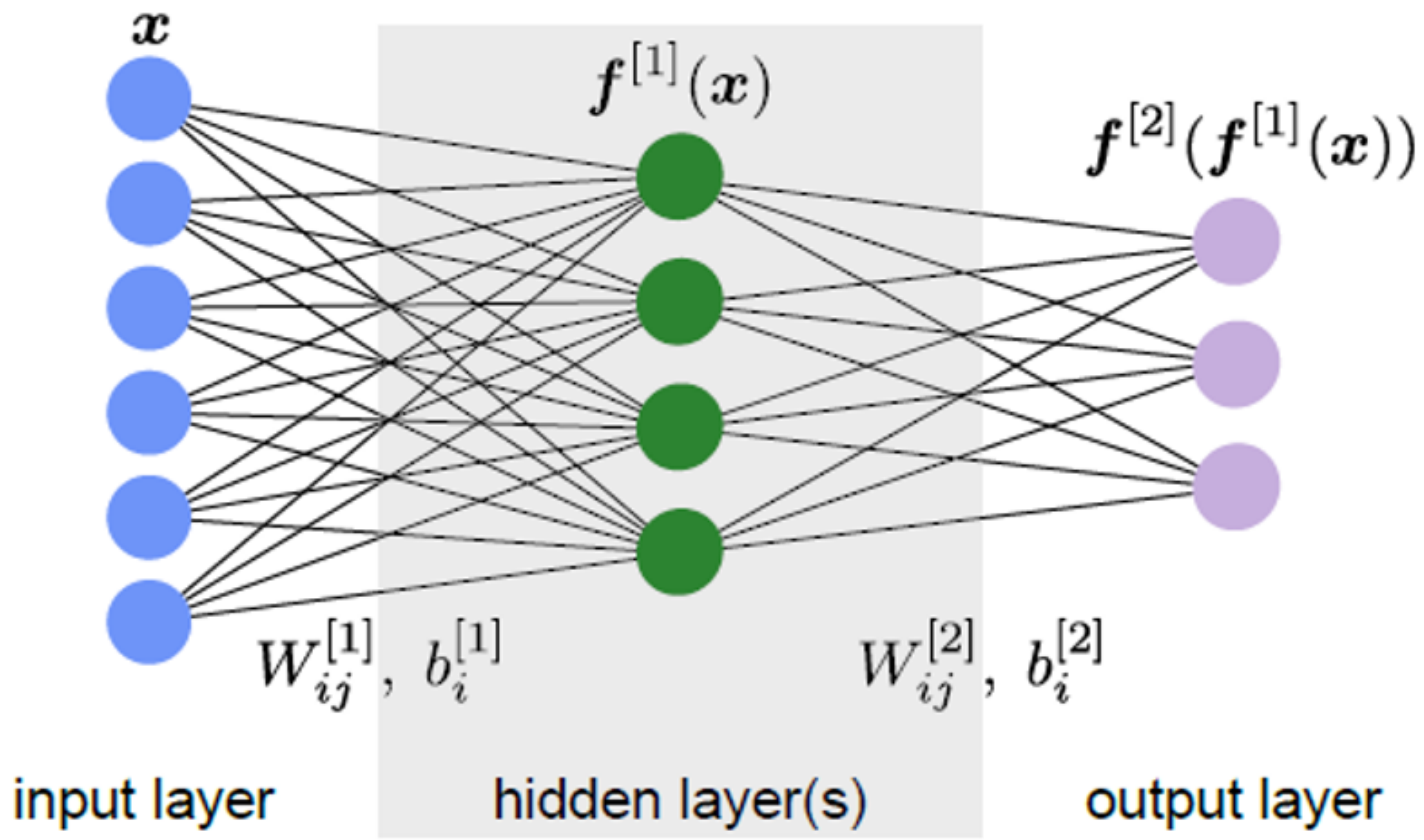
What happens in a neuron?



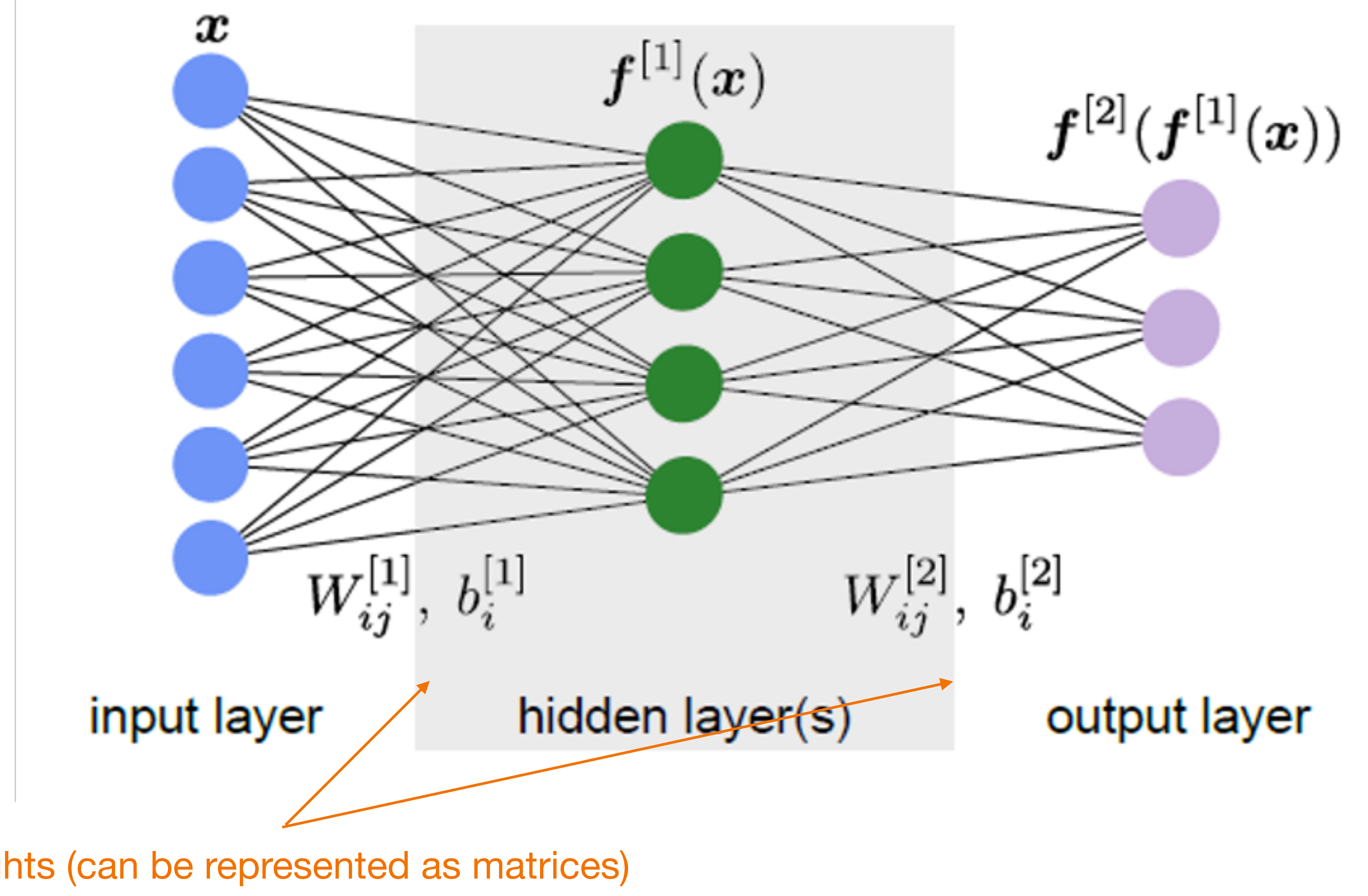
From https://cs231n.github.io

QMAI | TU Delft

Putting neurons in the networks

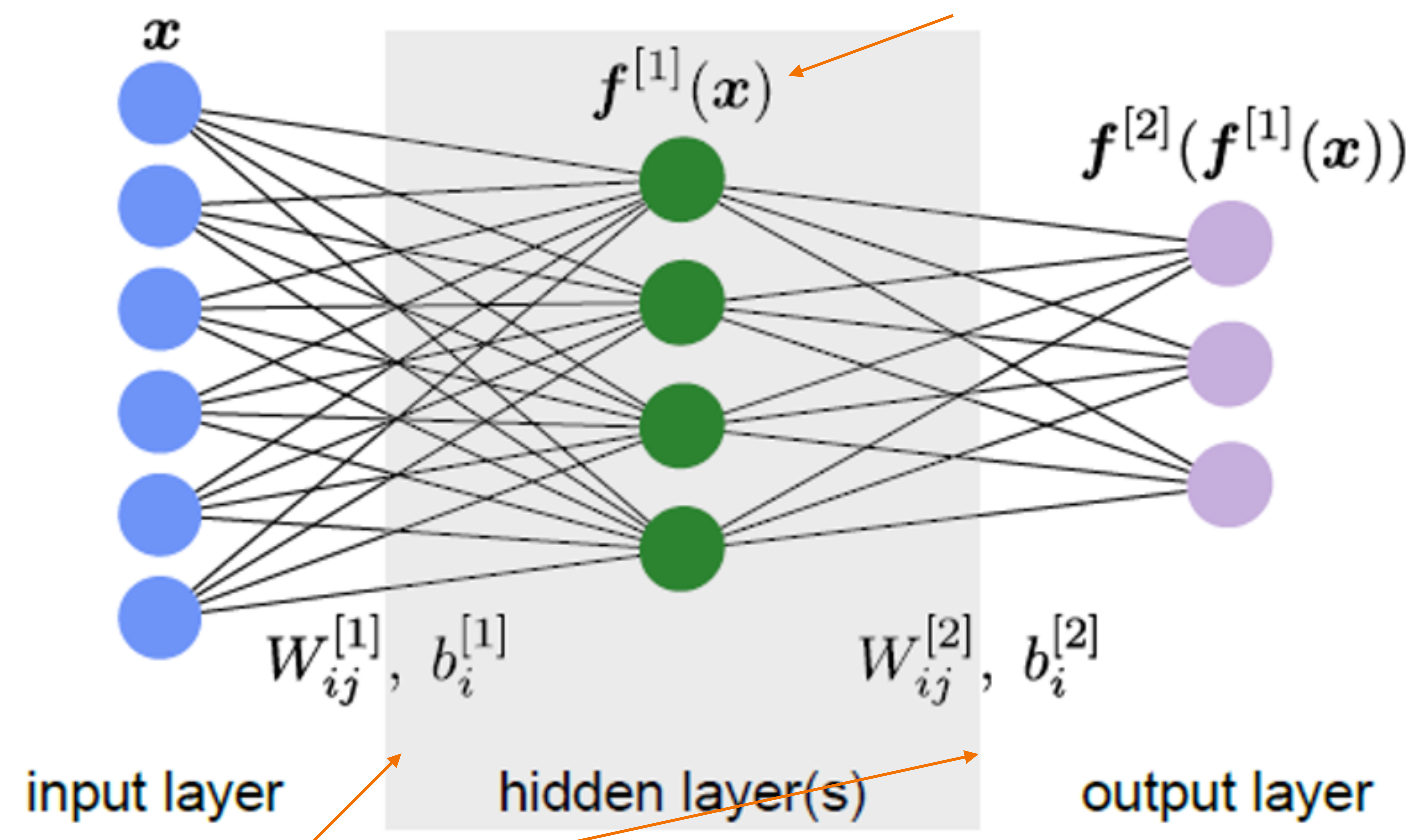


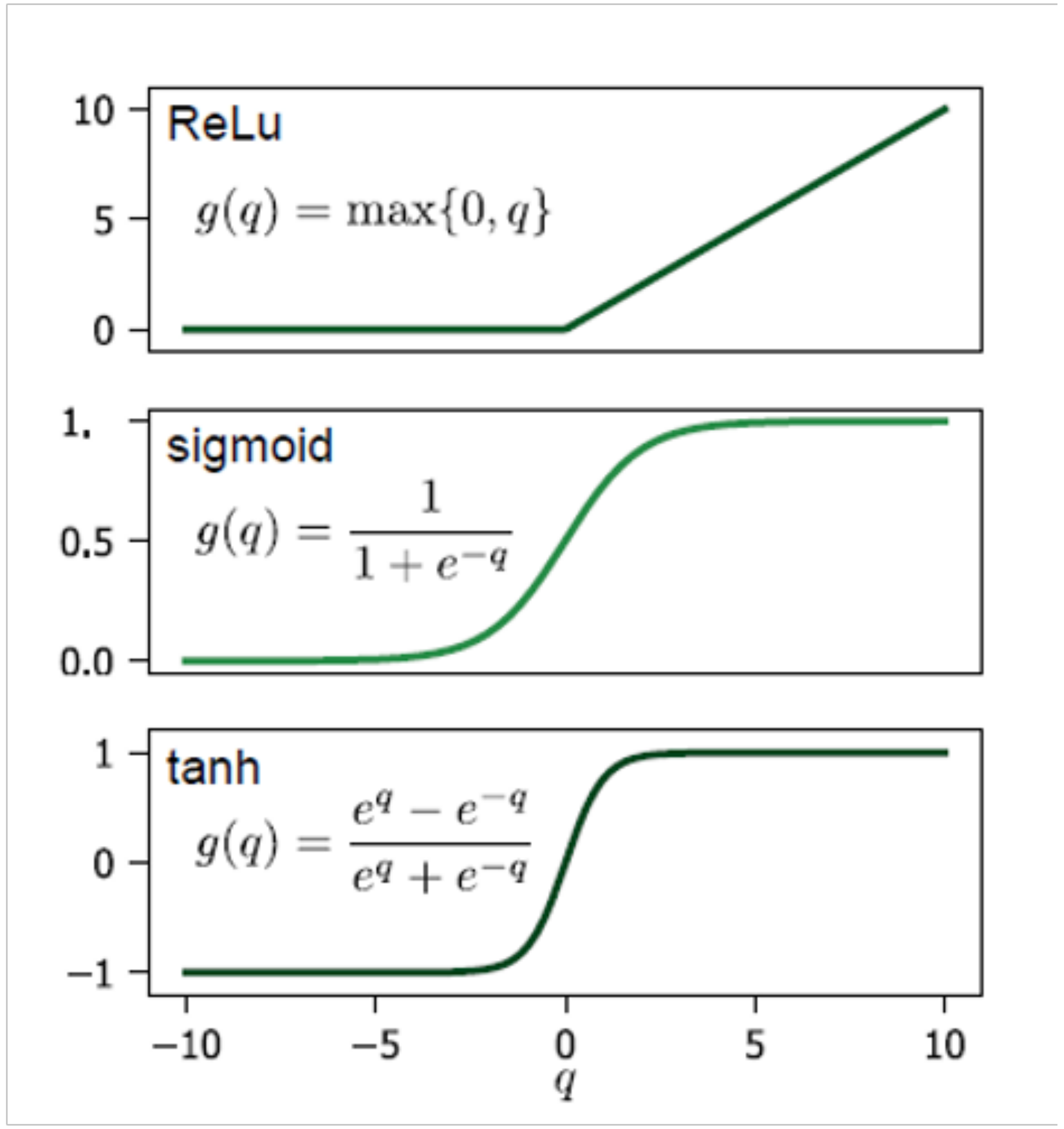
Putting neurons in the networks



Putting neurons in the networks

Activation functions

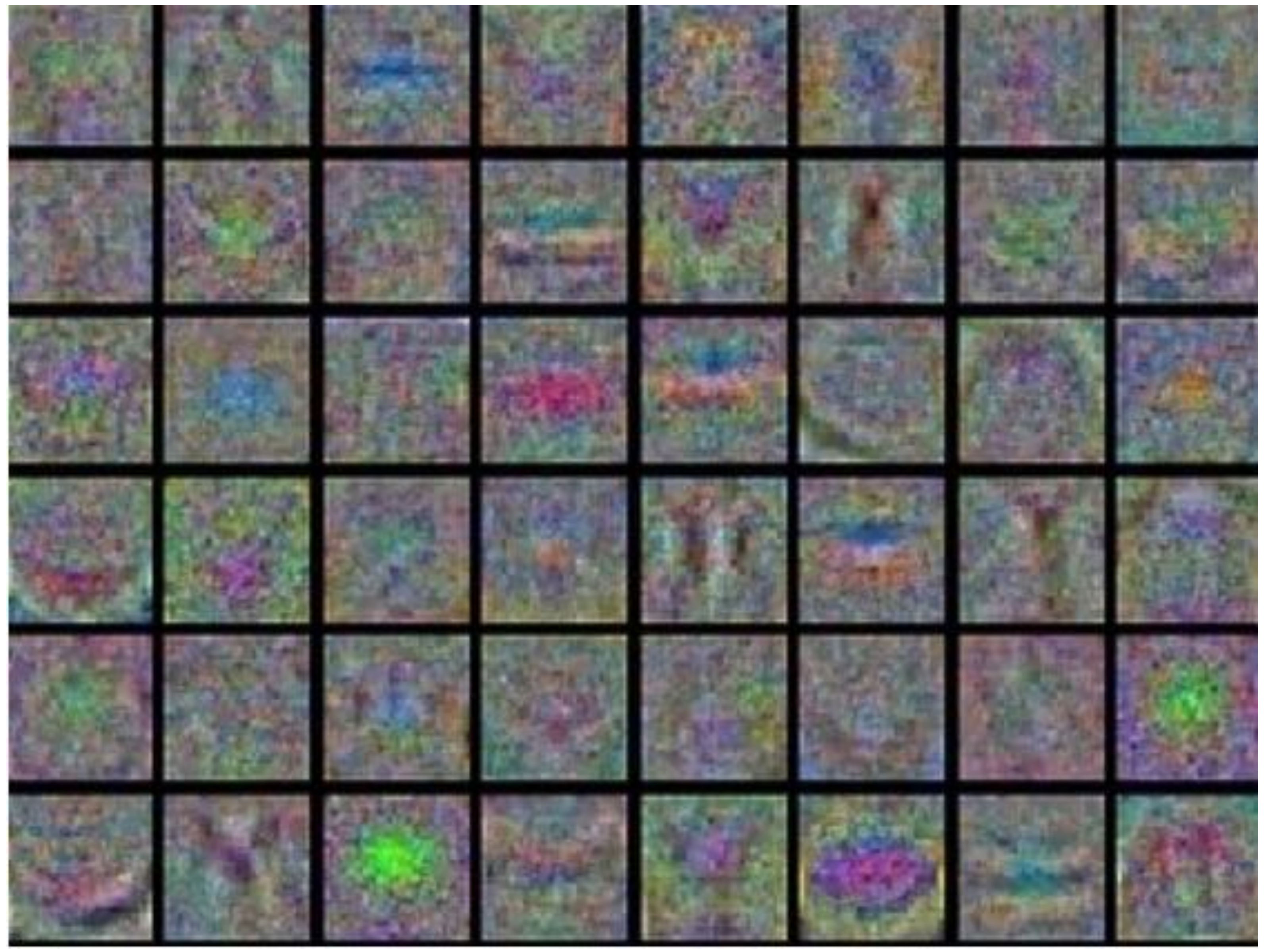


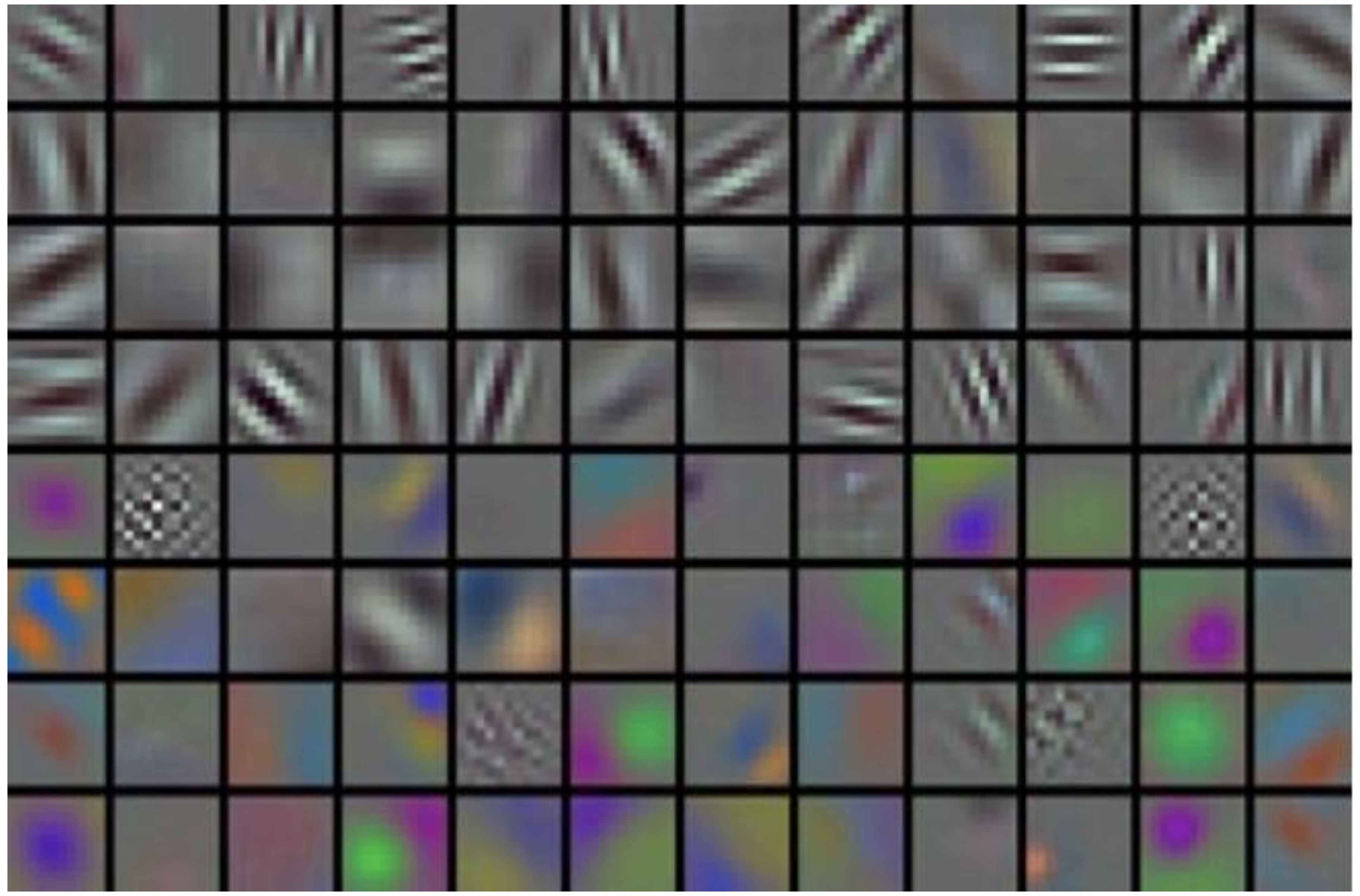


Weights (can be represented as matrices)

QMAI | TU Delft

How do the weights look like?





Weight matrices from a image recognition network:

- Left: badly trained network – the weights look noisy
- Right: well trained network – the weights clearly extract specific features in the data

From https://cs231n.github.io

How do we train neural network?

- Define the loss function L(x) [x = our data] that we will minimise during training
- Calculate the derivative of L wrt weights and biases

$$W_{ij} \rightarrow W_{ij} - \epsilon \frac{\partial L}{\partial W_{ij}}$$

How do we train neural network?

- Define the loss function L(x) [x = our data] that we will minimise during training
- Calculate the derivative of L wrt weights and biases

$$W_{ij} \rightarrow W_{ij} - \epsilon \frac{\partial L}{\partial W_{ij}}$$

learning rate

How do we calculate the derivative?

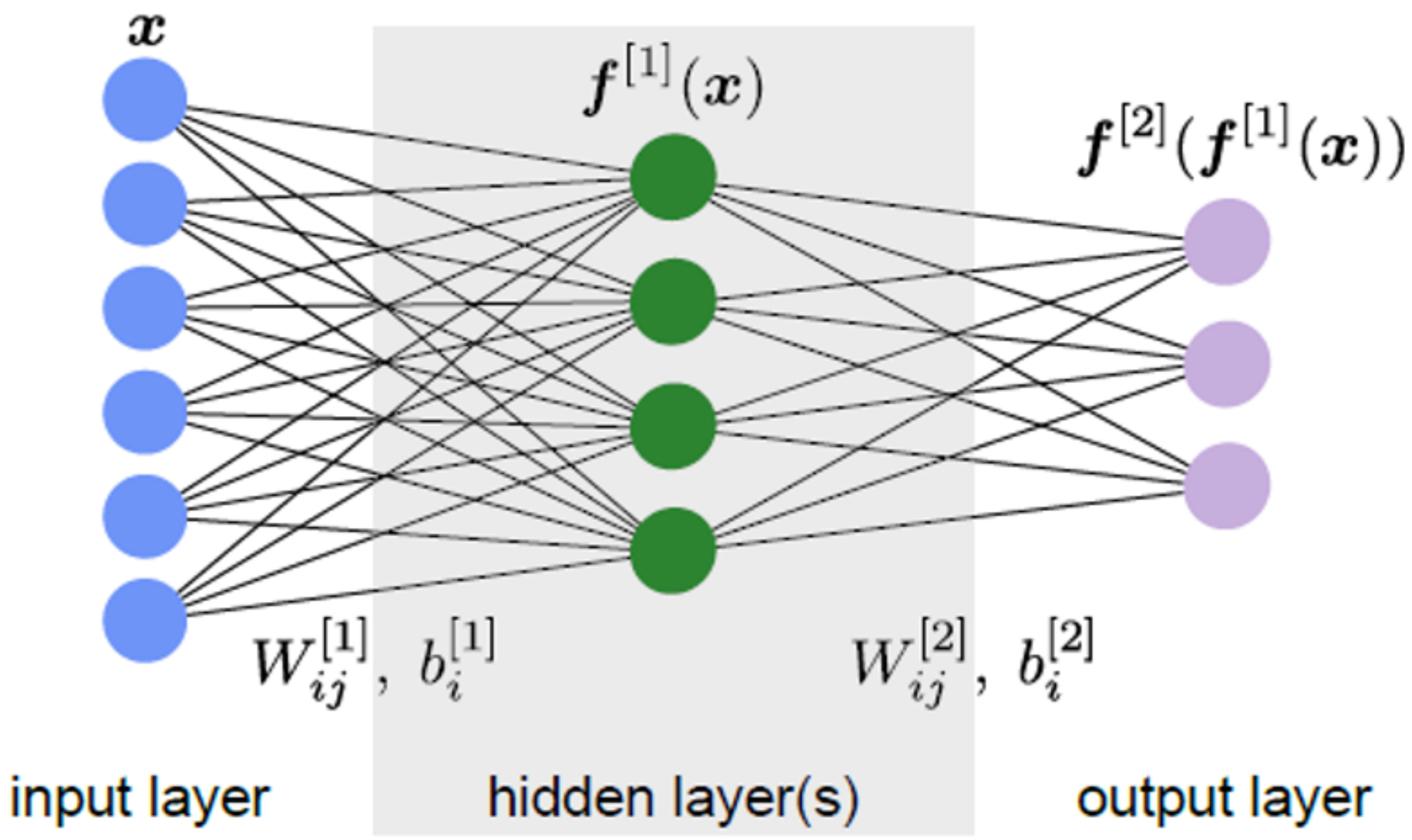
- CHAIN RULE REMINDER:

$$f(x) = g(h(x)) => \frac{df}{dx} = \frac{dg}{dh} * \frac{dh}{dx}$$

How do we calculate the derivative?

- CHAIN RULE REMINDER: $f(x) = g(h(x)) => \frac{df}{dx} = \frac{dg}{dh} * \frac{dh}{dx}$

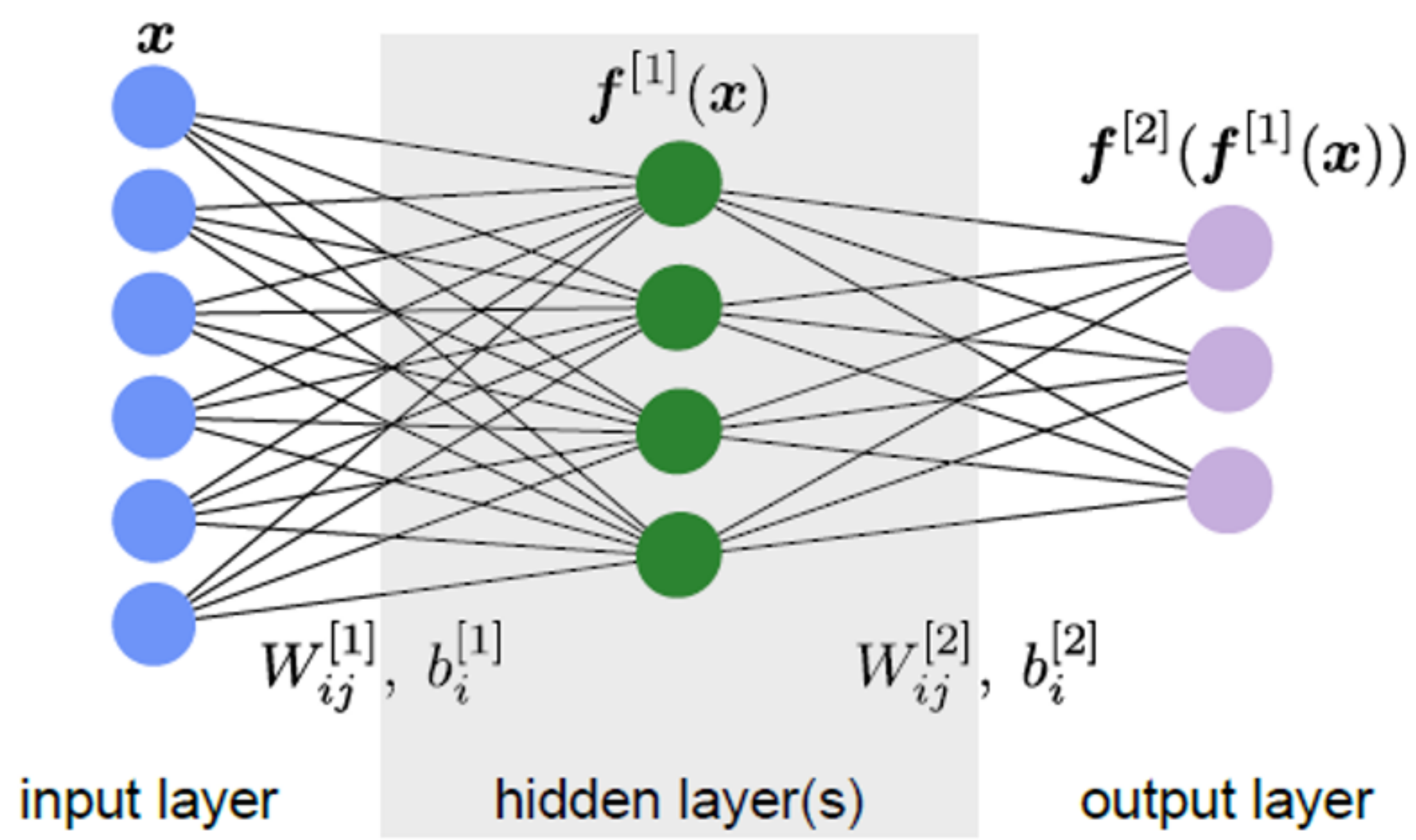
- APPLY TO NEURAL NET:



How do we calculate the derivative?

$$\frac{dL(f^{[2]})}{dW_{ij}^{[1]}} = \frac{dL}{df^{[2]}} \frac{df^{[2]}}{df^{[1]}} \frac{df^{[1]}}{dW^{[1]}}$$

- APPLY TO NEURAL NET:

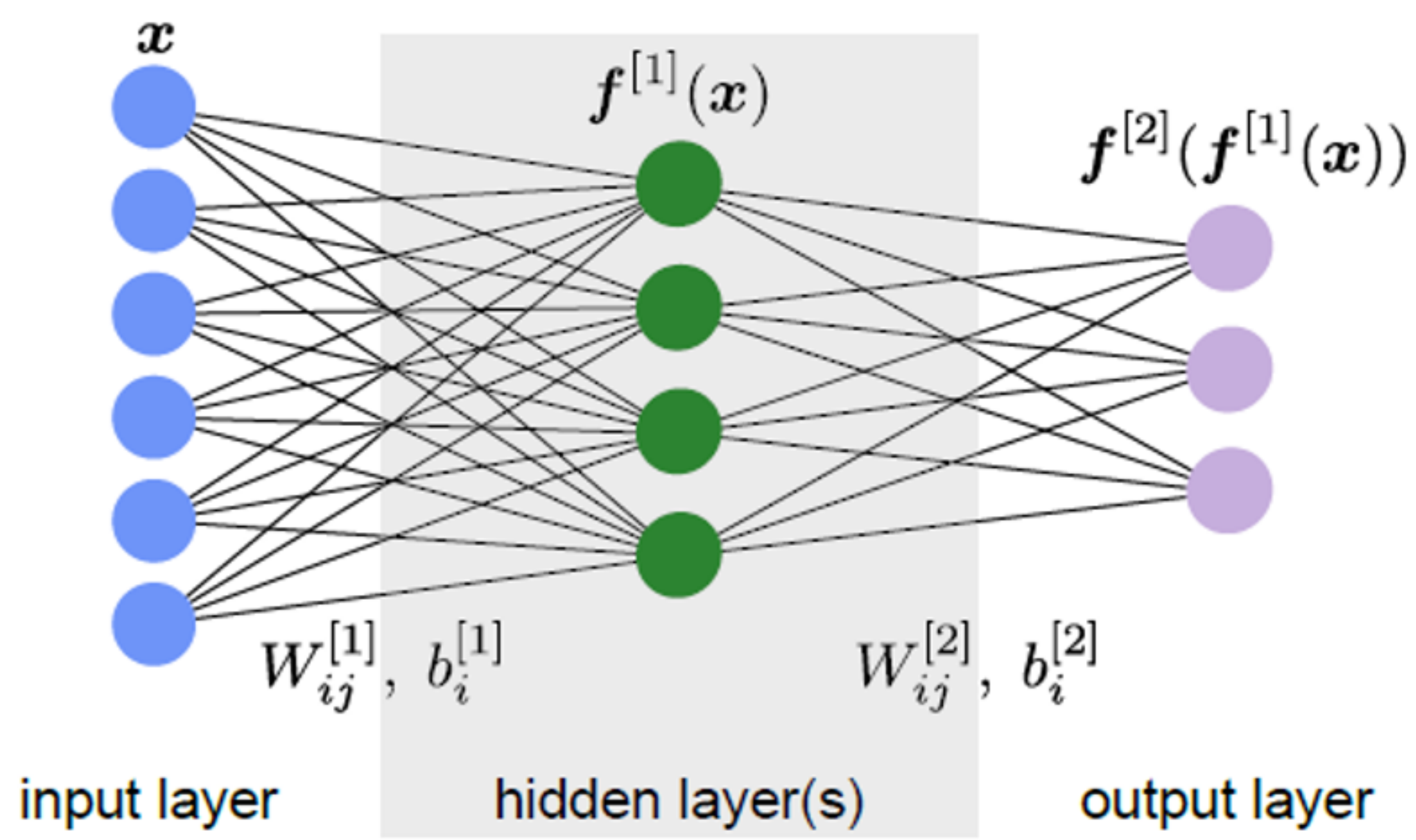


How do we calculate the derivative?

$$\frac{dL(f^{[2]})}{dW_{ij}^{[1]}} = \frac{dL}{df^{[2]}} \frac{df^{[2]}}{df^{[1]}} \frac{df^{[1]}}{dW^{[1]}}$$

First calculate this using the network's output

- APPLY TO NEURAL NET:



QMAI | TU Delft

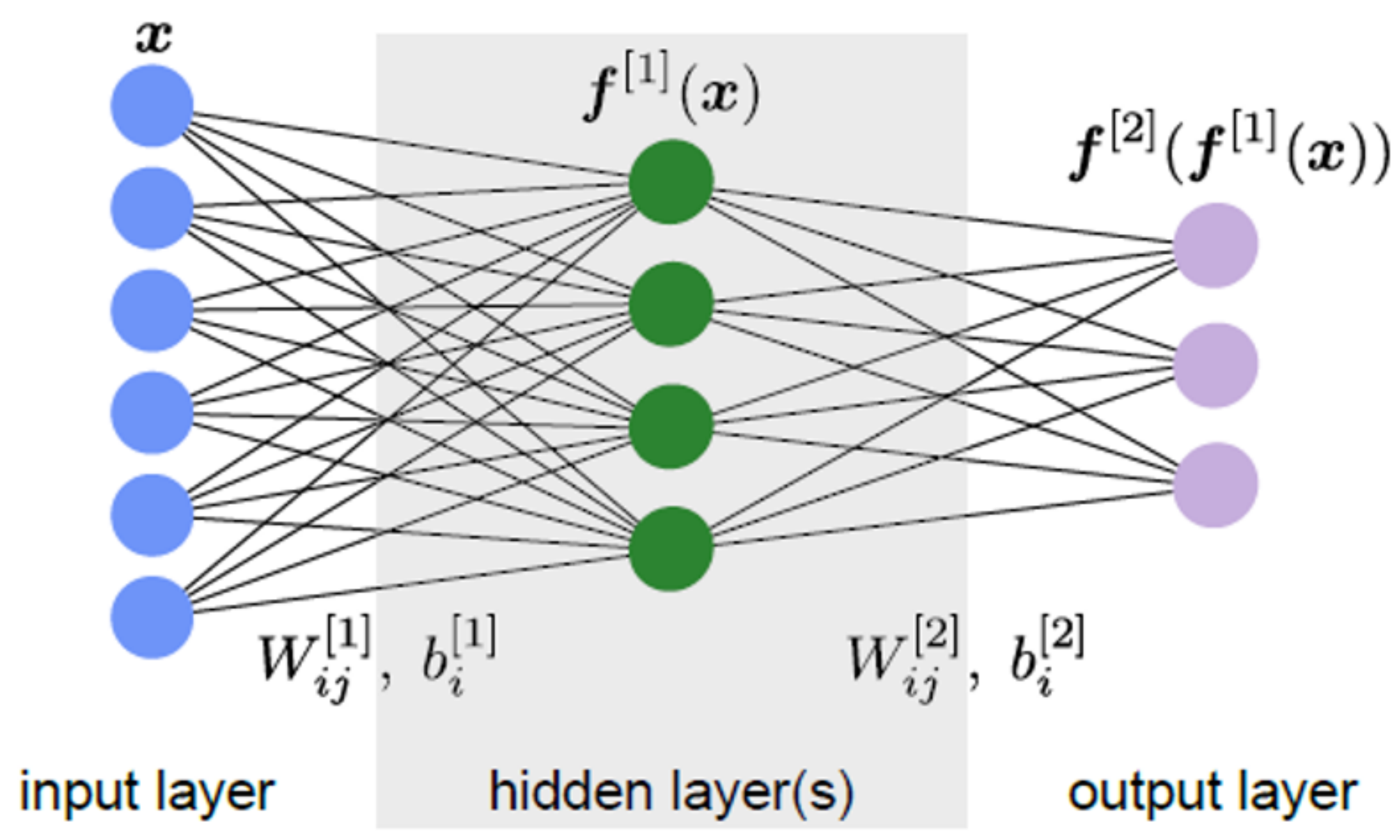
How do we calculate the derivative?

Calculate this using estimate of f[2] from previous step

$$\frac{dL(f^{[2]})}{dW_{ij}^{[1]}} = \frac{dL}{df^{[2]}} \frac{df^{[2]}}{df^{[1]}} \frac{df^{[1]}}{dW^{[1]}}$$

First calculate this using the network's output

- APPLY TO NEURAL NET:



How do we calculate the derivative?

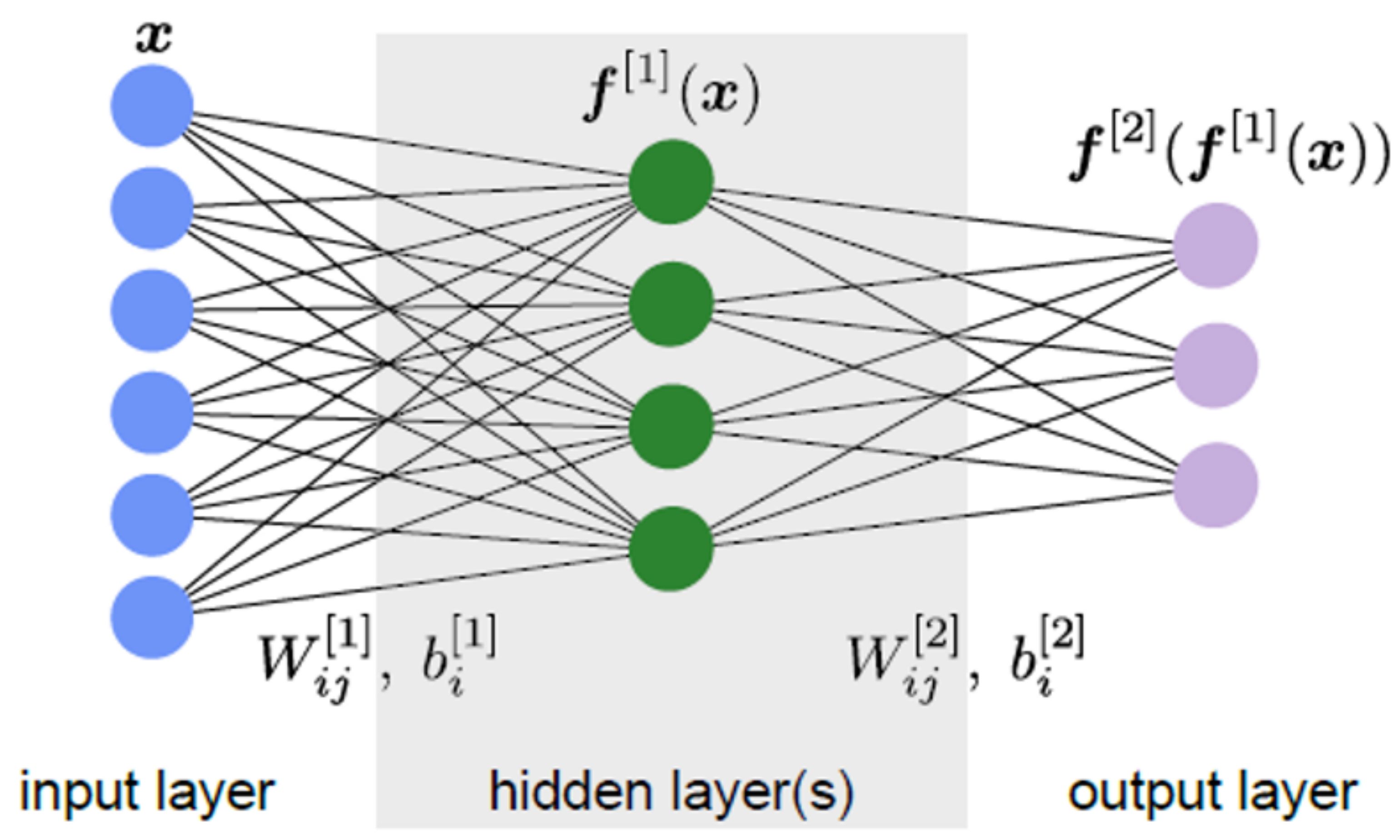
Calculate this using estimate of f[2] from previous step

$$\frac{dL(f^{[2]})}{dW_{ij}^{[1]}} = \frac{dL}{df^{[2]}} \frac{df^{[2]}}{df^{[1]}} \frac{df^{[1]}}{dW^{[1]}}$$

First calculate this using the network's output

..and so on: we calculate each derivative from the back

- APPLY TO NEURAL NET:



Backpropagation algorithm

1. Calculate forward propagation of the network
2. Calculate the backward phase

   a: Estimate the error in the final layer

   b: Propagate that error into the previous layer

   c: Evaluate the derivative of each parameter in the network
3. Combine all partial gradients into the final gradient
4. Update the weights using the calculated gradients to minimise the loss

Setting up the neural network

Classification: Does a picture belong into the class “A" or class “B”?

Build a network with two outputs that tell you the probability from which movie franchise your character is from.

Output and loss function

- LAST LAYER ACTIVATION FUNCTION:

  (normalises the output and maps it on the probability distribution)

$$f(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

- LOSS FUNCTION:

  (calculate the difference between predicted and correct outcome during training)

$$L = -\sum_x p(x) \log q(x)$$

Classification: a mini example

Label: p(x)

$$\begin{bmatrix}1\\0\end{bmatrix} \qquad \begin{bmatrix}0\\1\end{bmatrix}$$

Network output: q(x)

$$\begin{bmatrix}q\ (\text{class A})\\q\ (\text{class B})\end{bmatrix}$$

Loss: -1 log(q(class A)) - 0 log(q(class B))   -0 log(q(class A)) - 1 log(q(class B))

Classification: a mini example

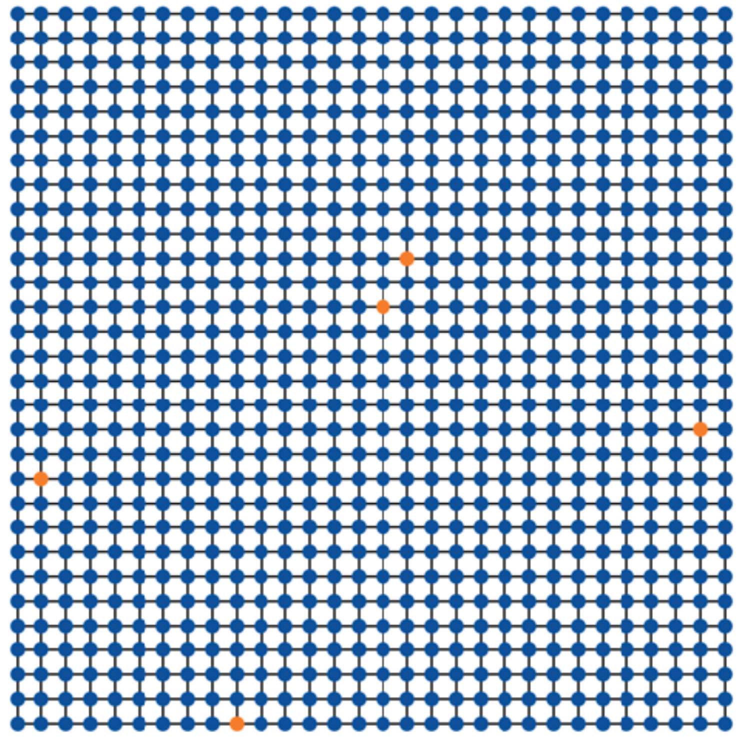
Label: p(x) $\begin{bmatrix}1\\0\end{bmatrix}$ $\begin{bmatrix}0\\1\end{bmatrix}$

Network output: q(x) $\begin{bmatrix}q\text{ (class A)}\\q\text{ (class B)}\end{bmatrix}$

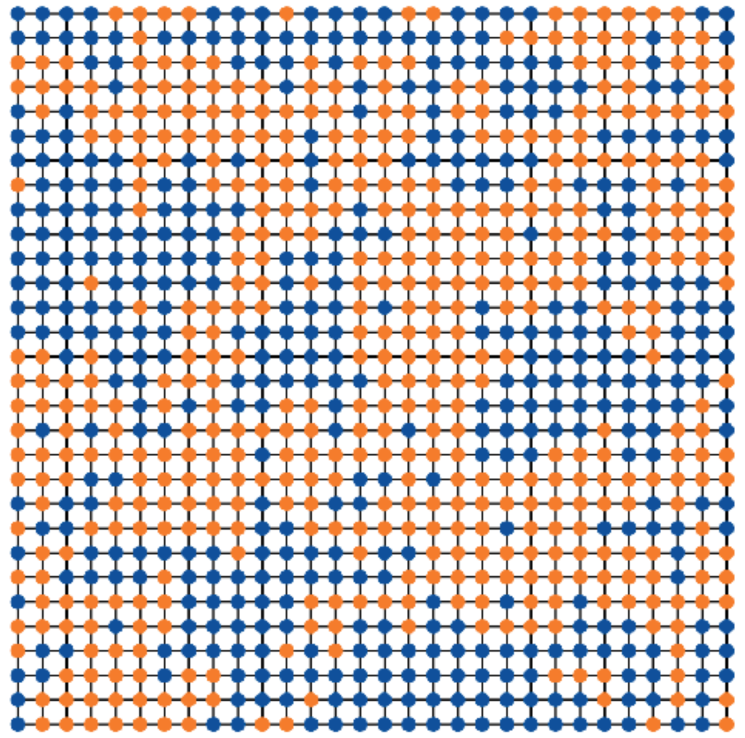
Loss: -1 log(q(class A)) - 0 log(q(class B)) -0 log(q(class A)) - 1 log(q(class B))

!the loss is minimal if q(x) matches the labels!

Back to physics



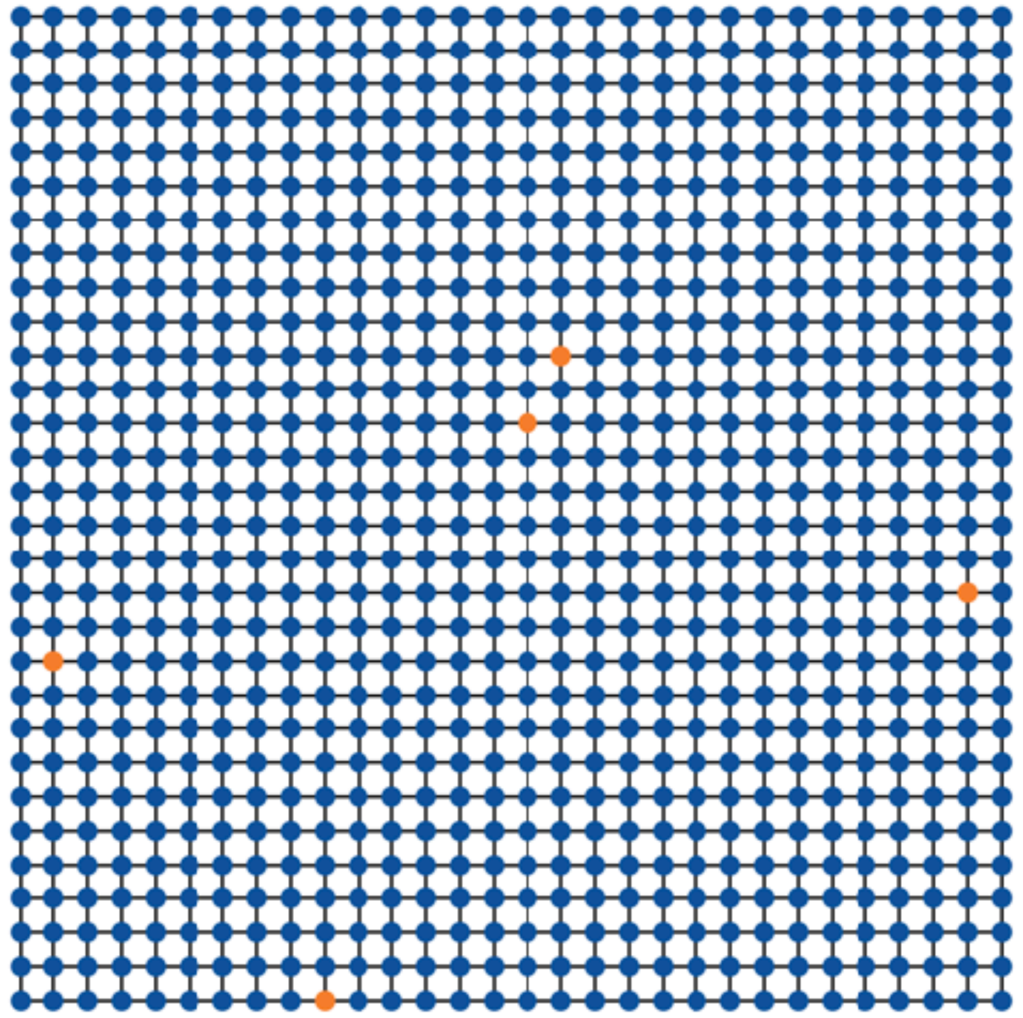
T=1.66



T=3.37

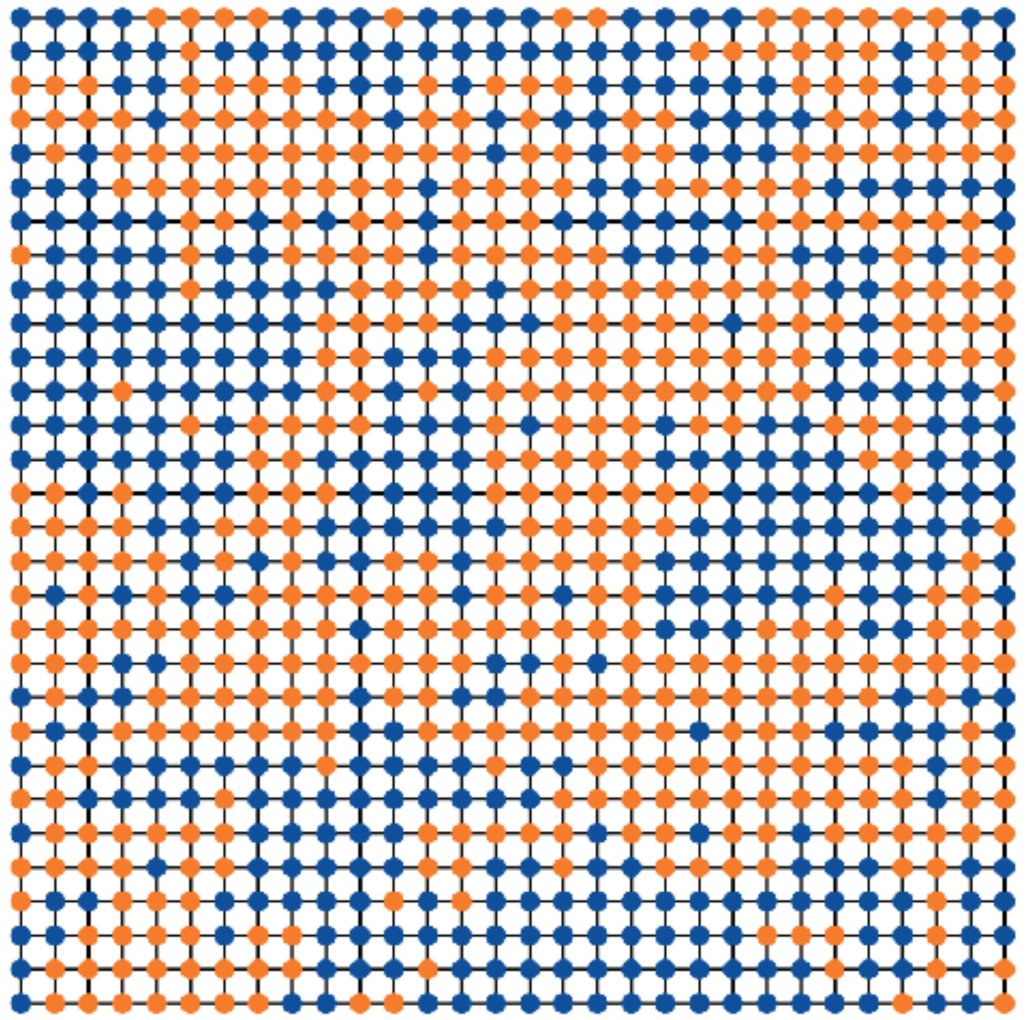
QMAI | TU Delft

Notebook 2: Supervised learning



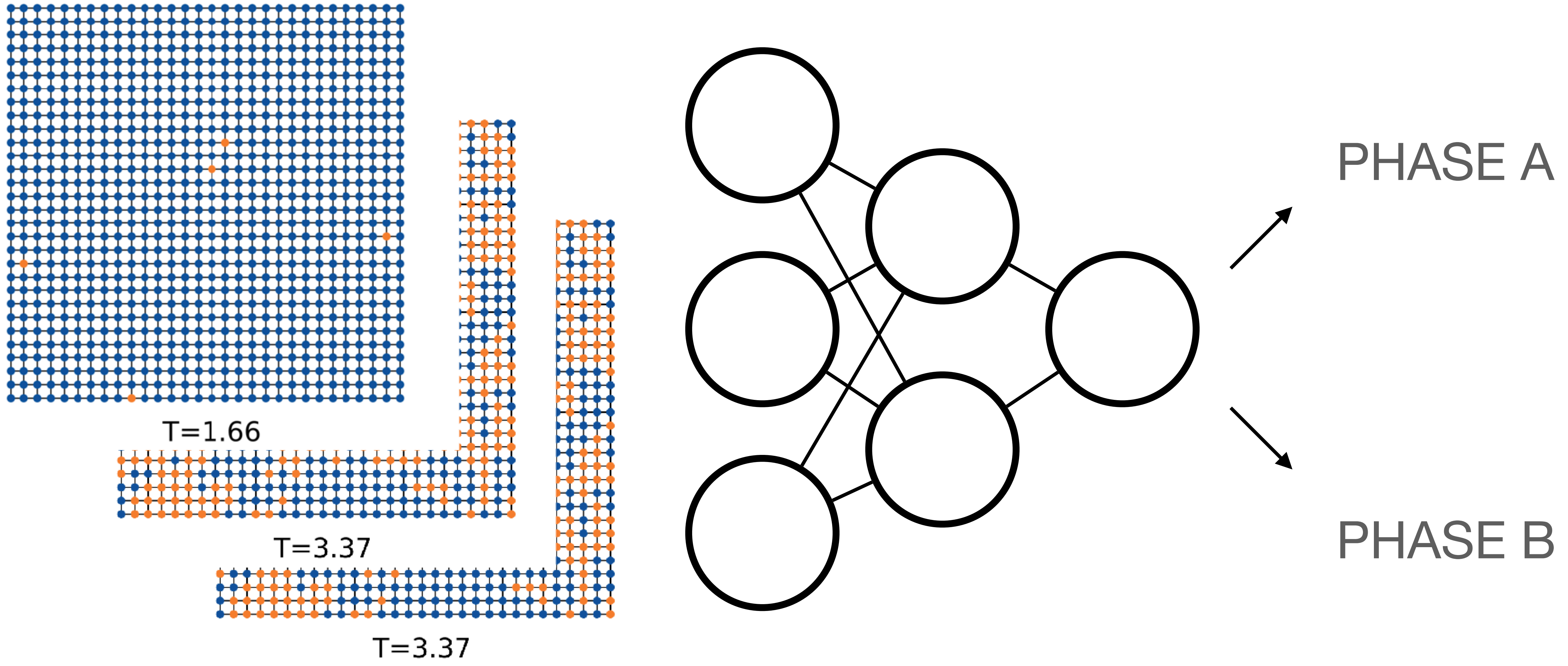
T=1.66

PHASE A



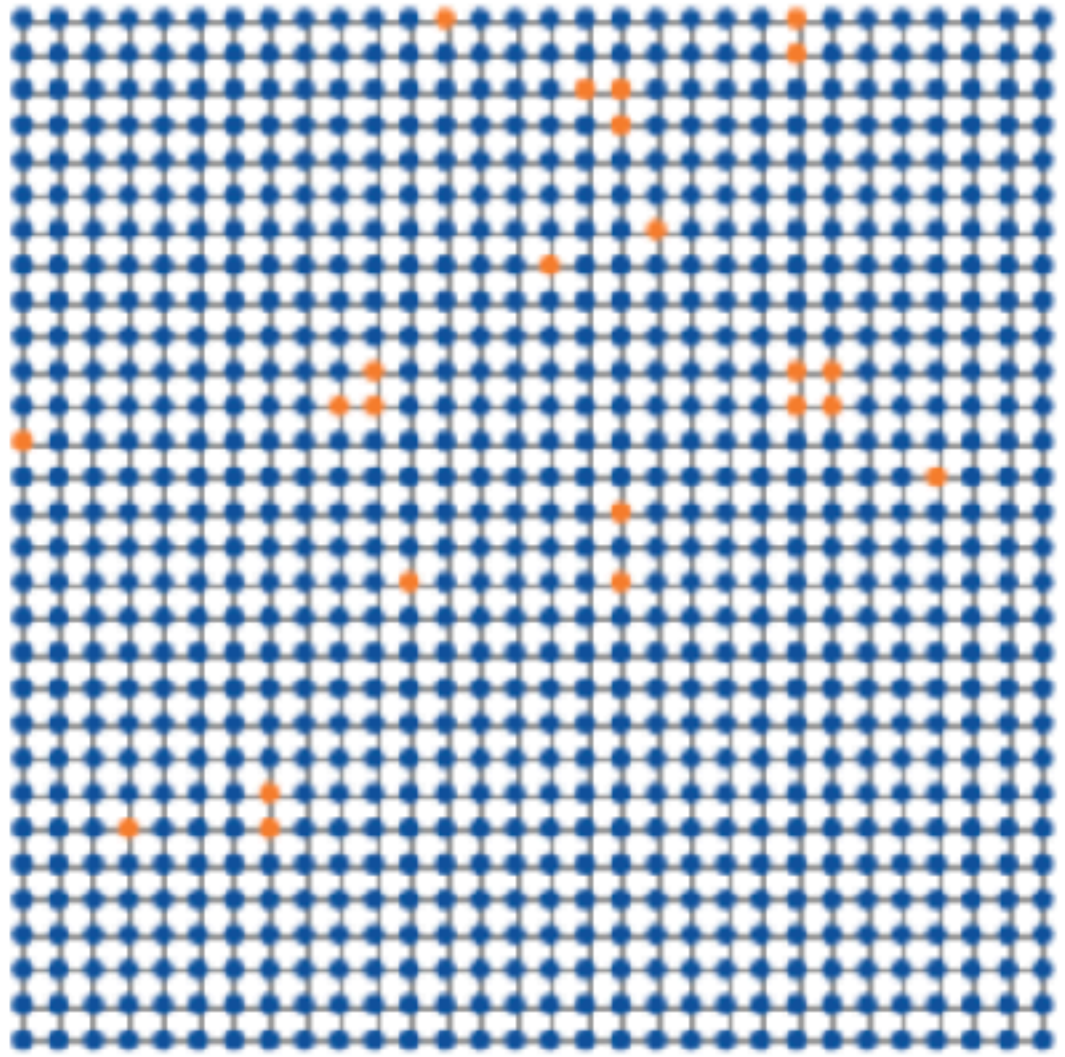
T=3.37

PHASE B

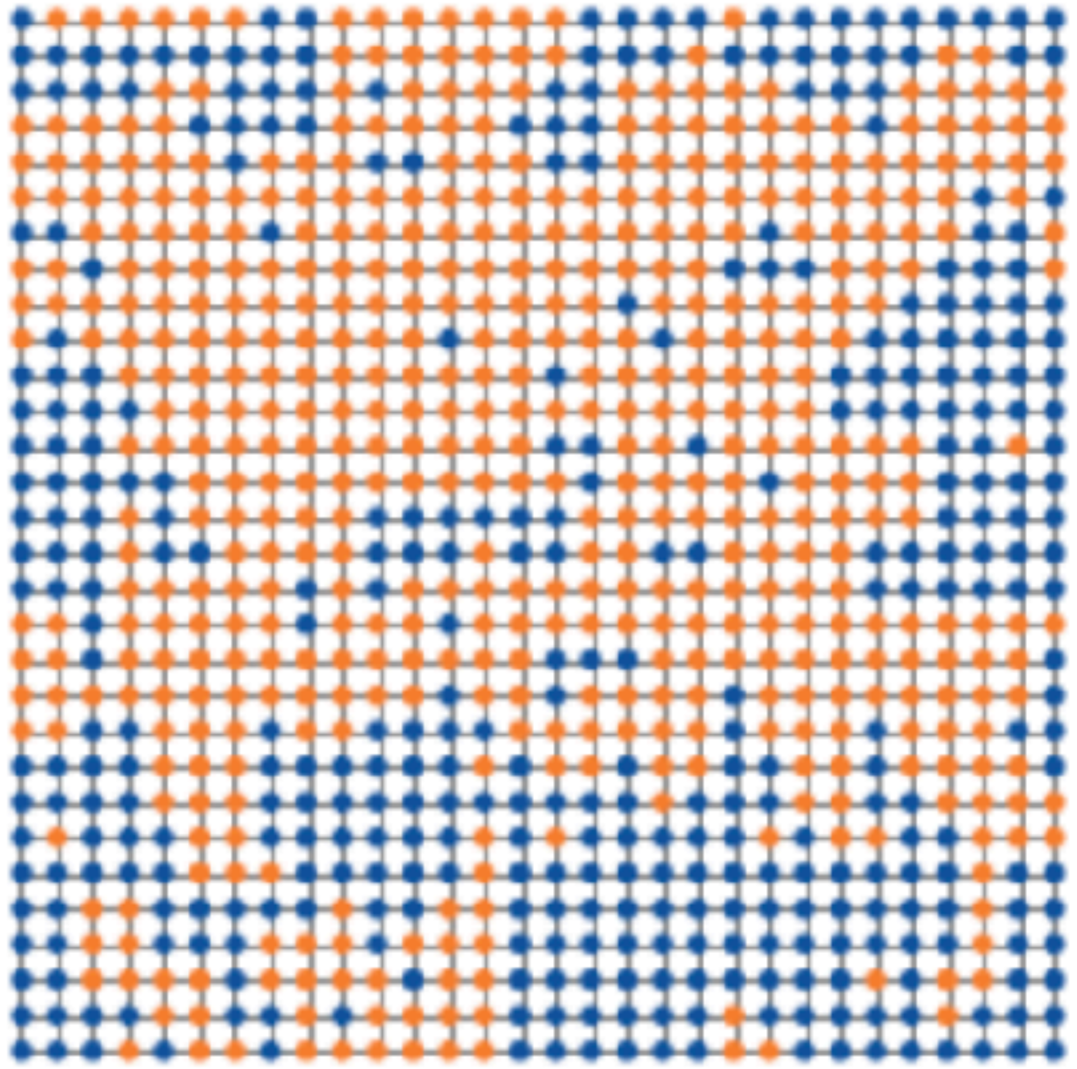


Quiz: Ising Classification

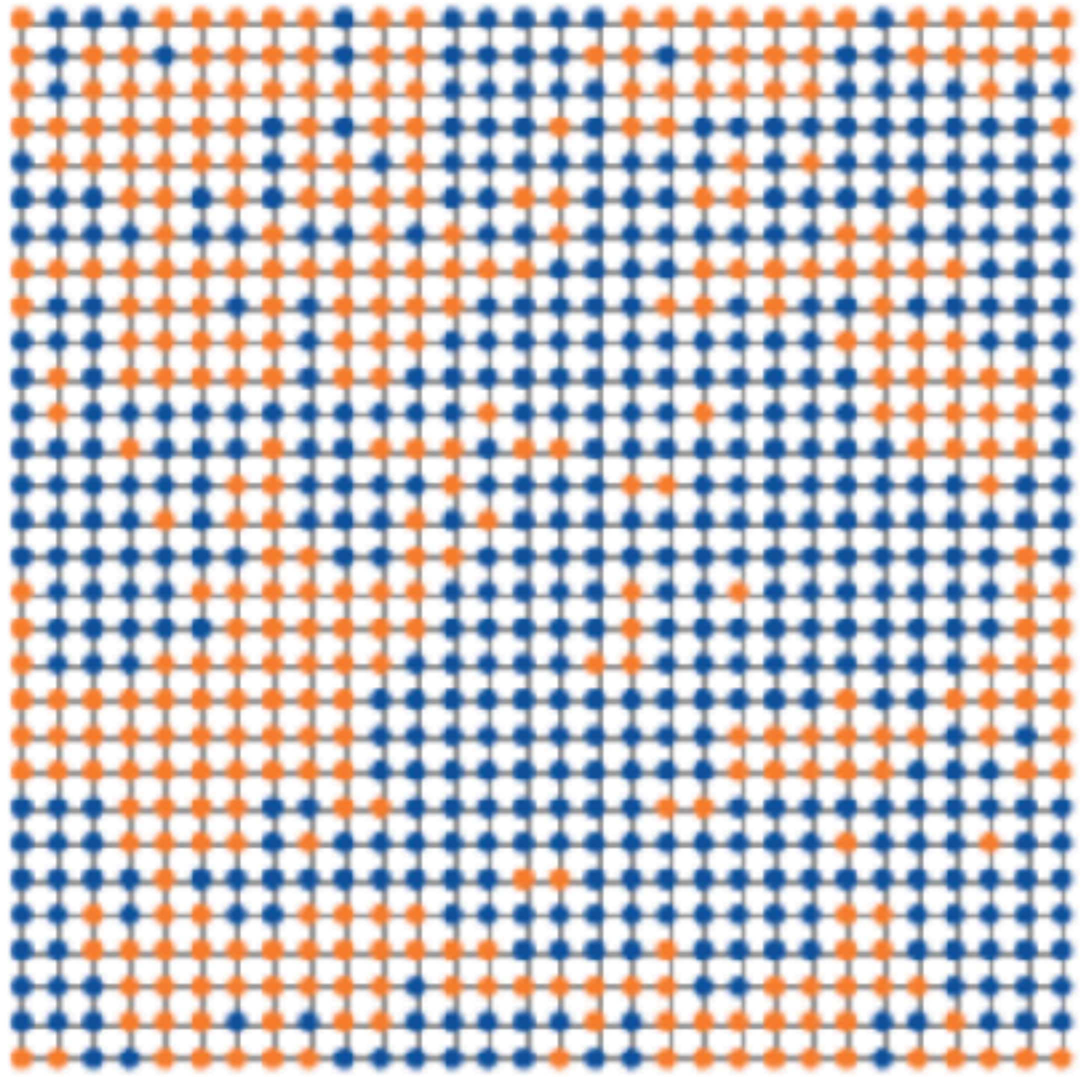
A



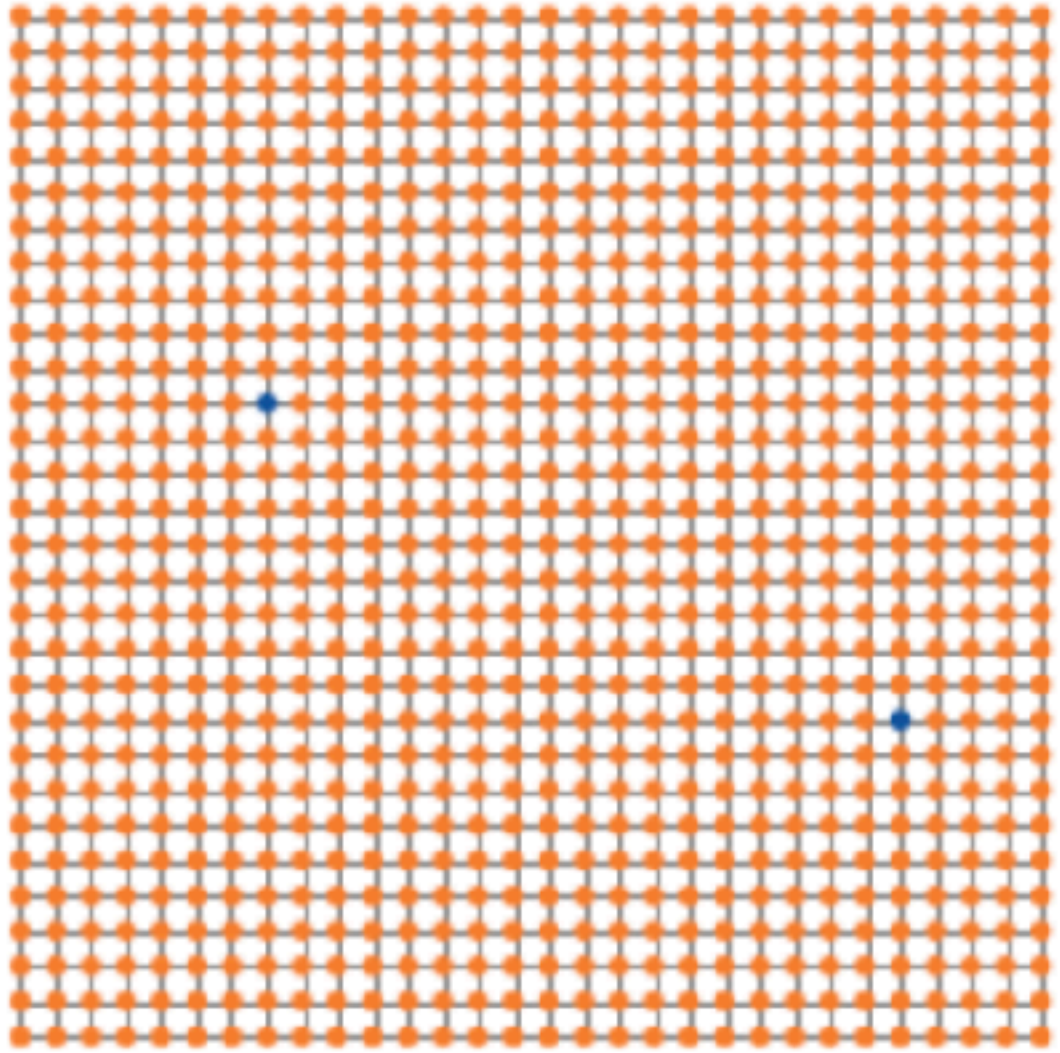
B



C



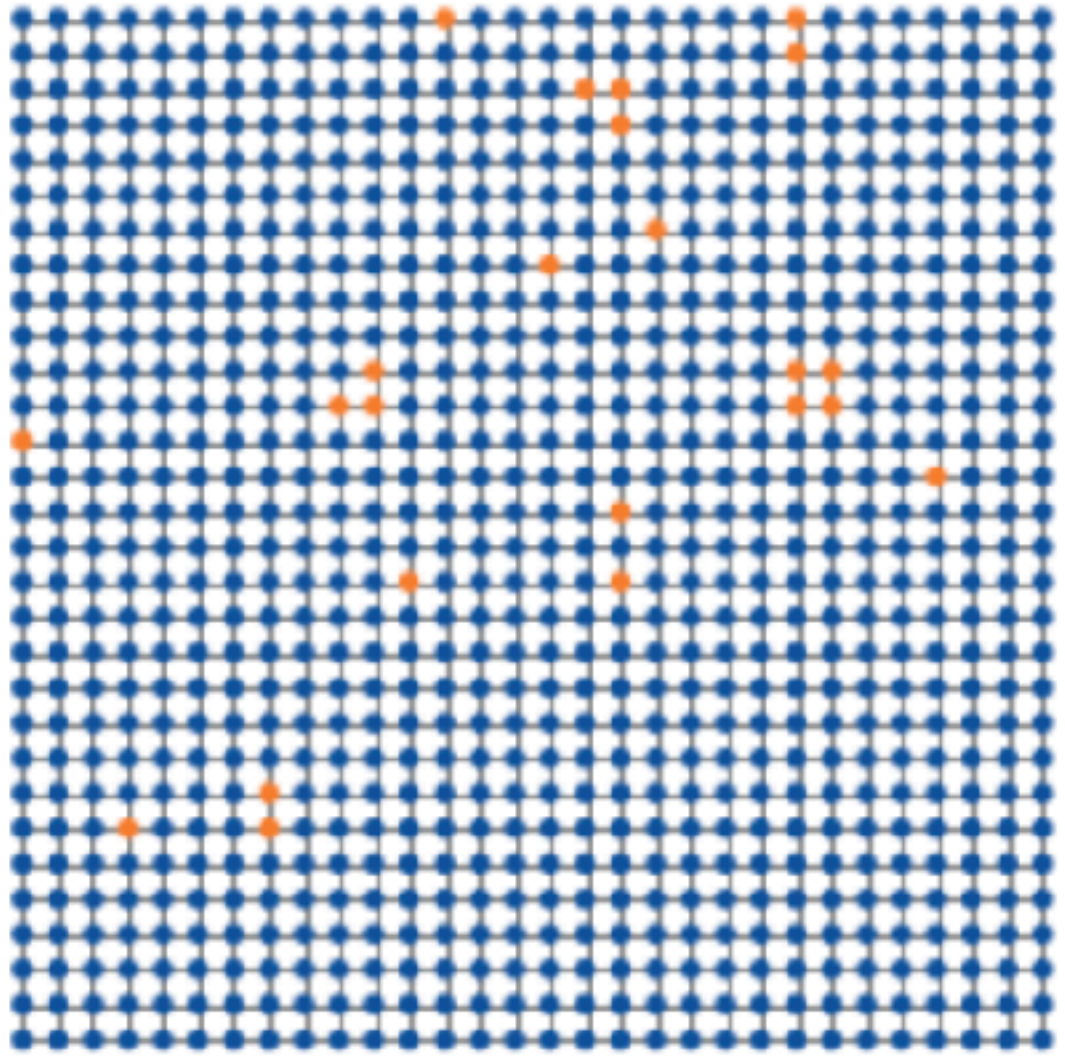
D



QMAI | TUDelft

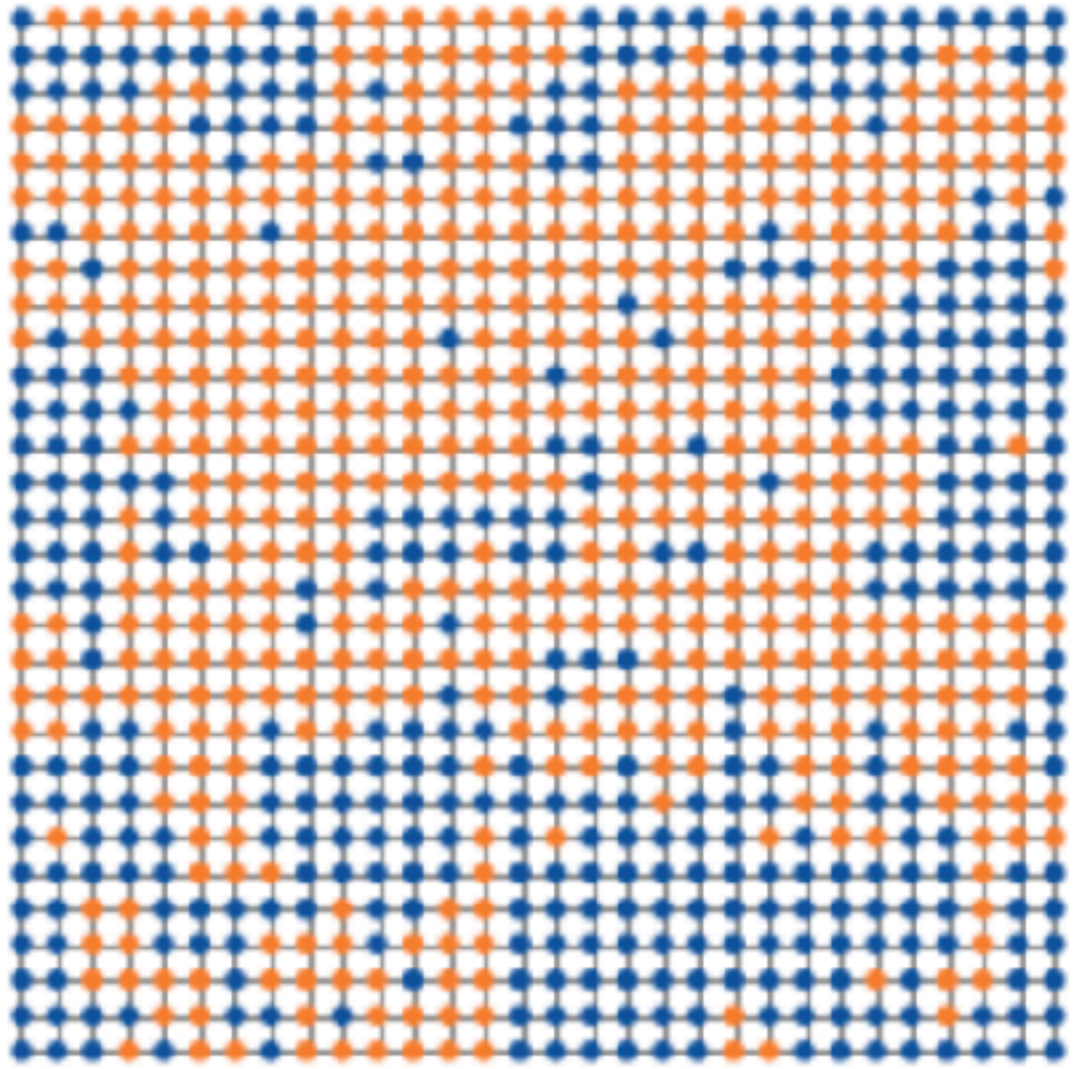
Quiz: Ising Classification

A



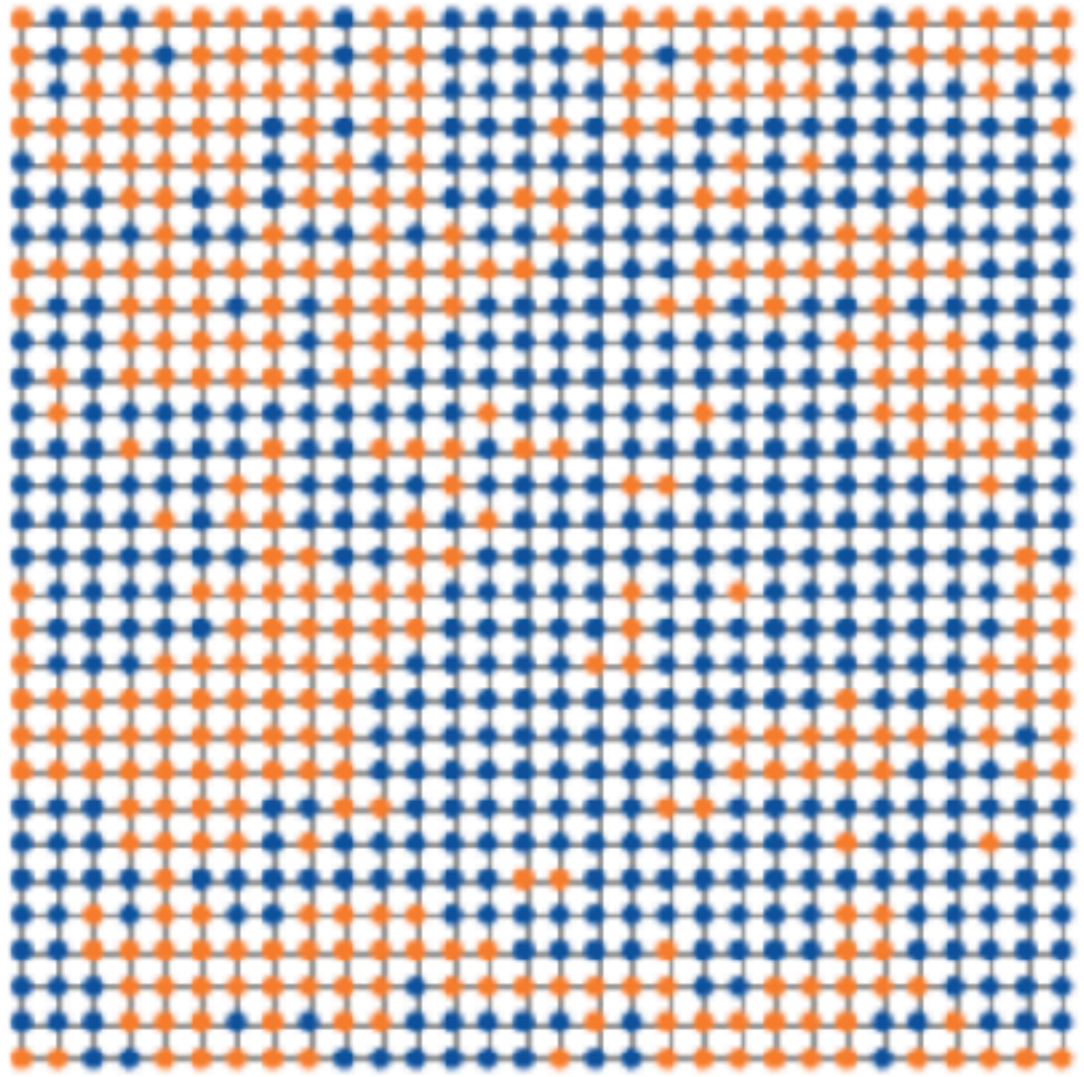
T=1.79

B



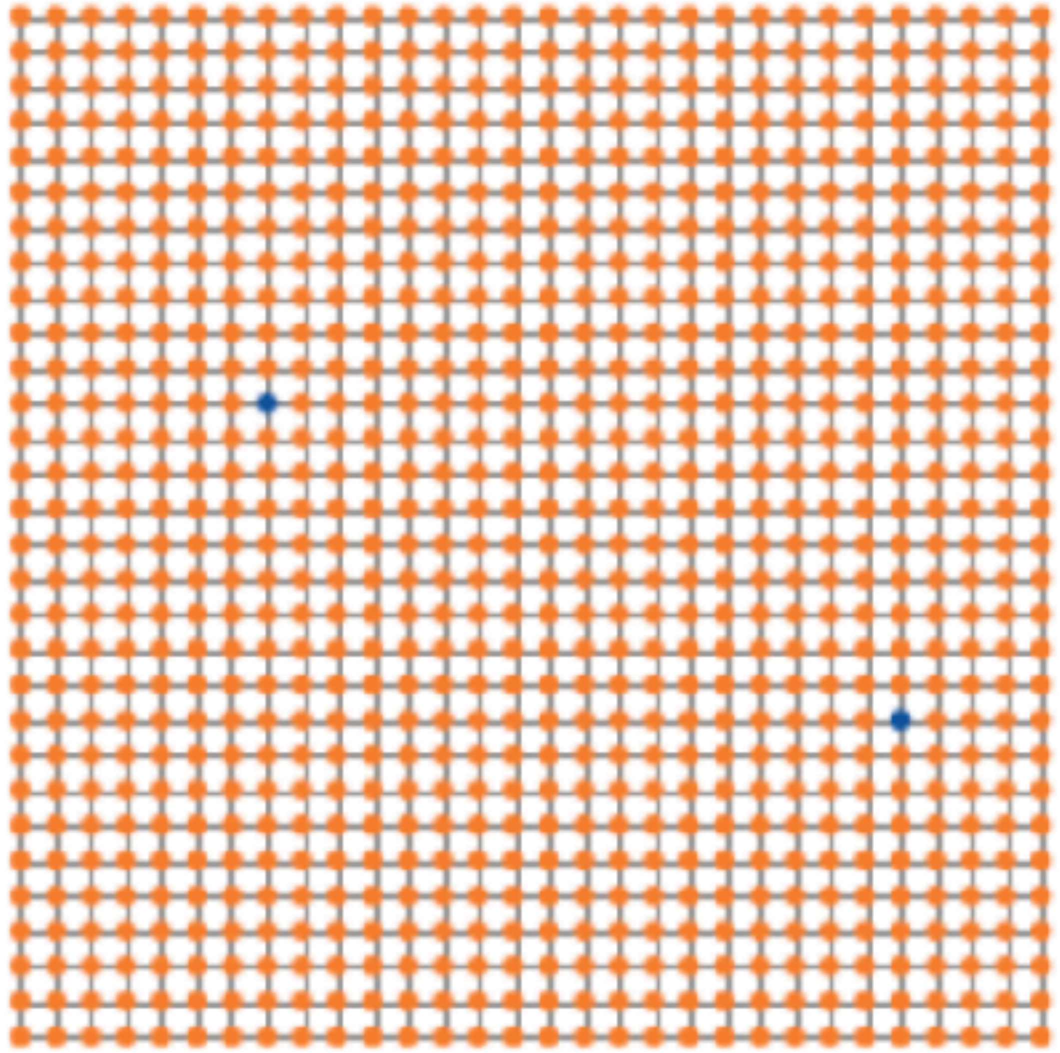
T=2.58

C



T=2.97

D

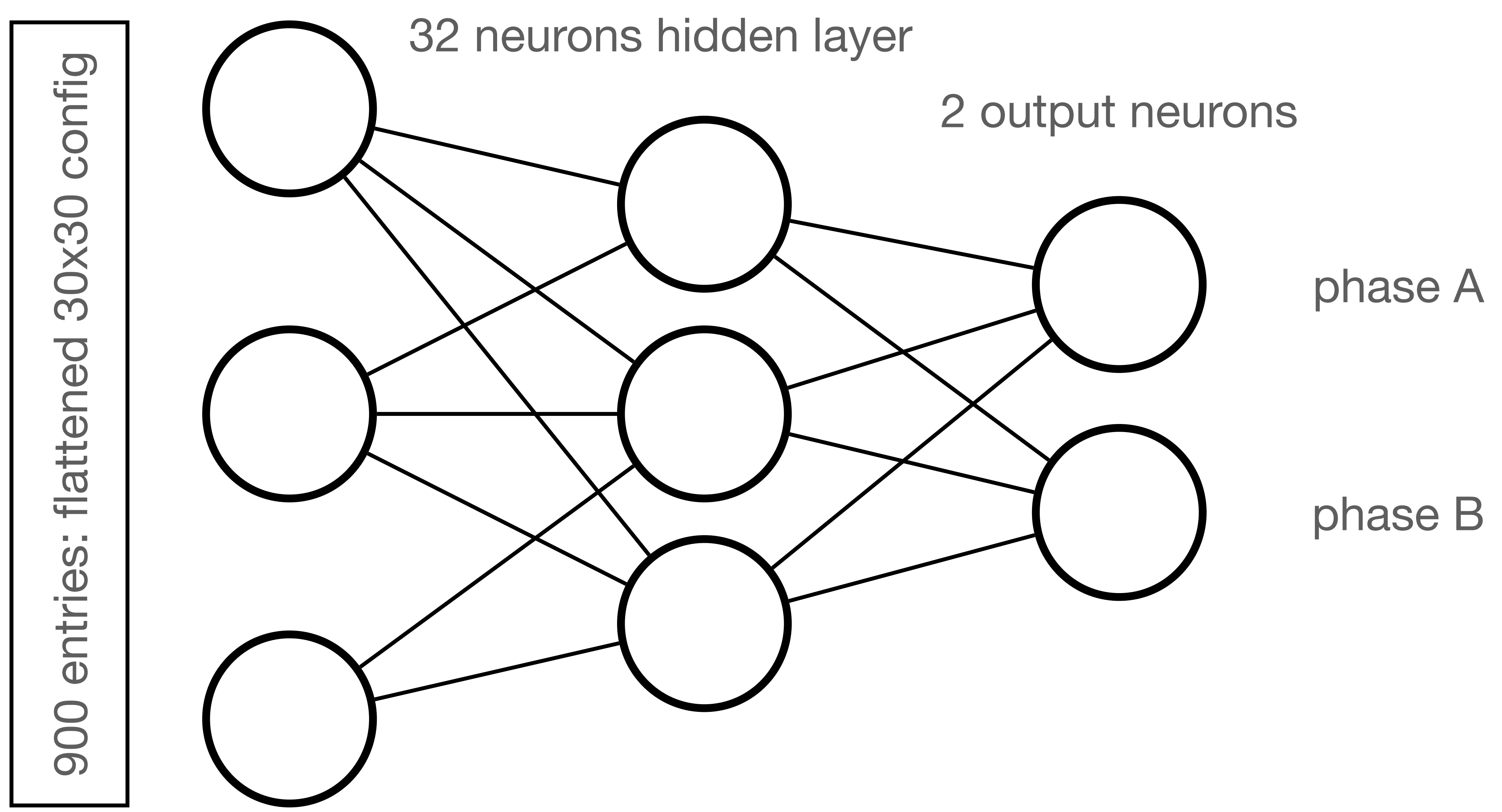


T=1.26

QMAI | TU Delft

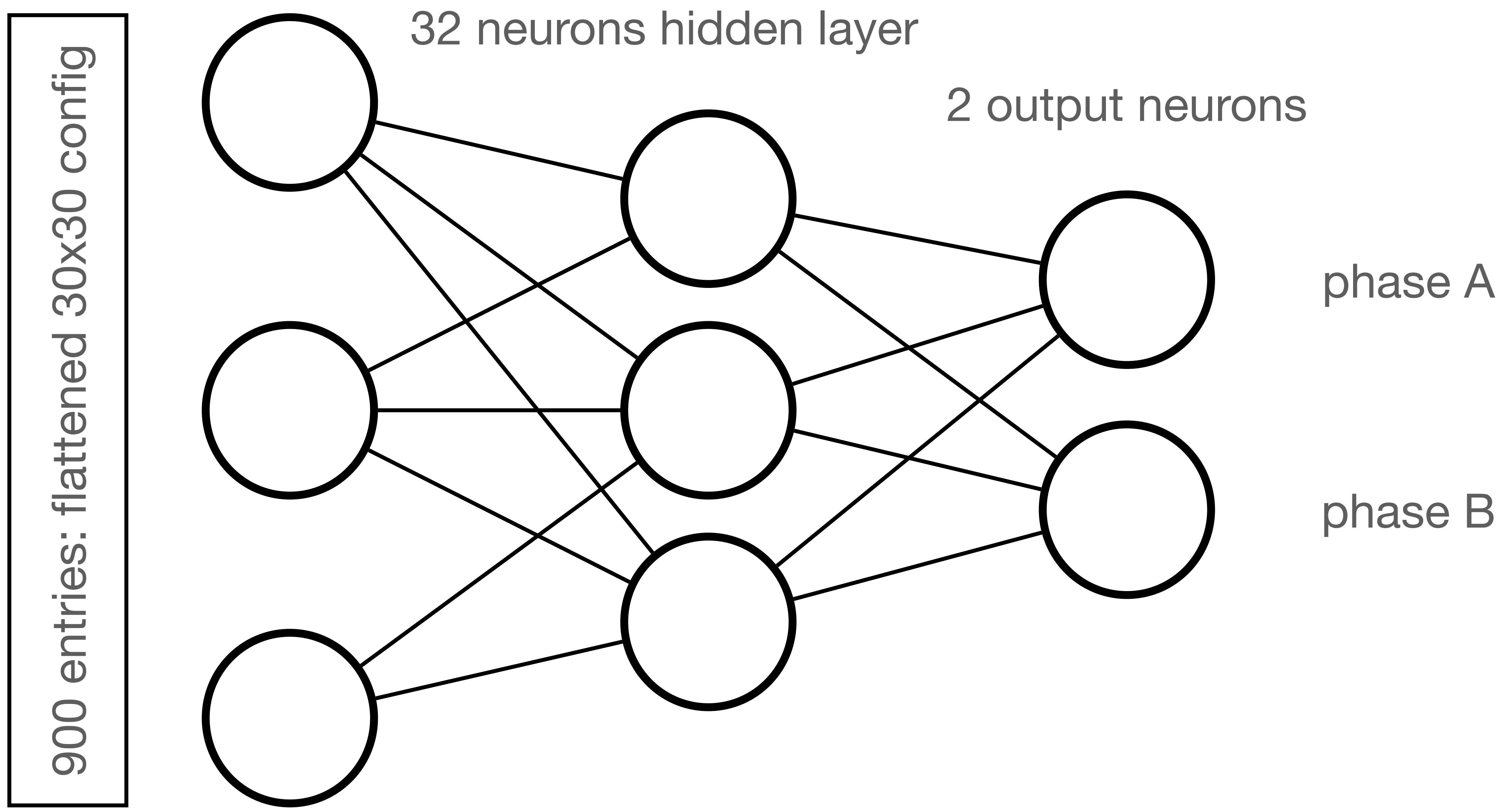
Notebook 2: Supervised learning

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten_1 (Flatten)          (None, 900)               0
_________________________________________________________________
dense_2 (Dense)              (None, 32)                28832
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 66
=================================================================
Total params: 28,898
Trainable params: 28,898
Non-trainable params: 0
_________________________________________________________________
```



Notebook 2: Supervised learning

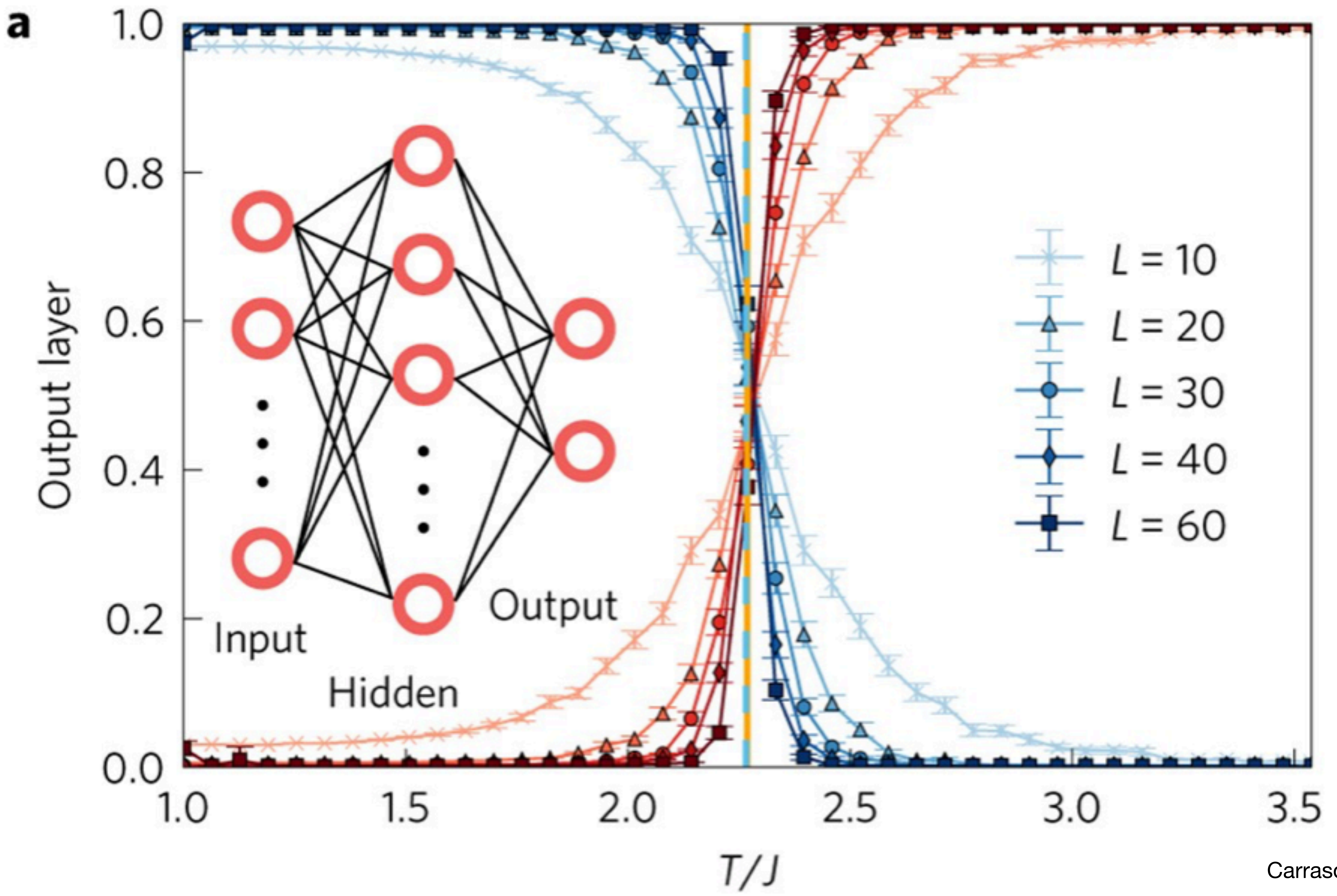
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten_1 (Flatten)          (None, 900)               0
_________________________________________________________________
dense_2 (Dense)              (None, 32)                28832
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 66
=================================================================
Total params: 28,898
Trainable params: 28,898
Non-trainable params: 0
_________________________________________________________________
```



Experiment with number of parameters!
Can you find a minimal model?

QMAI | TU Delft

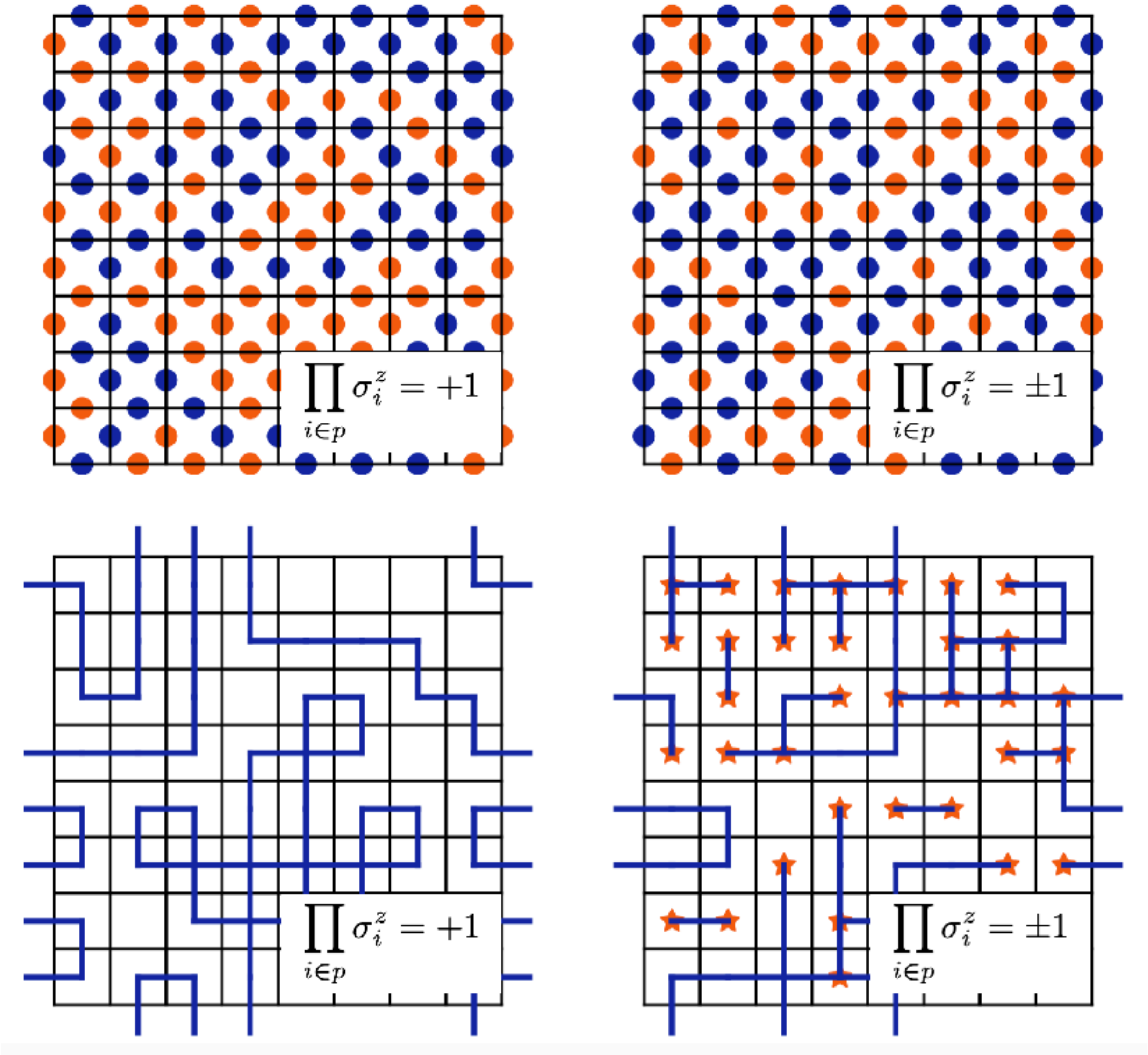
Notebook 2: Supervised learning



Carrasquilla&Melko, Nature Physics **13**, 431–434(2017)

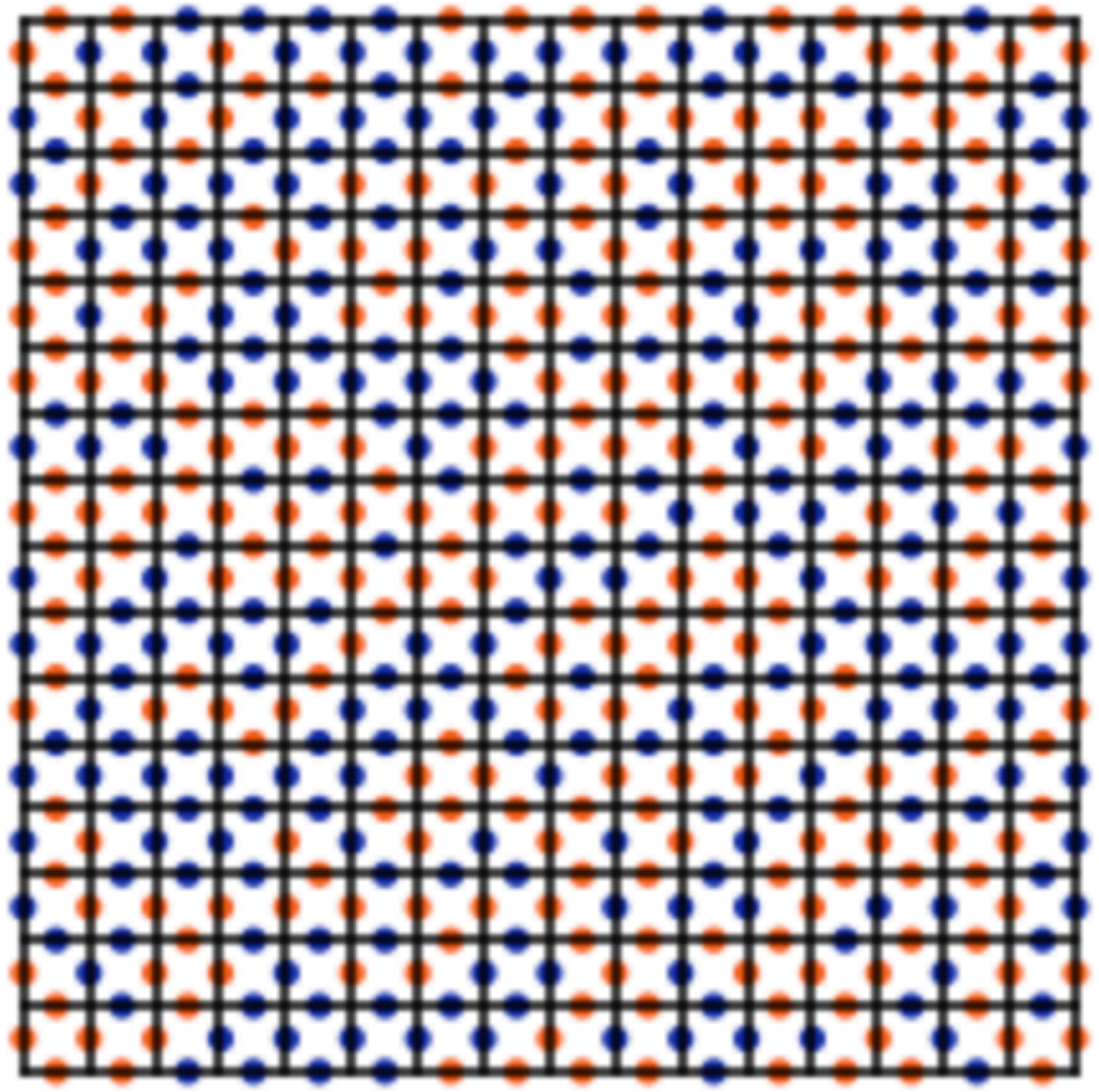
QMAI | TU Delft

Back to more challenging problem: IGT

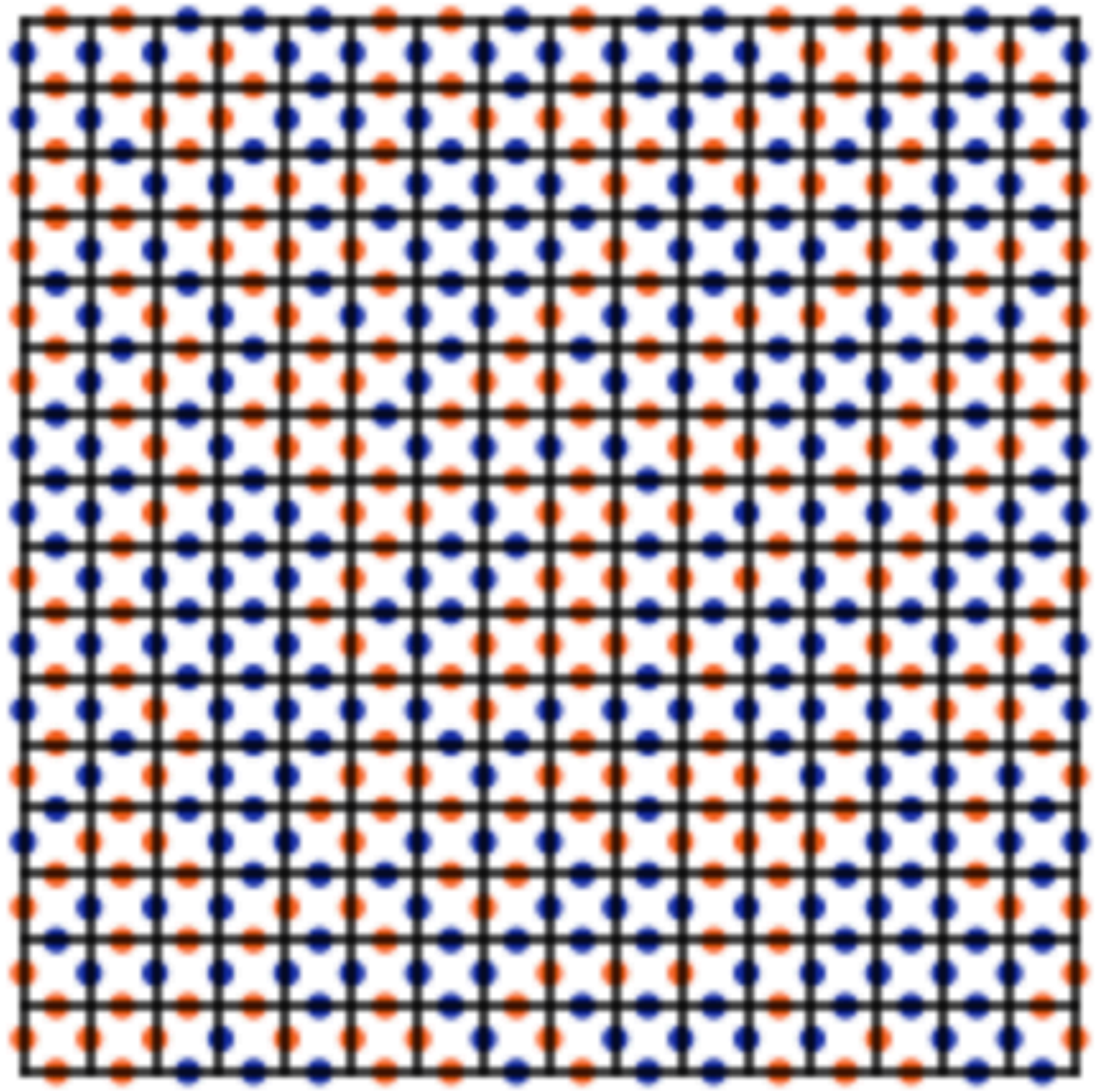


$$H_{IGT} = -J \sum_p \prod_{i \in p} \sigma_i^z$$

Quiz: IGT classification

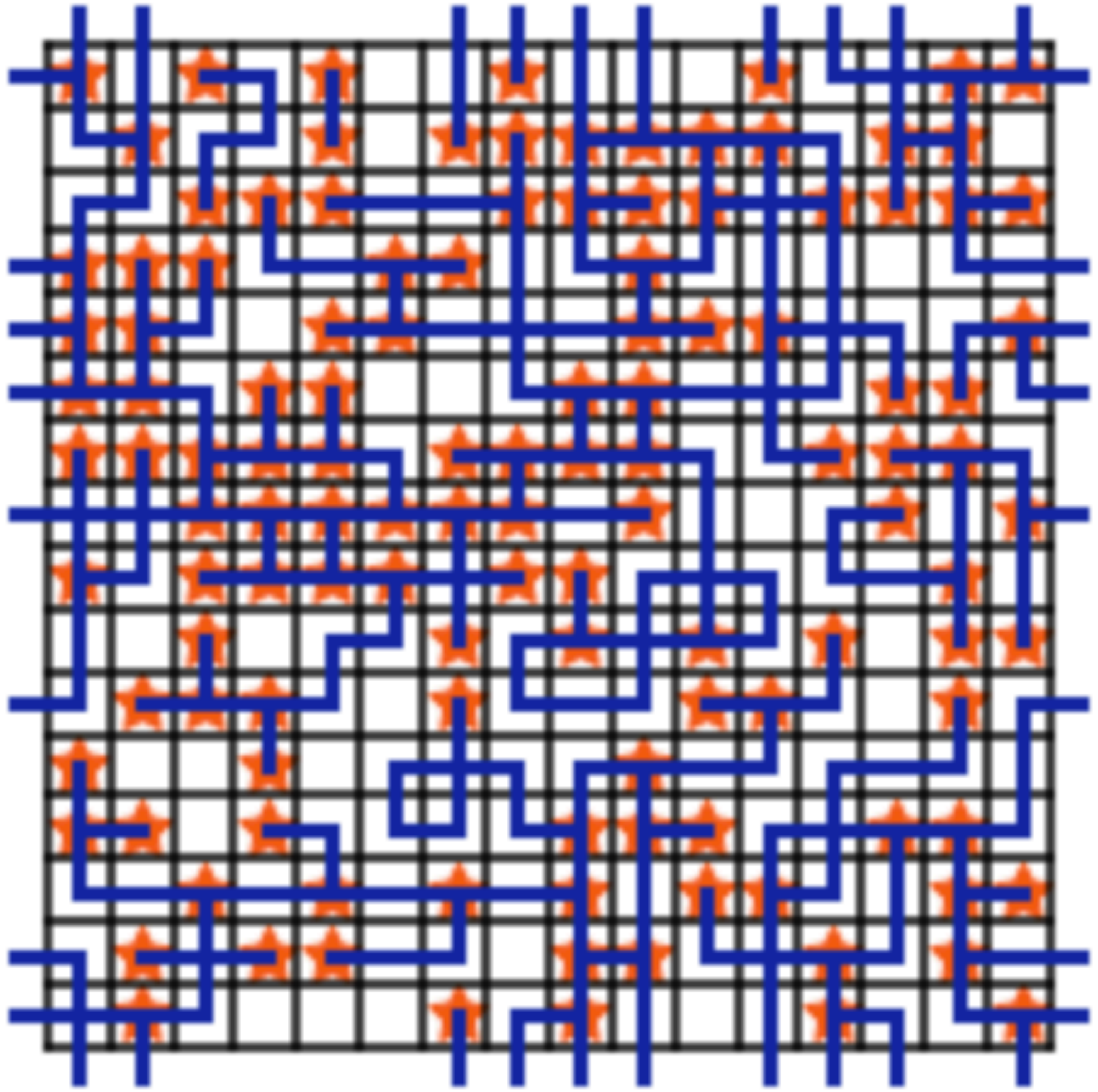


A

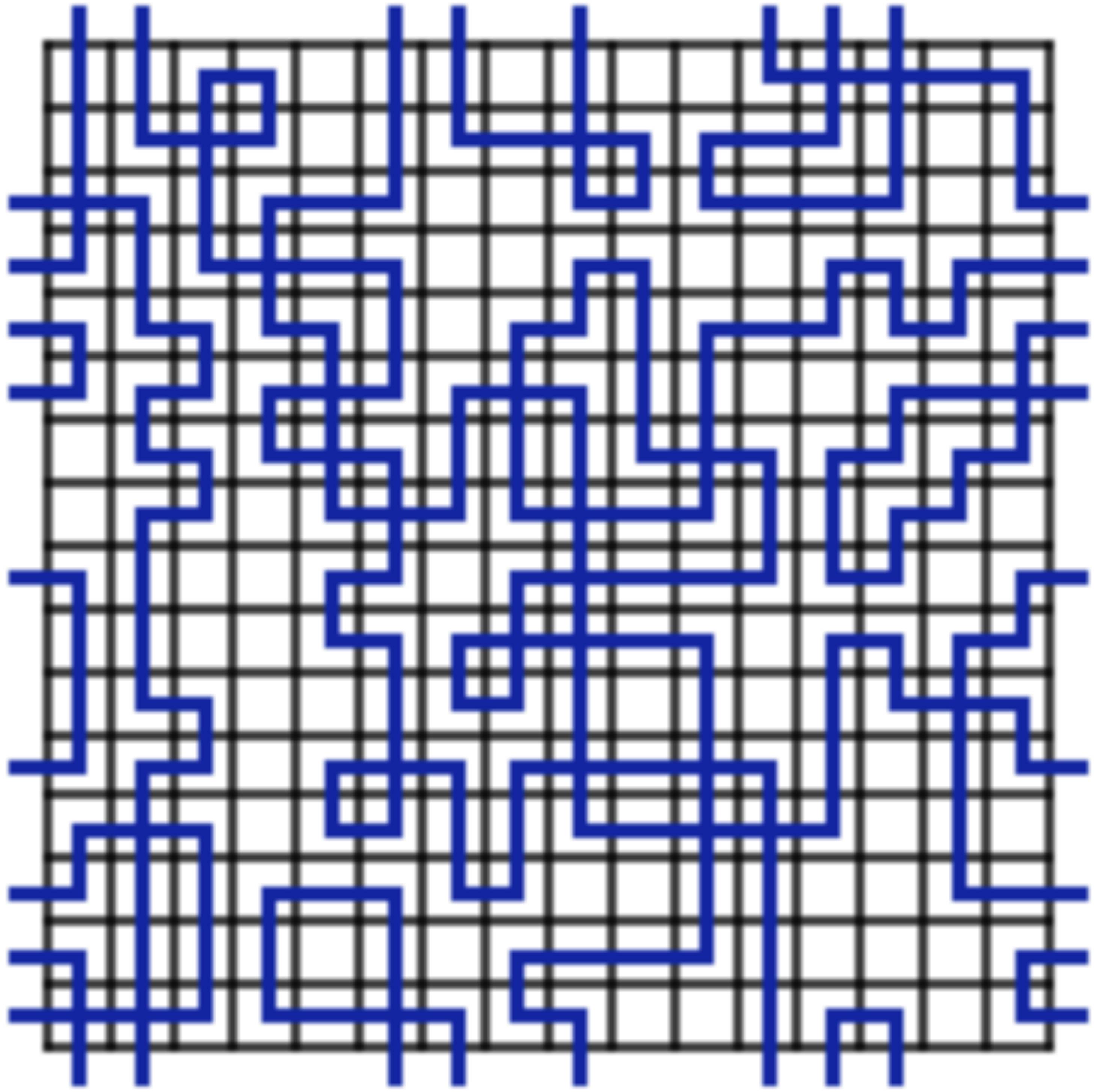


B

Quiz: IGT classification



A



B

QMAI | TU Delft

Notebook 2: IGT classification

STEP 1: Start with dense feed-forward network

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten (Flatten)            (None, 512)               0
_________________________________________________________________
dense (Dense)                (None, 100)               51300
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 202
=================================================================
Total params: 51,502
Trainable params: 51,502
Non-trainable params: 0
_________________________________________________________________
```

QMAI | TU Delft

Notebook 2: IGT classification

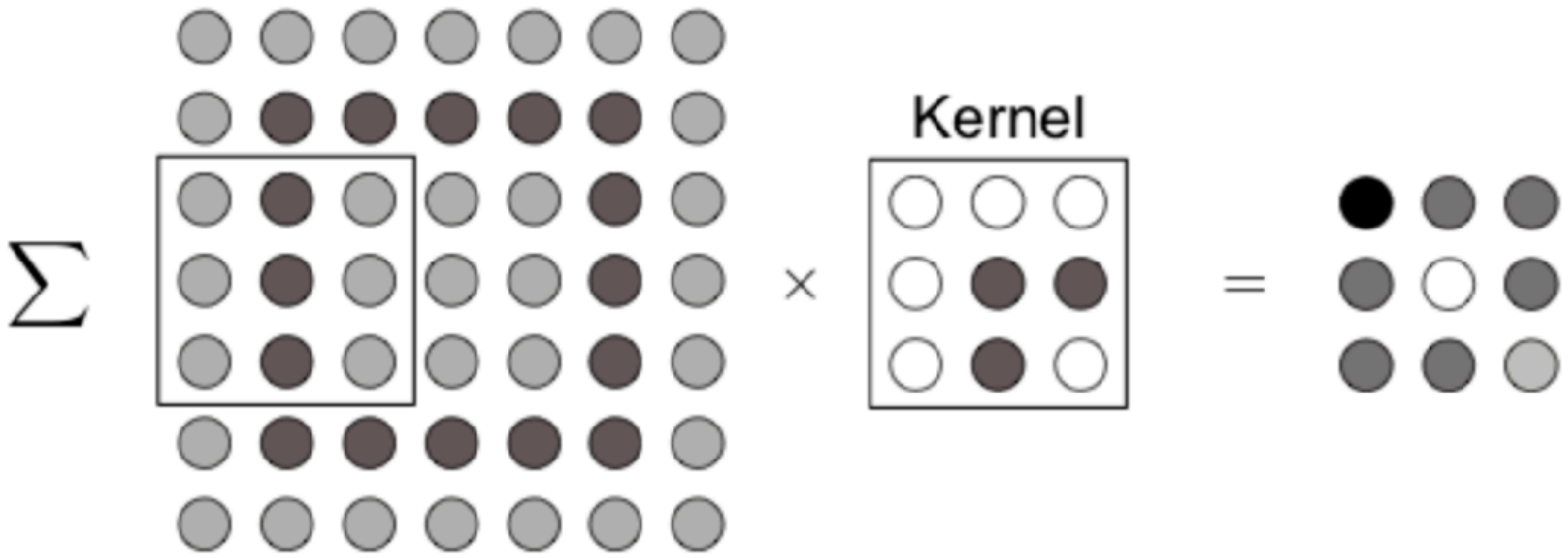
STEP 1: Start with dense feed-forward network

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten (Flatten)            (None, 512)               0
_________________________________________________________________
dense (Dense)                (None, 100)               51300
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 202
=================================================================
Total params: 51,502
Trainable params: 51,502
Non-trainable params: 0
_________________________________________________________________
```

Experiment with the architecture!
Does it ever work??

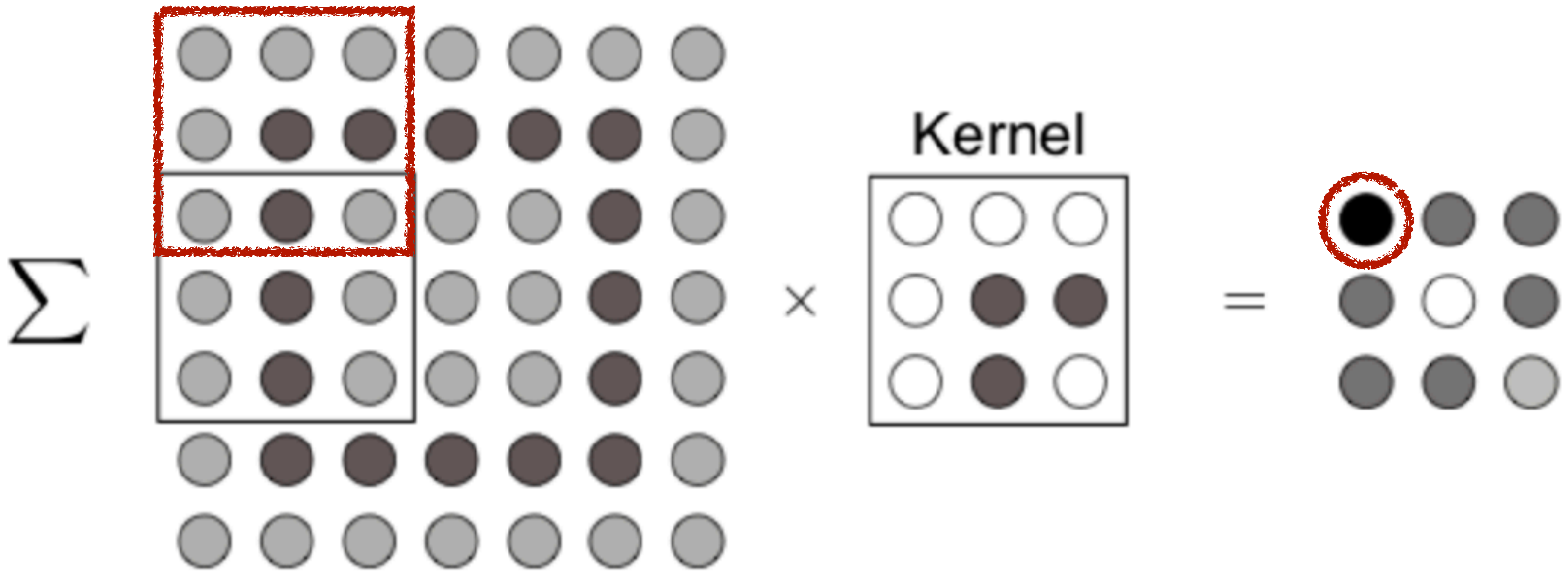
QMAI | TU Delft

We need more advanced layers: CONVOLUTIONS



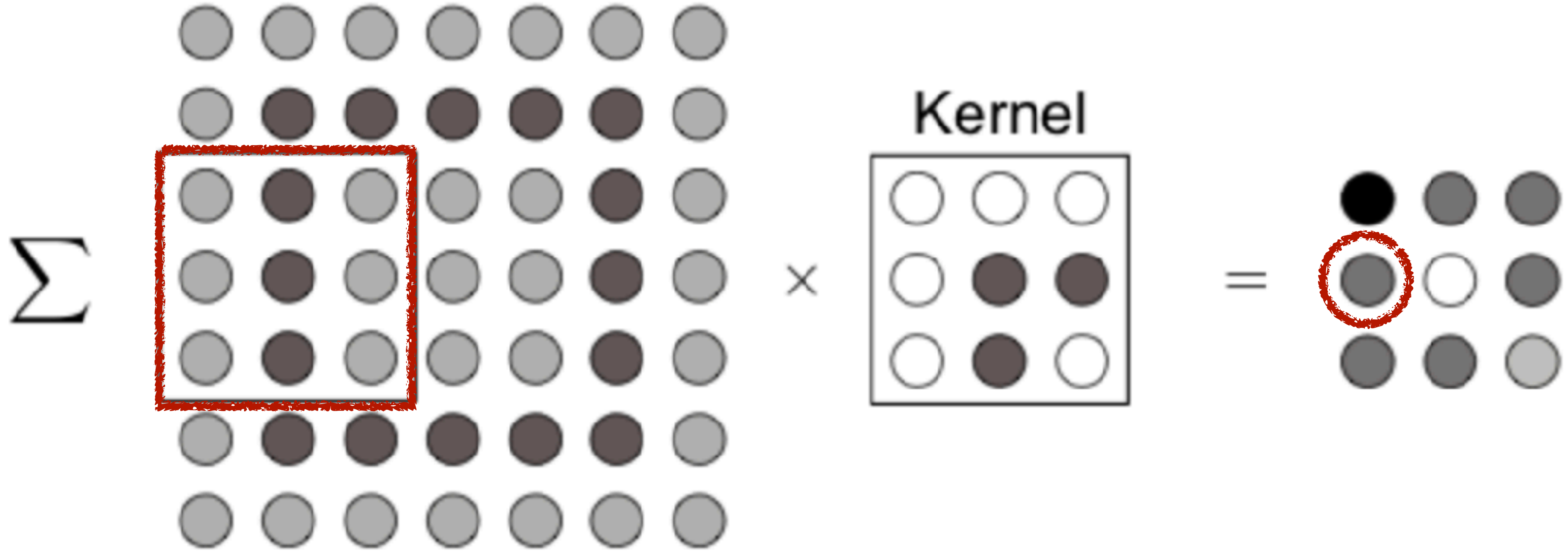
kernel = 3x3, stride = 2

We need more advanced layers: CONVOLUTIONS



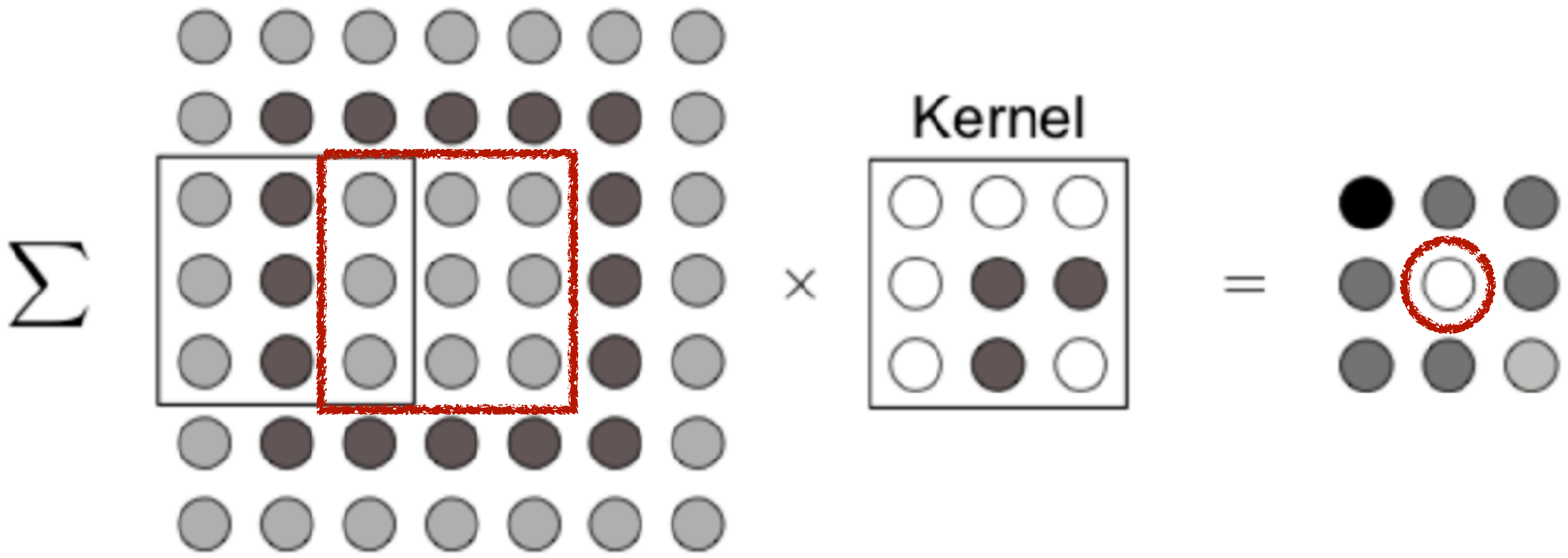
kernel = 3x3, stride = 2

We need more advanced layers: CONVOLUTIONS



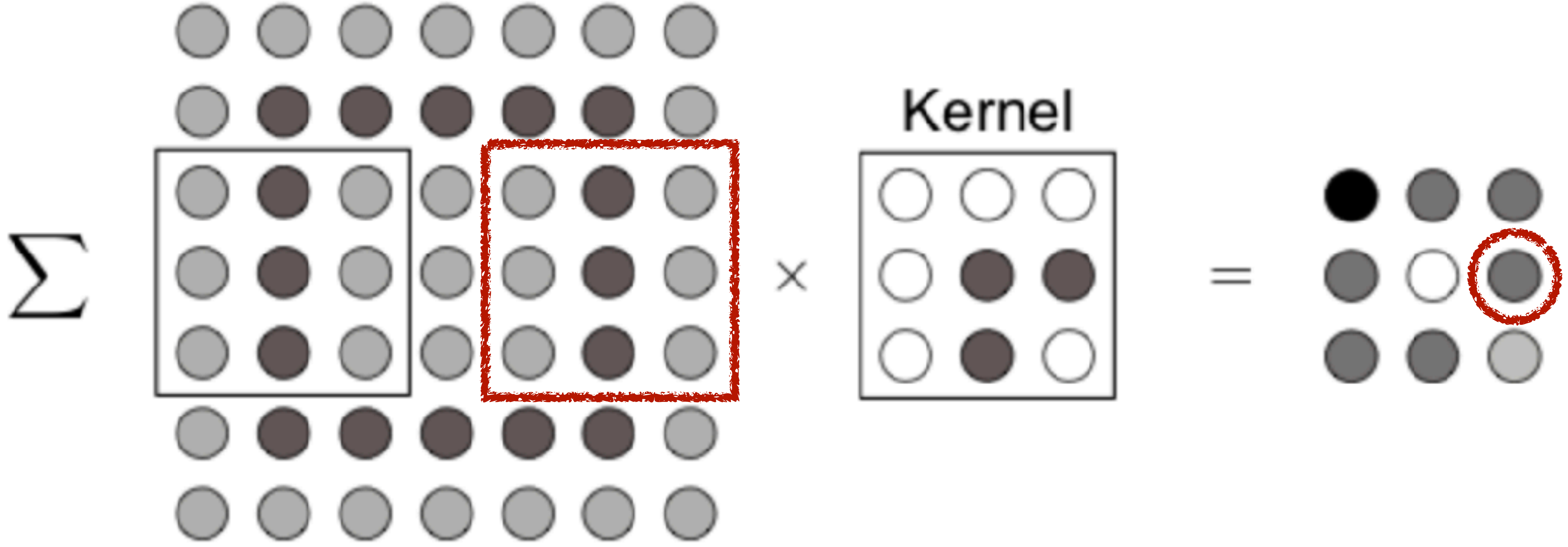
kernel = 3x3, stride = 2

We need more advanced layers: CONVOLUTIONS



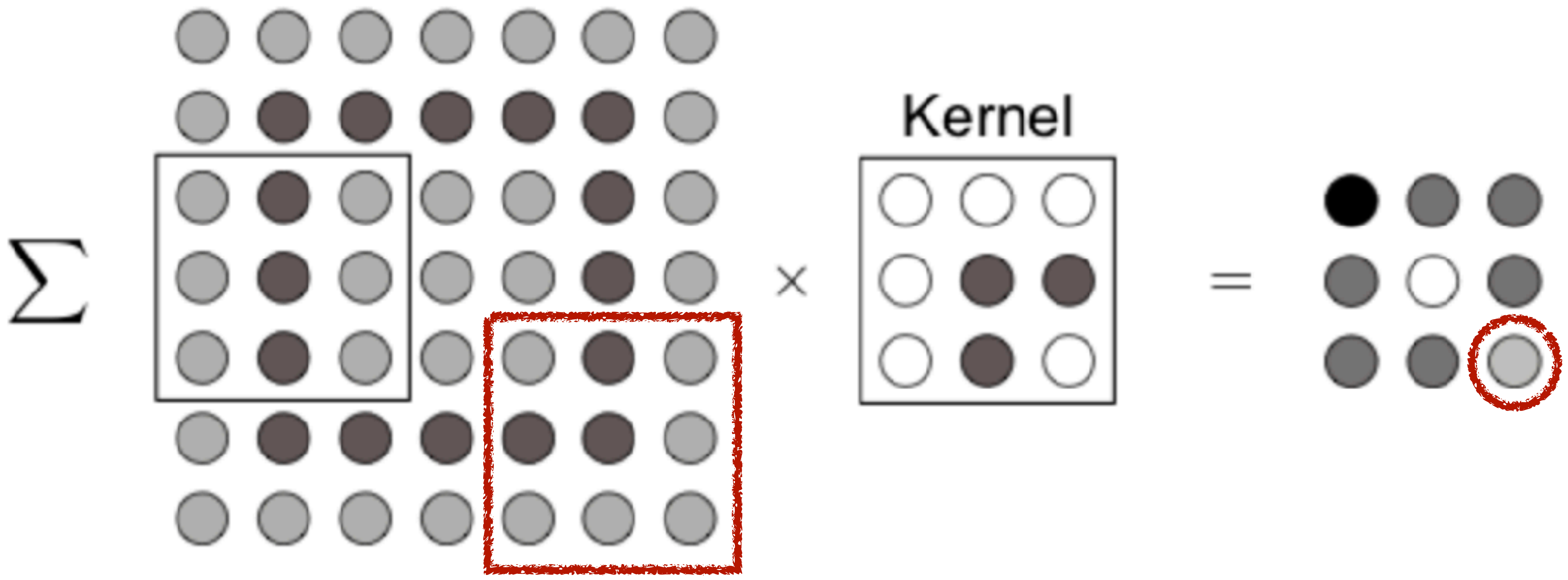
kernel = 3x3, stride = 2

We need more advanced layers: CONVOLUTIONS



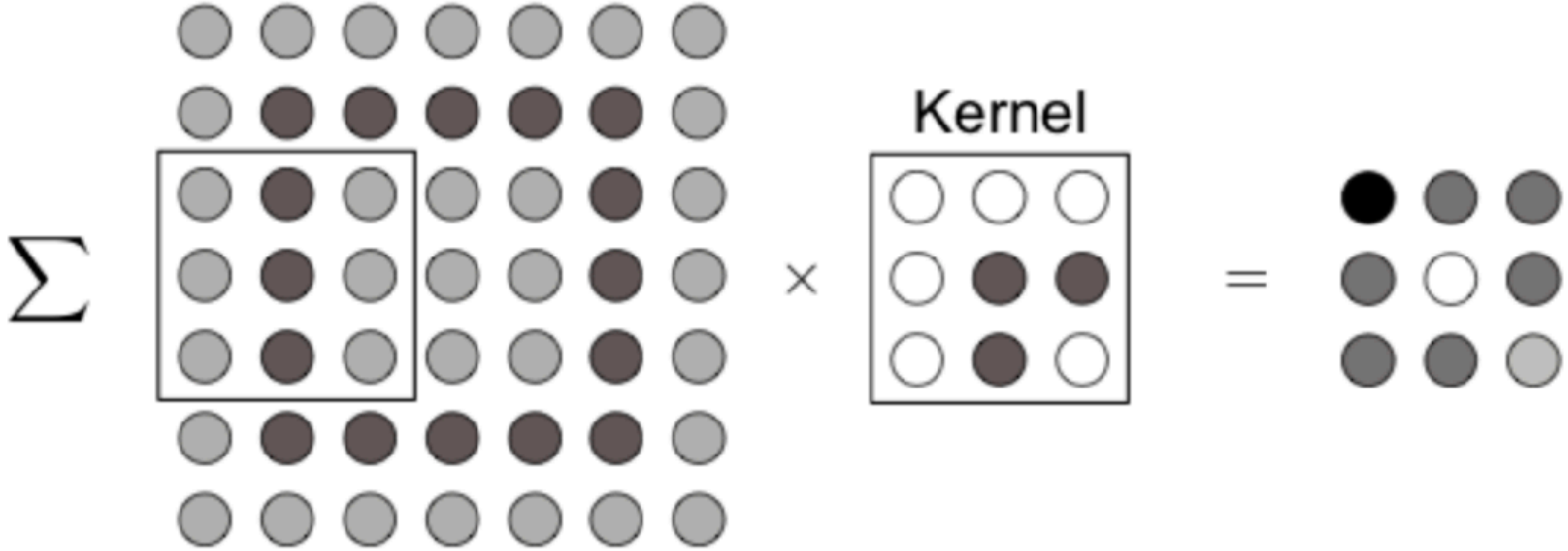
kernel = 3x3, stride = 2

We need more advanced layers: CONVOLUTIONS



kernel = 3x3, stride = 2

We need more advanced layers: CONVOLUTIONS



keras.layers.Conv2D(num_filters, (kernel_size, kernel_size), strides=(2,2), padding='Valid', input_shape=(), activation='relu')

We need more advanced layers: DROPOUT

- Overfitting on the training set is a serious practical issue
- Dropout is a great way to REGULARIZE

keras.layers.Dropout(0.3)

We need more advanced layers: DROPOUT

- Overfitting on the training set is a serious practical issue
- Dropout is a great way to REGULARIZE
- Other option: add reg terms to your loss function

$$R_{L1} = \frac{\lambda}{2} \sum_j |W_j|, \qquad R_{L2} = \frac{\lambda}{2} \sum_j W_j^2,$$

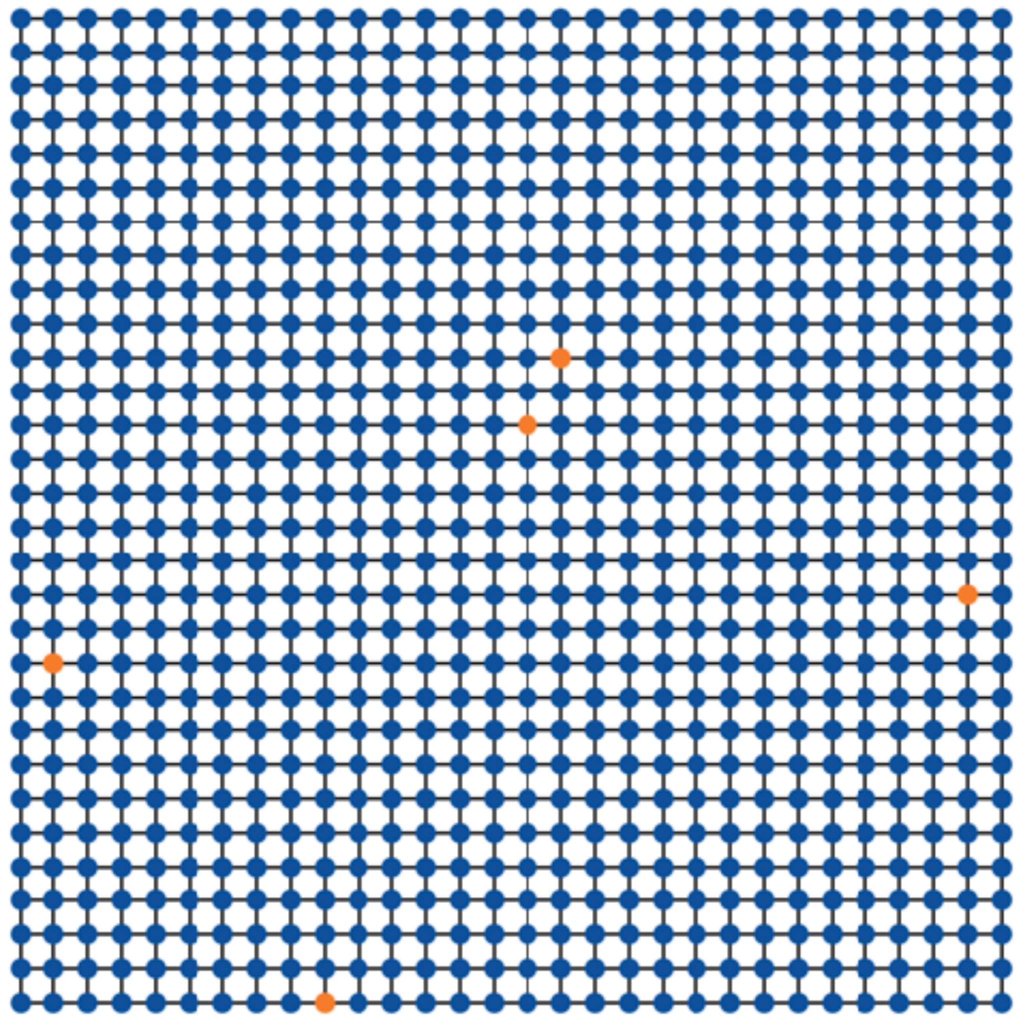
Notebook 2: Solving IGT

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 17, 17, 16)        144
_________________________________________________________________
flatten_2 (Flatten)          (None, 4624)              0
_________________________________________________________________
dense_4 (Dense)              (None, 8)                 37000
_________________________________________________________________
dropout_1 (Dropout)          (None, 8)                 0
_________________________________________________________________
dense_5 (Dense)              (None, 2)                 18
=================================================================
Total params: 37,162
Trainable params: 37,162
Non-trainable params: 0
_________________________________________________________________
```

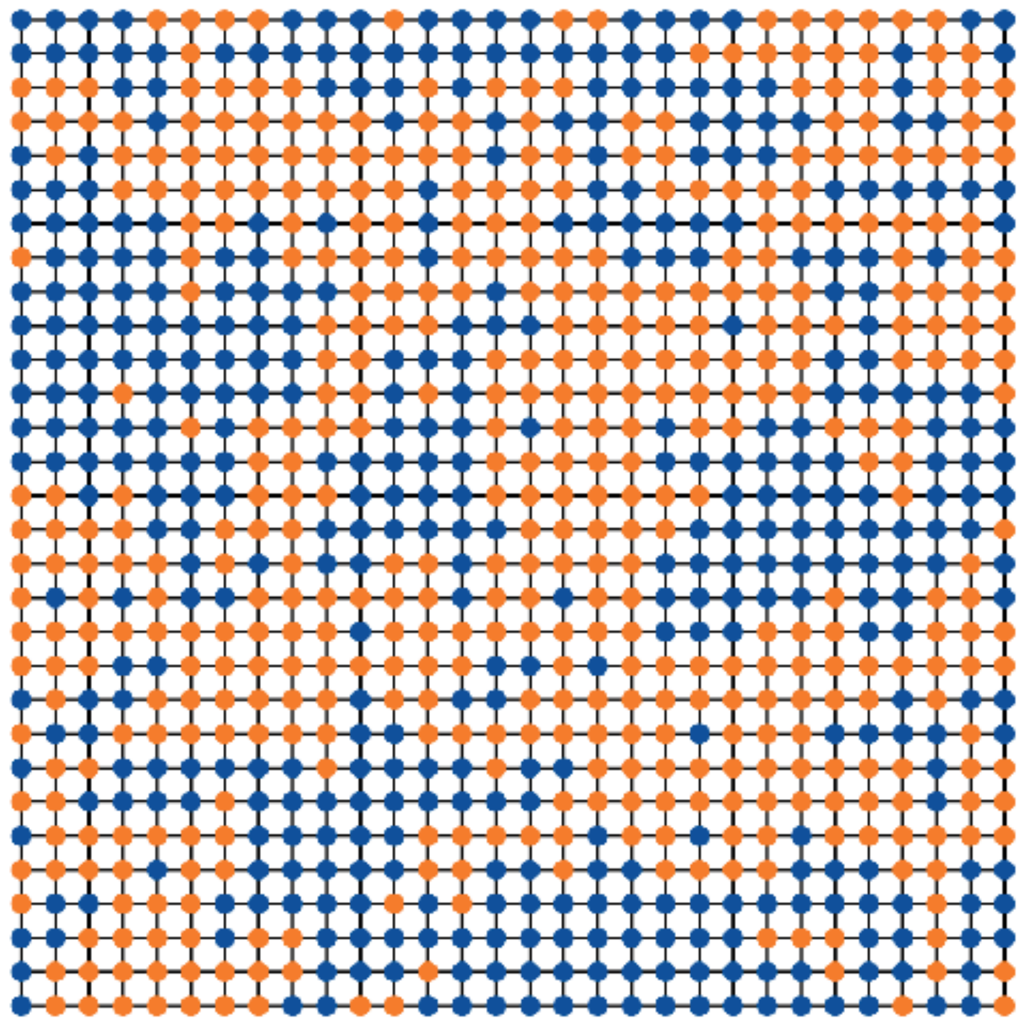
You will need both convolutions and dropout to make it work :)

BREAK 15 mins

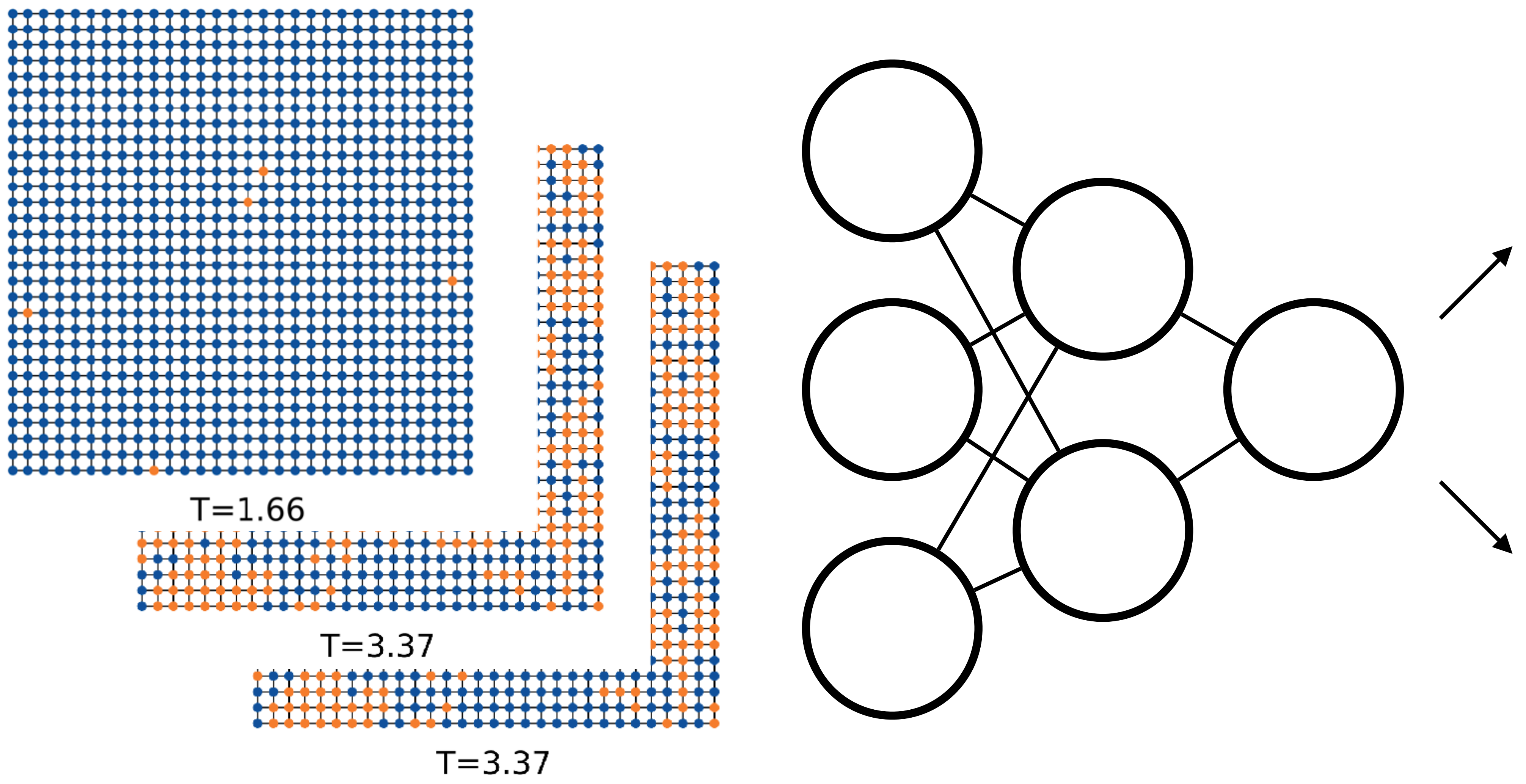
Notebook 2: Supervised learning



PHASE A



PHASE B



Neural nets: More sophisticated methods

Previously we have seen:

(1) clustering works elegantly for simple problems, does not generalise well for hard ones

(2) supervised learning works great for simple and hard problems BUT if we had to label it first are we learning something new?

Learning by confusion

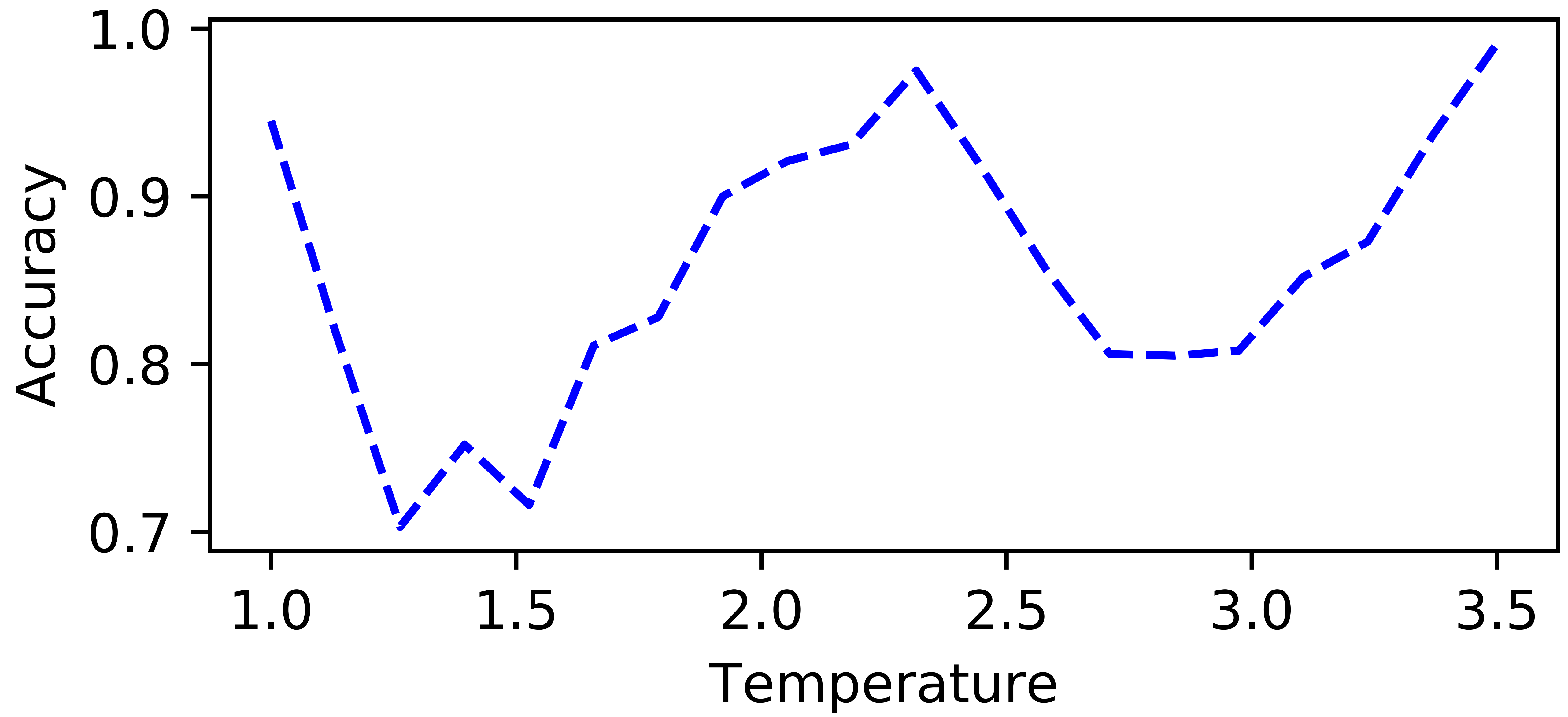
STEP 1: make a guess of phase transition temperature T_c

STEP 2: train a classification model assuming your guess in step 1 in a correct temperature

STEP 3: train many models as many guess of T_c as you like

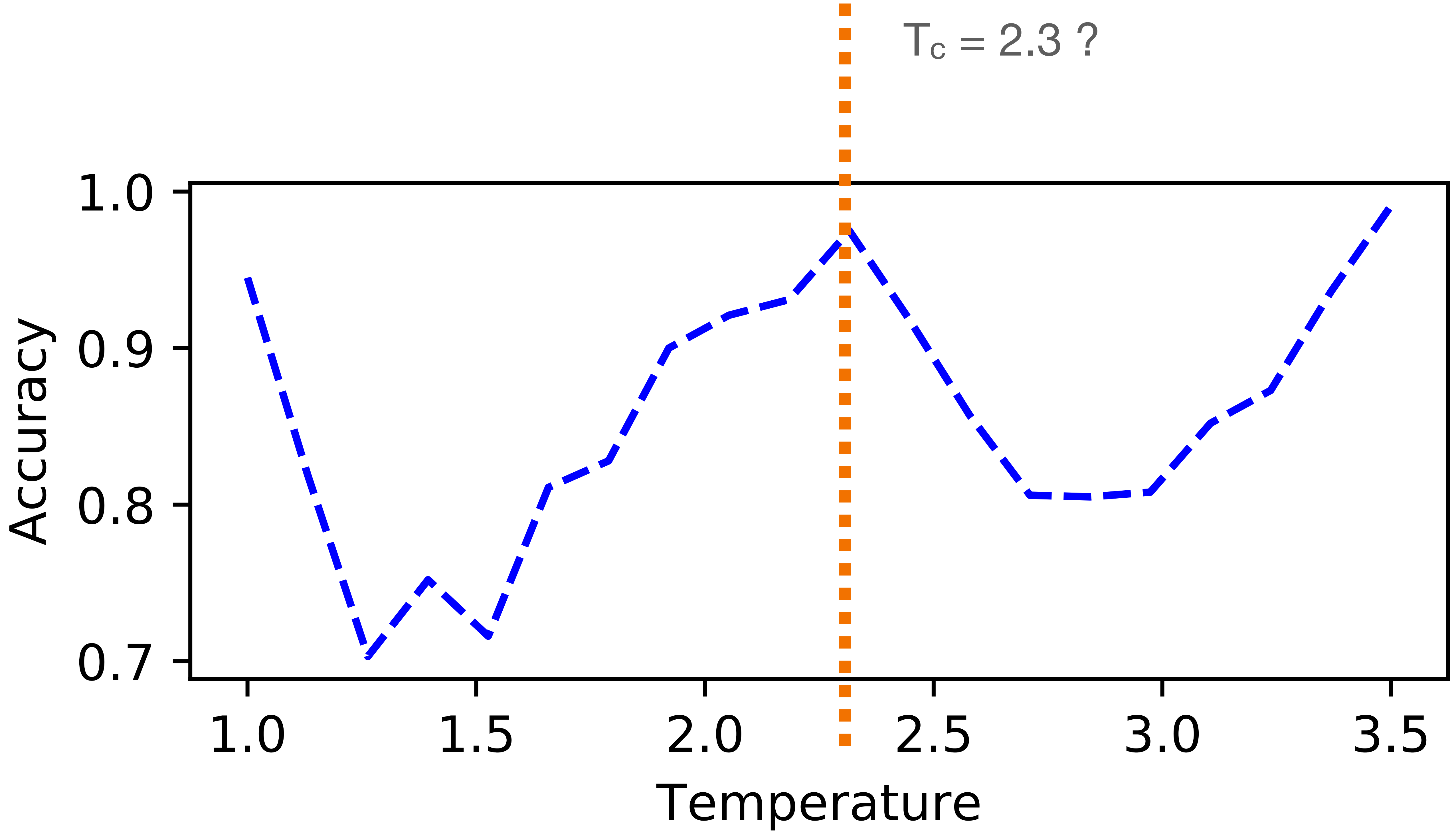
STEP 4: look at the accuracy of your model as a function of T_c

Learning by confusion: Notebook 3

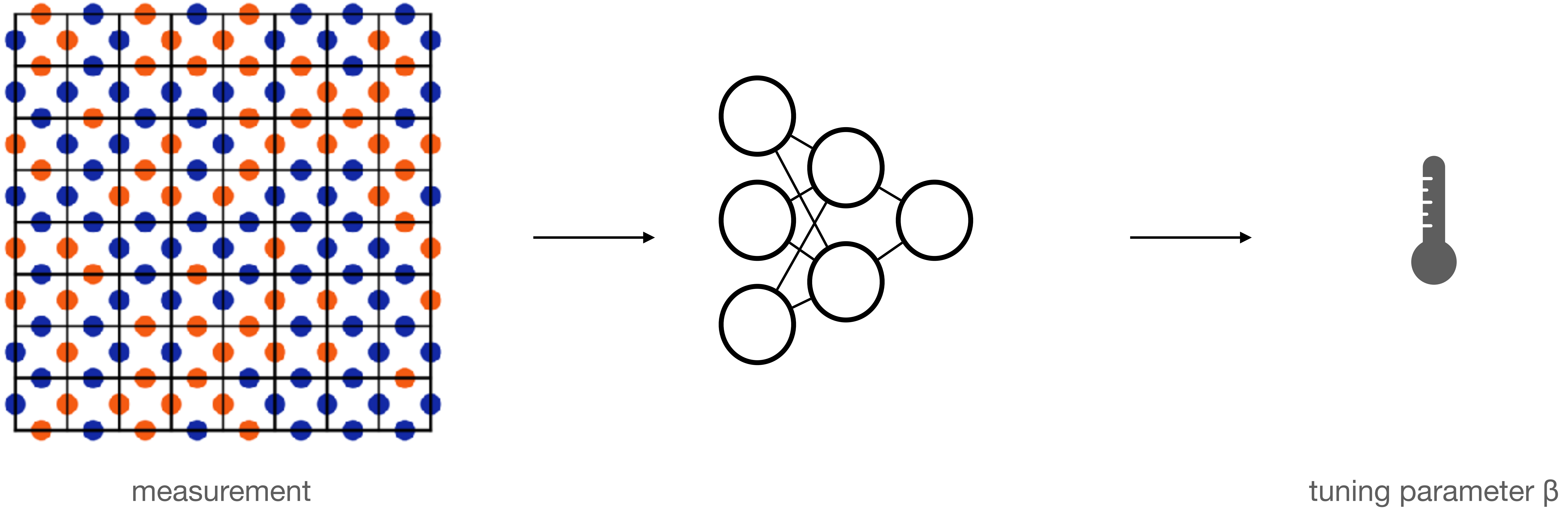


QMAI | TU Delft

Learning by confusion: Notebook 3



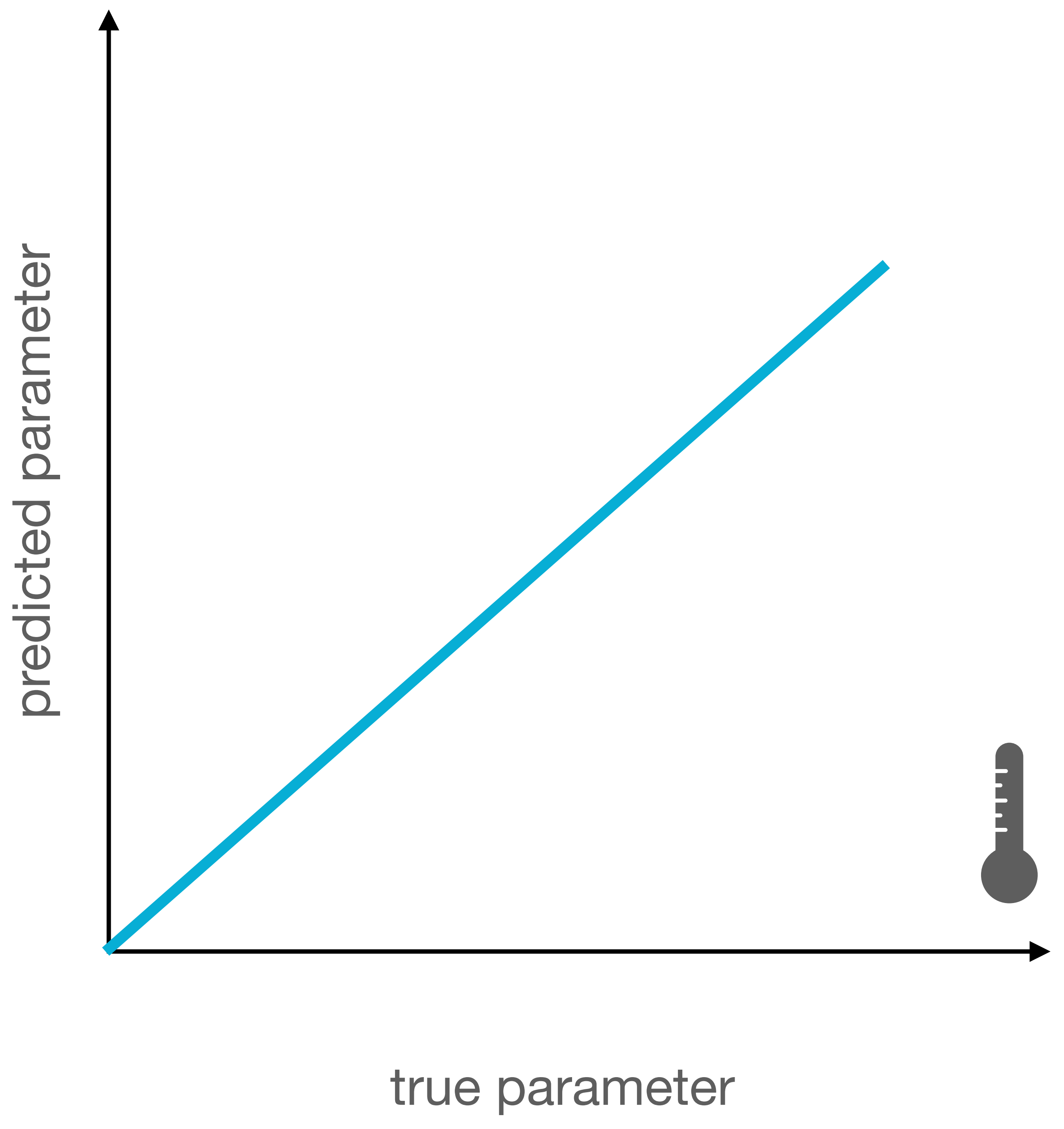
Unsupervised learning with a predictive model

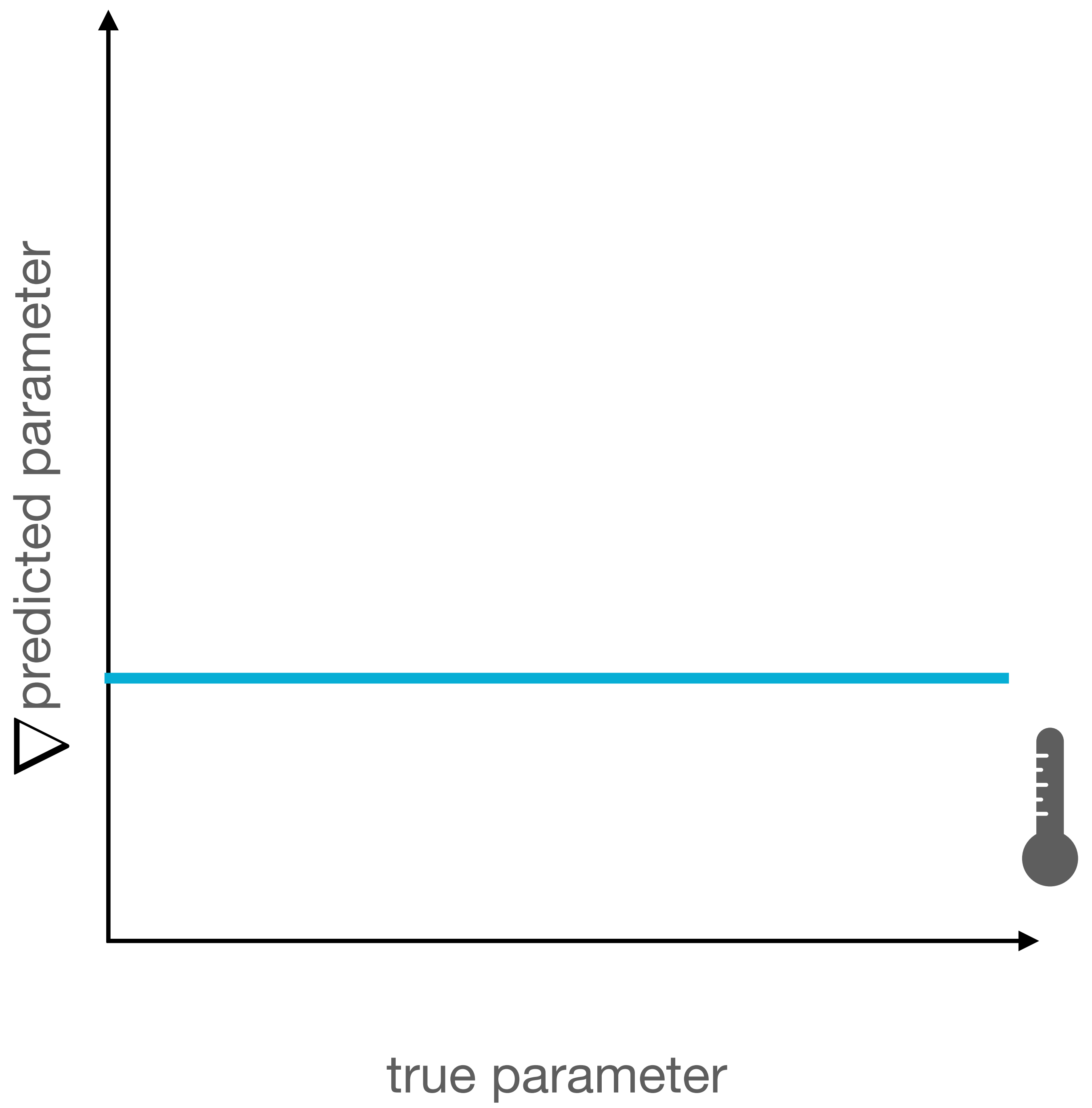


EG, A. Valenti, G. Boschung, F. Schäfer, N. Lörch, S. Huber, New J. Phys. 22 045003 (2020)
F. Schäfer, N. Lörch, Phys. Rev. E 99, 062107 (2019)

QMAI | TU Delft

Unsupervised learning:
Can we discover new phases just from data?

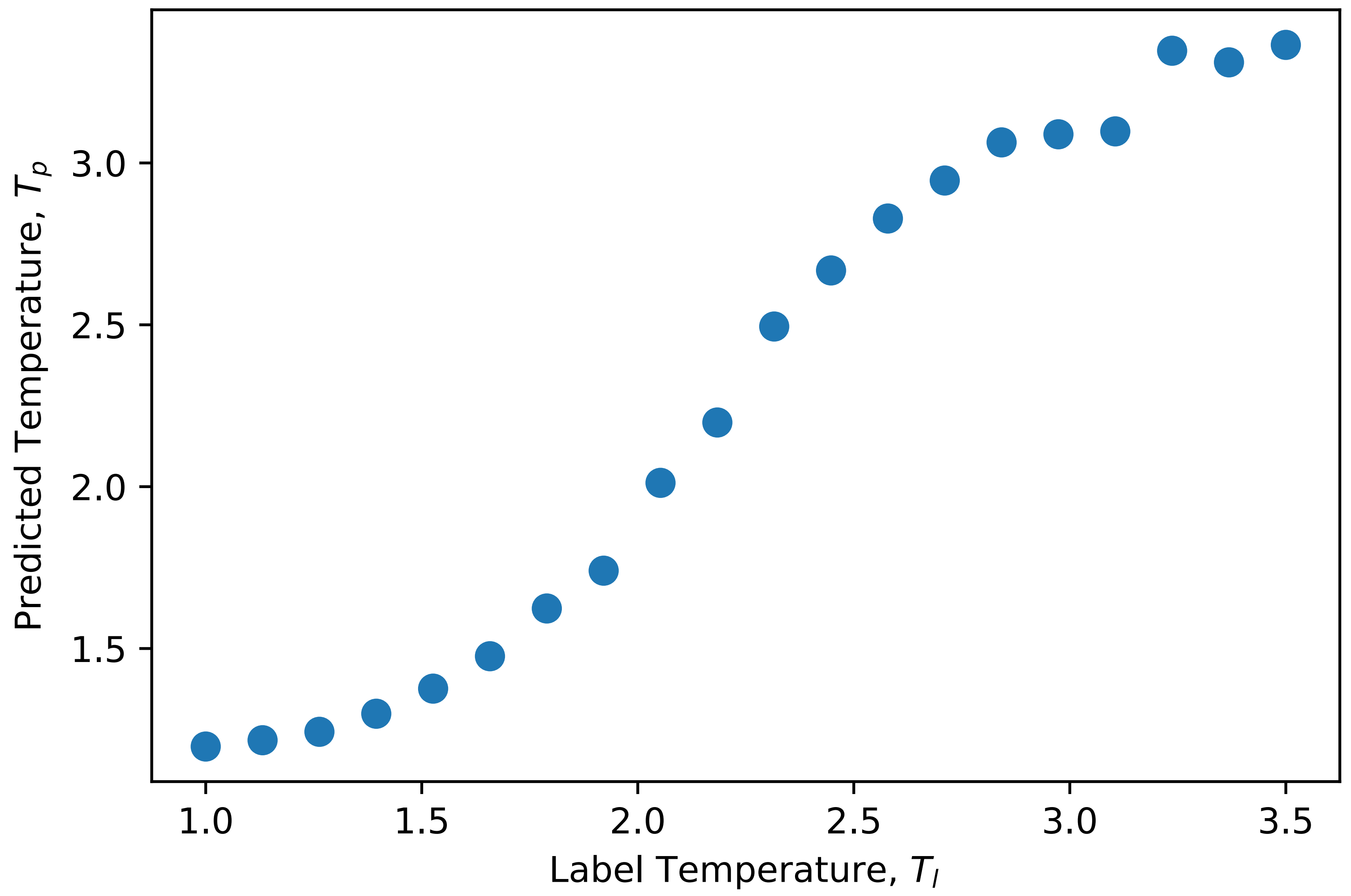


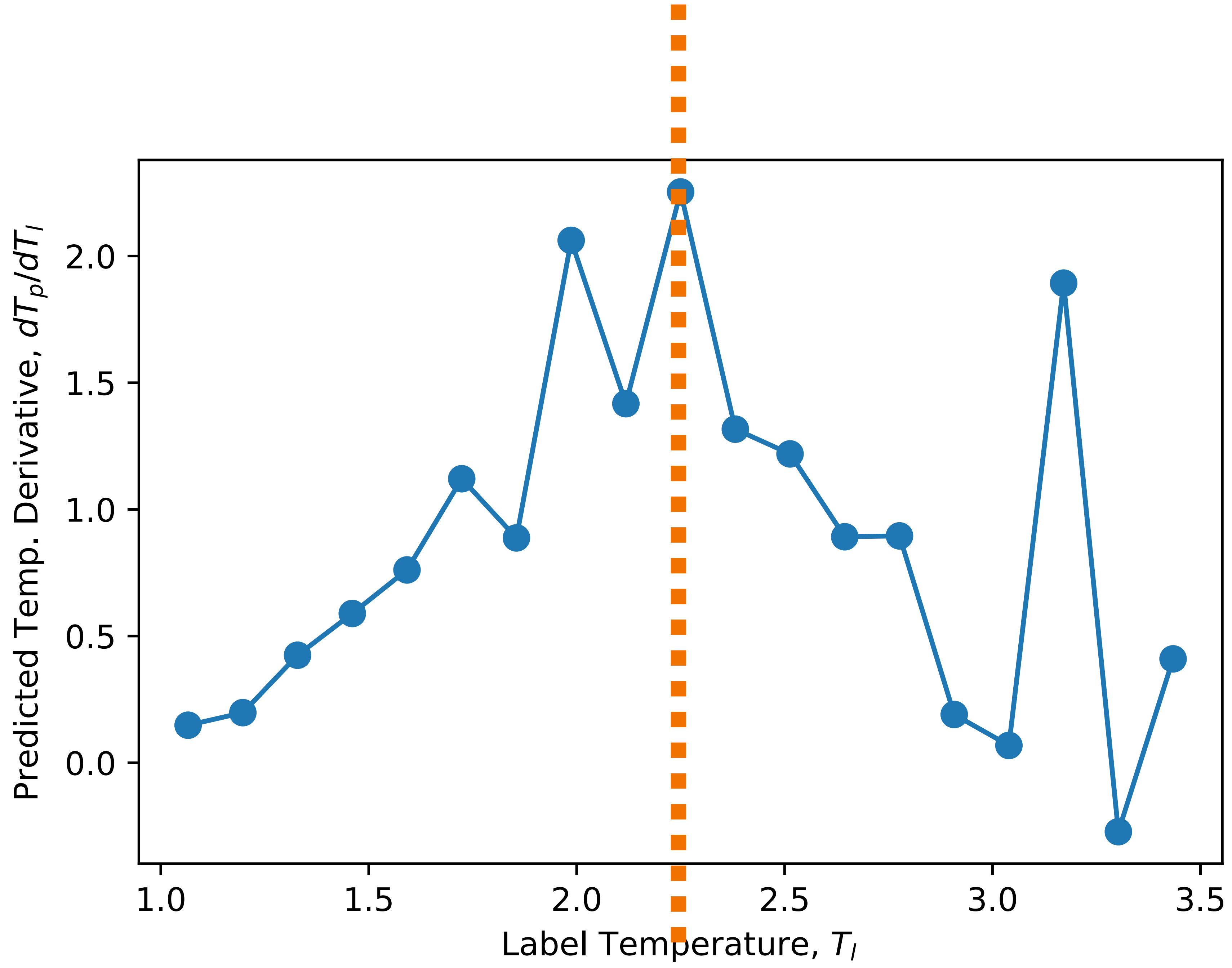


EG, A. Valenti, G. Boschung, F. Schäfer, N. Lörch, S. Huber, New J. Phys. 22 045003 (2020)
F. Schäfer, N. Lörch, Phys. Rev. E 99, 062107 (2019)

QMAI | TU Delft

Unsupervised learning:
Can we discover new phases just from data?





EG, A. Valenti, G. Boschung, F. Schäfer, N. Lörch, S. Huber, New J. Phys. 22 045003 (2020)
F. Schäfer, N. Lörch, Phys. Rev. E 99, 062107 (2019)

QMAI | TUDelft

Unsupervised learning: Can we discover new phases just from data?

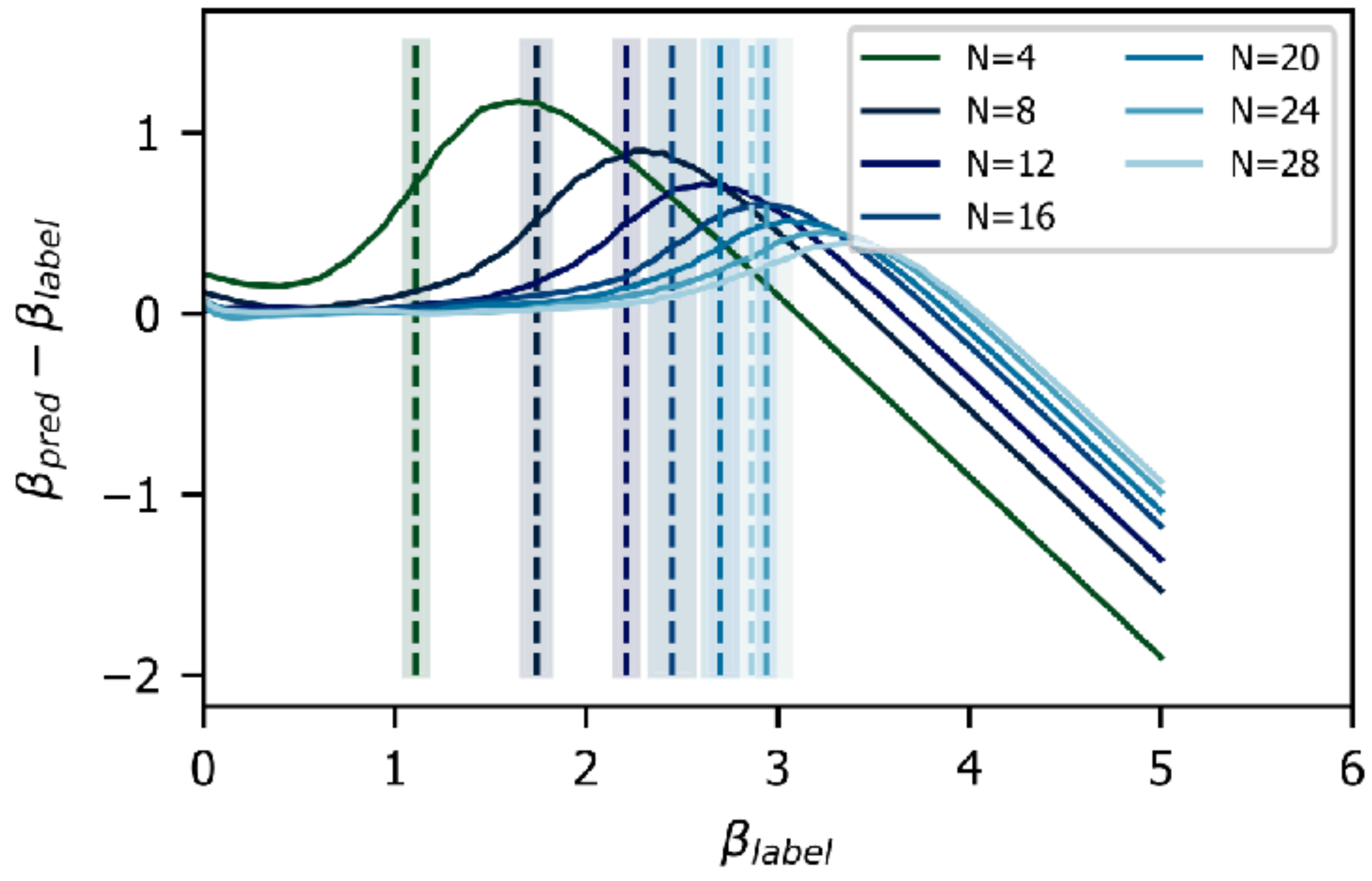
STEP 1: train your network to predict the parameter you know: temperature of the sample

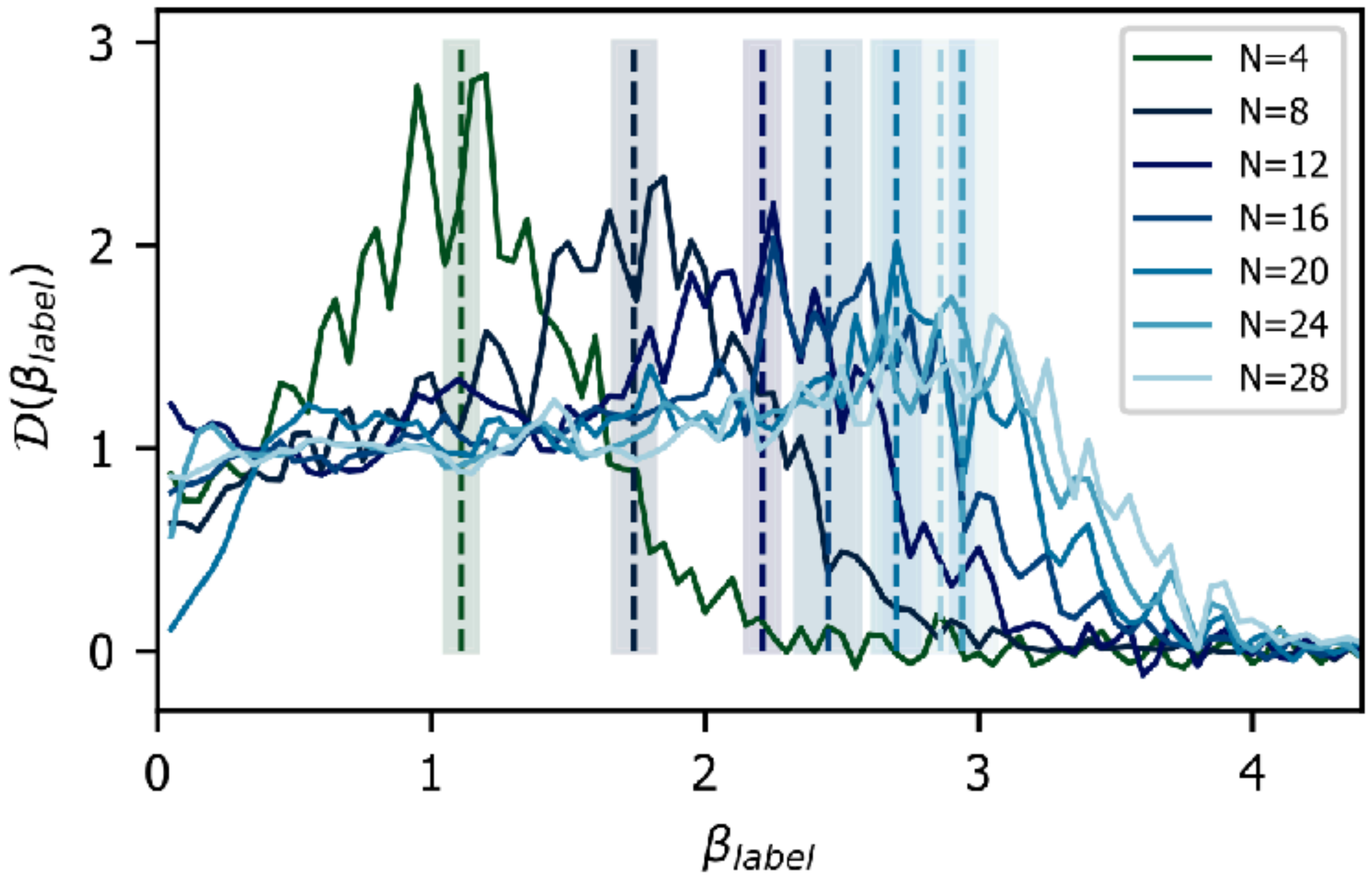
STEP 2: for the validation data plot correct temperature vs predicted temperature

STEP 3: take numerical derivative of STEP 2

STEP 4: what is the temperature for which the network make the biggest mistake?

Unsupervised learning: IGT



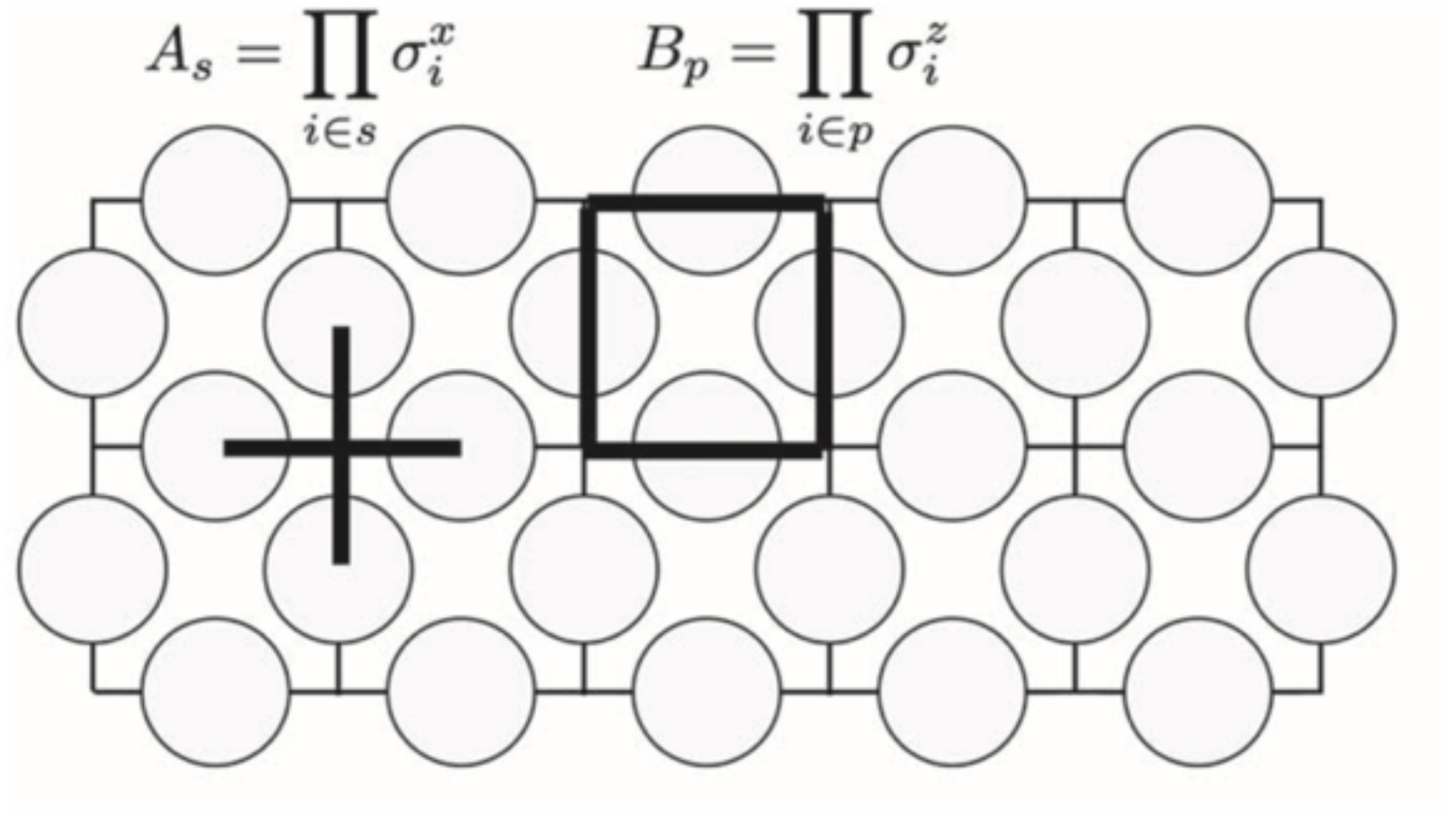


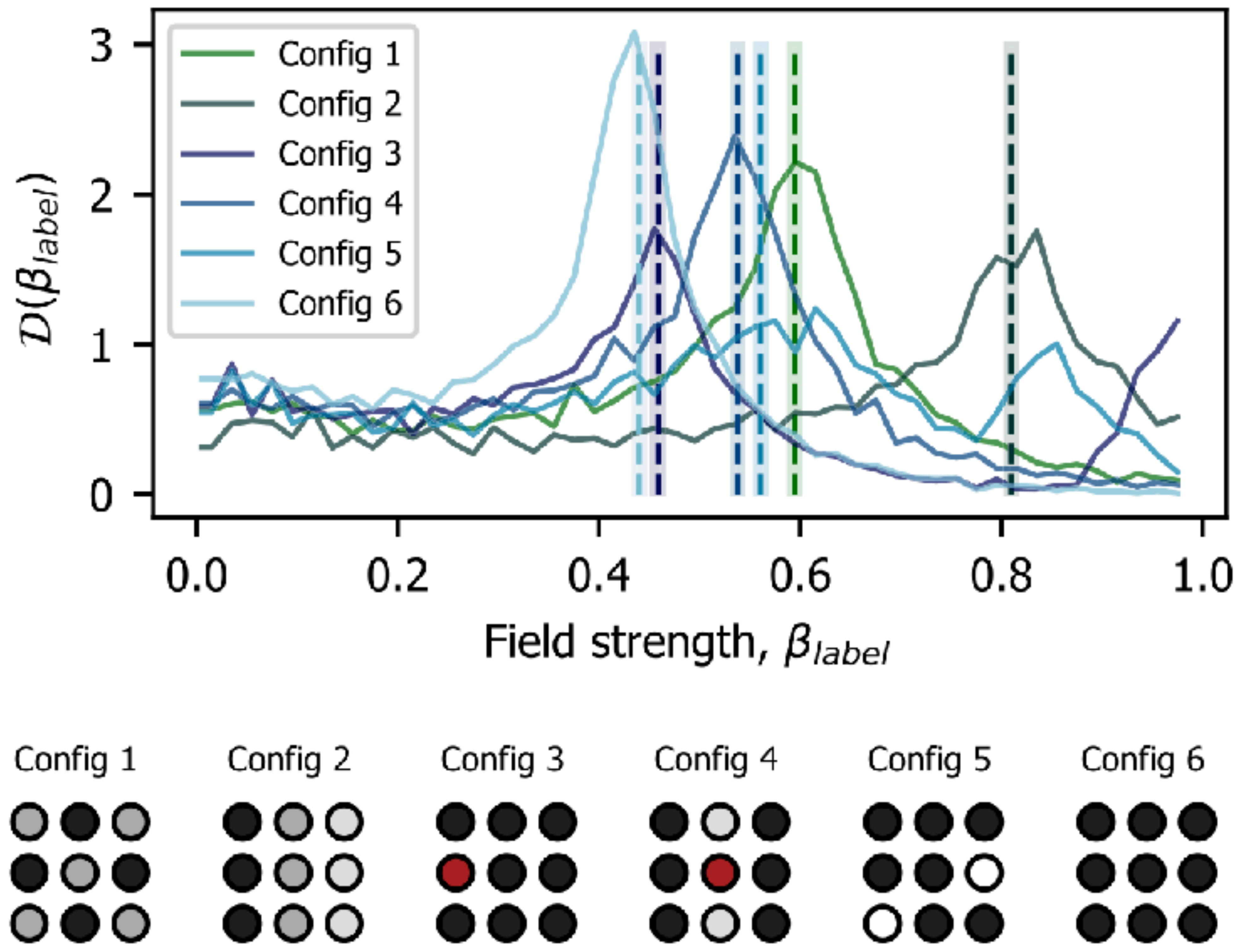
EG, A. Valenti, G. Boschung, F. Schäfer, N. Lörch, S. Huber, New J. Phys. 22 045003 (2020)
F. Schäfer, N. Lörch, Phys. Rev. E 99, 062107 (2019)

Unsupervised learning

For funky quantum topological phase transitions it works too!

TORIC CODE

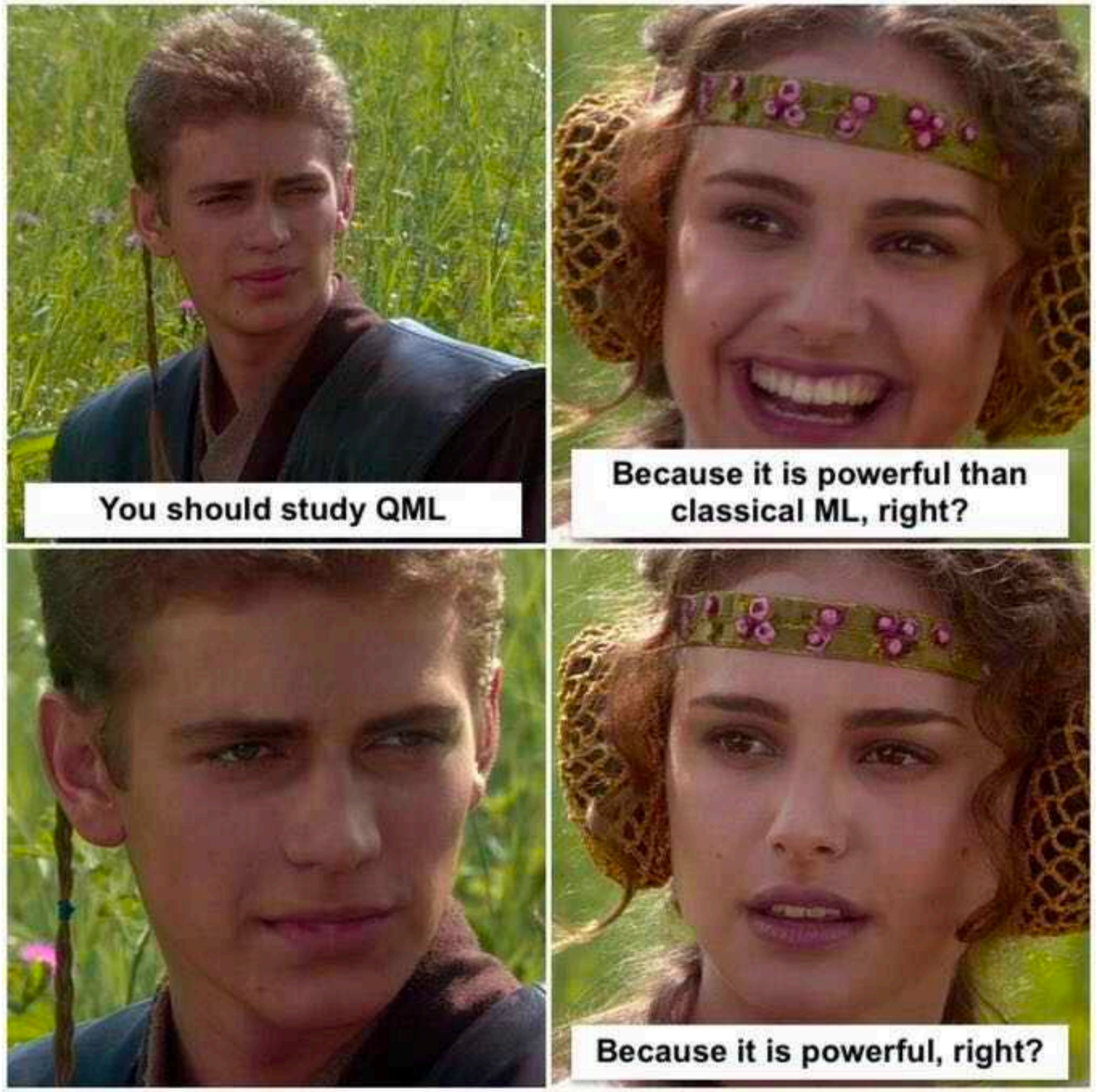




EG, A. Valenti, G. Boschung, F. Schäfer, N. Lörch, S. Huber, New J. Phys. 22 045003 (2020)
F. Schäfer, N. Lörch, Phys. Rev. E 99, 062107 (2019)

QMAI | TU Delft

Break



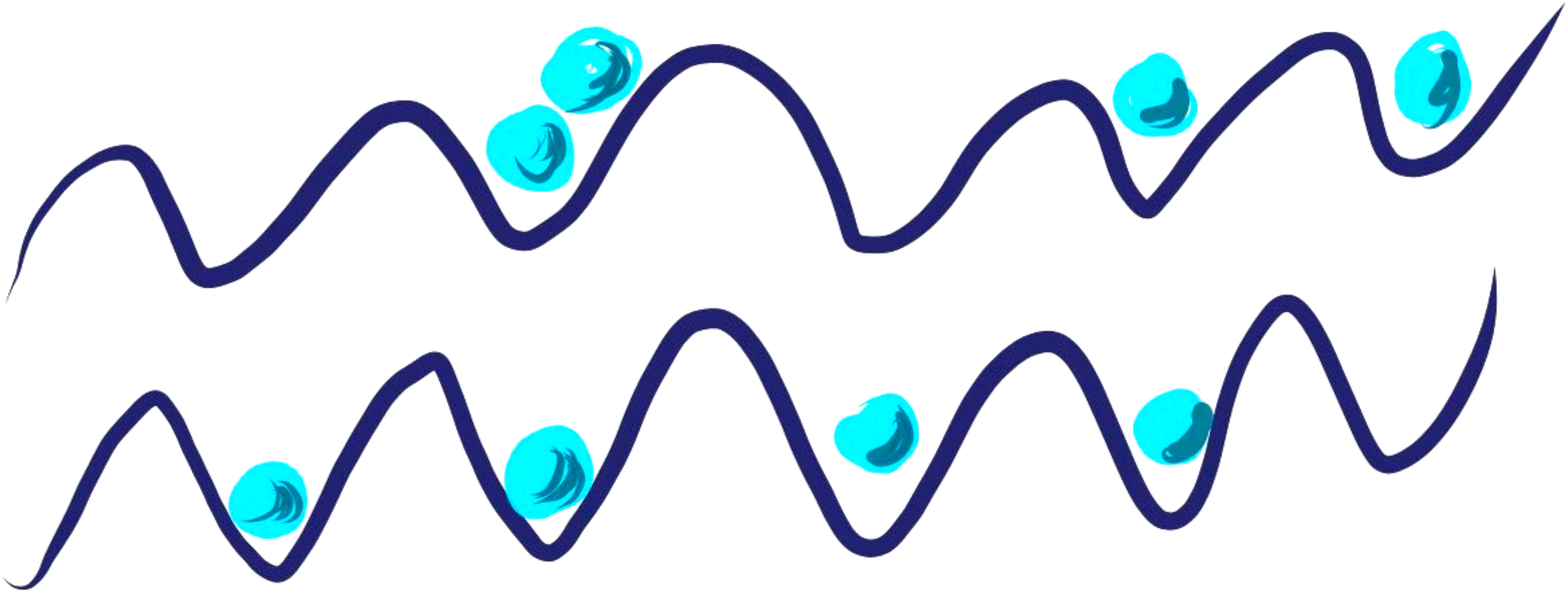
“We have a device that is more complex than anything classical computers can simulate - great!

Q: How do we then verify the device is doing what it should and producing correct results if there is no other computer on Earth that can simulate that exact same physics?”

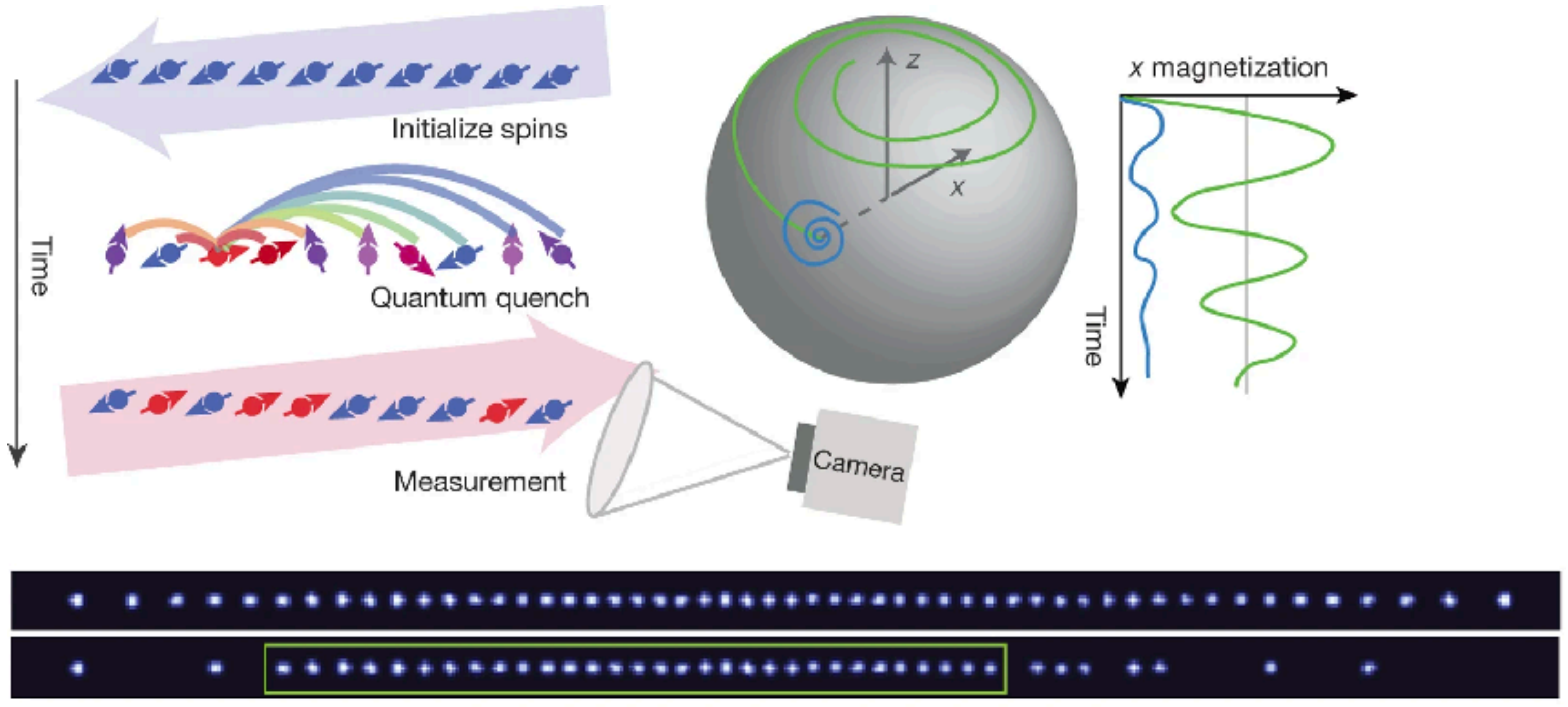
QMAI | TUDelft

Neural net learning + State of art experiments

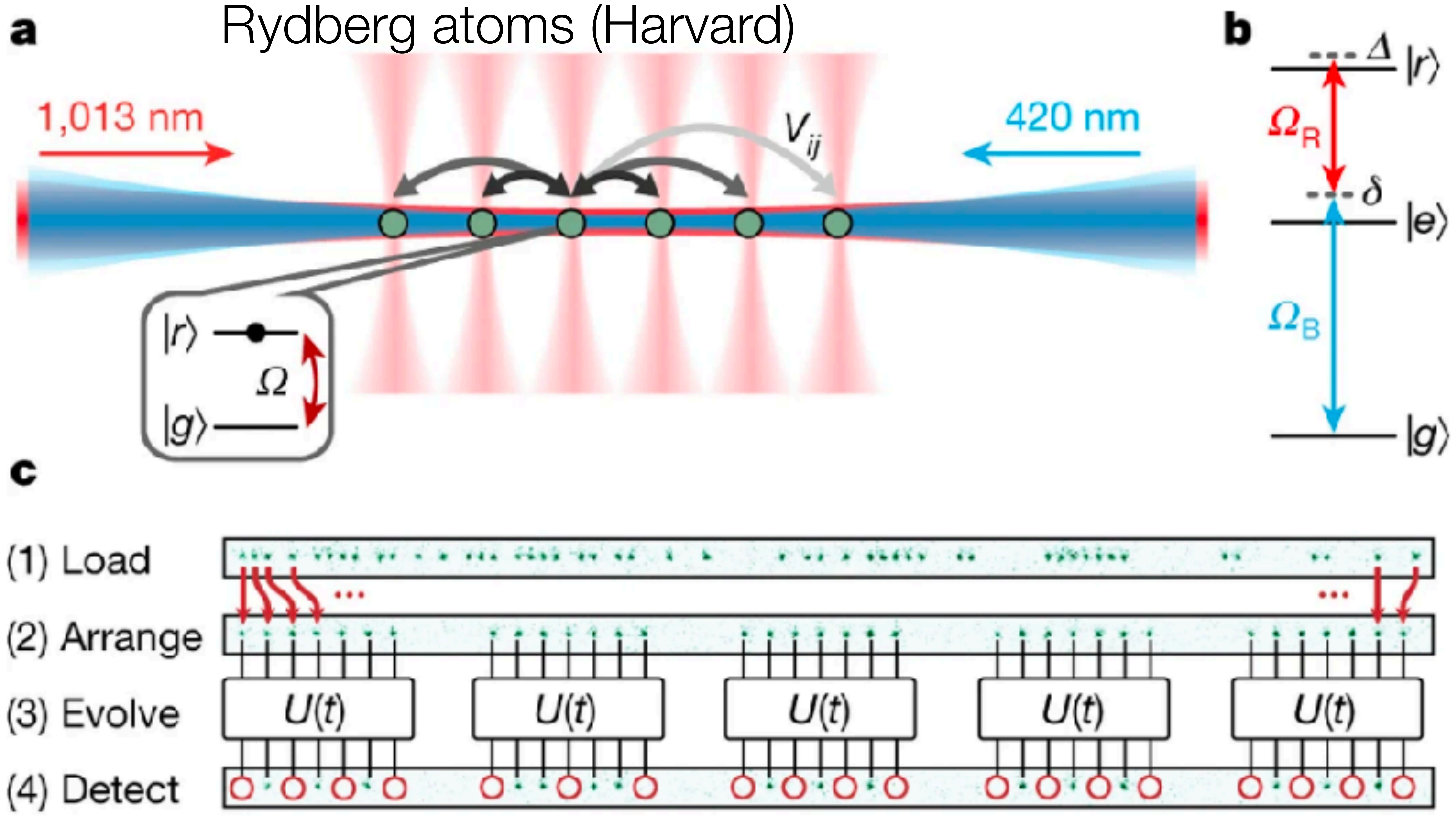
Large Scale Quantum Simulation



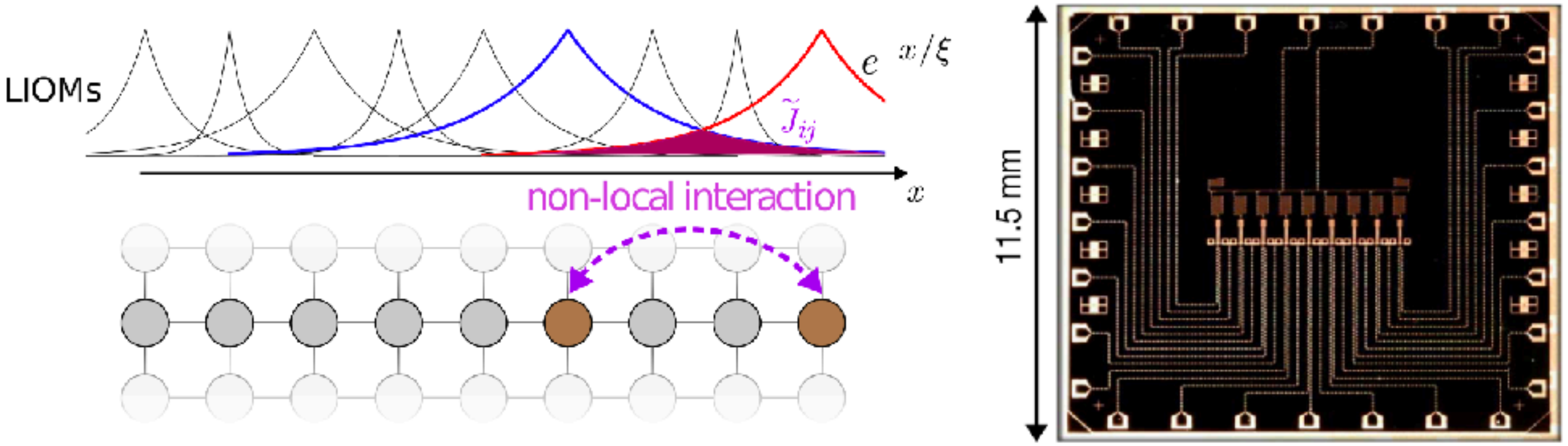
Zhang et al, Nature 551, 601–604 (2017)
Trapped Ions (Maryland)



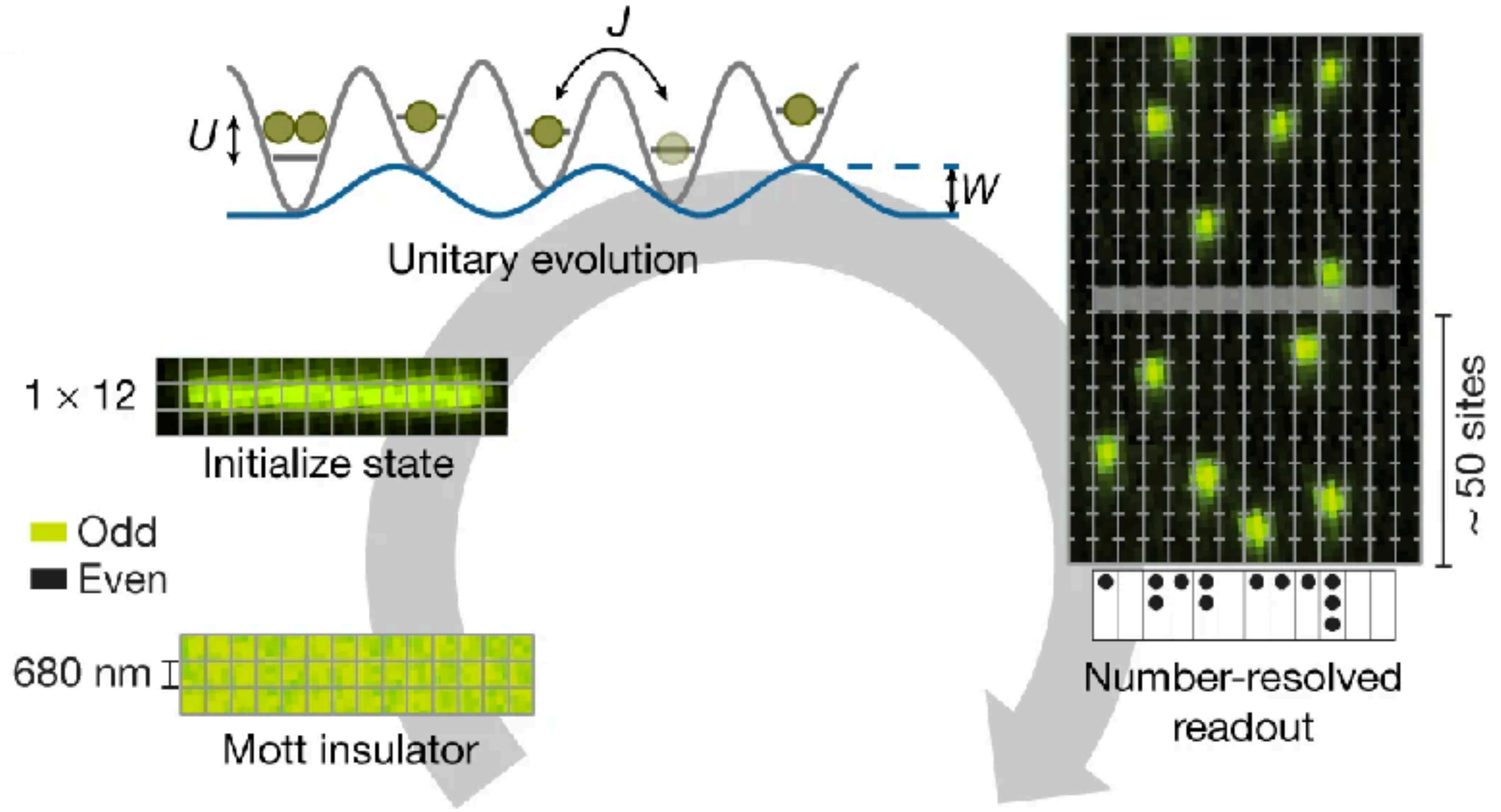
Bernien et al, Nature **551**,579–584 (2017)
Rydberg atoms (Harvard)



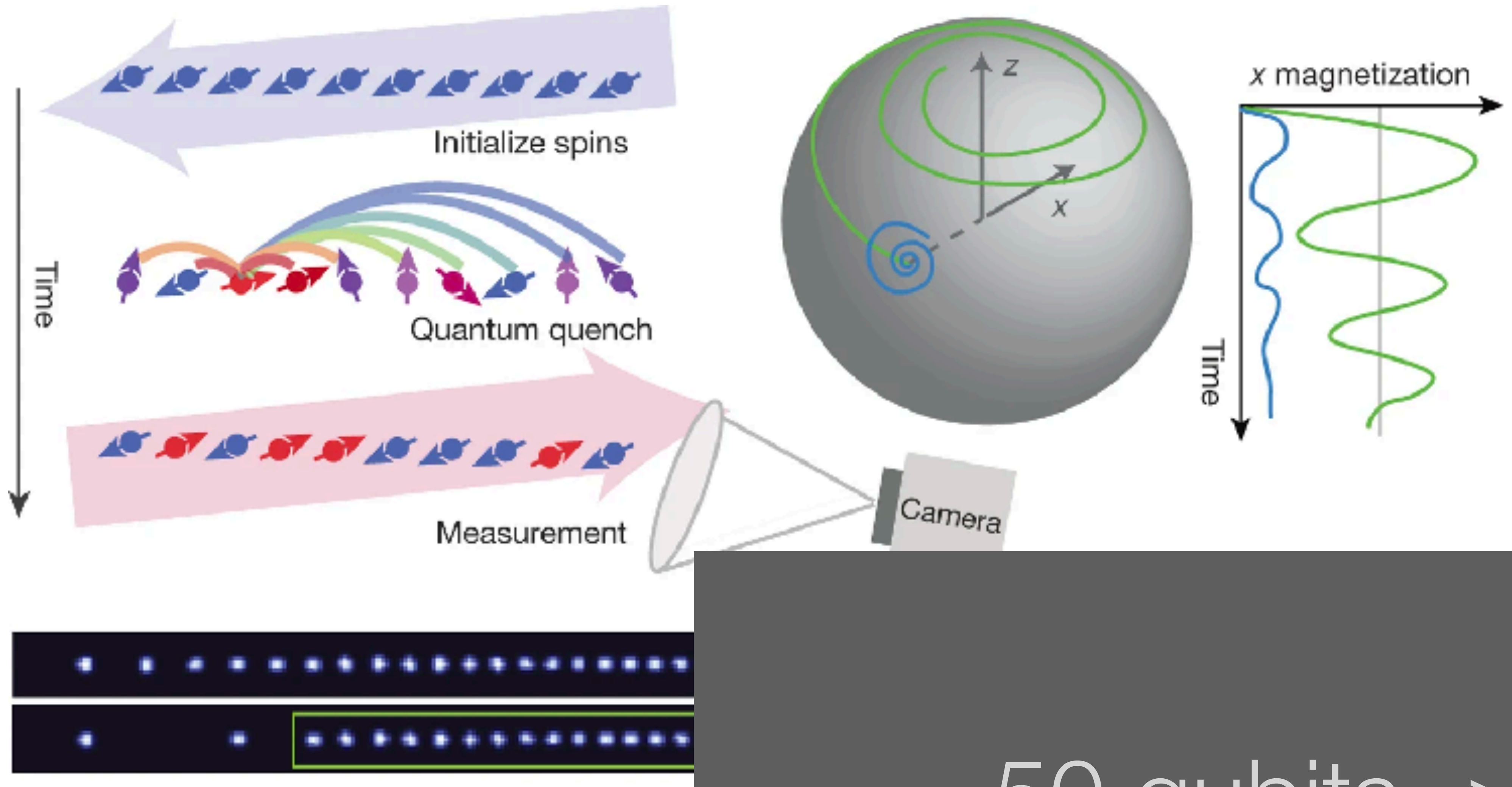
Chiaro et al, arXiv:1910.06024 (2019)
Superconducting qubits (Google)



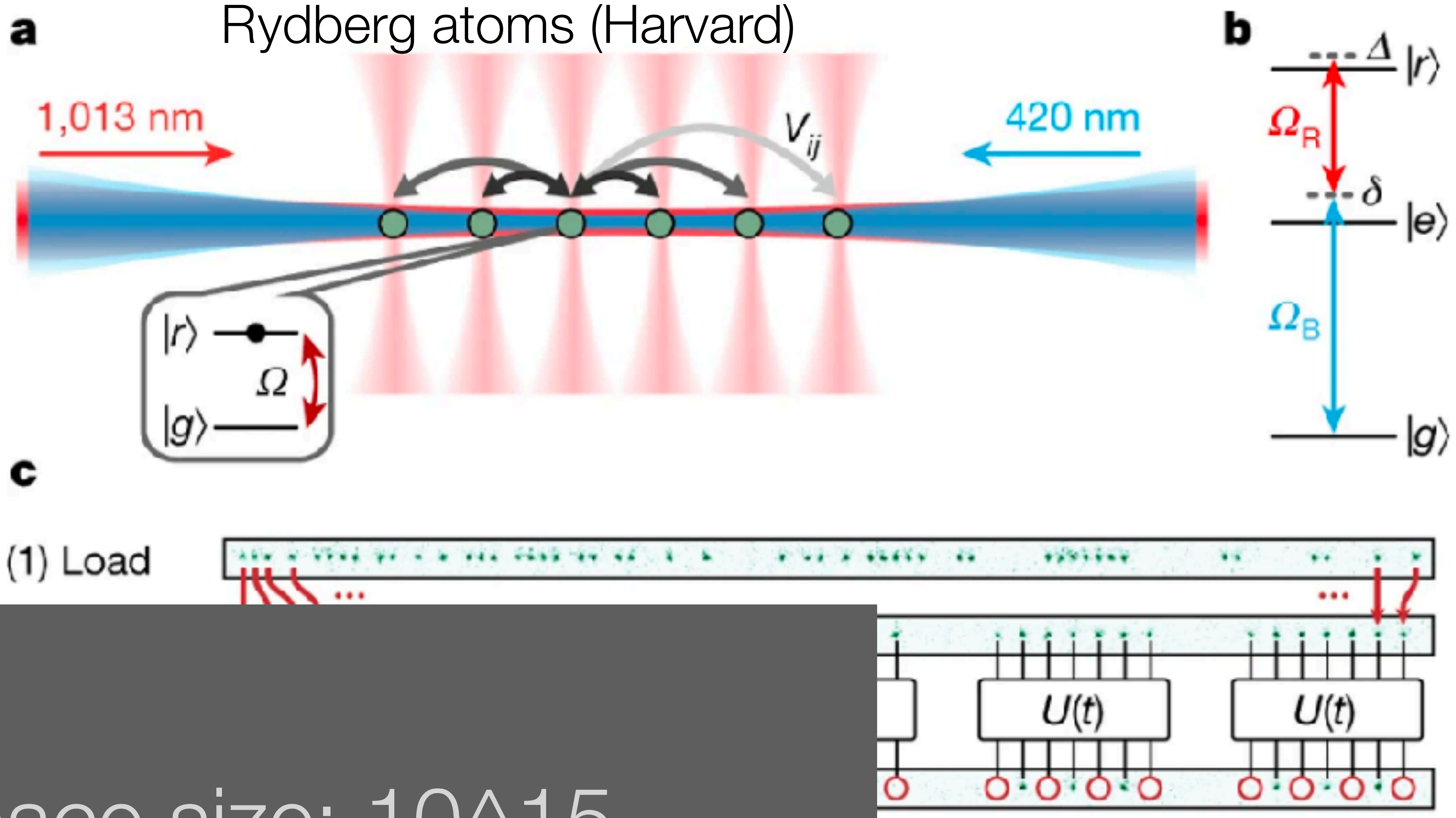
Rispoli et al, Nature **573**, 385–389 (2019)
Cold atoms in an optical lattice (Harvard)



Zhang et al, Nature 551, 601–604 (2017)
Trapped Ions (Maryland)

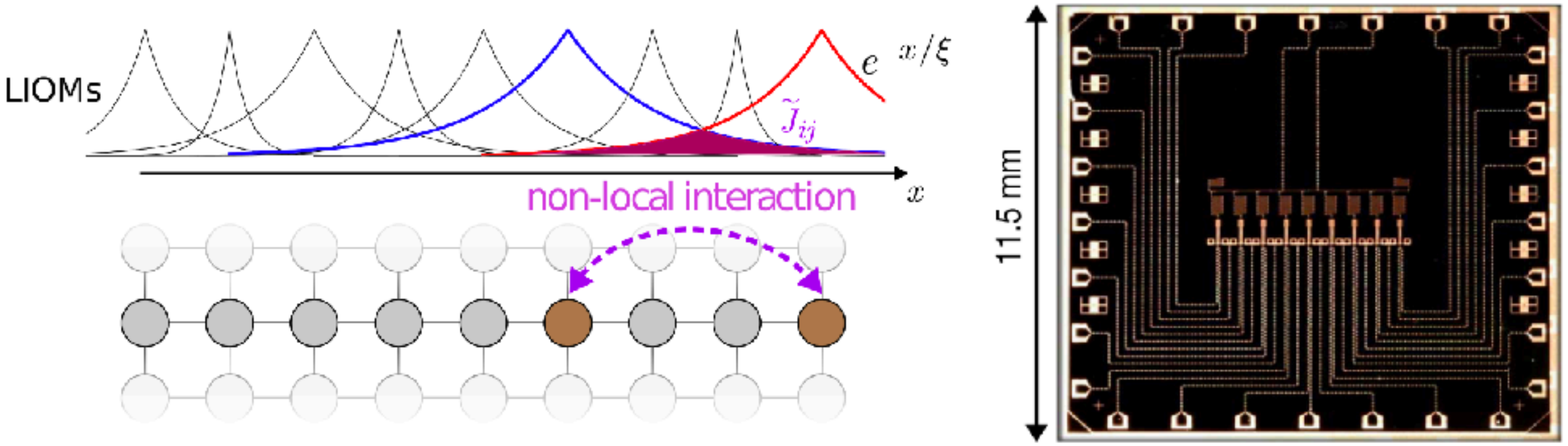


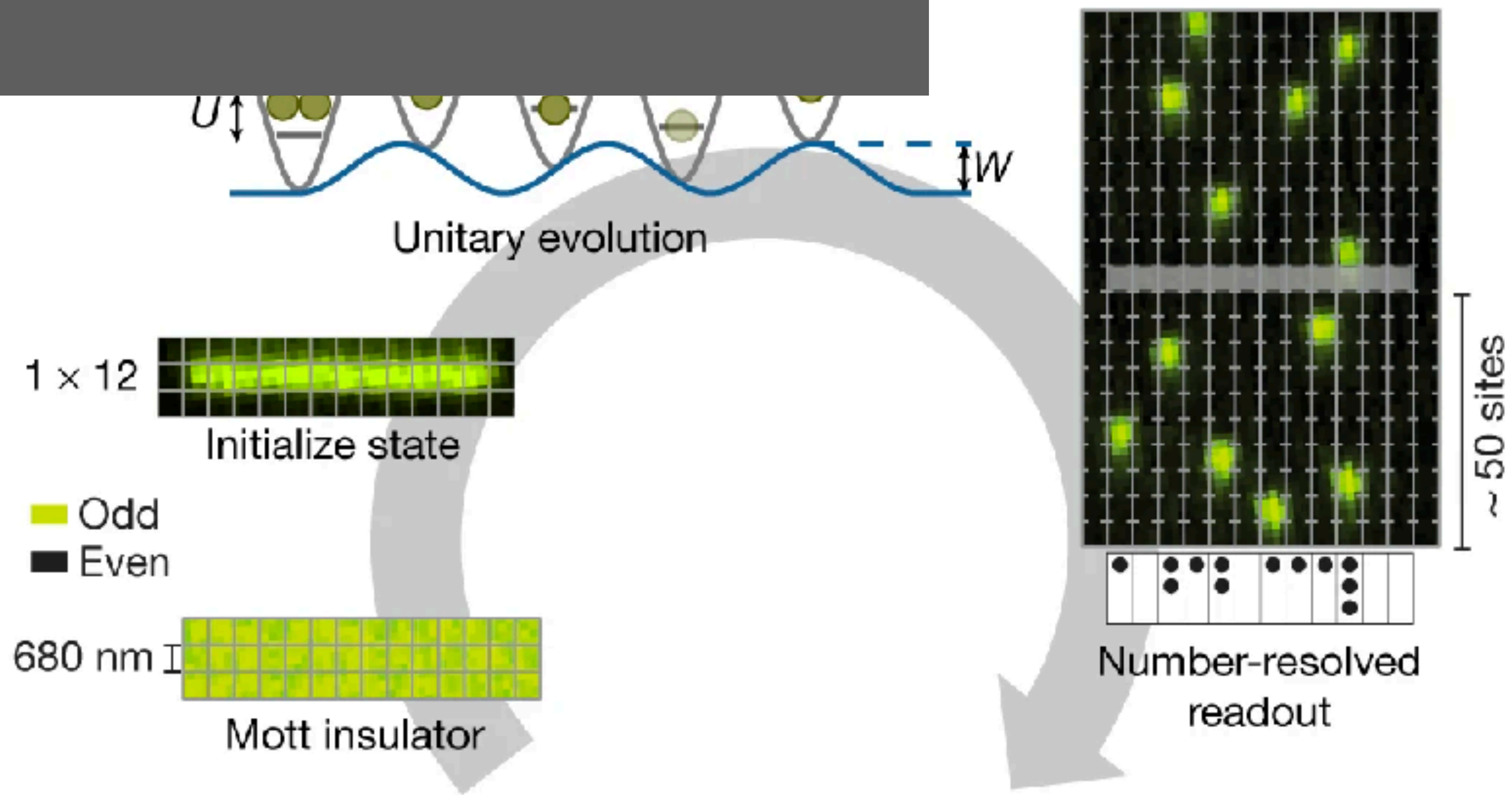
Bernien et al, Nature **551**,579–584 (2017)
Rydberg atoms (Harvard)



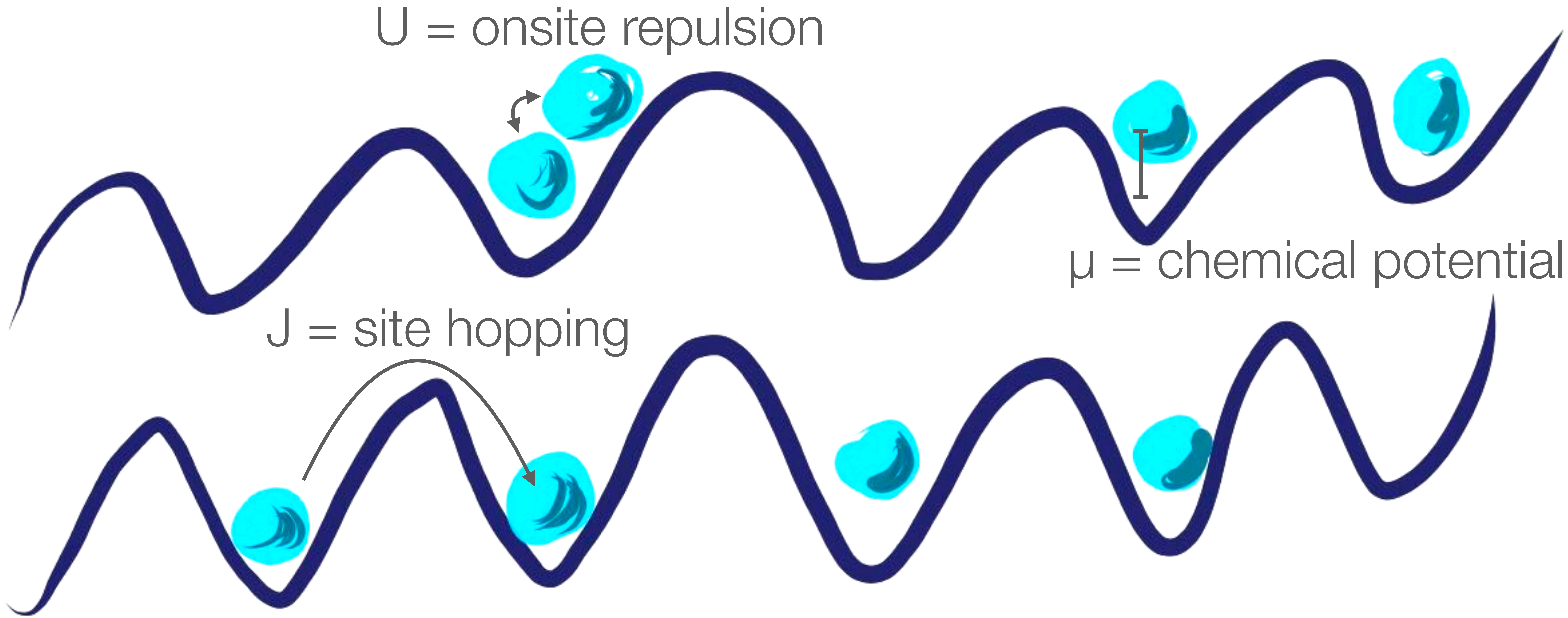
50 qubits -> Hilbert space size: 10^15

Chiaro et al, arXiv:1910.06
Superconducting qubits (G





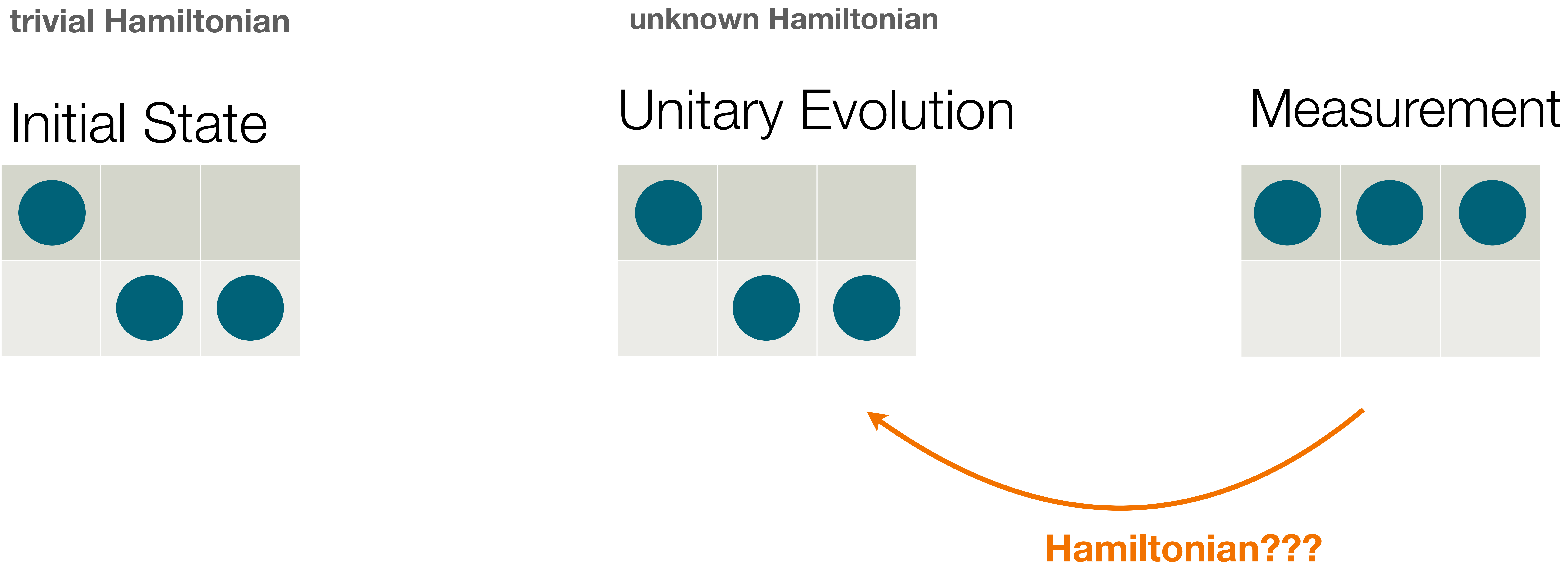
Quantum Simulation



How to learn the parameters governing the physics of quantum simulators as precisely as possible using experimentally accessible information?

Experimental sequence

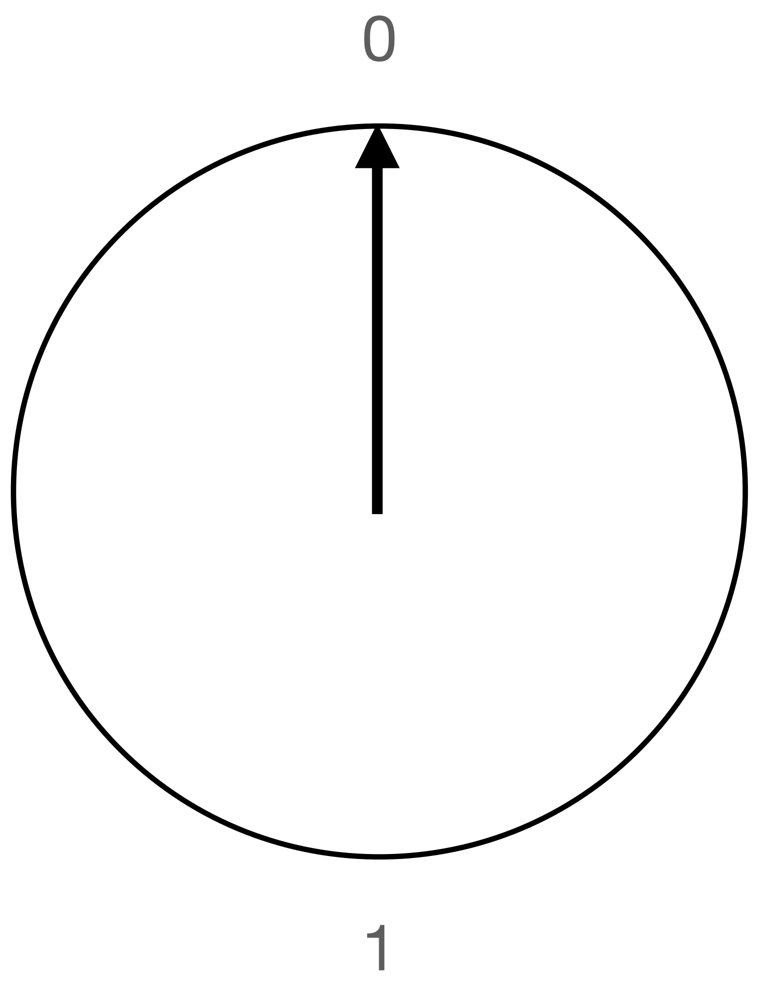
QMAI | TU Delft



Experimental sequence

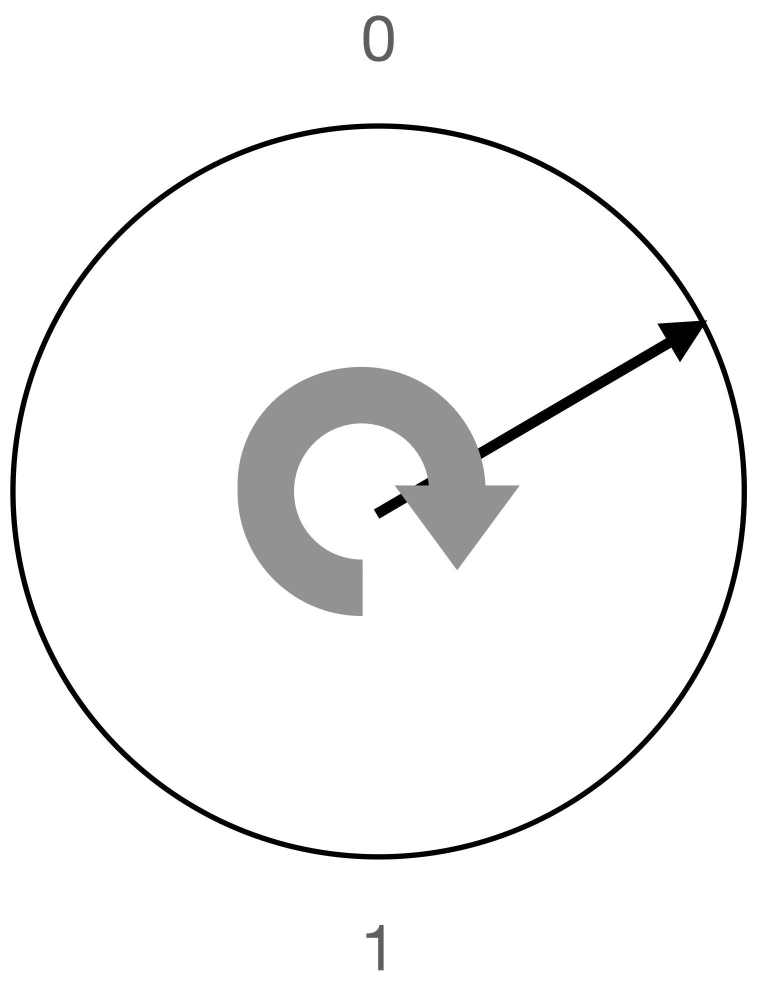
trivial Hamiltonian

Initial State



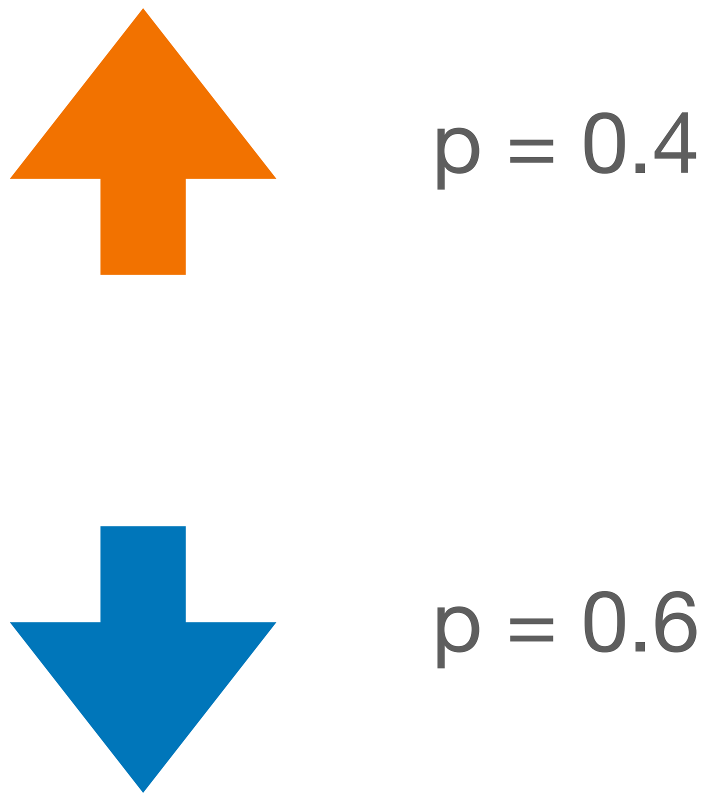
unknown Hamiltonian

Unitary Evolution



$$U = \exp(-i\omega\sigma_y t)$$

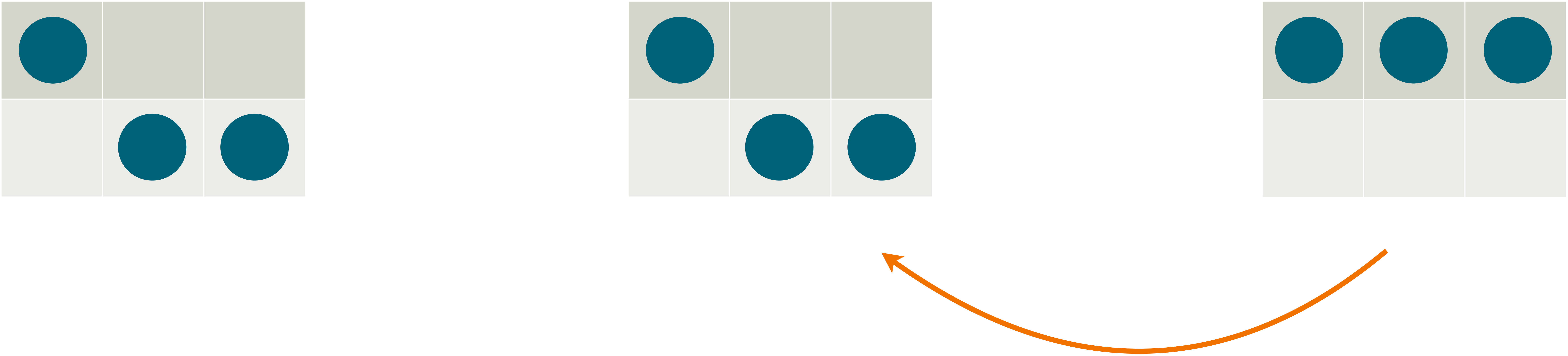
Measurement



Experimental sequence

trivial Hamiltonian | unknown Hamiltonian

Initial State | Unitary Evolution | Measurement



Hamiltonian???

QMAI | TU Delft

Experimental system

QMAI | TU Delft

$$H_{BH} = -J \sum_{\langle i,j \rangle} \hat{a}_i^\dagger \hat{a}_j + \frac{U}{2} \sum_i \hat{a}_i^\dagger \hat{a}_i (\hat{a}_i^\dagger \hat{a}_i - 1) - \mu \sum_i \hat{a}_i^\dagger \hat{a}_i.$$

2x50 sites

QMAI | TUDelft

$$H_{BH} = -\sum_{\langle i,j \rangle} J_{i,j} \hat{a}_i^\dagger \hat{a}_j + \sum_i \frac{U_i}{2} \hat{a}_i^\dagger \hat{a}_i (\hat{a}_i^\dagger \hat{a}_i - 1) - \sum_i \mu_i \hat{a}_i^\dagger \hat{a}_i$$

2x50 sites

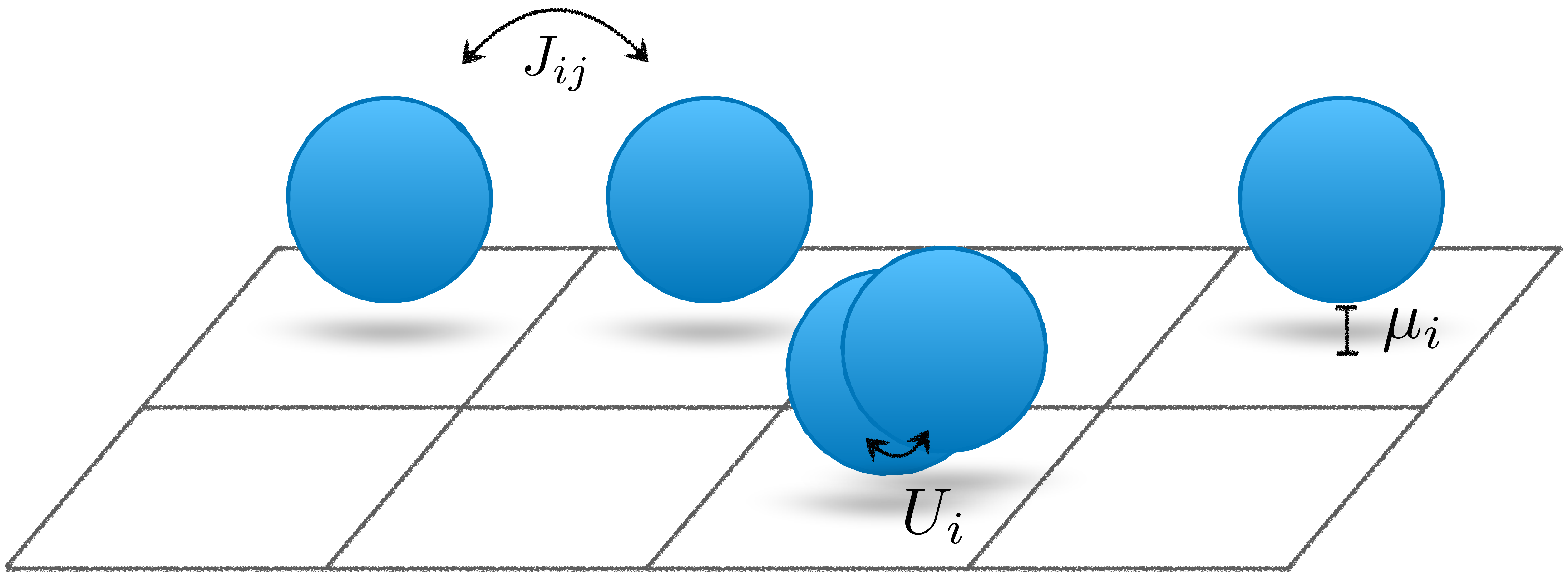
QMAI | TUDelft

$$H_{BH} = -\sum_{\langle i,j \rangle} J_{i,j} \hat{a}_i^\dagger \hat{a}_j + \sum_i \frac{U_i}{2} \hat{a}_i^\dagger \hat{a}_i (\hat{a}_i^\dagger \hat{a}_i - 1) - \sum_i \mu_i \hat{a}_i^\dagger \hat{a}_i$$

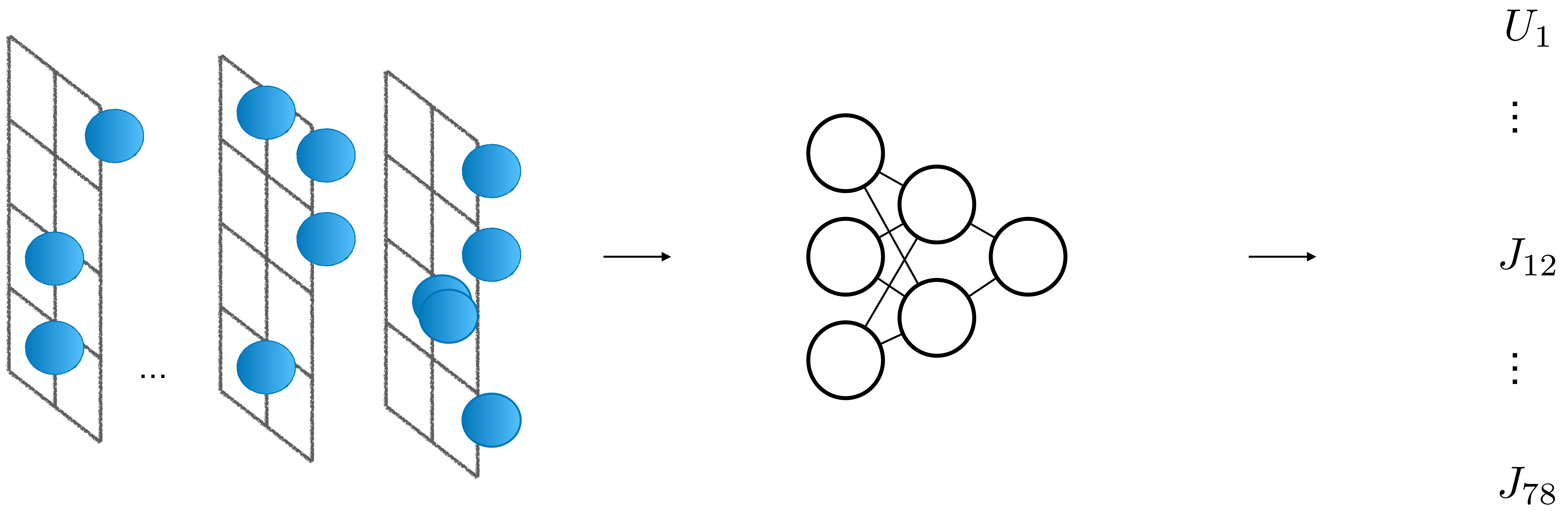
10 particles -> Hilbert space ~ 10^13

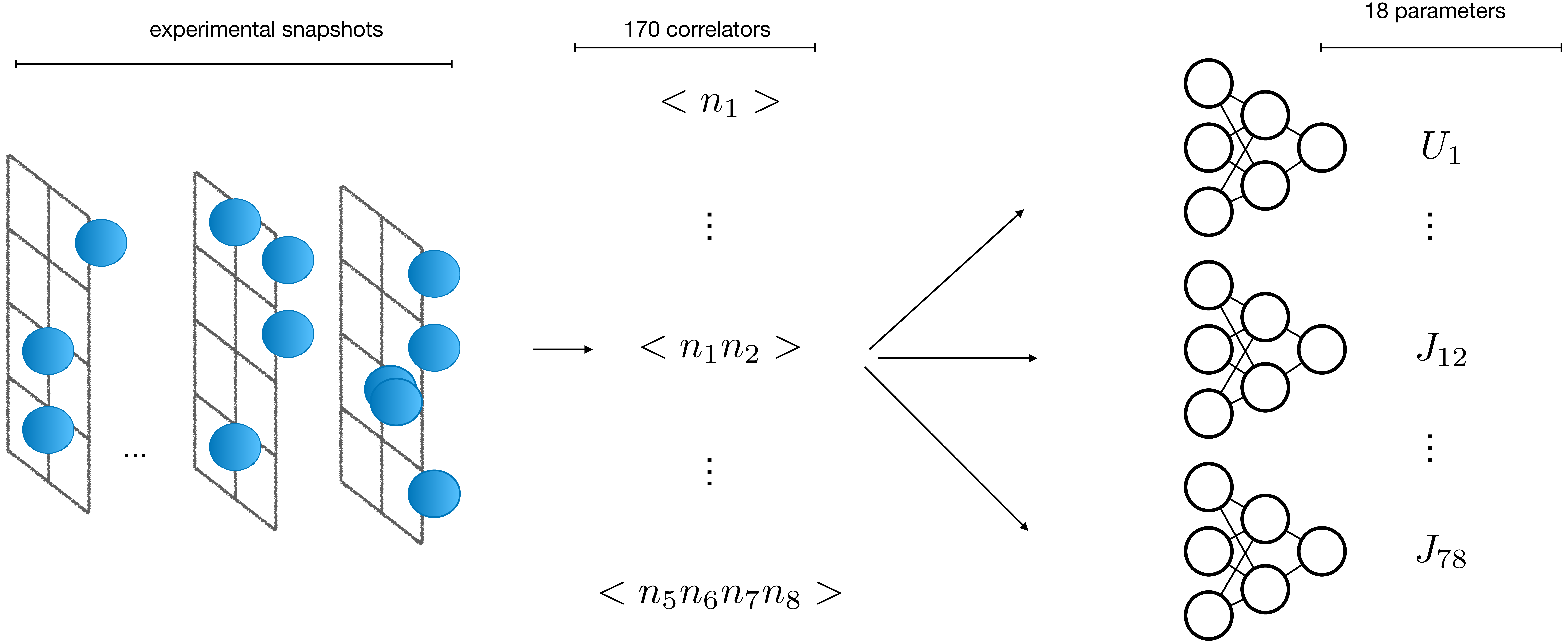
350 parameters to estimate

$$H_{BH} = -\sum_{\langle i,j \rangle} J_{i,j}\hat{a}_i^\dagger \hat{a}_j + \sum_i \frac{U_i}{2}\hat{a}_i^\dagger \hat{a}_i(\hat{a}_i^\dagger \hat{a}_i - 1) - \sum_i \mu_i \hat{a}_i^\dagger \hat{a}_i$$



4 particles -> Hilbert space dim = 330
25 parameters to estimate


...
U_1
⋮
J_{12}
⋮
J_{78}


experimental snapshots
...
170 correlators
$< n_1 >$
$\vdots$
$< n_1 n_2 >$
$\vdots$
$< n_5 n_6 n_7 n_8 >$
18 parameters
U_1
$\vdots$
J_{12}
$\vdots$
J_{78}

QMAI | TU Delft

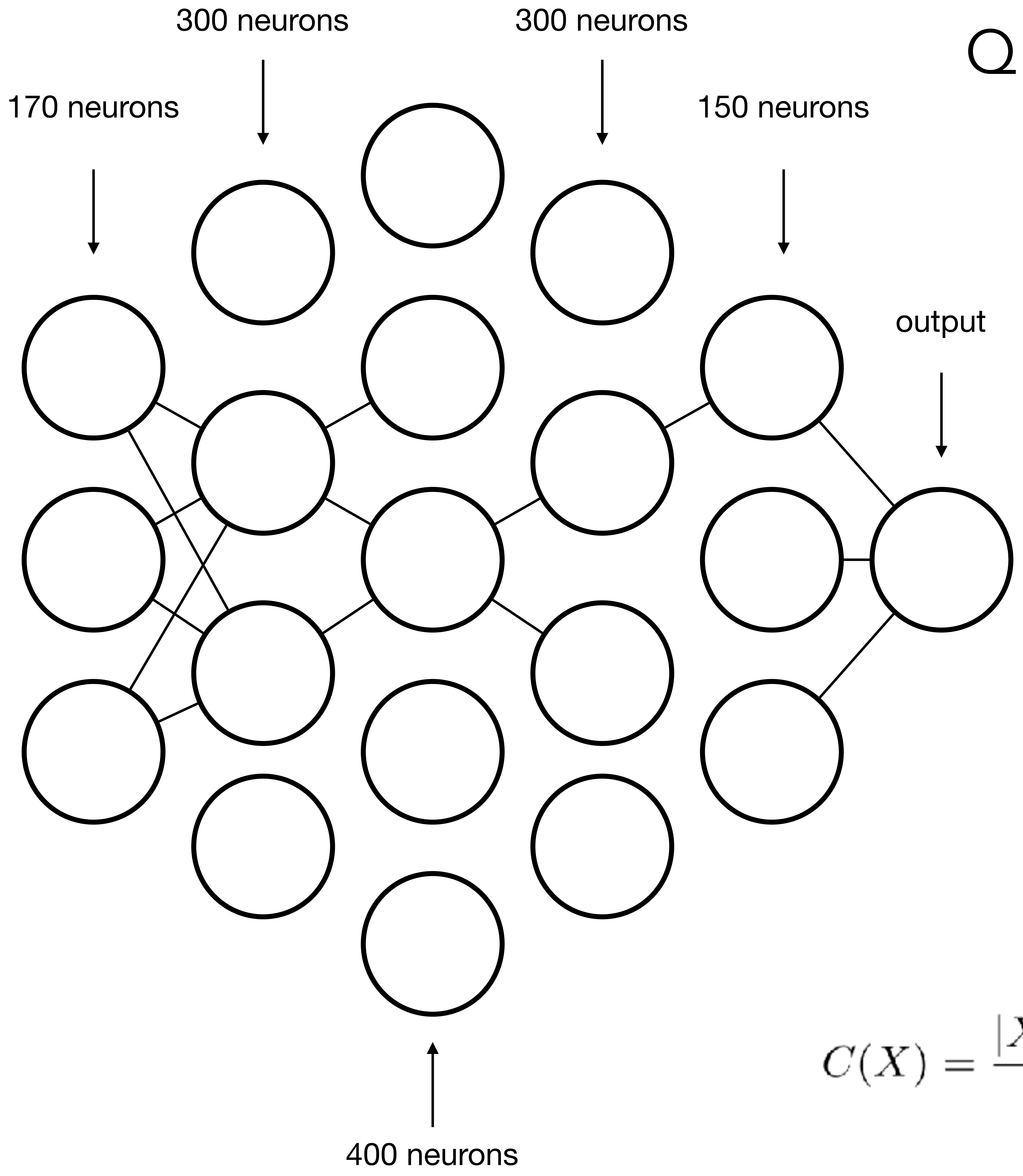
$< n_1 >$

$\vdots$

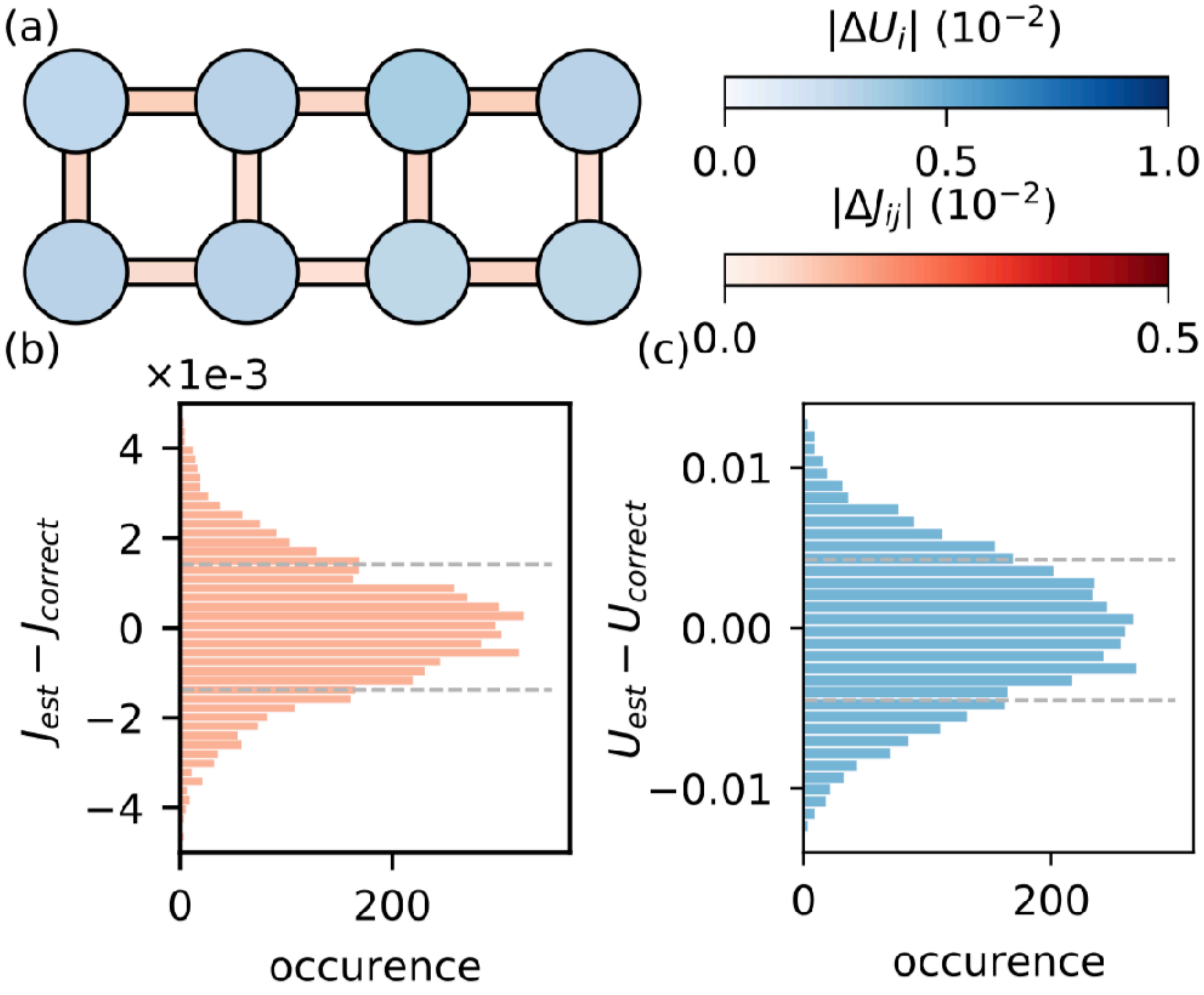
$< n_1 n_2 >$

$\vdots$

$< n_5 n_6 n_7 n_8 >$



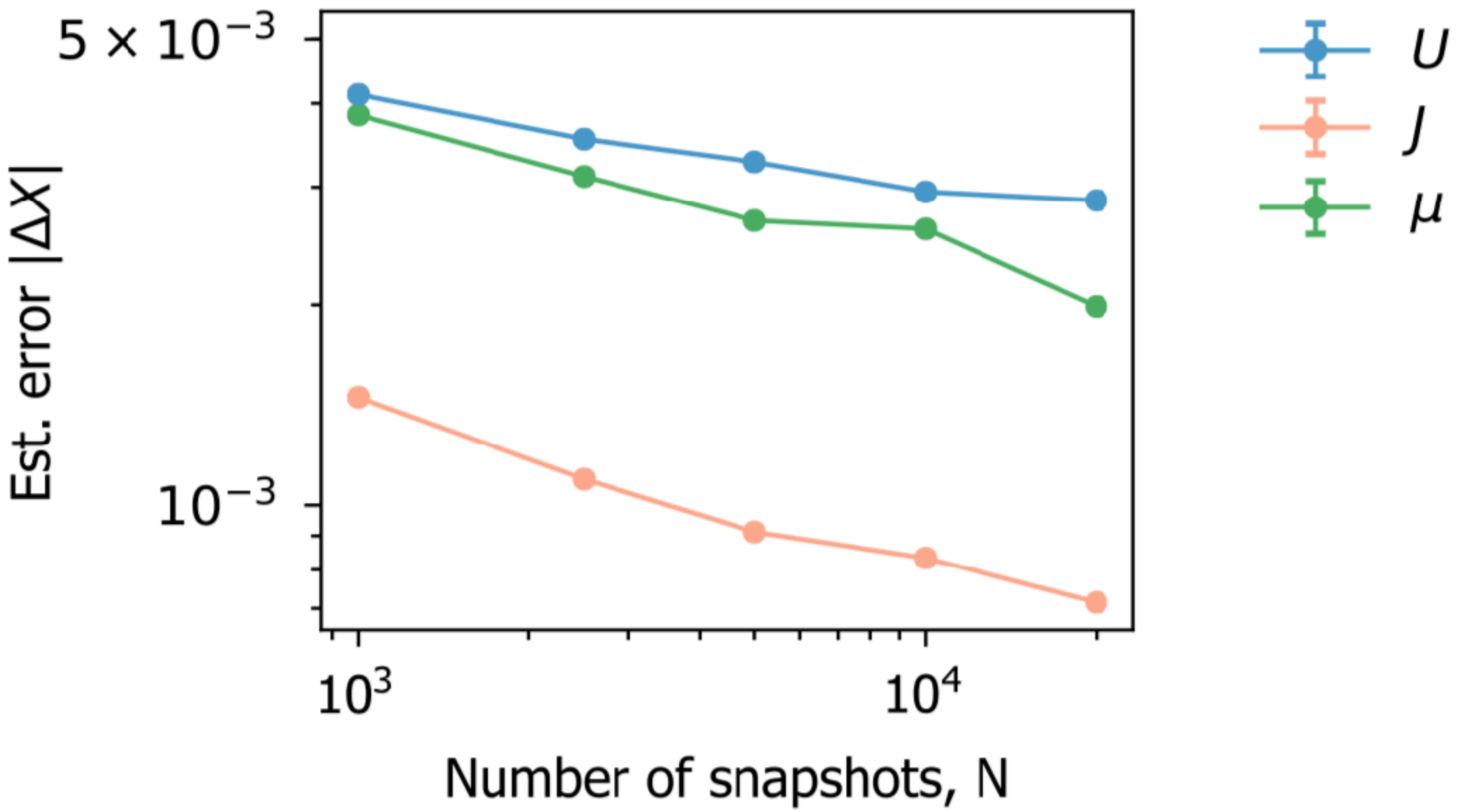
$$C(X) = \frac{|X^{\text{pred}} - X^{\text{label}}|^2}{N_{\text{Batch}}}.$$



Results for 2500 experimental snapshots

0.1% average estimation error

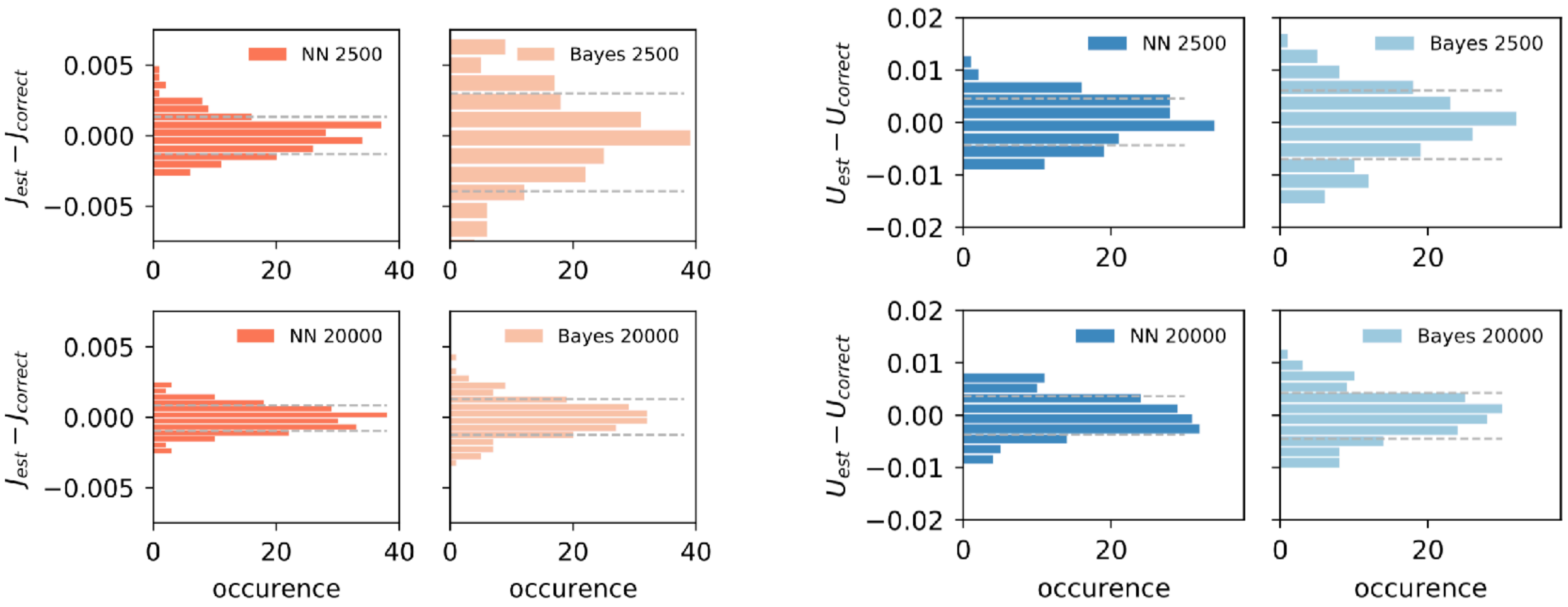
Error scaling wt number of measurements



<u>For 2500 snapshots:</u>
ΔJ = 0.001
ΔU = 0.0035
Δμ = 0.003

<u>For 20 000 snapshots:</u>
ΔJ = 0.0007
ΔU = 0.003
Δμ = 0.0025

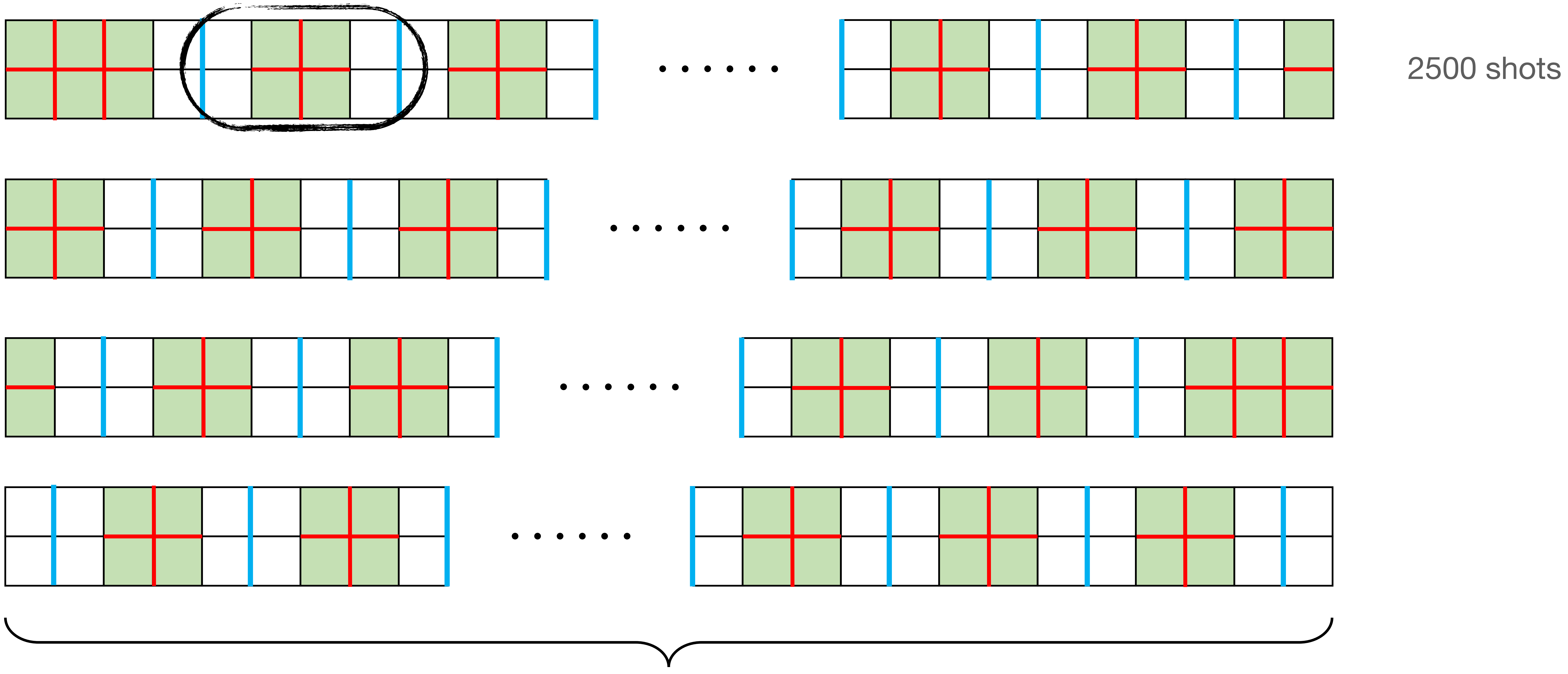
Bayesian Benchmark



Scaling to larger system sizes

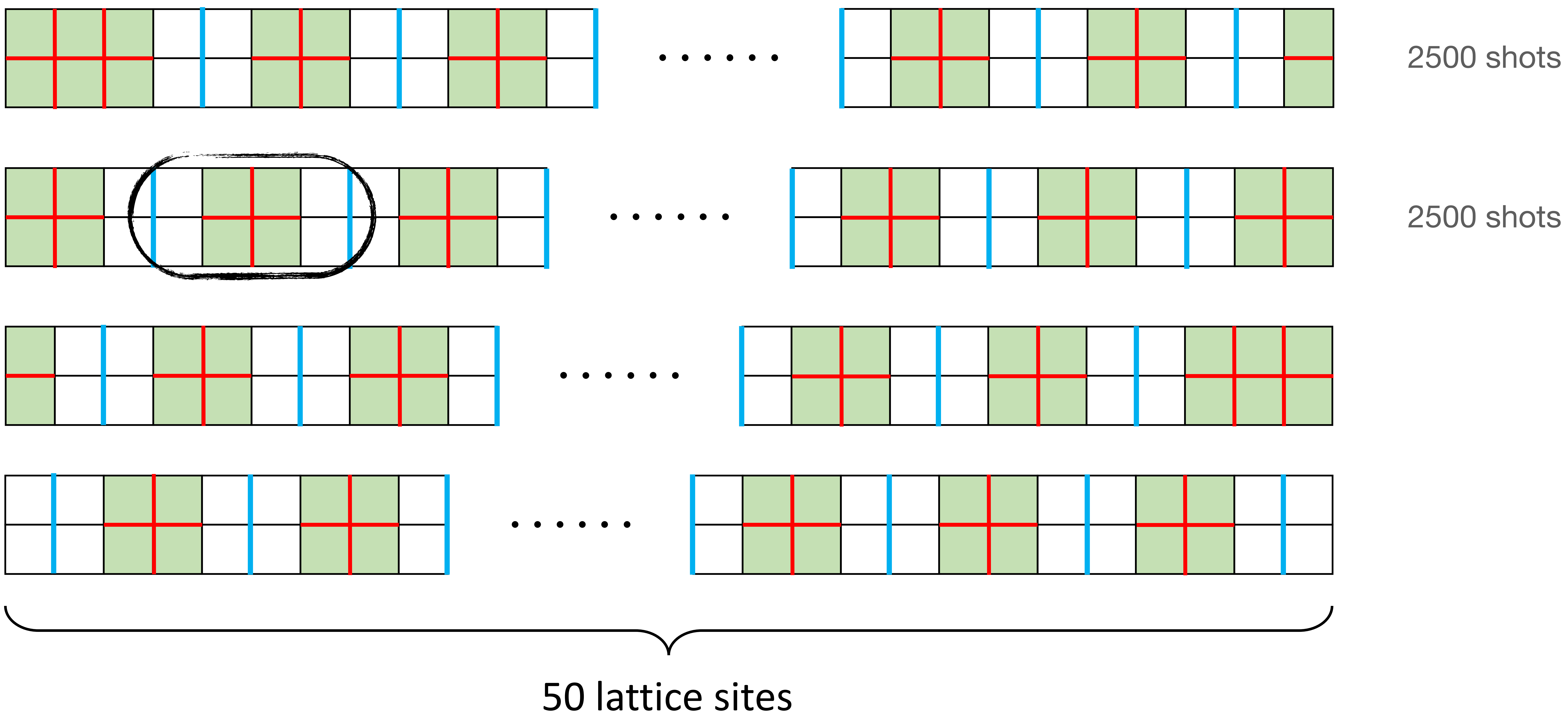
Scaling

Effective parameters: U_i J_{ij}



Scaling

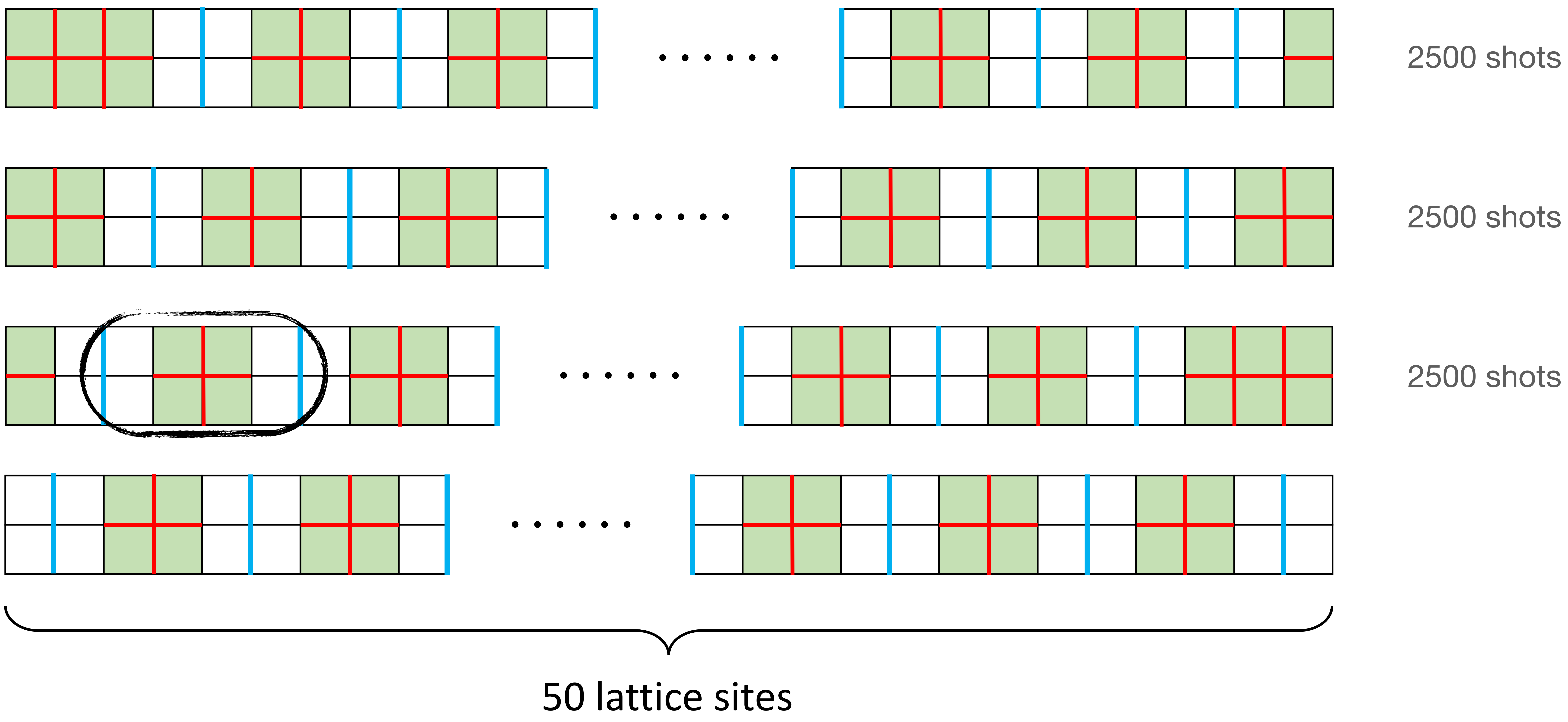
Effective parameters: U_i J_{ij}



Scaling

QMAI | TU Delft

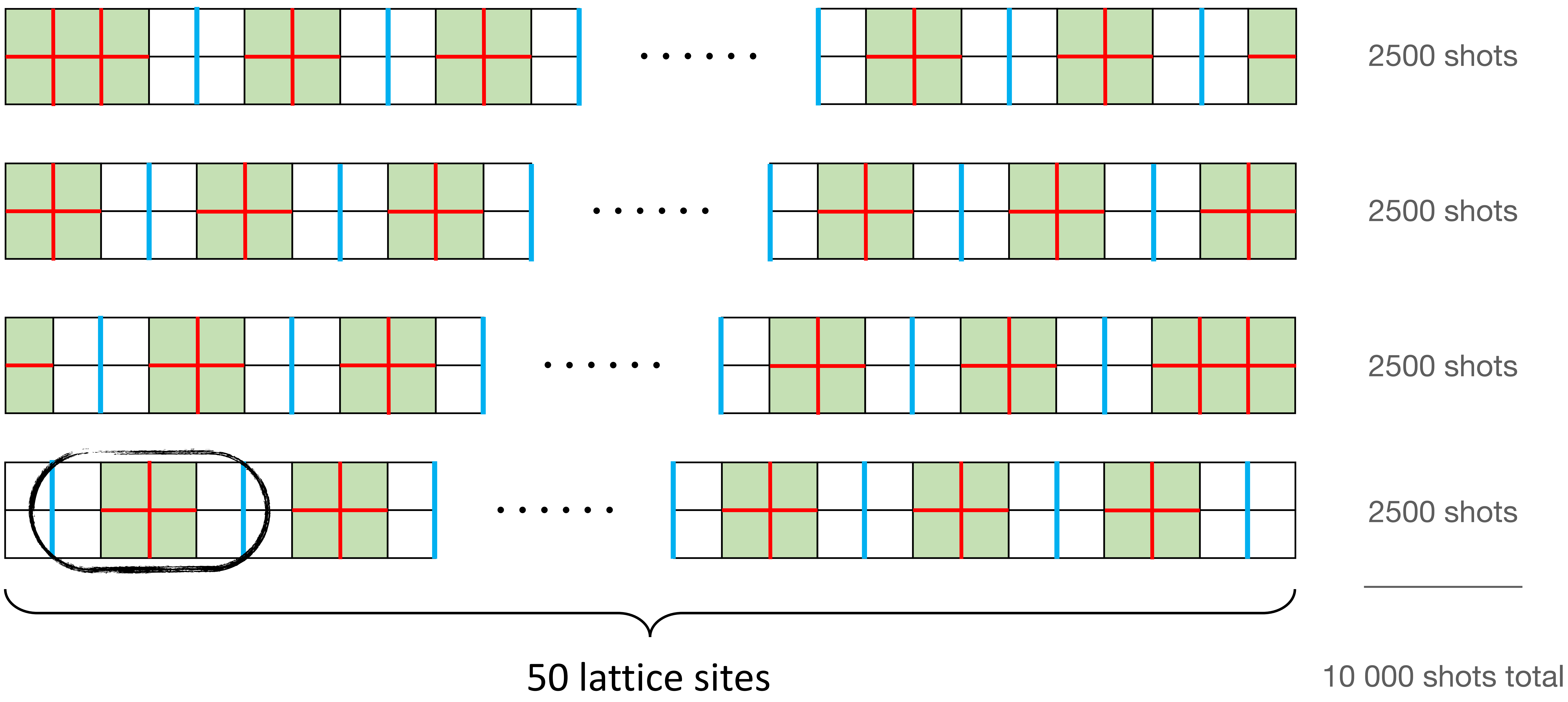
Effective parameters: U_i J_{ij}



Scaling

QMAI | TU Delft

Effective parameters: U_i J_{ij}



QMAI | TU Delft

$$H_{BH} = -\sum_{\langle i,j \rangle} J_{i,j} \hat{a}_i^\dagger \hat{a}_j + \sum_i \frac{U_i}{2} \hat{a}_i^\dagger \hat{a}_i (\hat{a}_i^\dagger \hat{a}_i - 1) - \sum_i \mu_i \hat{a}_i^\dagger \hat{a}_i$$

- practical scalable Hamiltonian learning customised to relevant experiment
- out-of-equilibrium Hamiltonian learnable with ~0.1% - 0.3% precision from 10 000 experimental snapshots

Applying ML to quantum experiments

Recap QUIZ

What are two-dimensional materials?

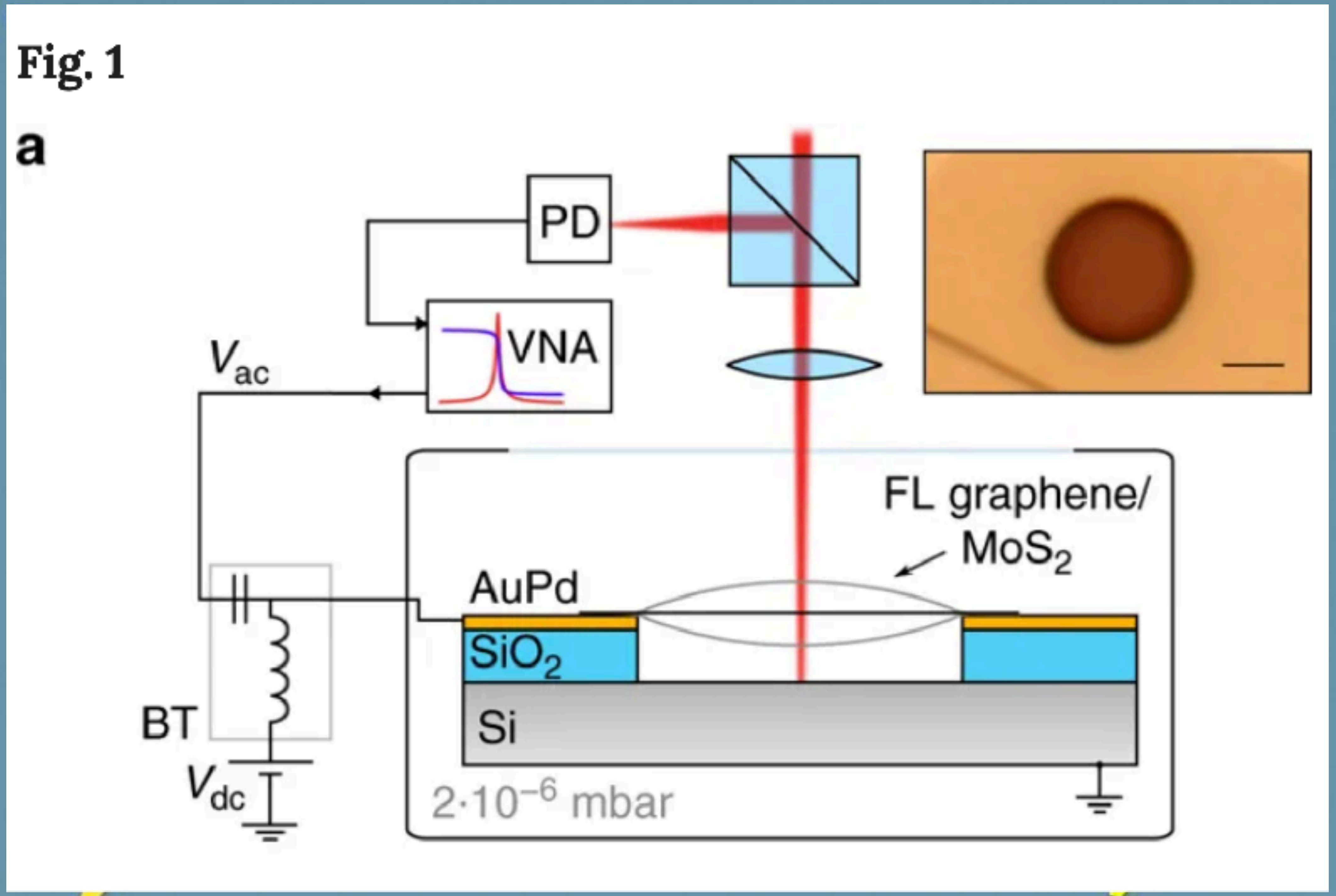
EG, Carolin Gold, Benedikt Kratochwil, Tim Davatz, Riccardo Pisoni, Annika Kurzmann, Peter Rickhaus, Mark H. Fischer, Thomas Ihn, Sebastian Huber arXiv: 1910.00066 (2019)

QMAI | TU Delft

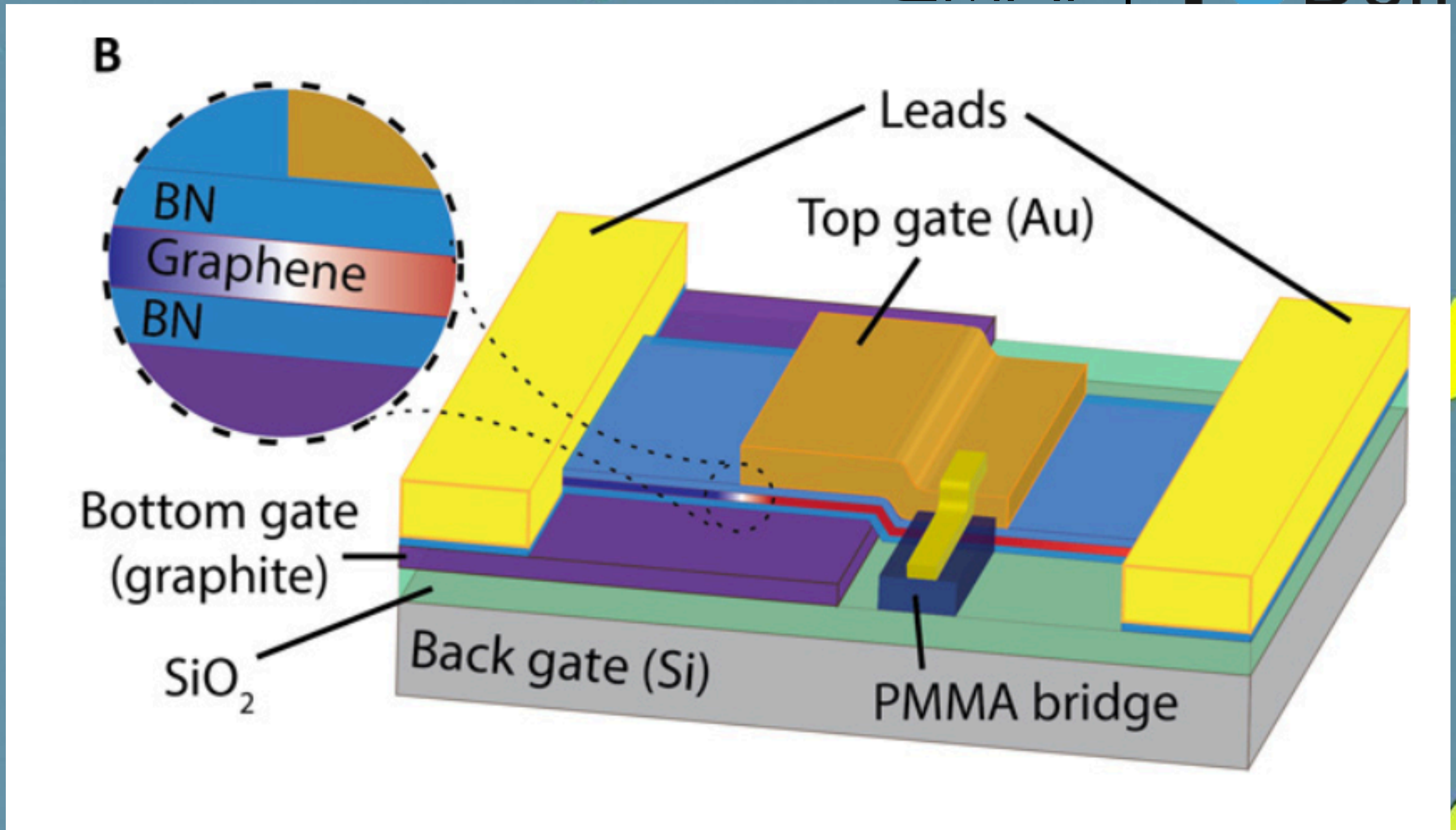
Thin nanomaterials are key constituents of many modern quantum devices.

EG, Carolin Gold, Benedikt Kratochwil, Tim Davatz, Riccardo Pisoni, Annika Kurzmann, Peter Rickhaus, Mark H. Fischer, Thomas Ihn, Sebastian Huber arXiv: 1910.00066 (2019)

QMAI | TU Delft

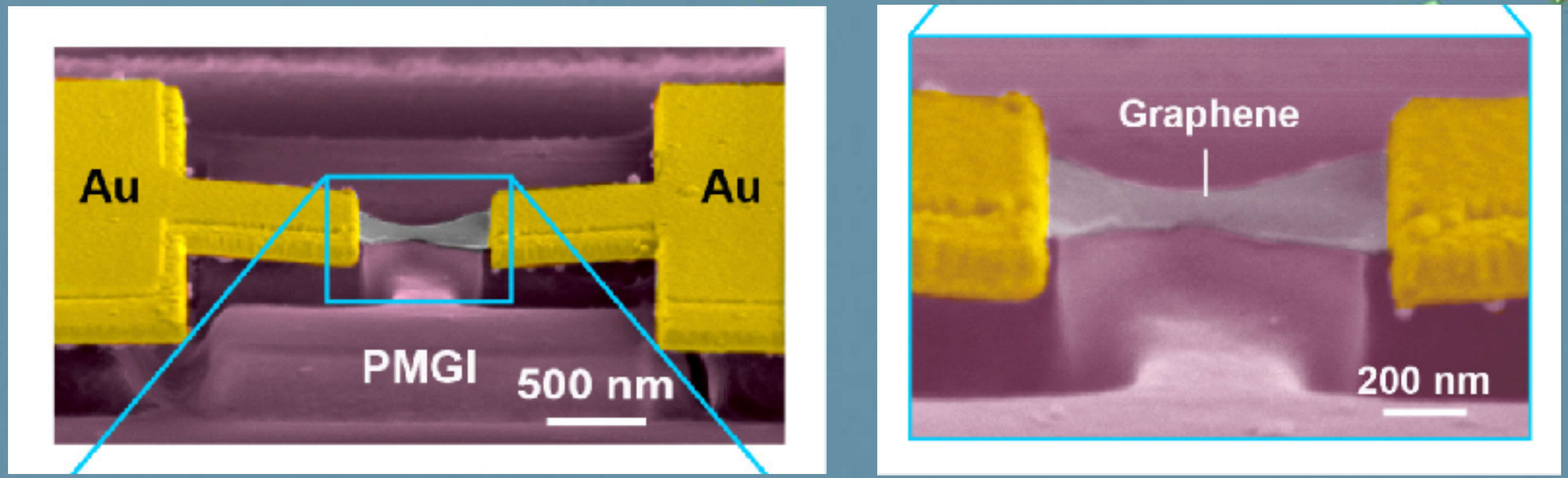


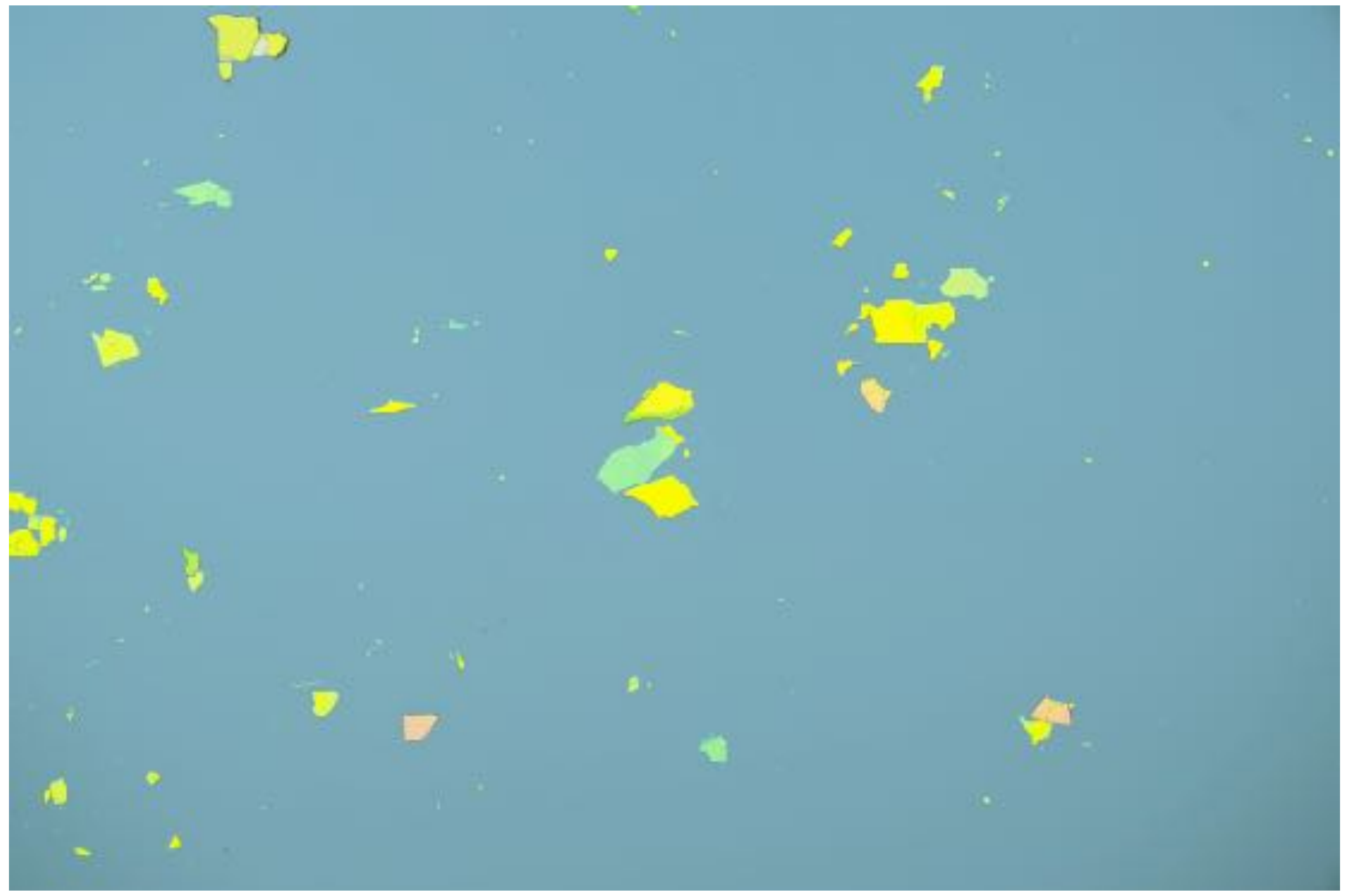
D. Davidovikj et.al., Nature Communications 8, 1253 (2017)

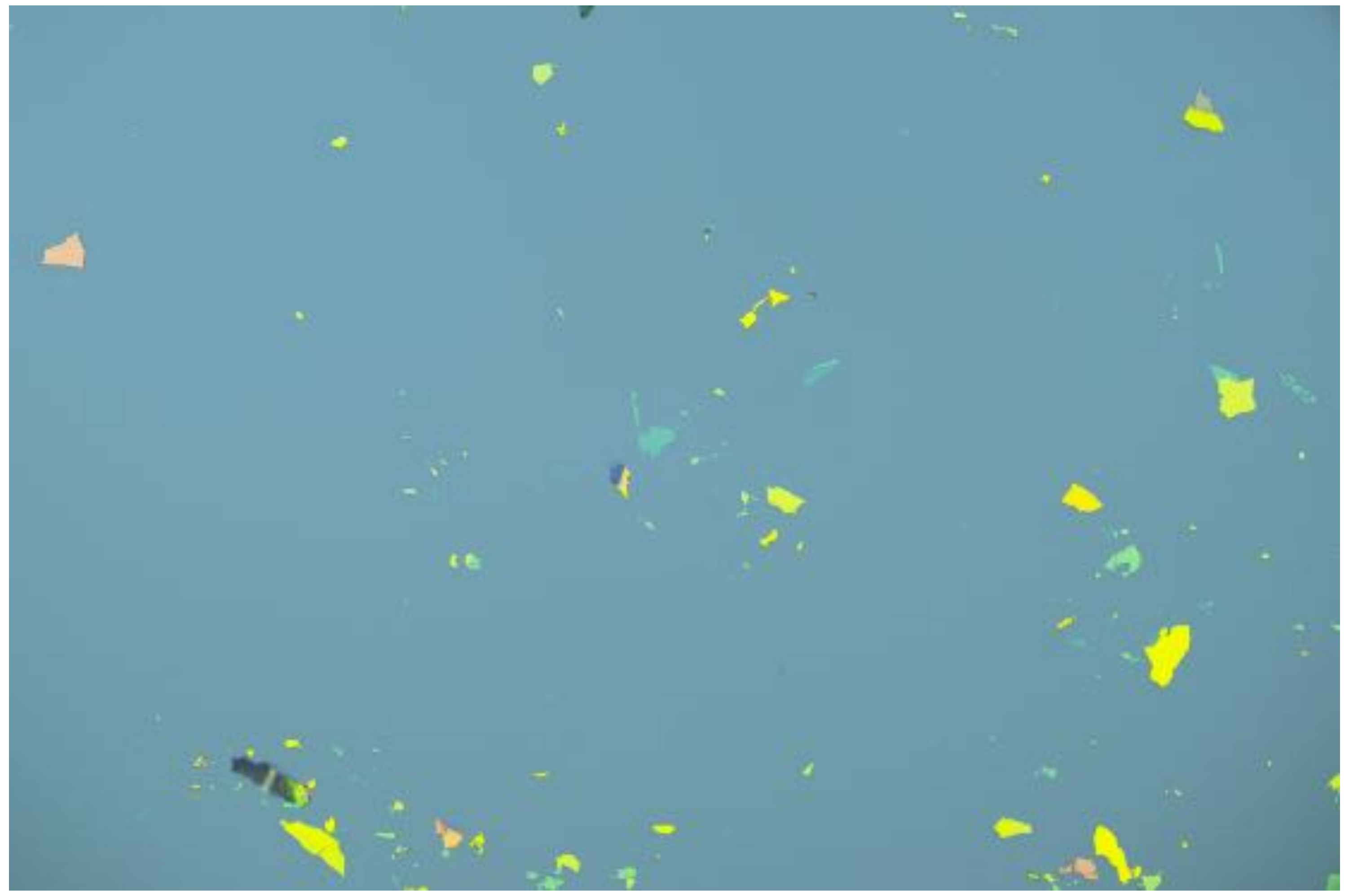


D.S. Wei et.al., Science Advances, Vol. 3, no. 8, e1700600 (2017)

S. Canava et.al., Nature Nanotechnology 13, 1126–1131 (2018)







EG, Carolin Gold, Benedikt Kratochwil, Tim Davatz, Riccardo Pisoni, Annika Kurzmann, Peter Rickhaus, Mark H. Fischer, Thomas Ihn, Sebastian Huber arXiv: 1910.00066 (2019)

QMAI | TU Delft

This task is hard to automise due to diversity of the data.

EG, Carolin Gold, Benedikt Kratochwil, Tim Davatz, Riccardo Pisoni, Annika Kurzmann, Peter Rickhaus, Mark H. Fischer, Thomas Ihn, Sebastian Huber arXiv: 1910.00066 (2019)


QMAI | TU Delft


QMAI | TUDelft

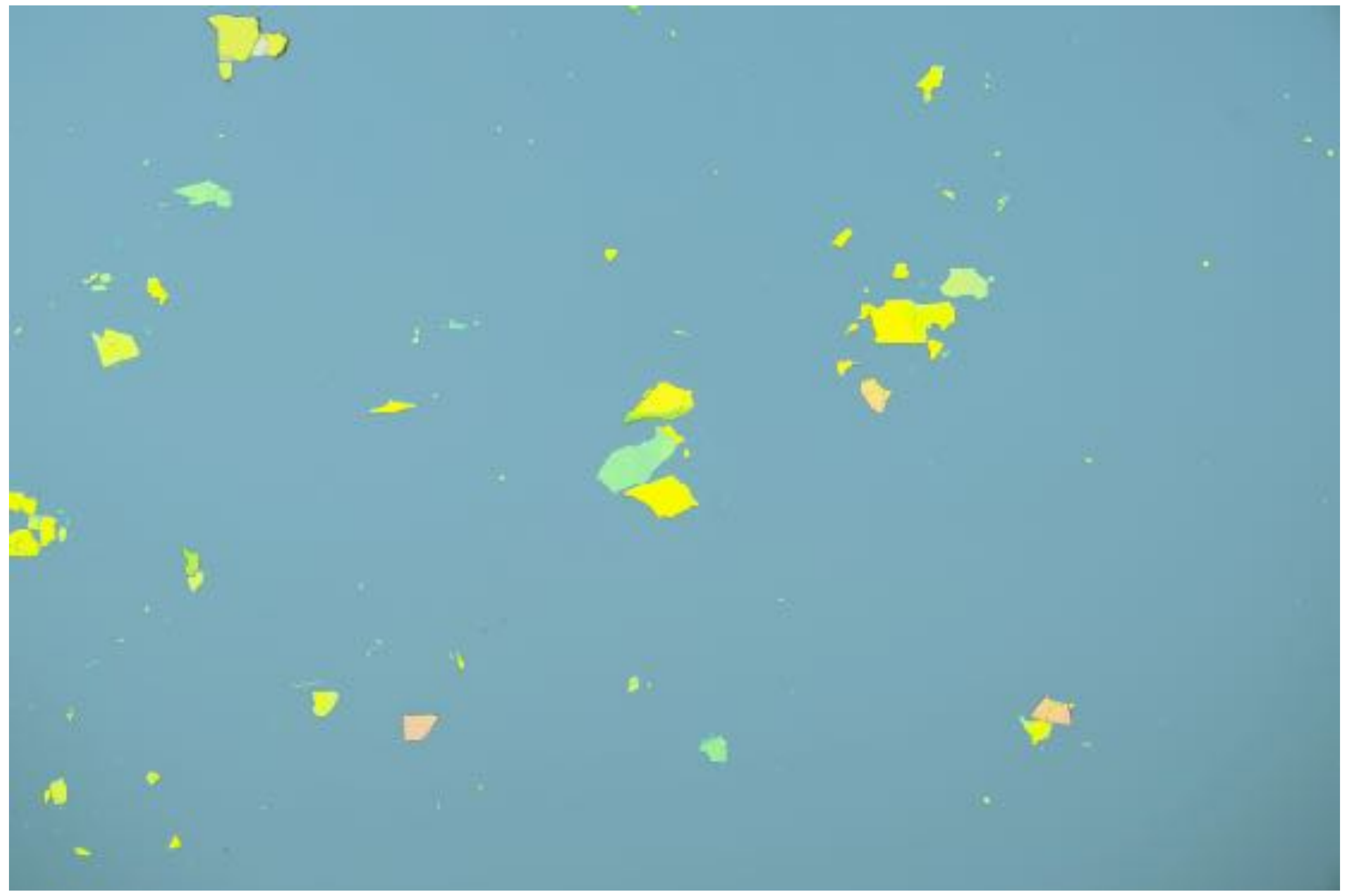

QMAI | TU Delft

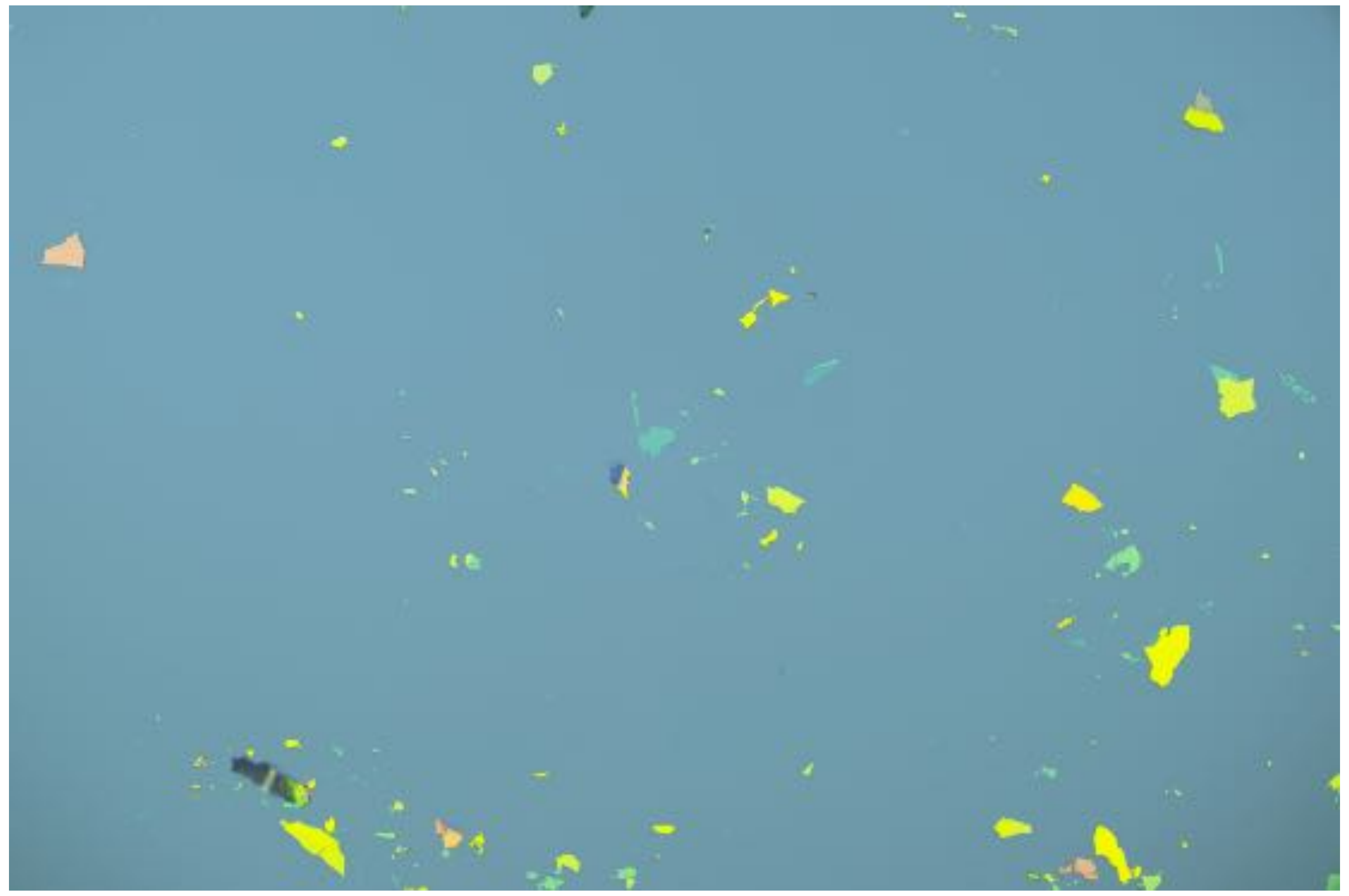
A person would move the camera, took pictures, and look through these manually saving coordinates of the good flake candidates.

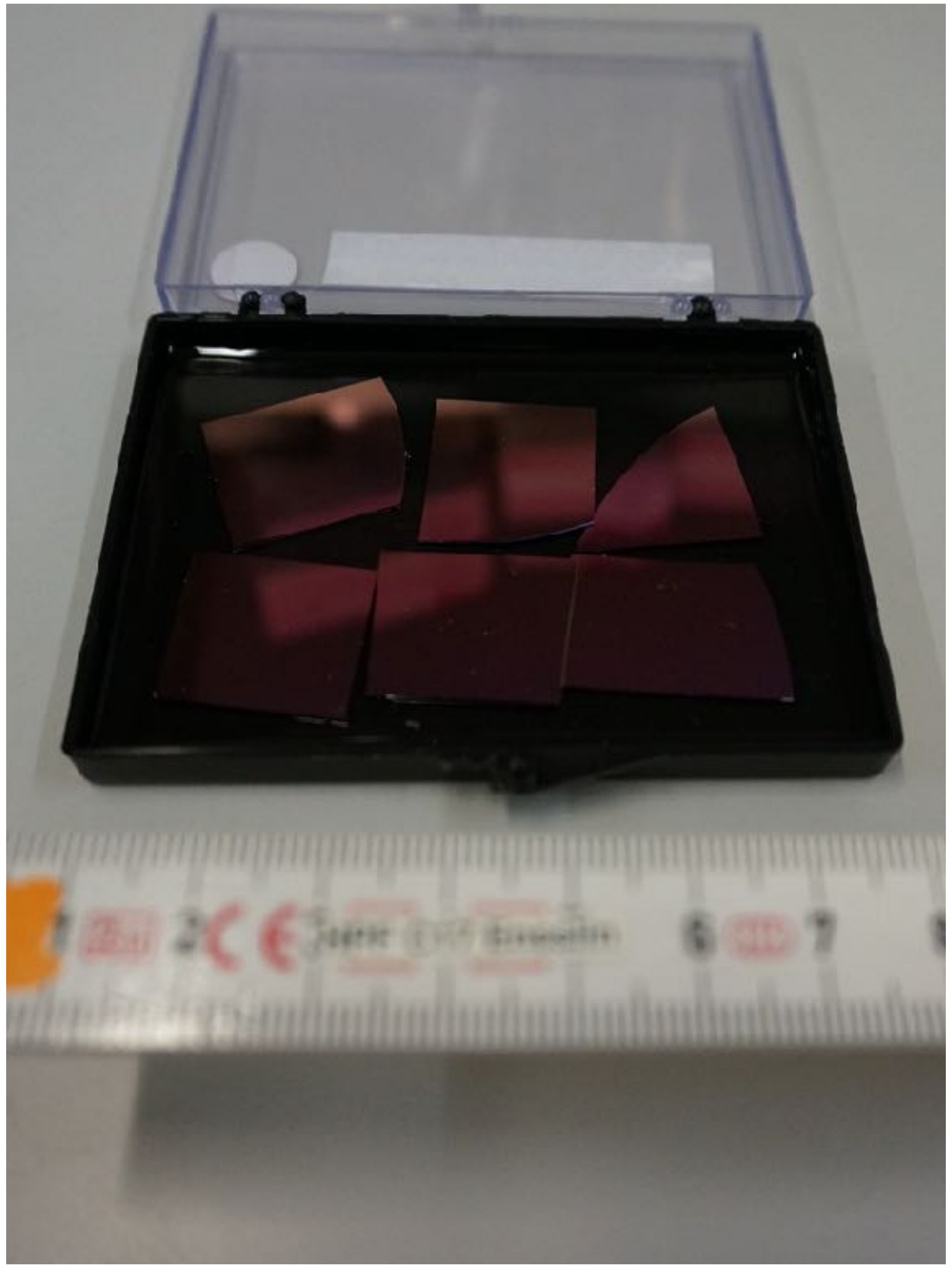
We wish to automatise this process.

QMAI | TU Delft

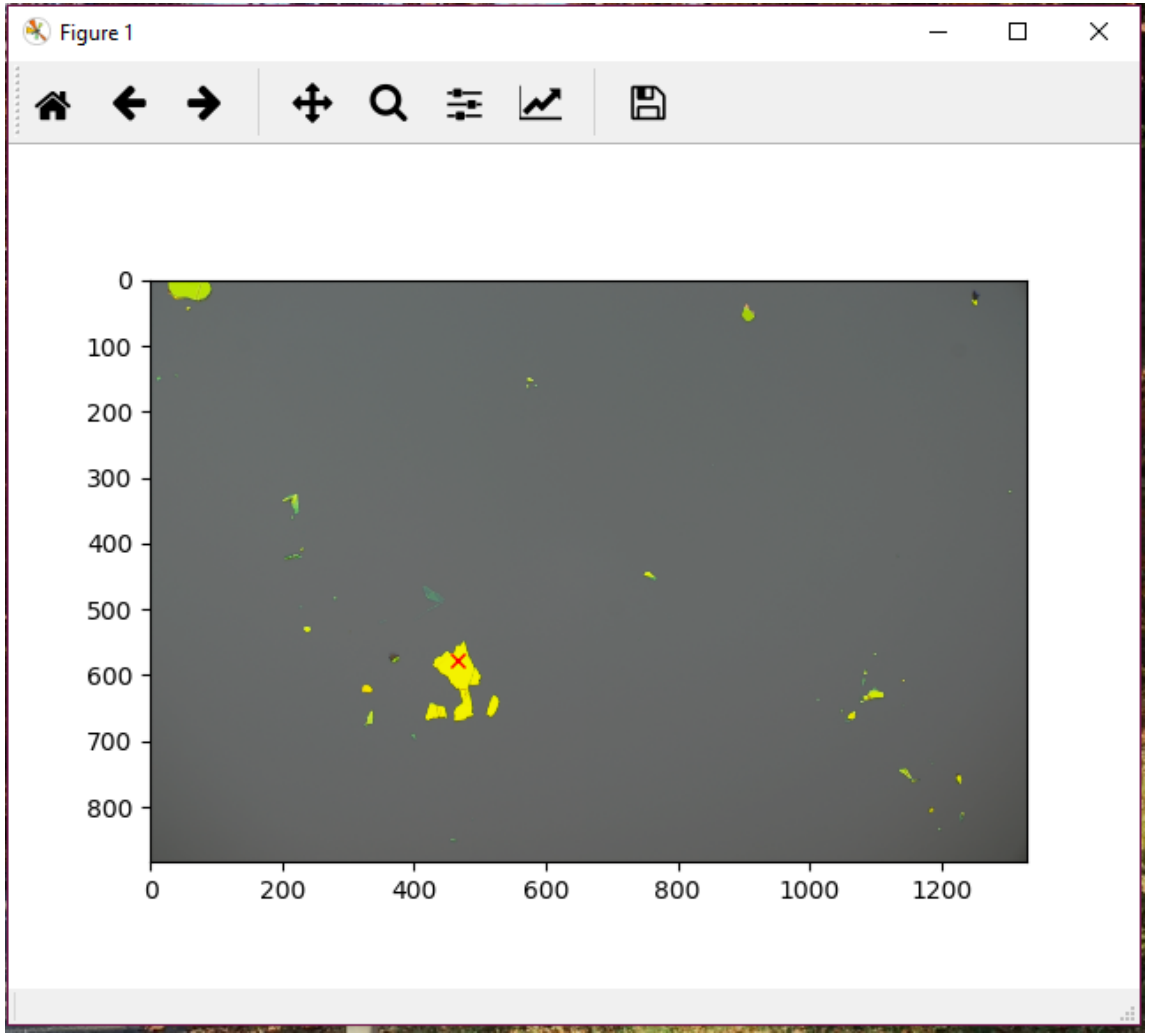
https://github.com/cmt-qo/cm-flakes

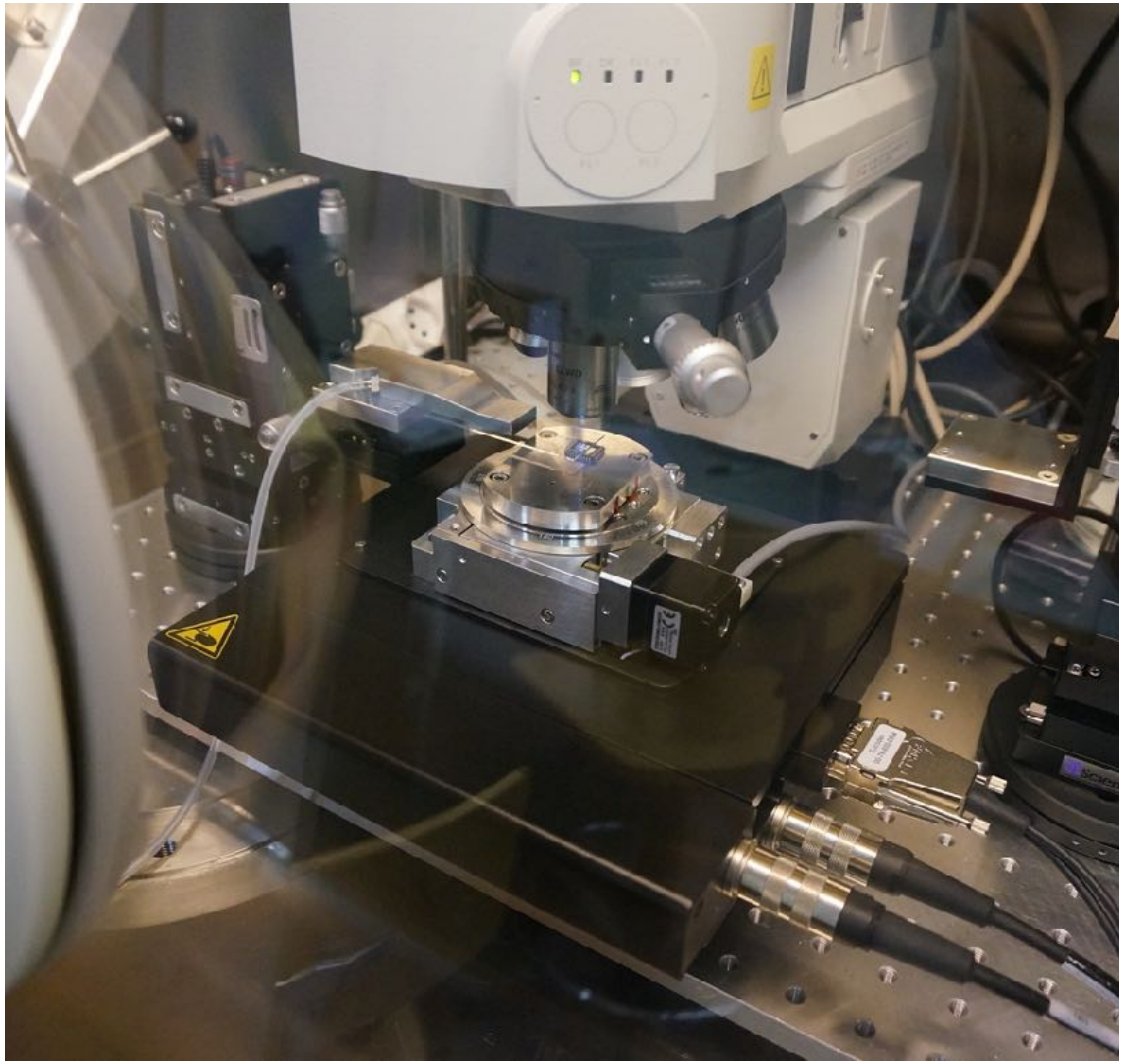


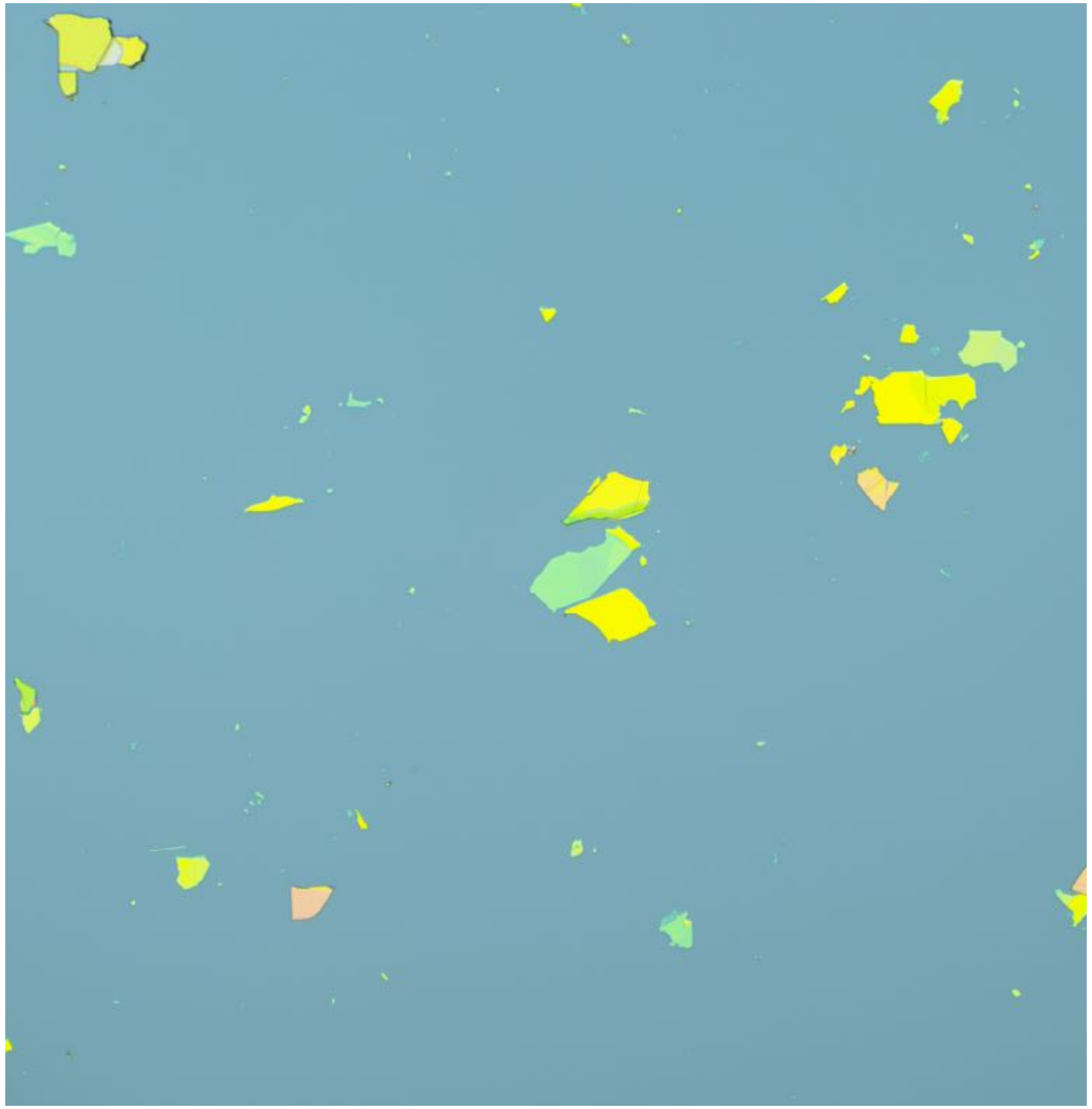




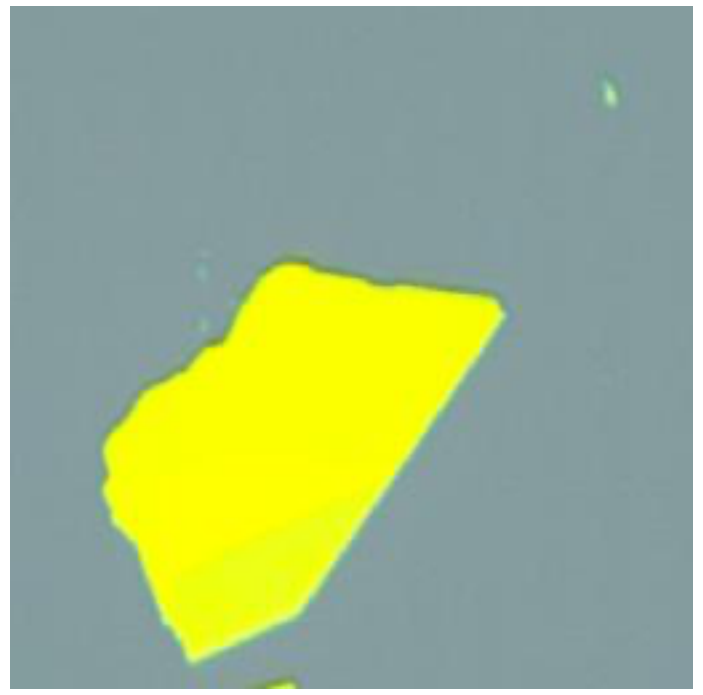
QMAI | TUDelft







GOOD



$$\begin{bmatrix}1\\0\end{bmatrix}$$

PIL LIBRARY - standard deviation criterion

BAD

$$\begin{bmatrix}0\\1\end{bmatrix}$$

* switch to notebook here *

QMAI | TU Delft

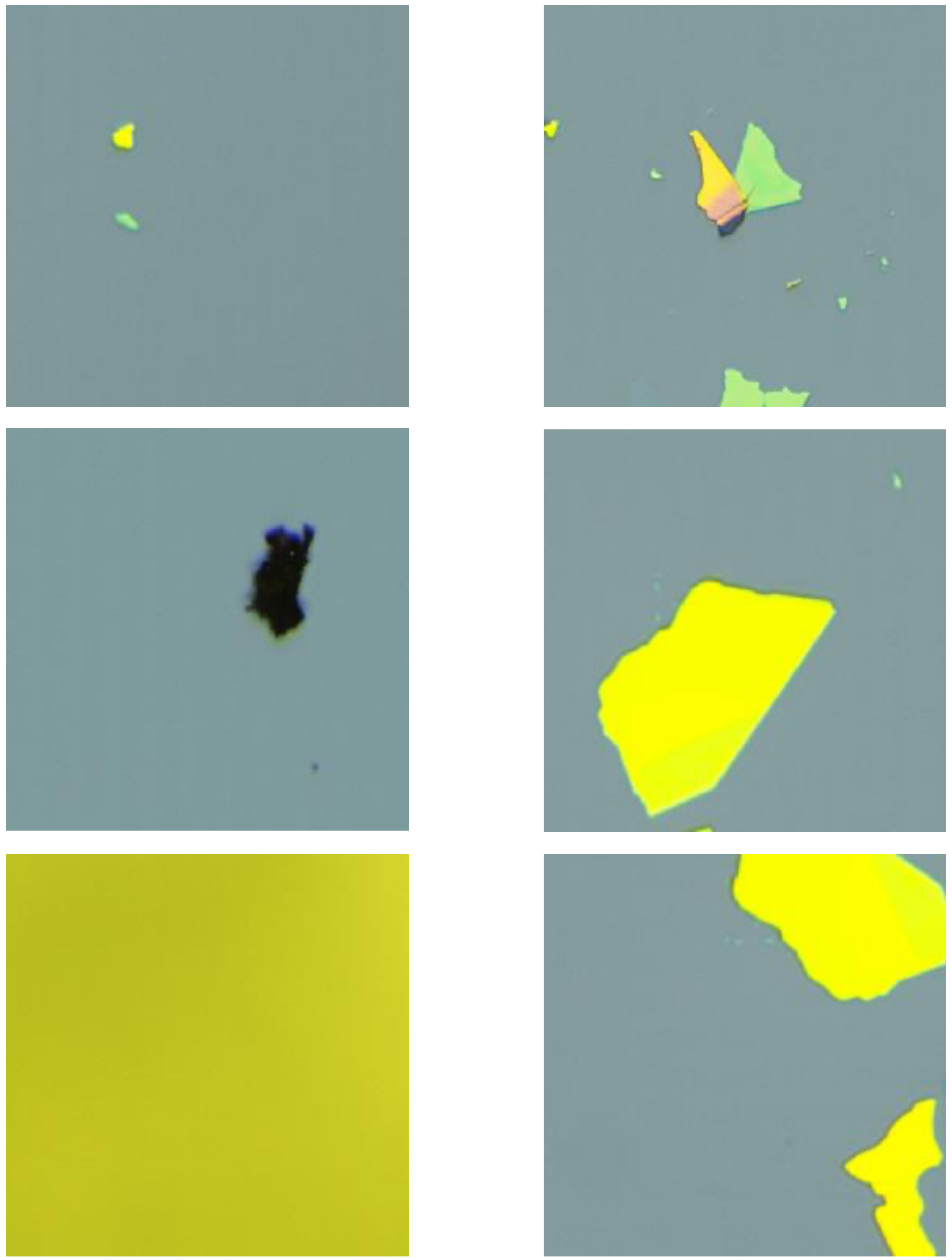
Flakes in total: ~ 100 000

GOOD flakes: 2000

turn into 6x2000 = 12 000 by rotations/mirroring using PIL library

‘sub-optimal distribution can be overcome by strategic preparation of the training batches’

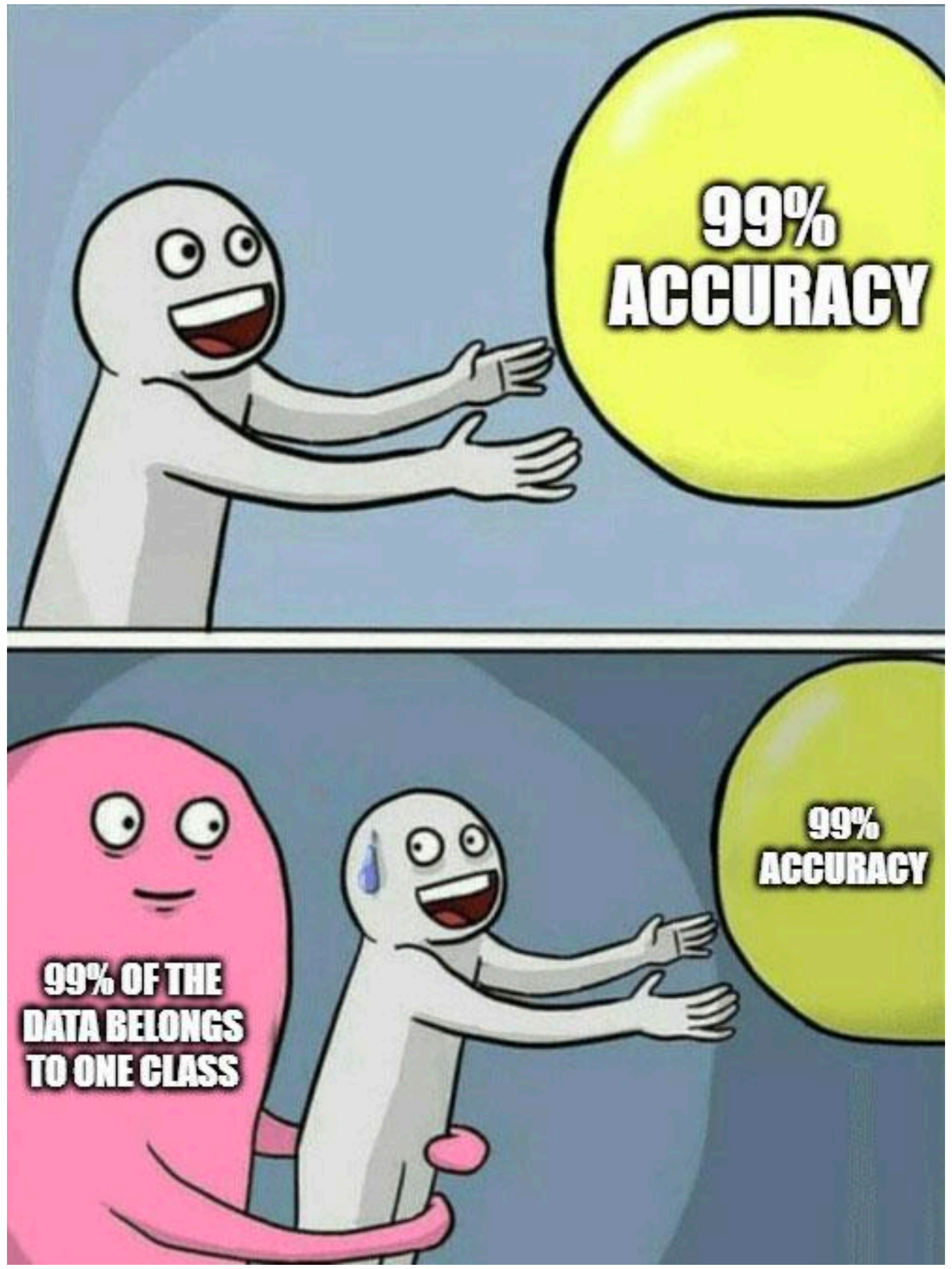
BAD **GOOD**



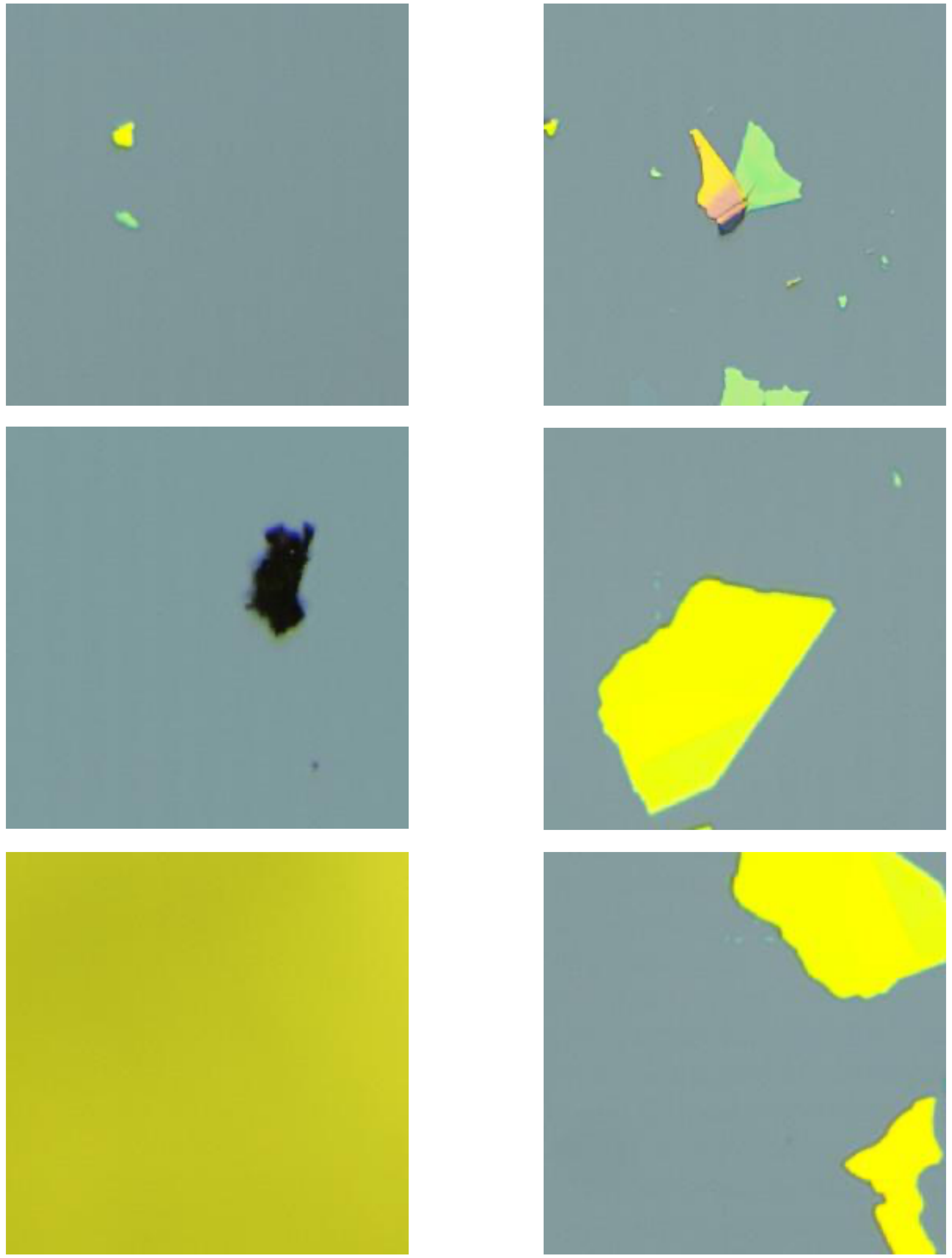
QMAI | TU Delft

BAD flakes: 90 000

GOOD flakes: 10 000

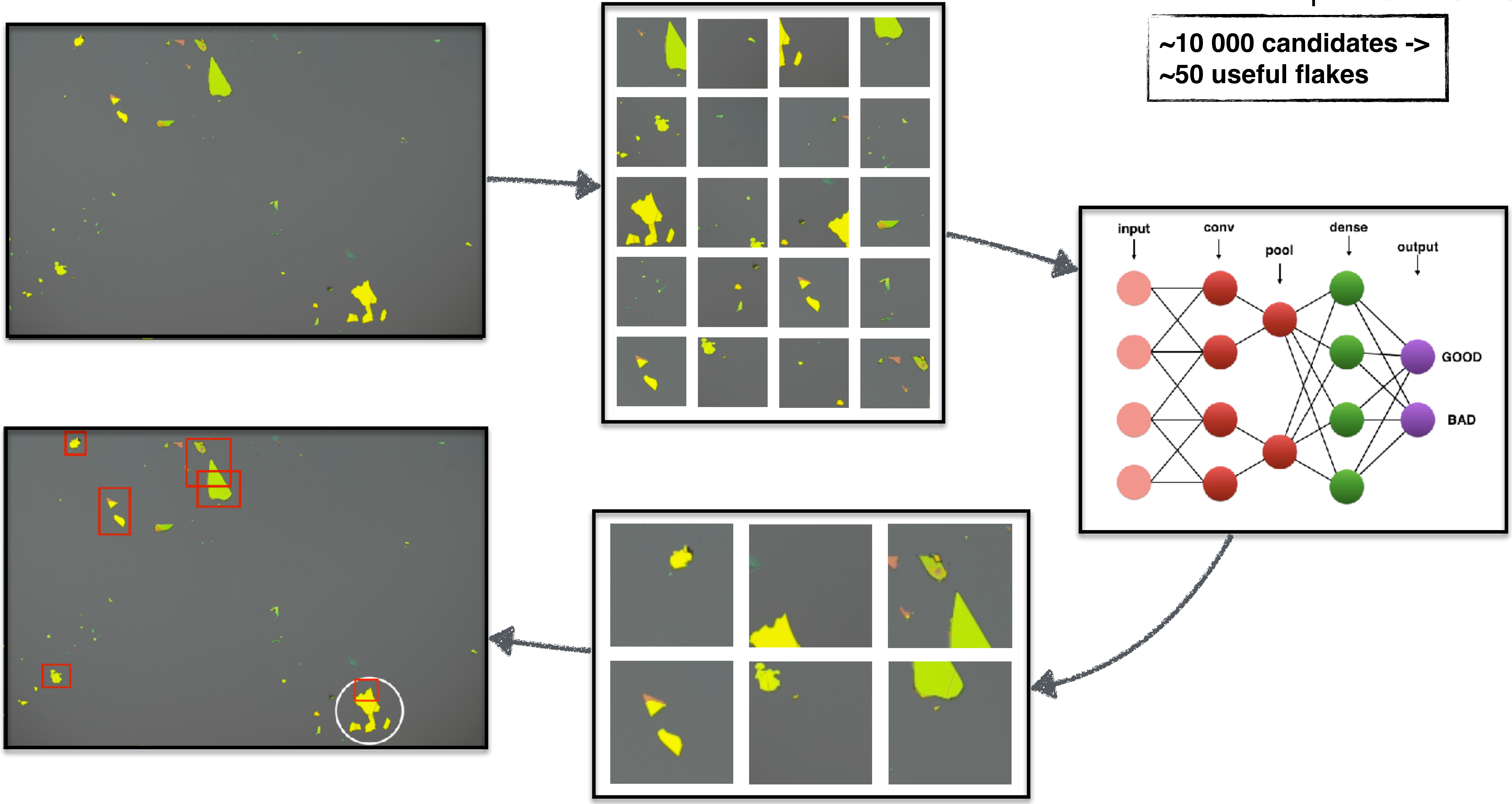


BAD **GOOD**



QMAI | TU Delft

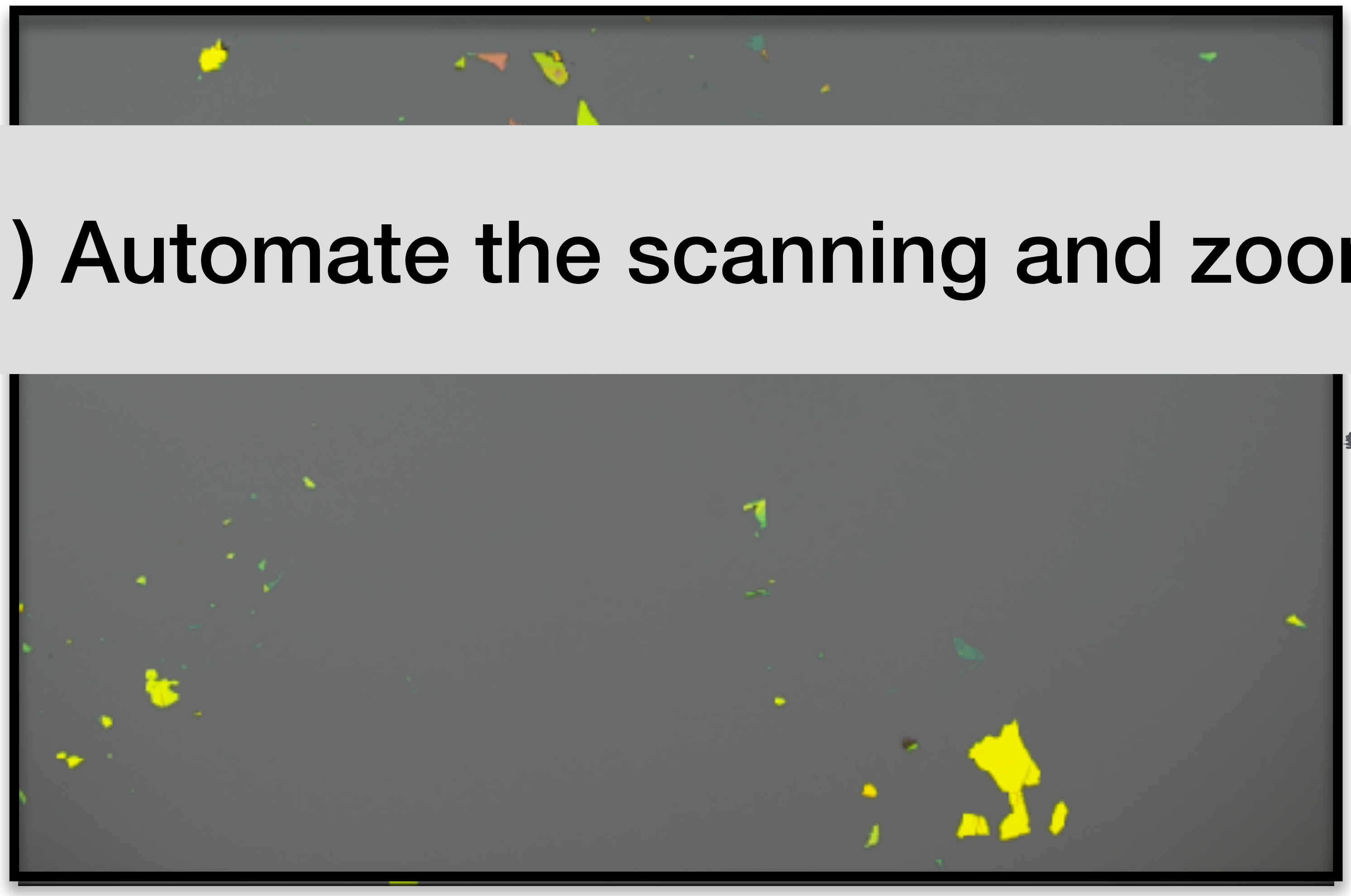
~10 000 candidates ->
~50 useful flakes

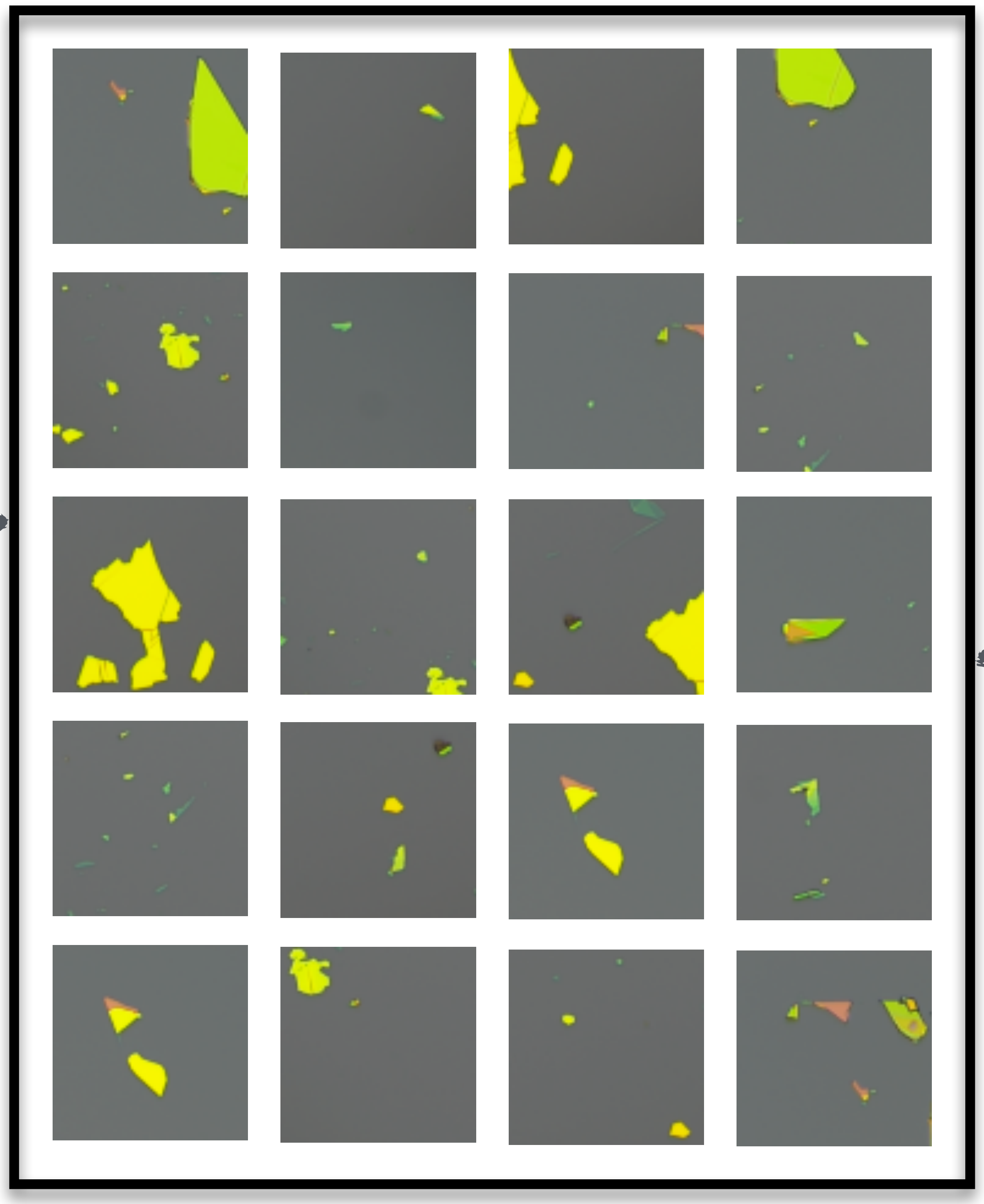


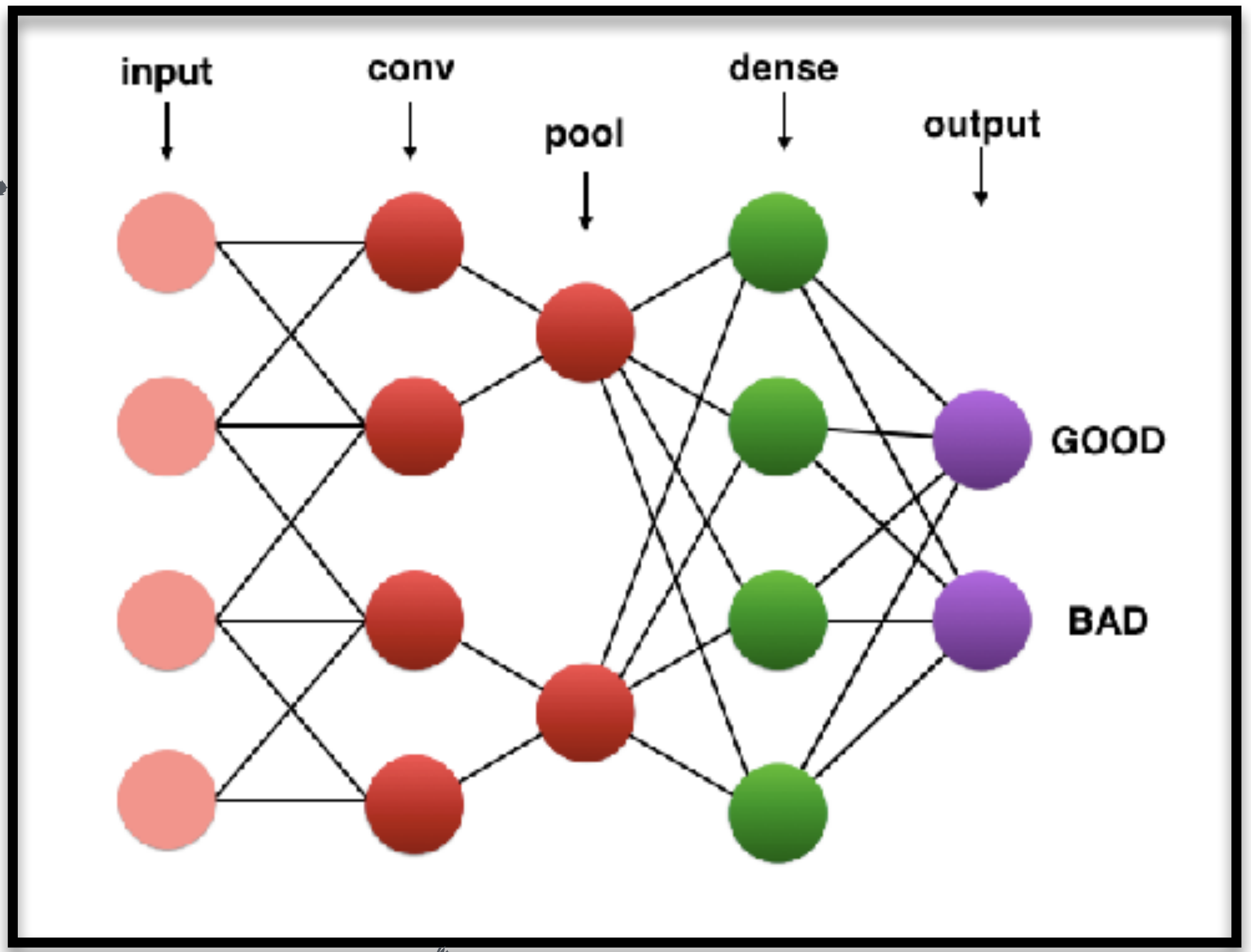
QMAI | TUDelft

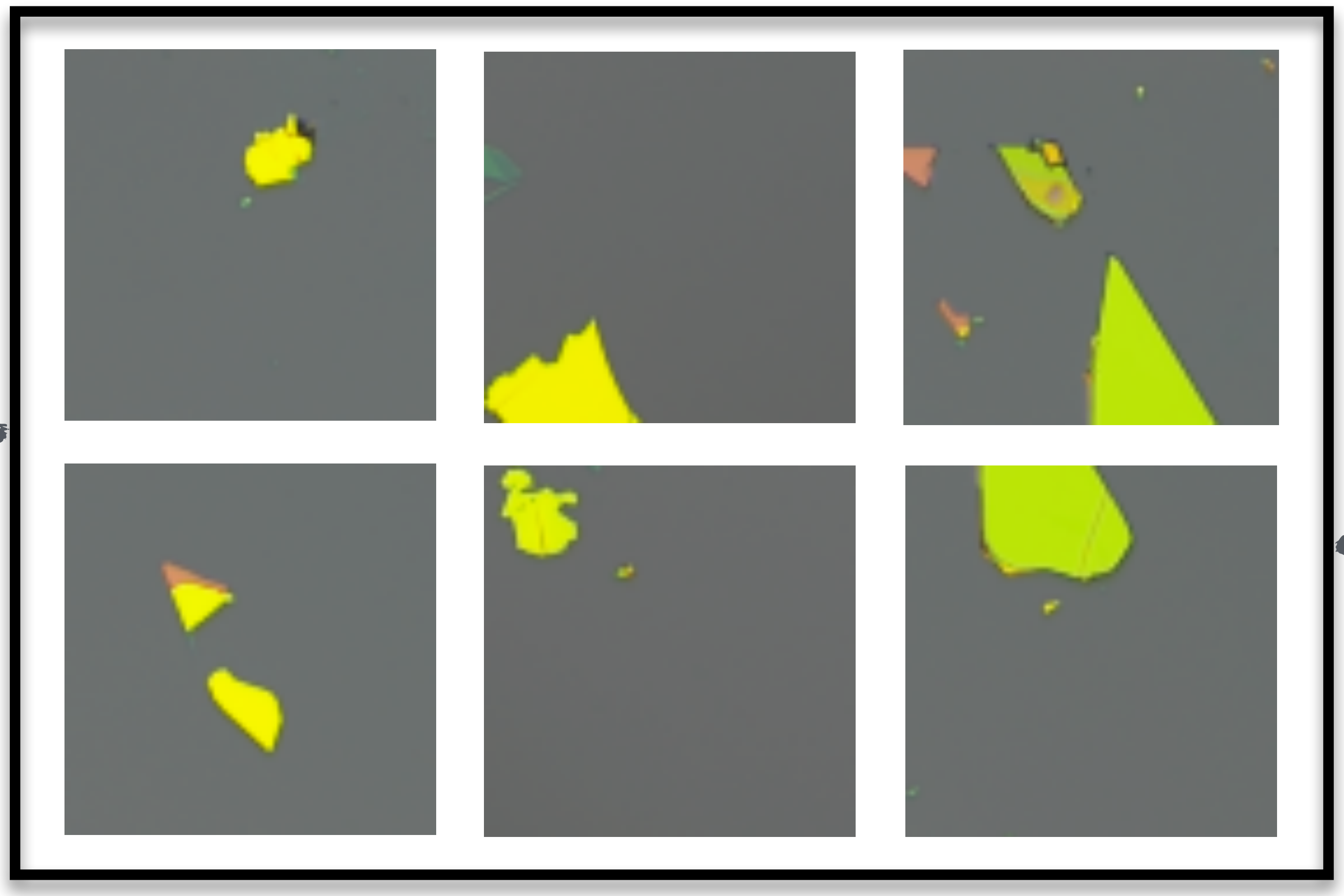
~10 000 candidates ->
~50 useful flakes

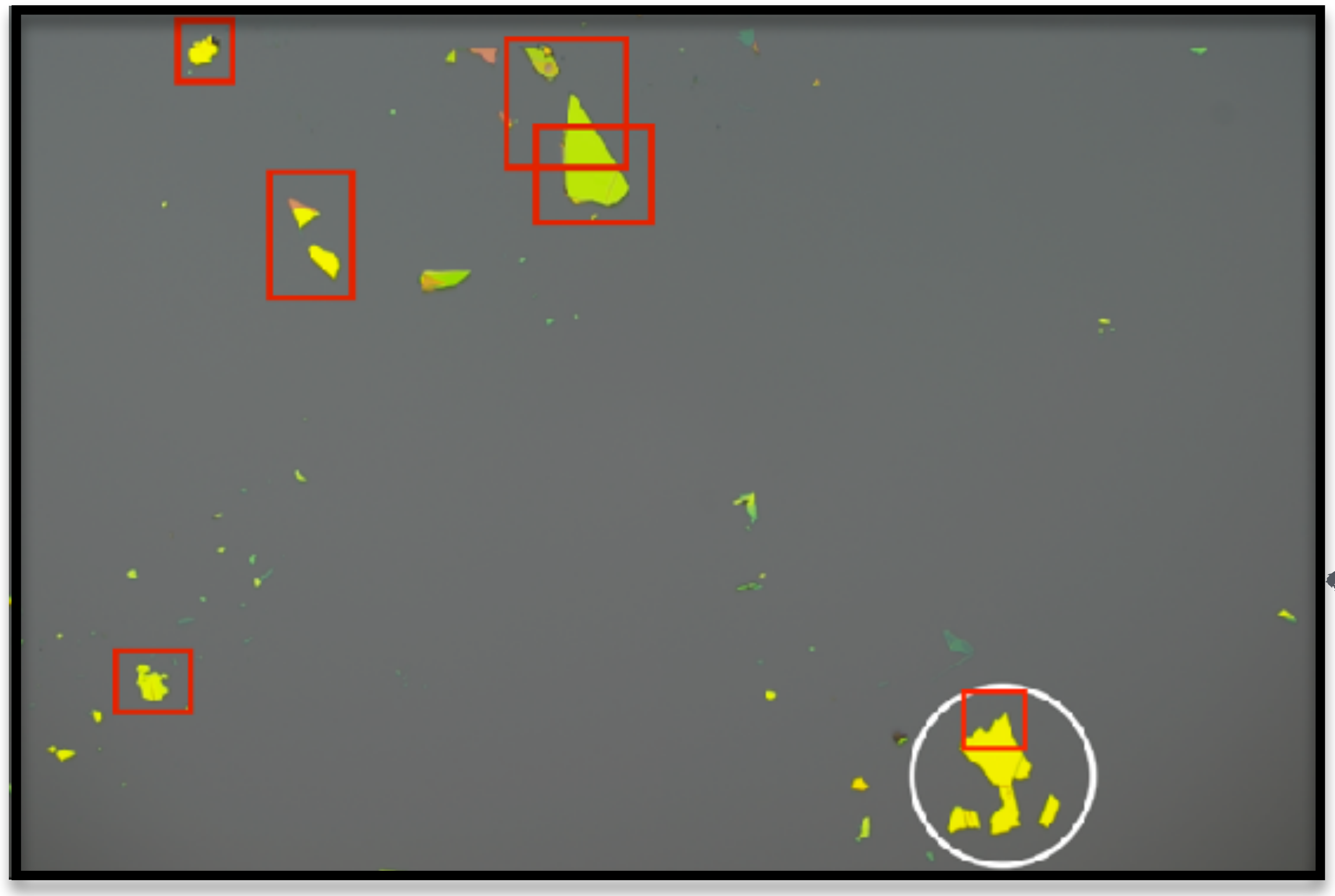
(1) Automate the scanning and zoom

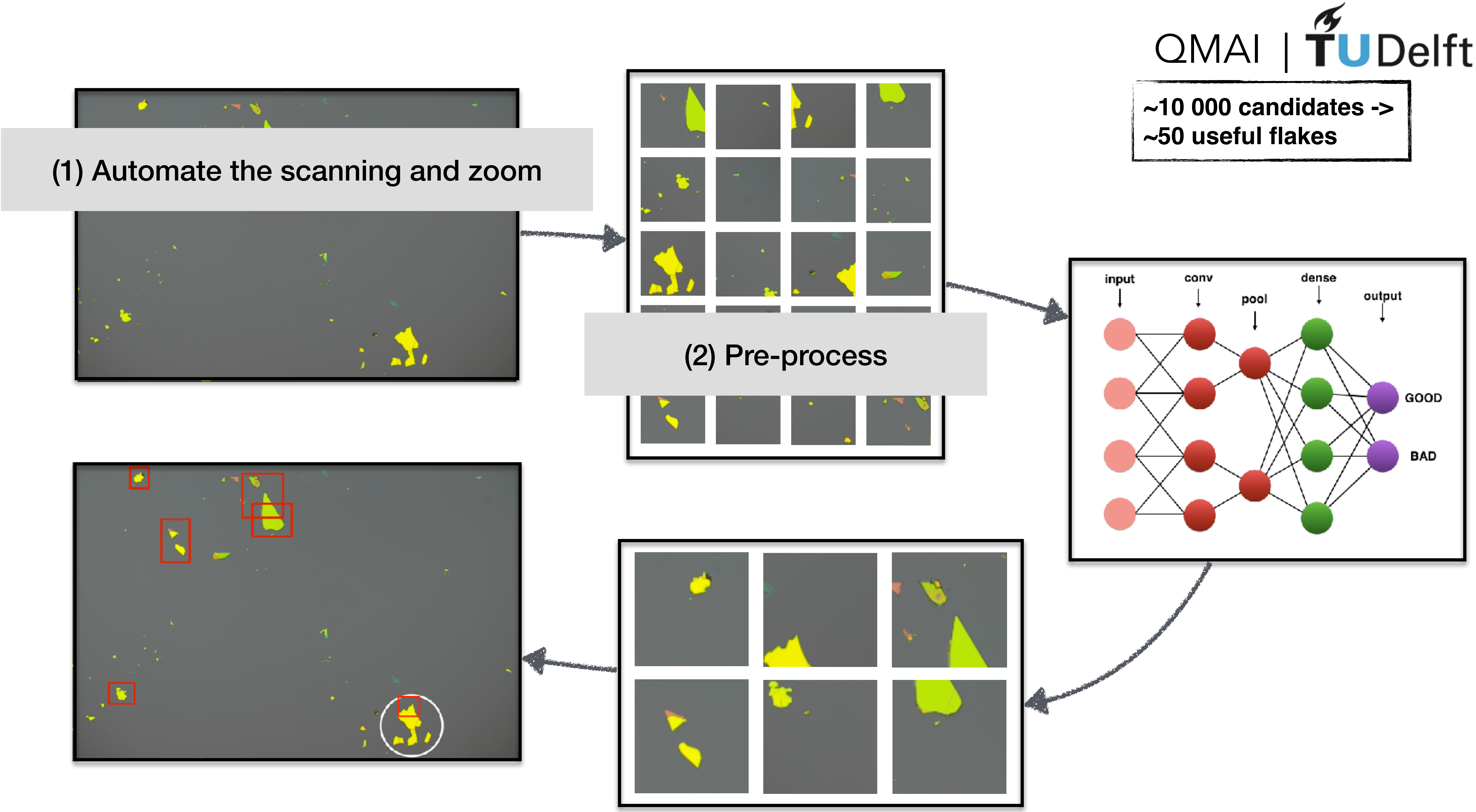






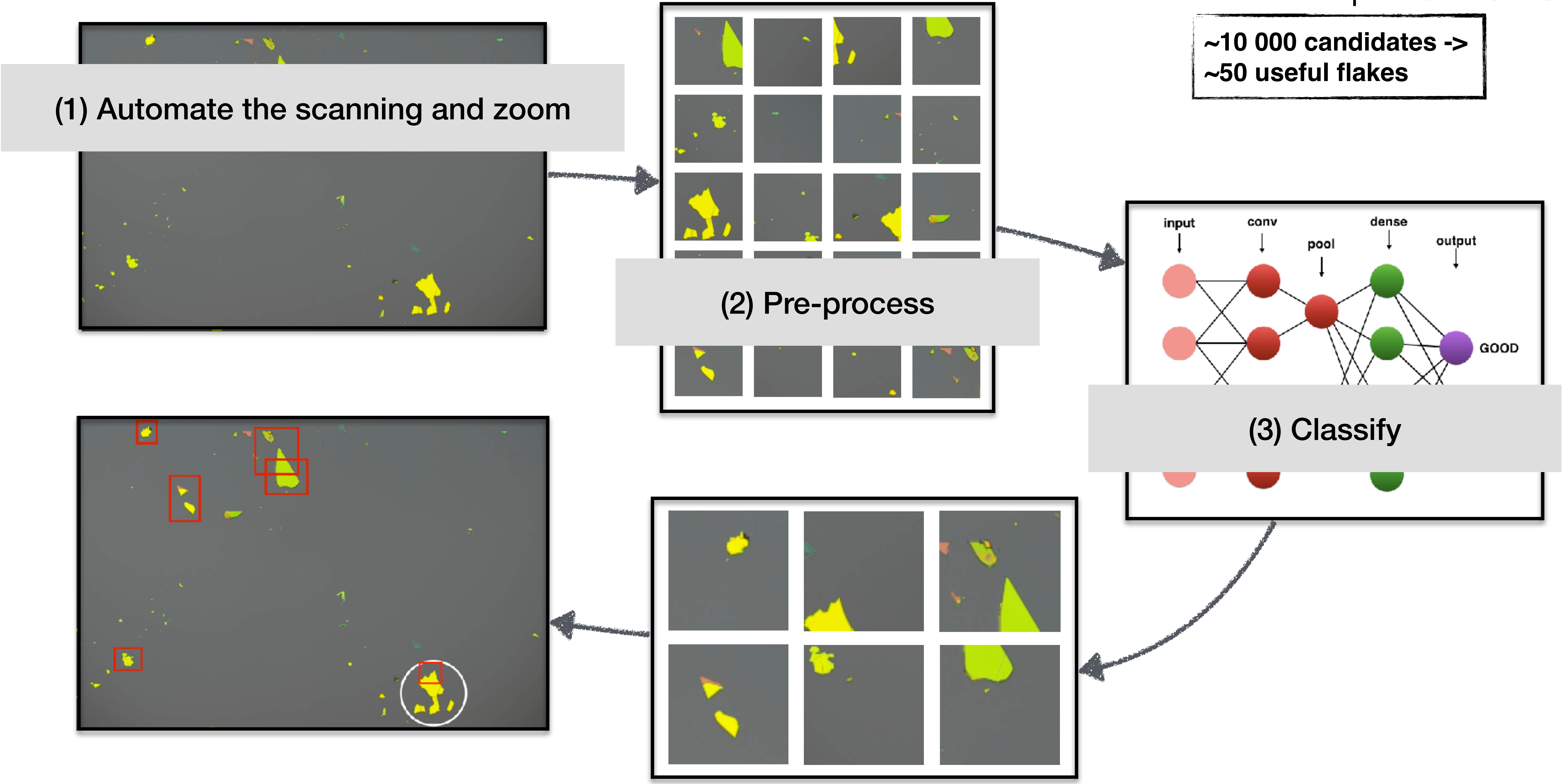




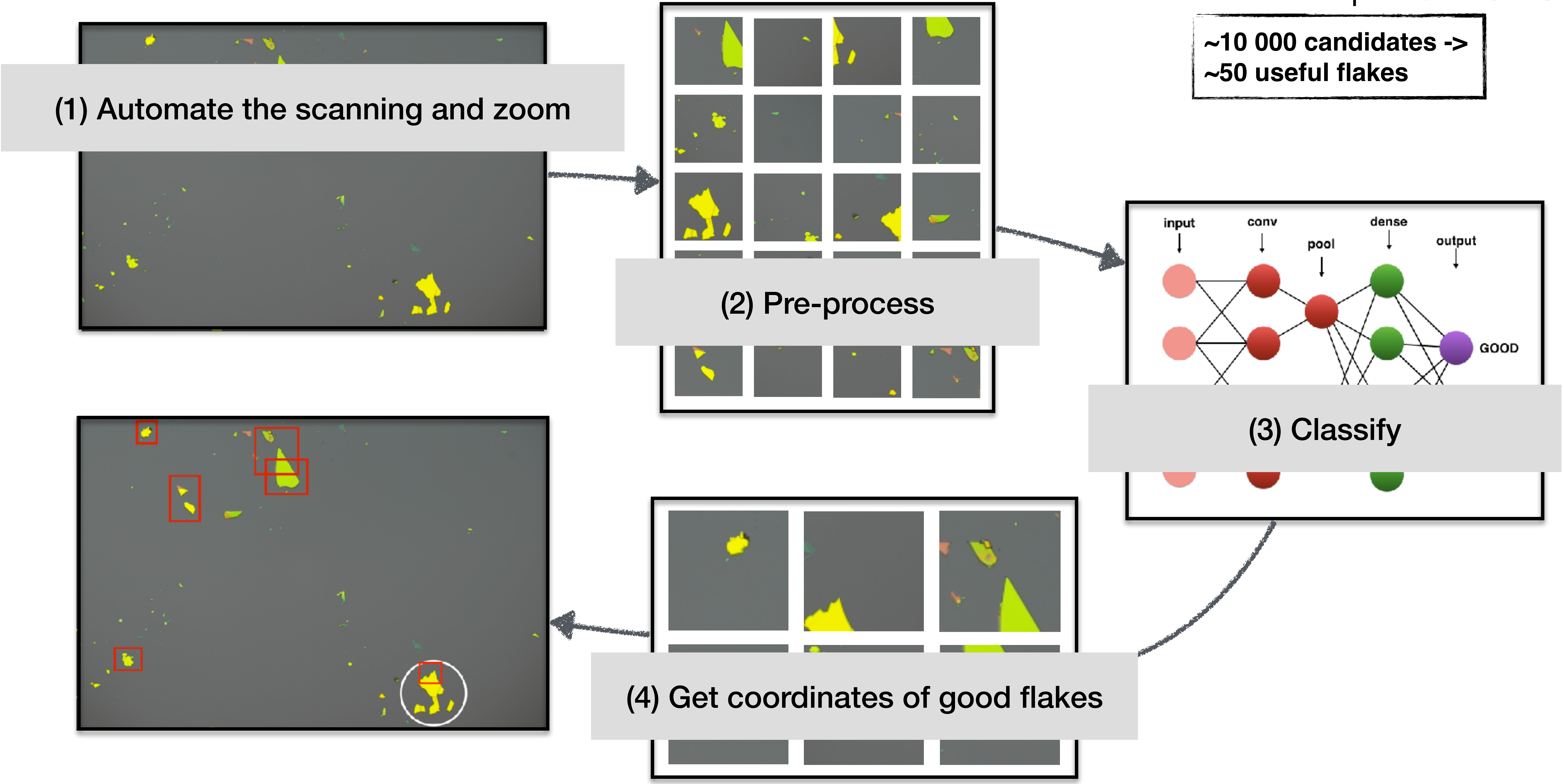

QMAI | TUDelft
~10 000 candidates ->
~50 useful flakes
(1) Automate the scanning and zoom
(2) Pre-process
input
conv
pool
dense
output
GOOD
BAD

QMAI | TU Delft

~10 000 candidates ->
~50 useful flakes

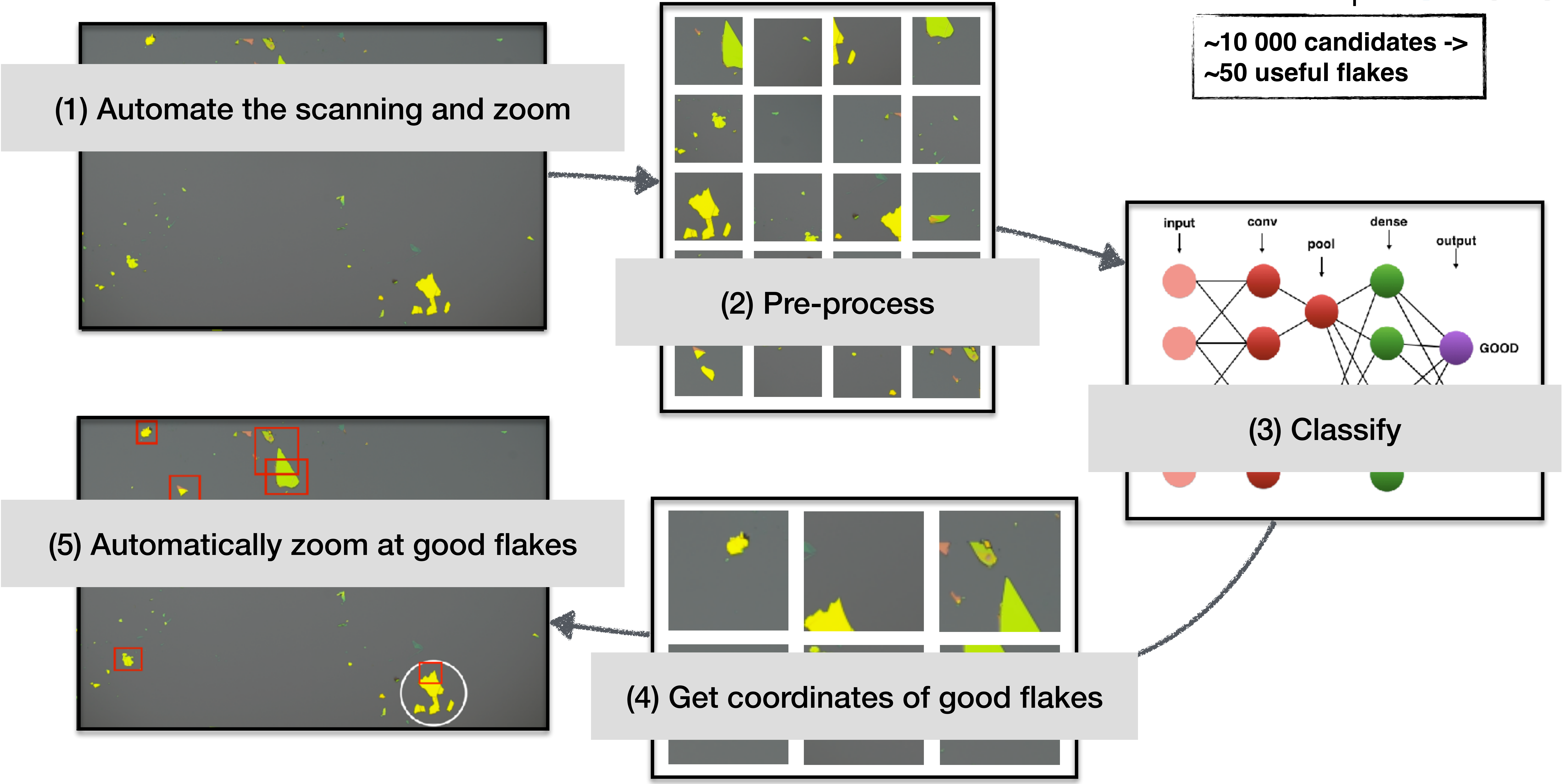


~10 000 candidates ->
~50 useful flakes



QMAI | TUDelft

~10 000 candidates ->
~50 useful flakes



QMAI | TU Delft

Our solution downsamples each 10 000 to ca 200 candidates.

EG, Carolin Gold, Benedikt Kratochwil, Tim Davatz, Riccardo Pisoni, Annika Kurzmann, Peter Rickhaus, Mark H. Fischer, Thomas Ihn, Sebastian Huber arXiv: 1910.00066 (2019)

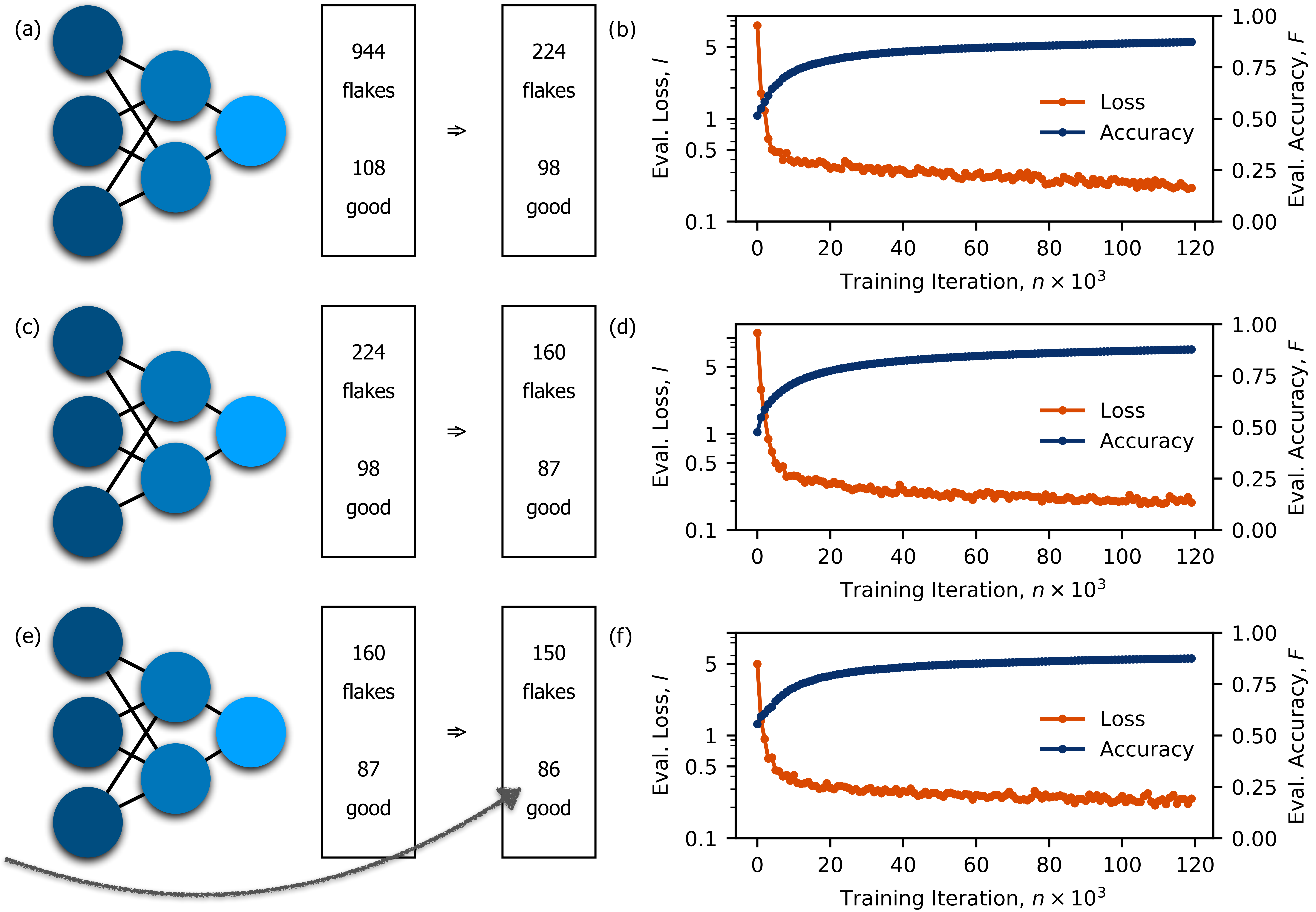
QMAI | TU Delft

Out of the 200 candidates over a 100 is useful to an experimentalist.

EG, Carolin Gold, Benedikt Kratochwil, Tim Davatz, Riccardo Pisoni, Annika Kurzmann, Peter Rickhaus, Mark H. Fischer, Thomas Ihn, Sebastian Huber arXiv: 1910.00066 (2019)

QMAI | TU Delft

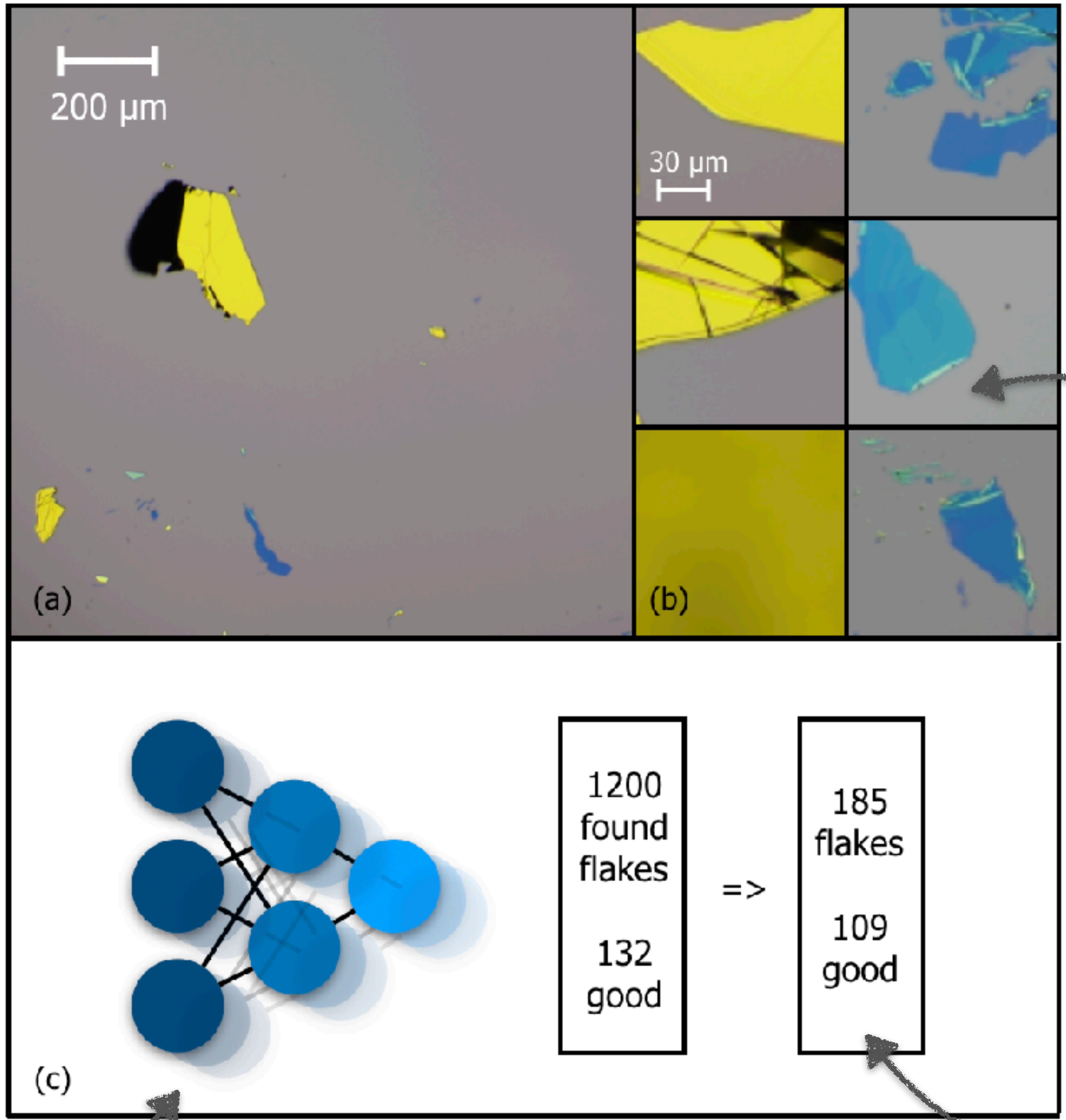
hBN performance



Every SECOND flake an experimentalist looks at is USEFUL

Graphite performance

BAD GOOD

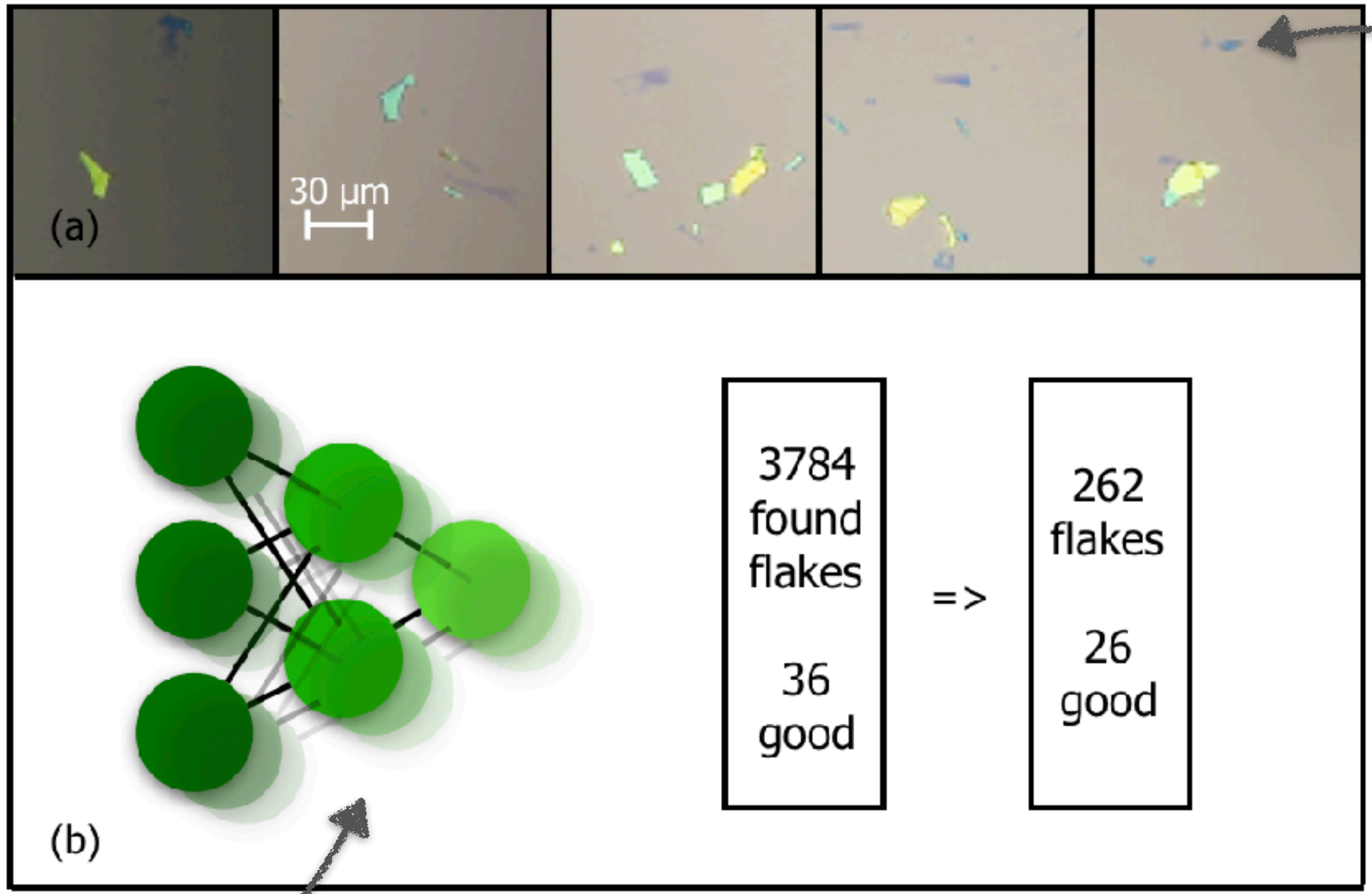


GOOD graphite flakes look quite different!

Models retrained on ca 200 GOOD graphite flakes

Every SECOND flake an experimentalist looks at is USEFUL

Bilayer Graphene performance

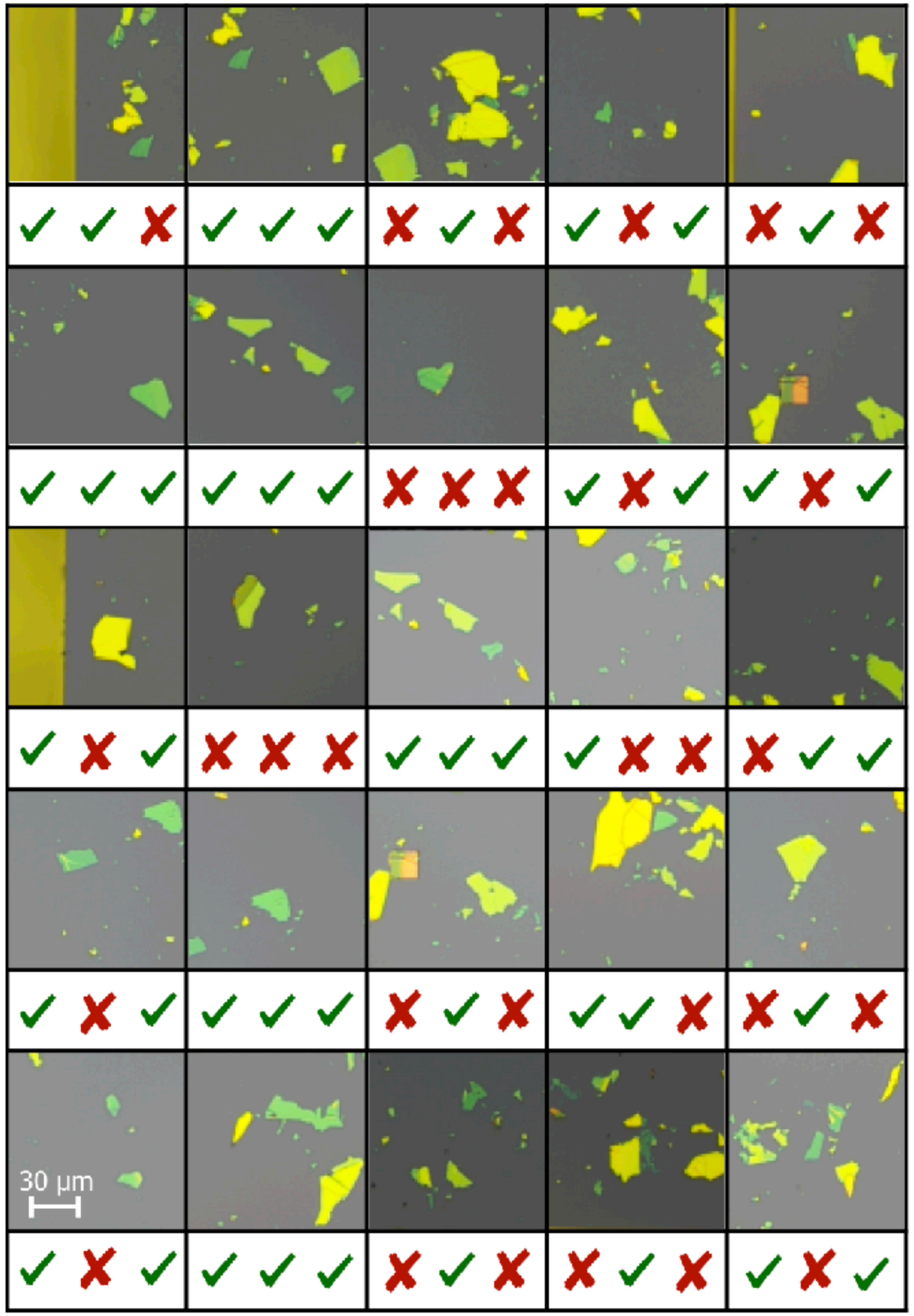


GOOD graphene flakes are hard to spot

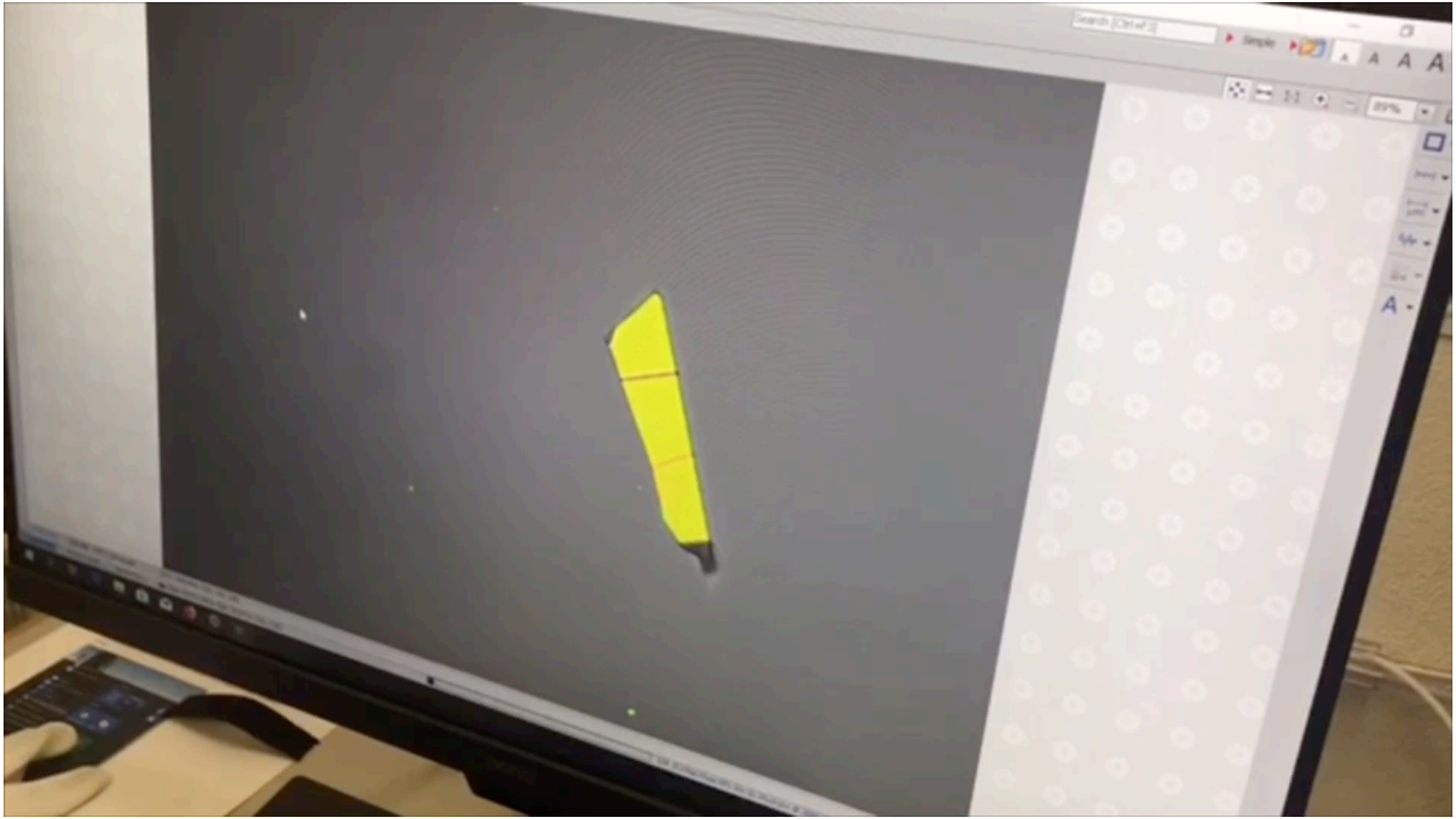
Models retrained on ca 40 GOOD graphene flakes

10% of flakes an experimentalist looks at is USEFUL

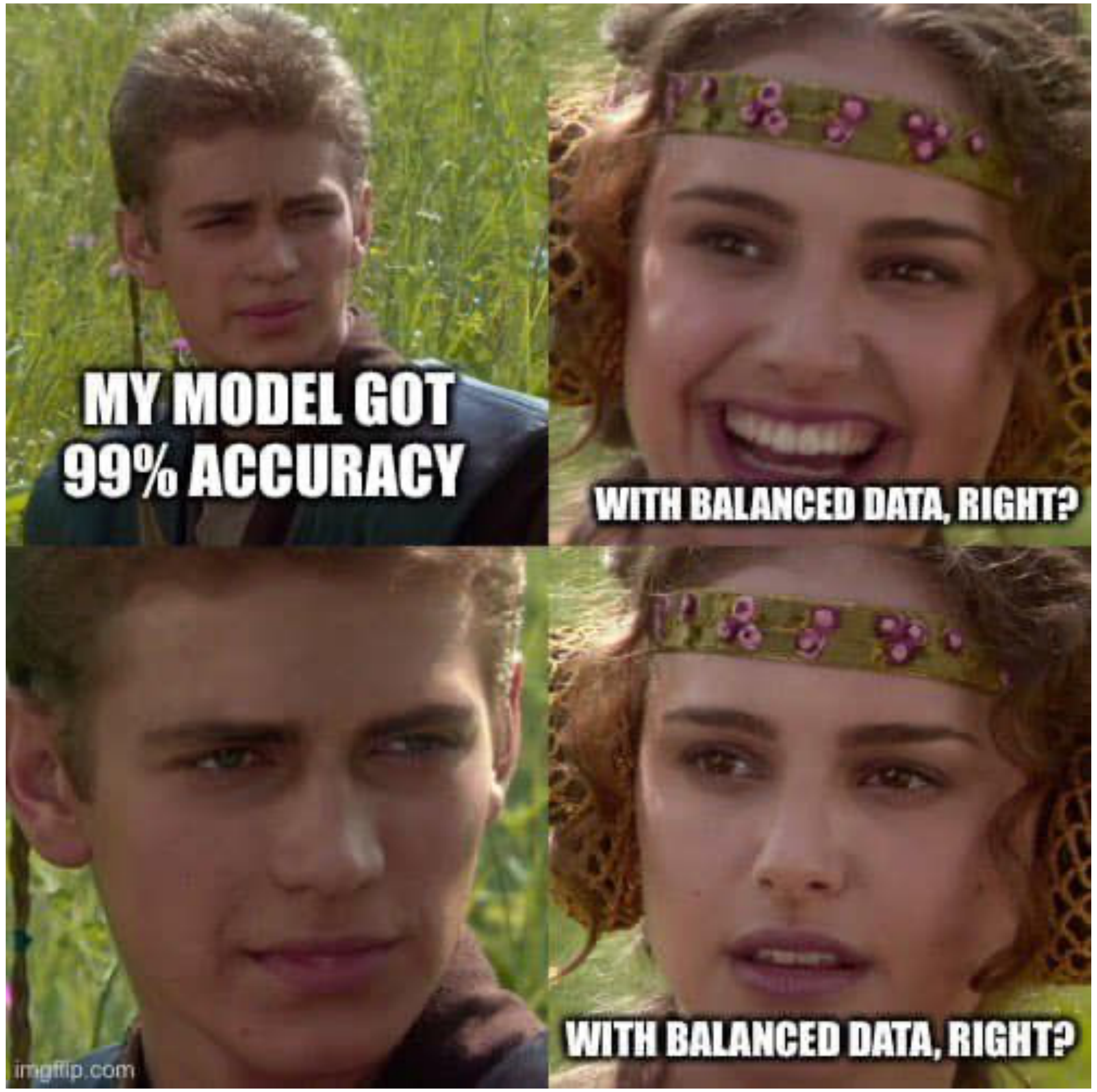
Human labelling



The diversity in the labels given by different human operators introduces a bound on the efficiency of machine learning model.

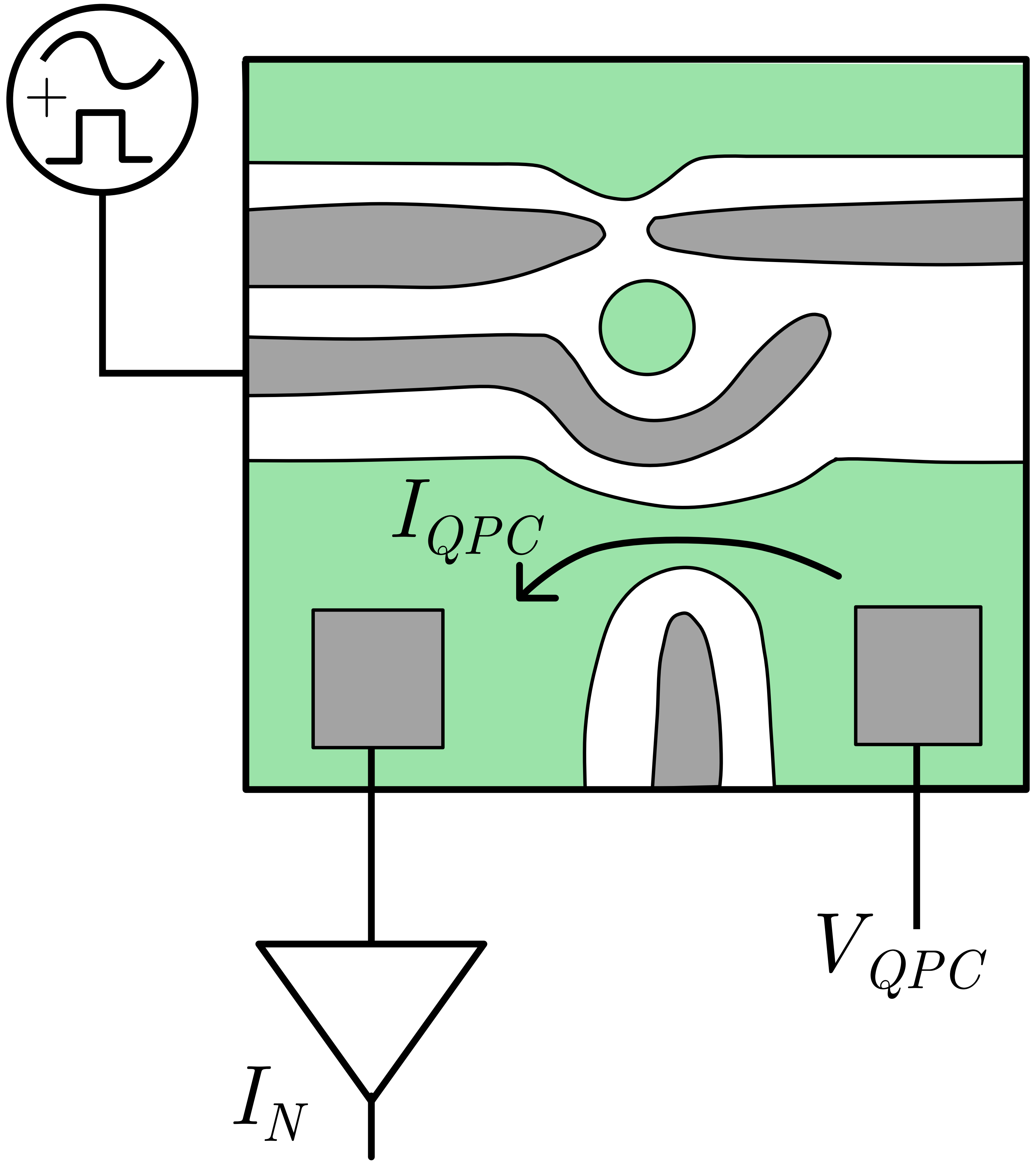


15 minutes break



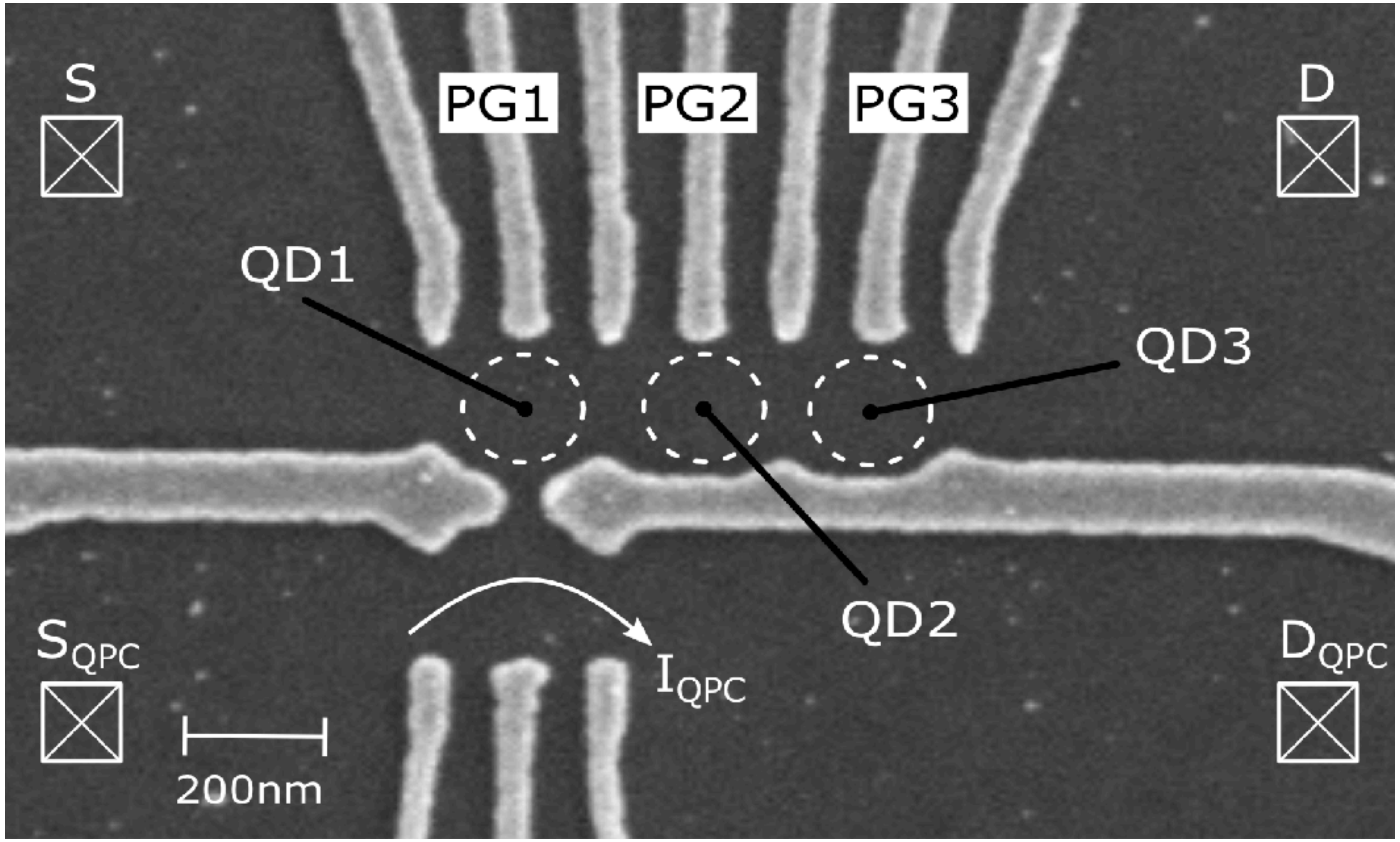
Quantum Dots

QMAI | TU Delft



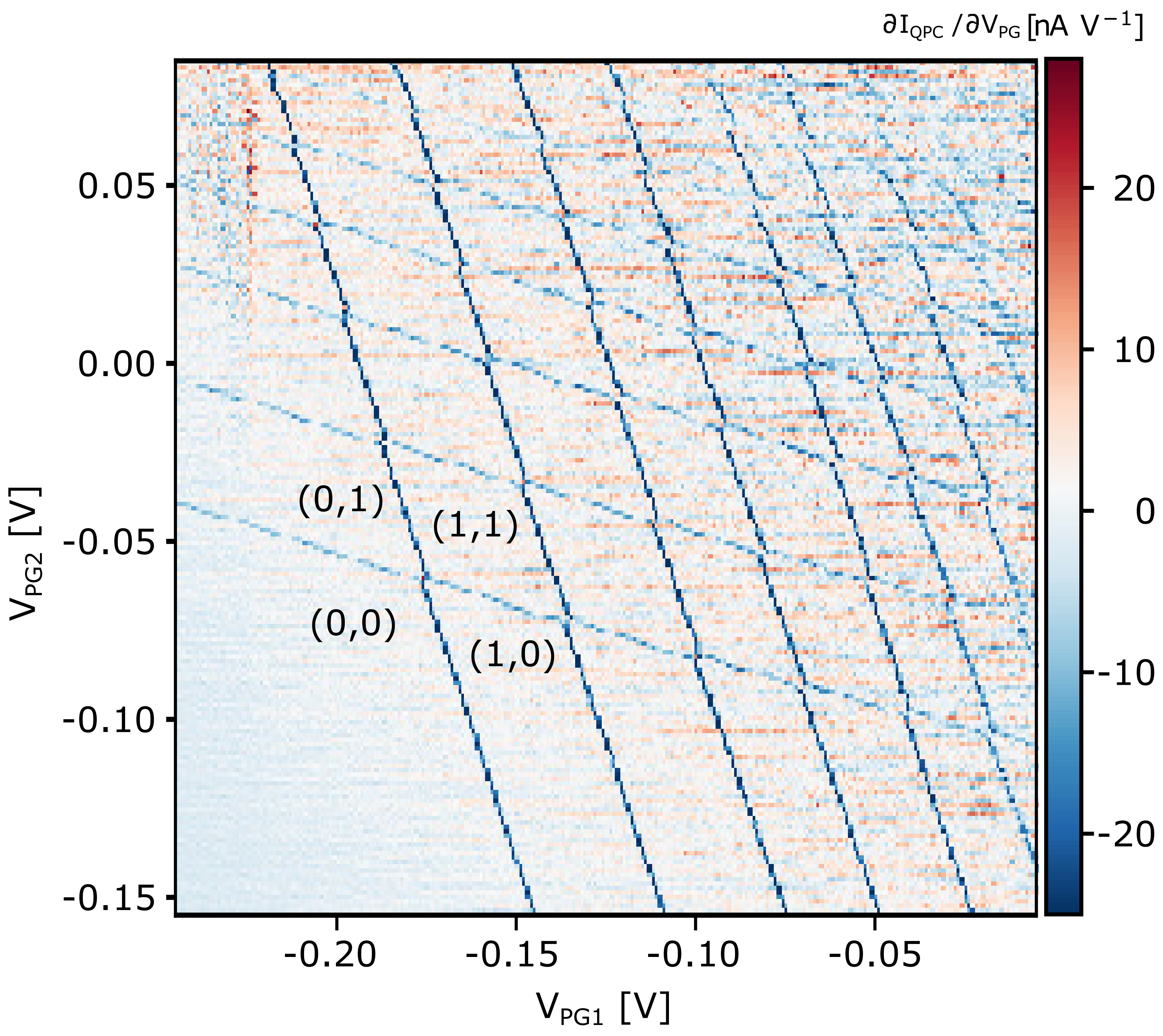
The Device

QMAI | TU Delft



R Durrer, B Kratochwil, JV Koski, AJ Landig, C Reichl, W Wegscheider, T Ihn, **EG**, Phys. Rev. Applied 13, 054019 (2020)

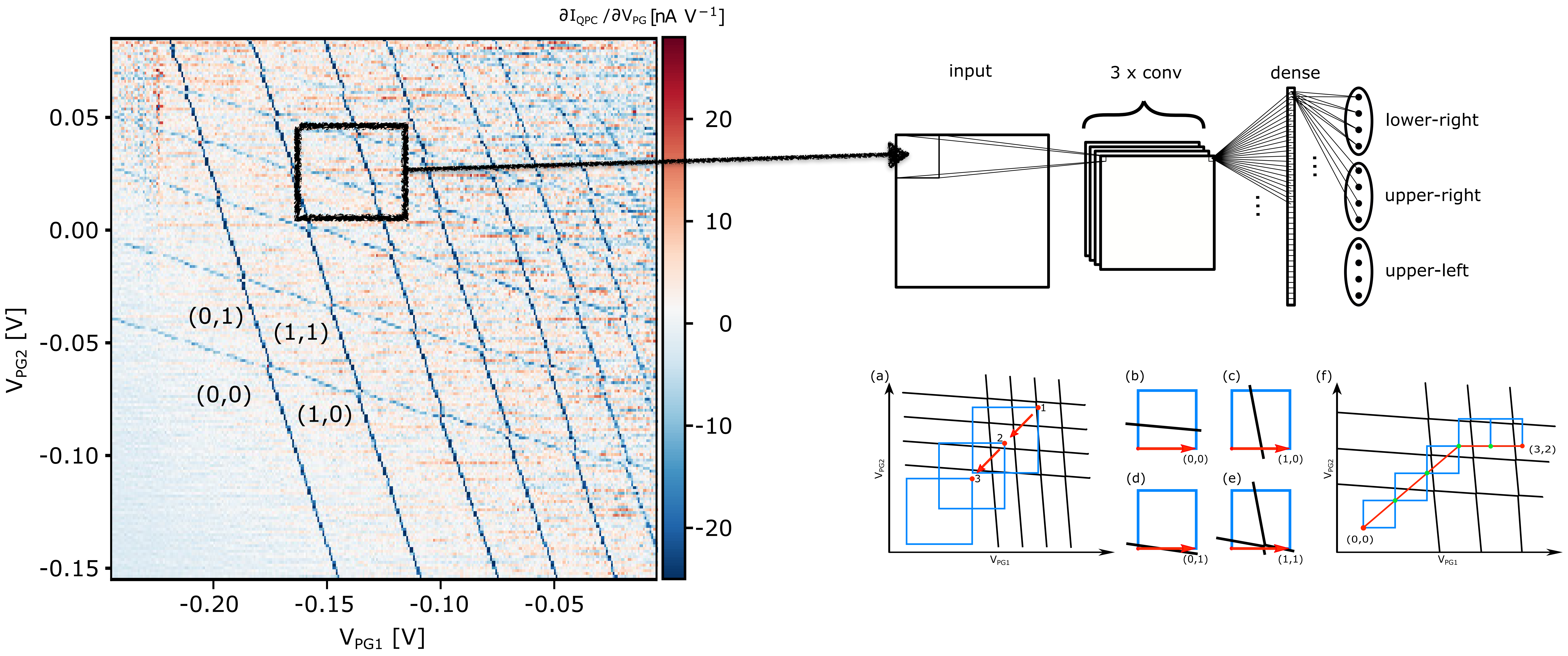
Sampling Charge-Stability Diagrams



R Durrer, B Kratochwil, JV Koski, AJ Landig, C Reichl, W Wegscheider, T Ihn, **EG**, Phys. Rev. Applied 13, 054019 (2020)

Preparing Specific Charge States

QMAI | TU Delft



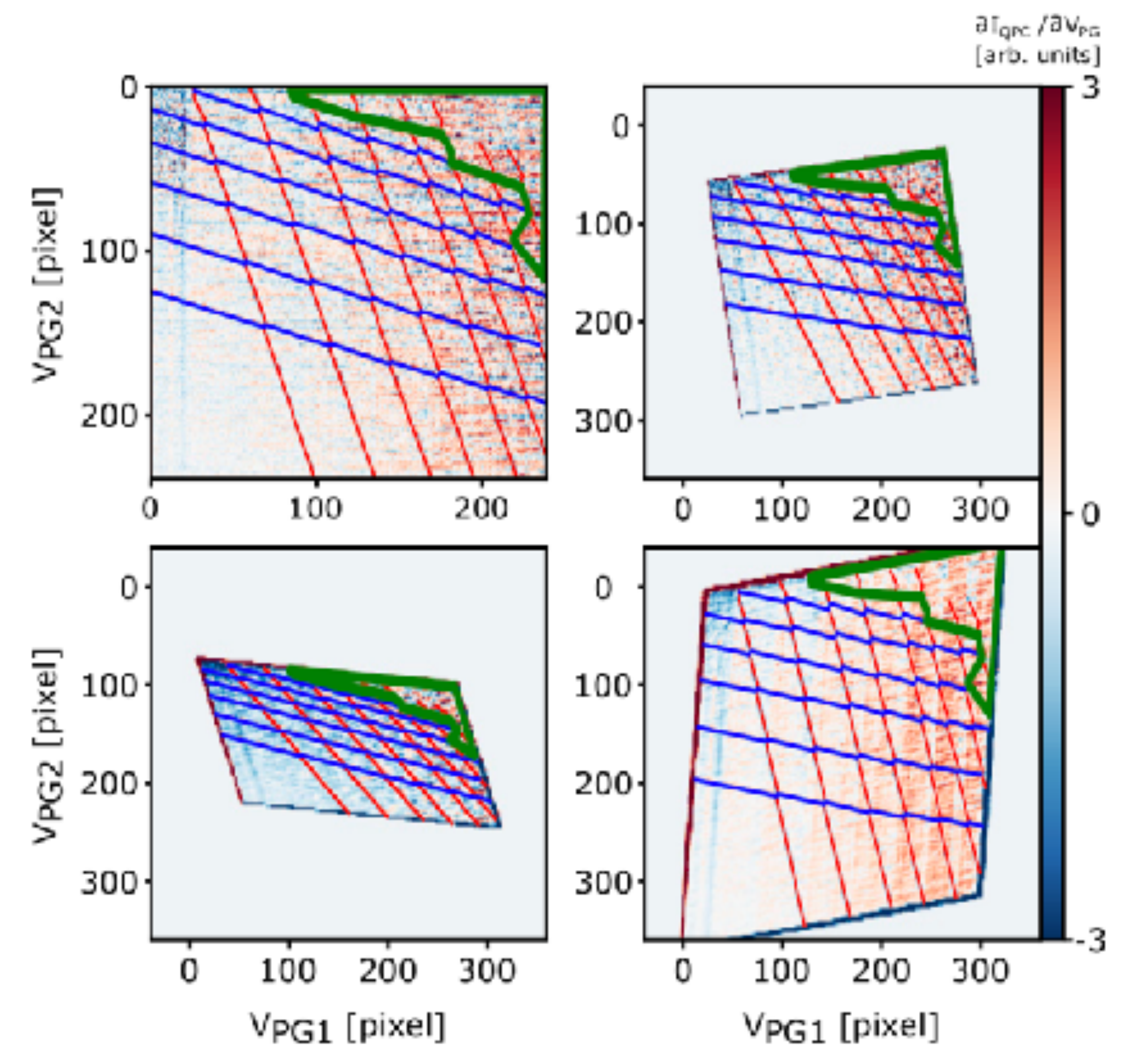
R Durrer, B Kratochwil, JV Koski, AJ Landig, C Reichl, W Wegscheider, T Ihn, **EG**, Phys. Rev. Applied 13, 054019 (2020)

Training set

128 full charge stability diagrams

+

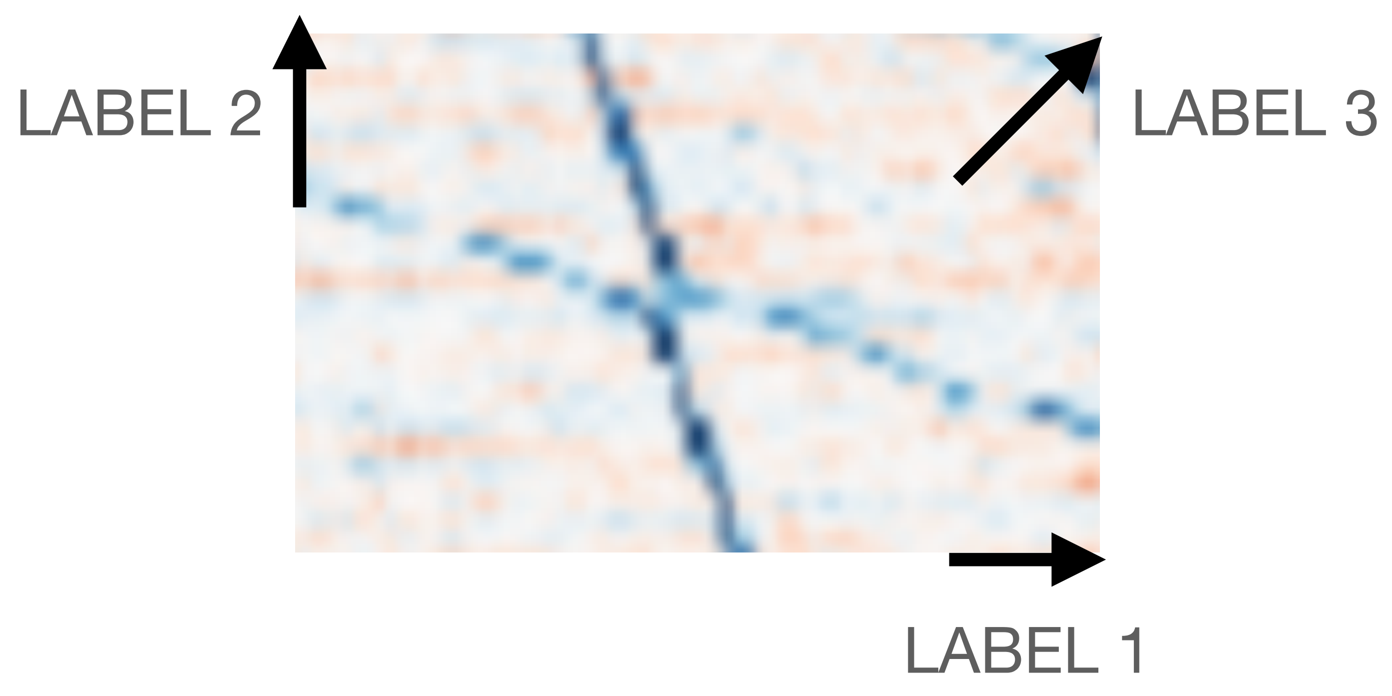
Augmentations

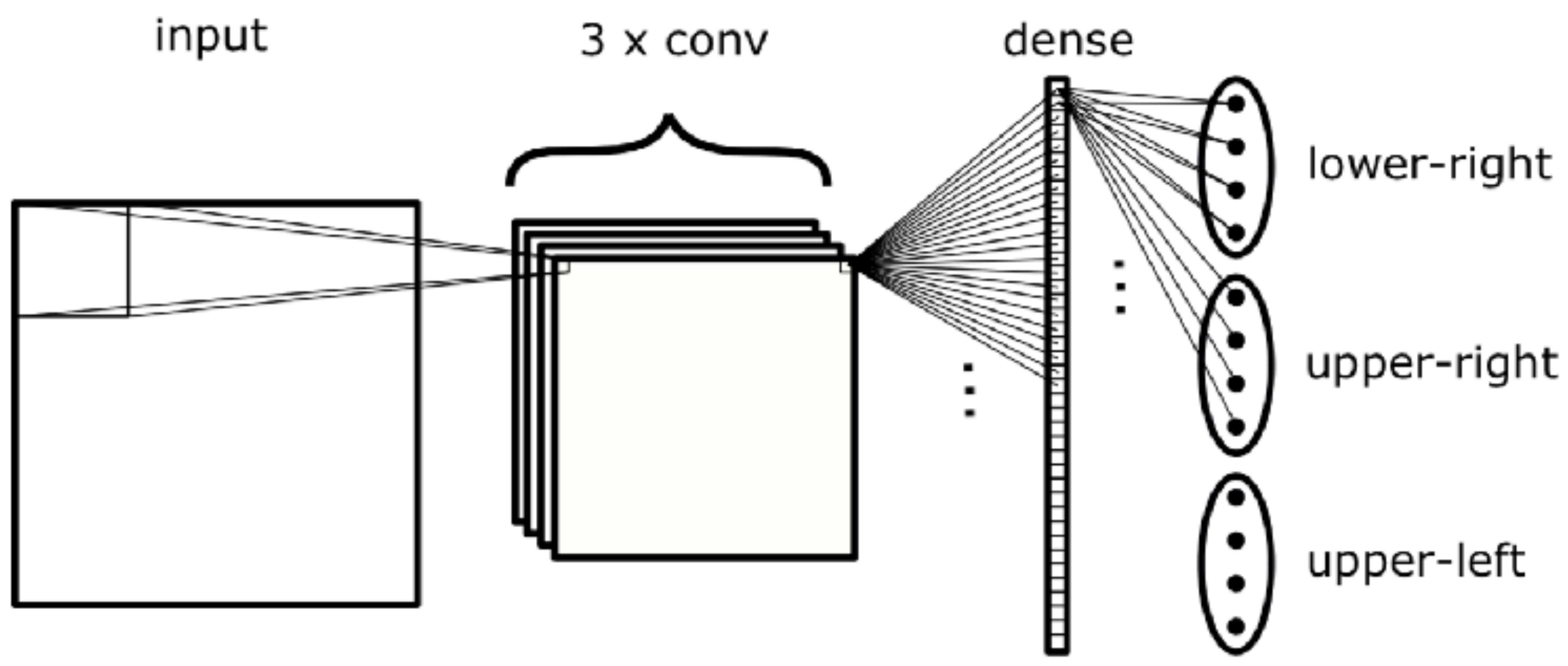


Training

QMAI | TU Delft

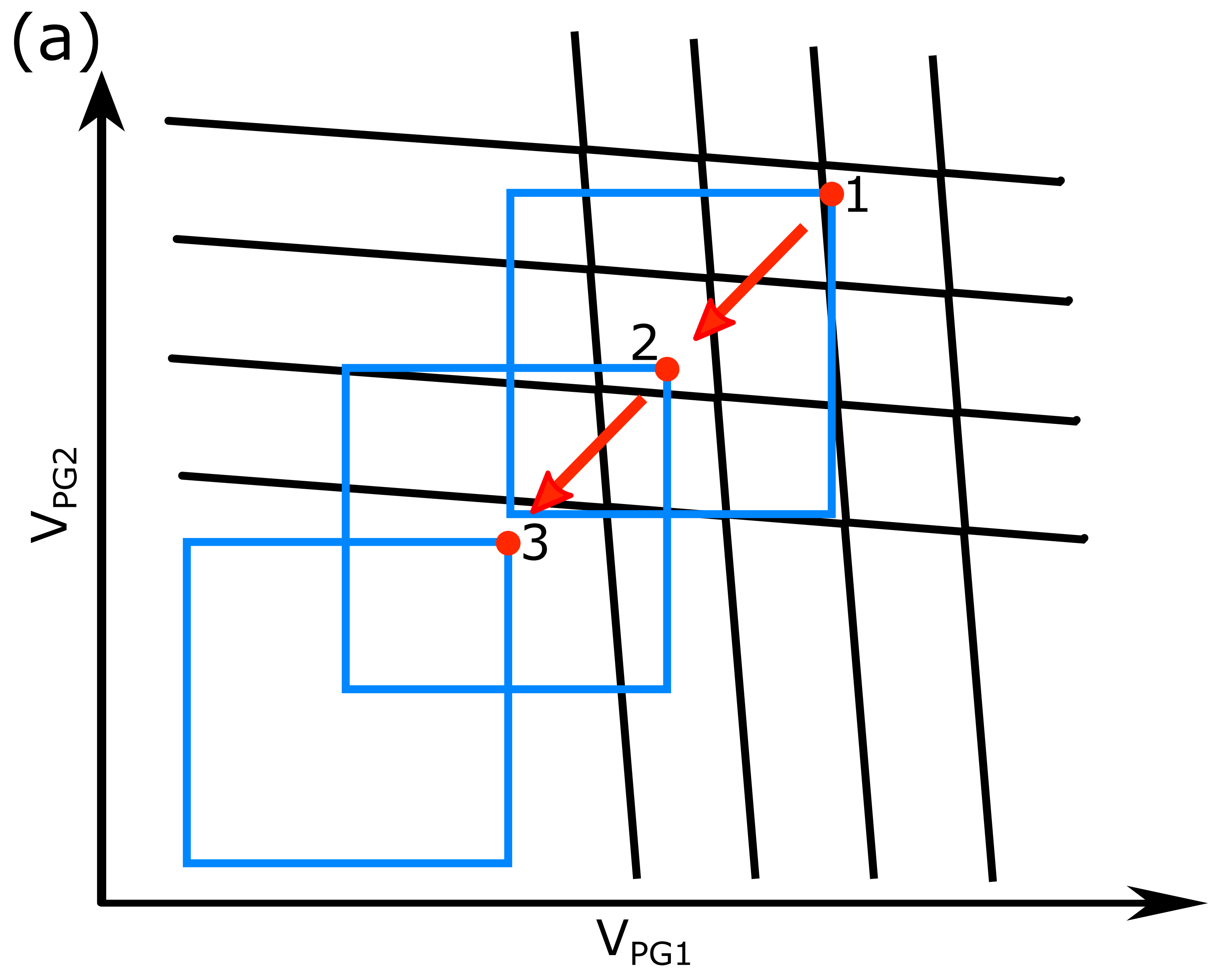
~600 000 labeled patches



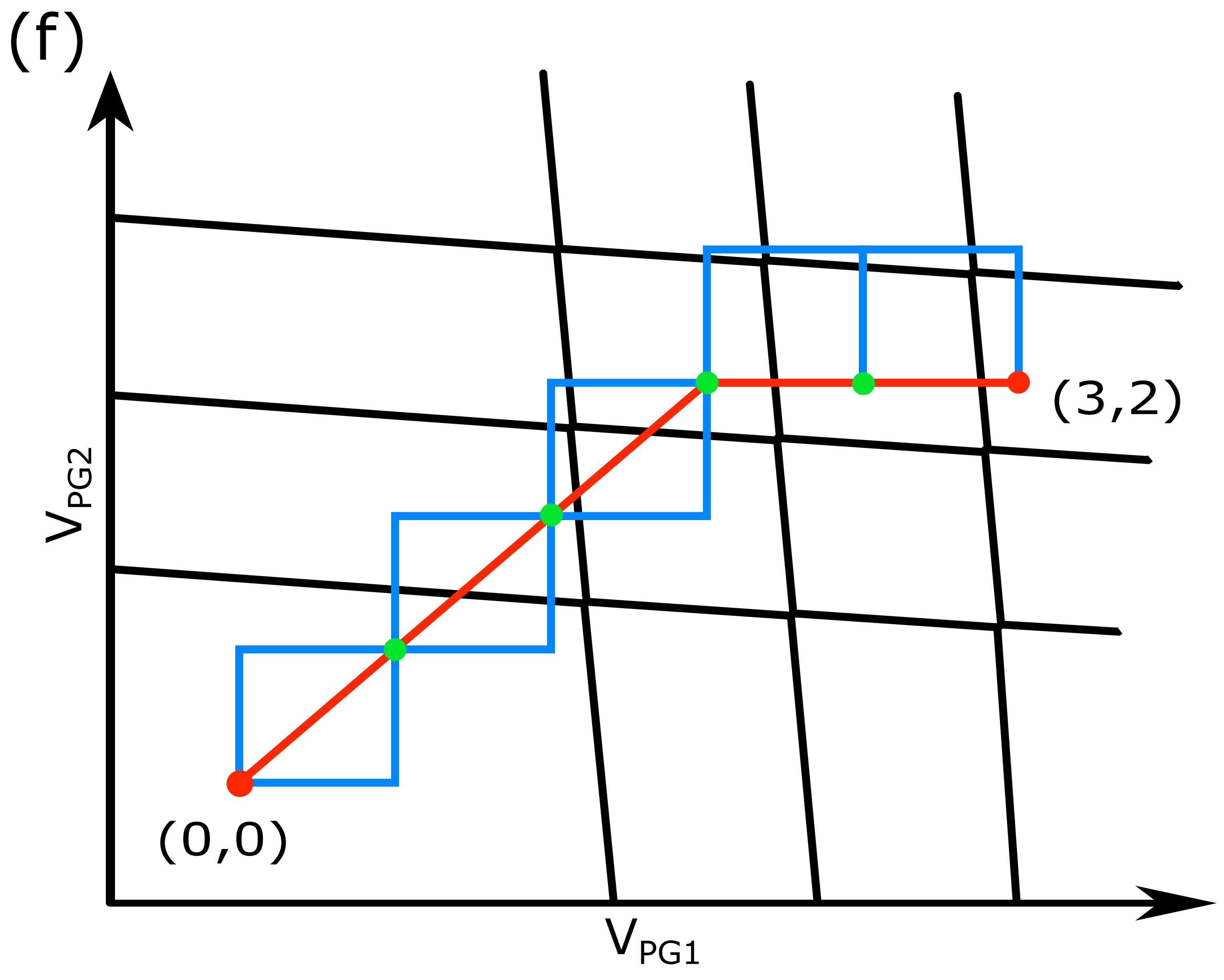


accuracy: 98.9%

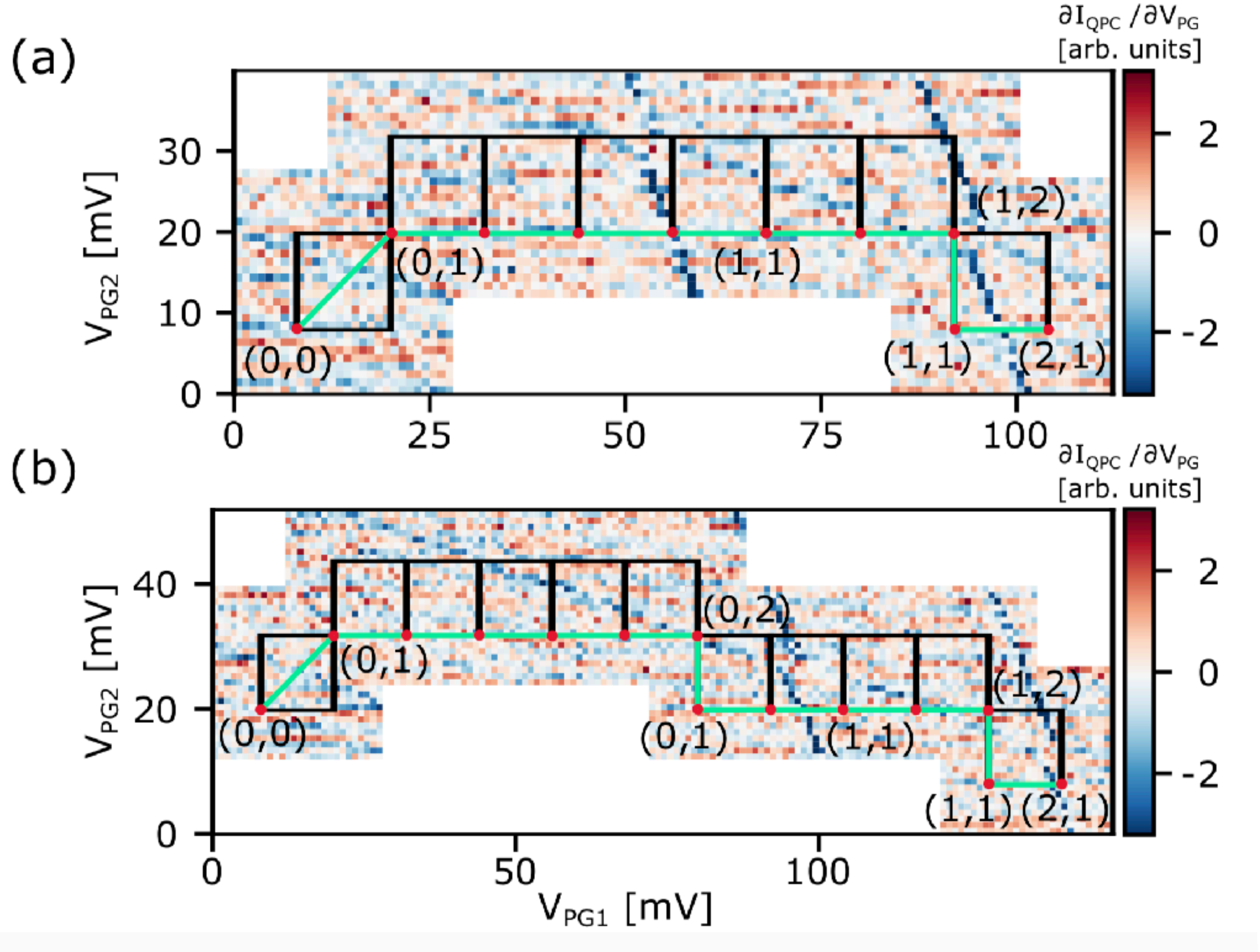
Part 1: Find (0,0)



Part 2: Find (m,n)



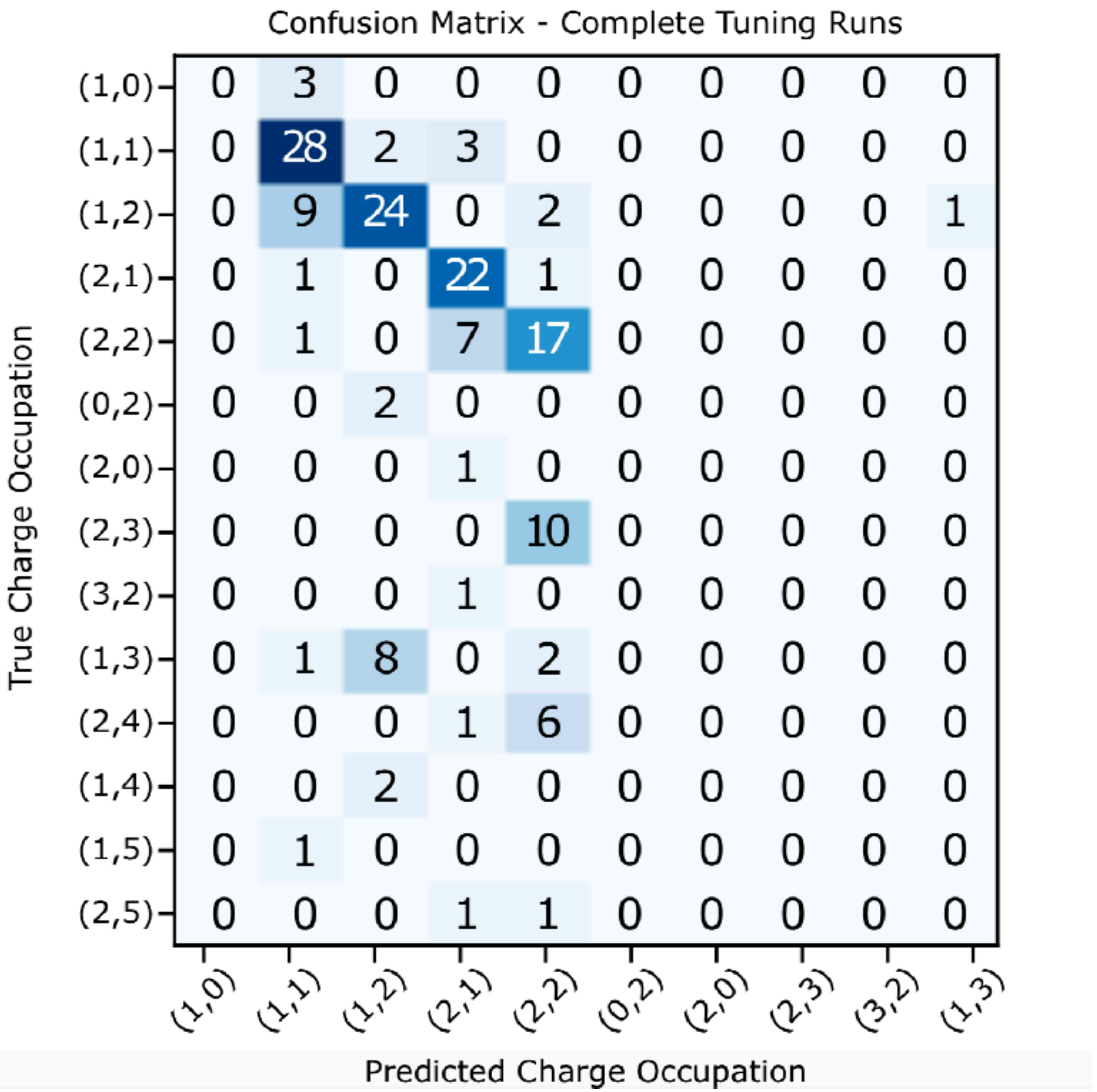
Tuning Runs on the Device



R Durrer, B Kratochwil, JV Koski, AJ Landig, C Reichl, W Wegscheider, T Ihn, **EG**, Phys. Rev. Applied 13, 054019 (2020)

Results

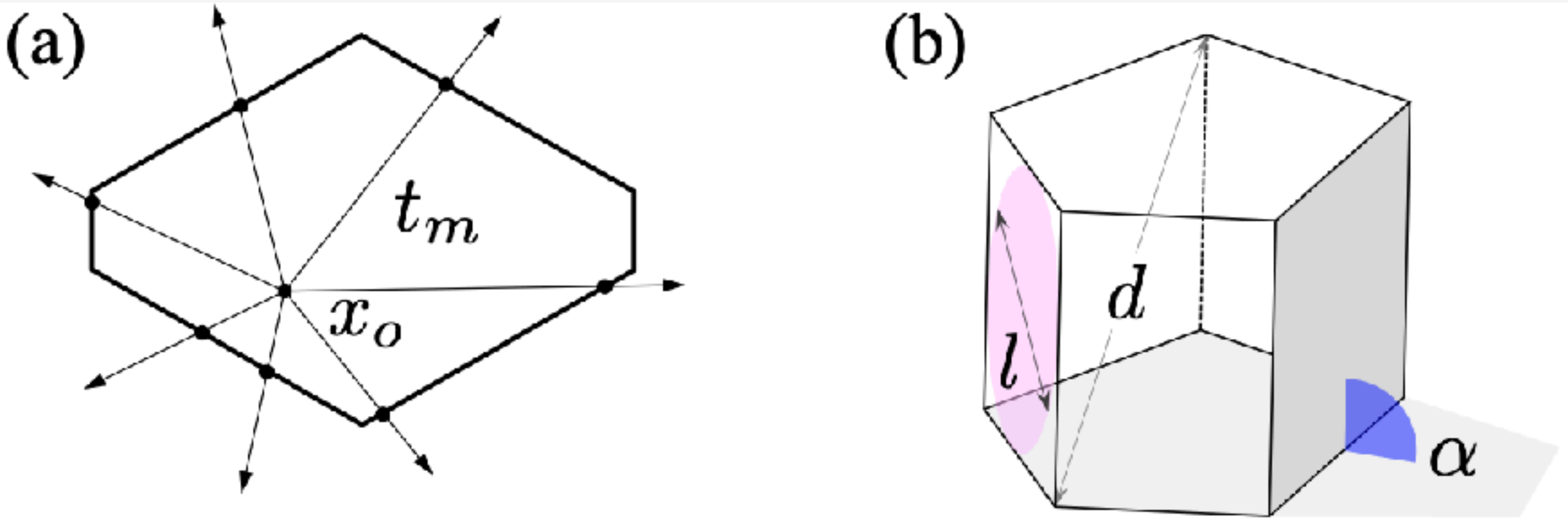
- 160 tuning runs
- Finding (0,0): 90% success
- Finding arbitrary charge state:
  ca 60% success
- same success rate on both DQDs



R Durrer, B Kratochwil, JV Koski, AJ Landig, C Reichl, W Wegscheider, T Ihn, **EG**, Phys. Rev. Applied 13, 054019 (2020)

Exciting related work

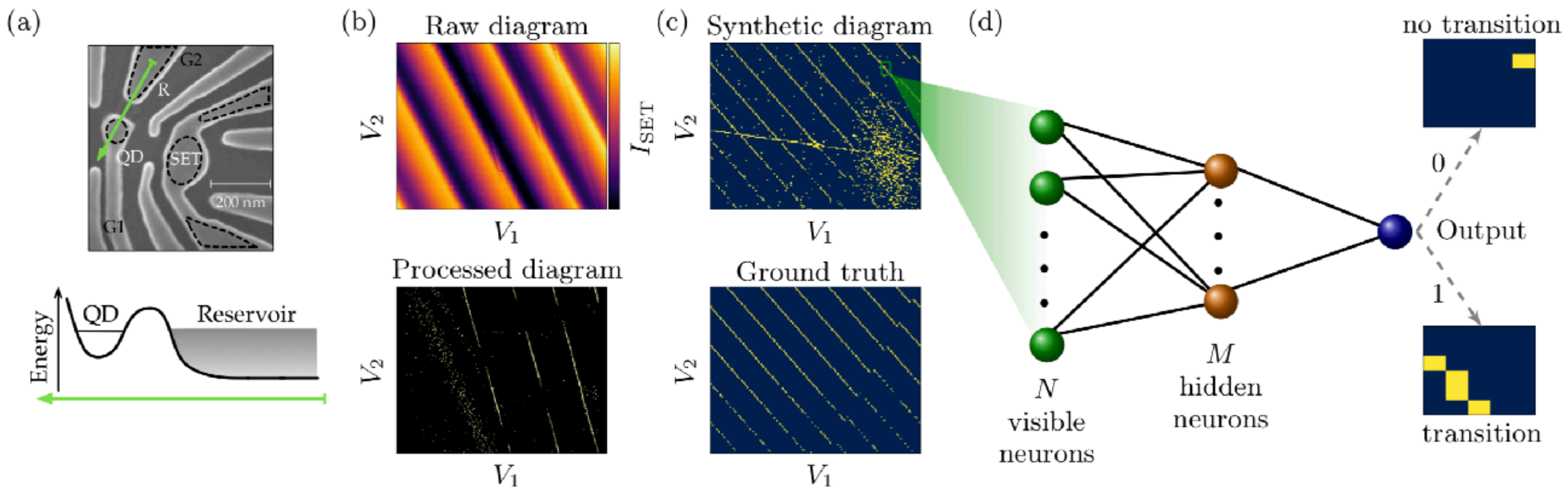
- less data for patch collection: ray method Justyna Zwolak



arXiv: 2103.09577

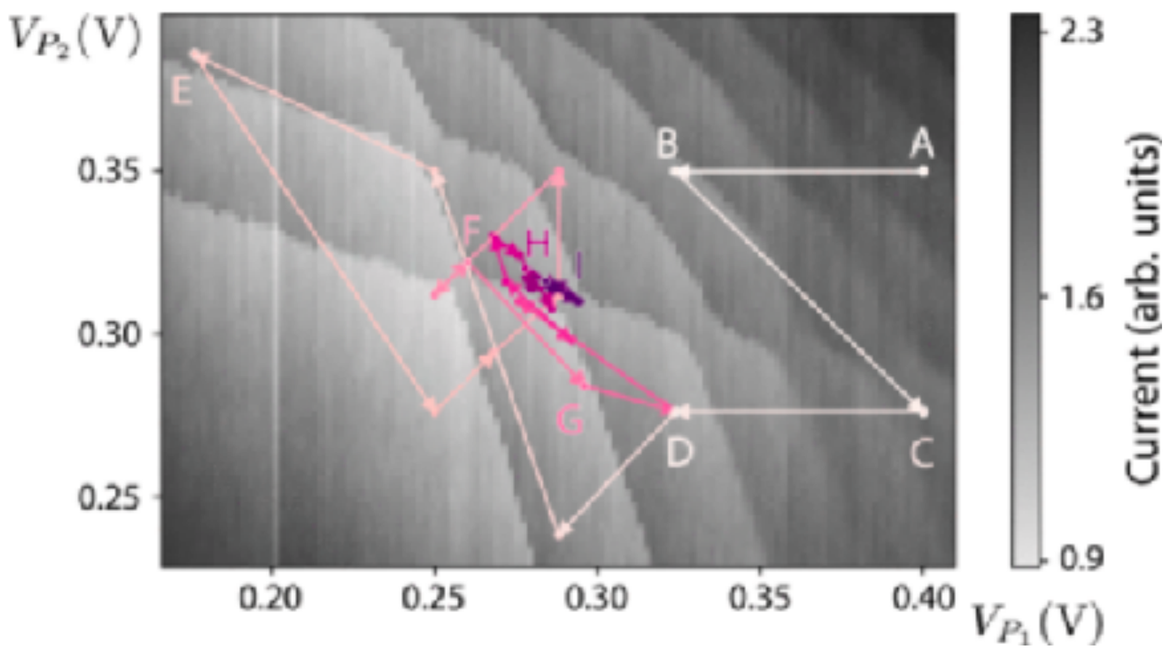
Exciting related work

- Stephanie Czischek (Roger Melko group) - NN miniaturisations



arXiv: 2101.03181

Auto-Tuning



Kalentre et al., npj QI 5, 6 (2019)
Zwolak et al., Phys. Rev. Applied 13, 034075 (2020)

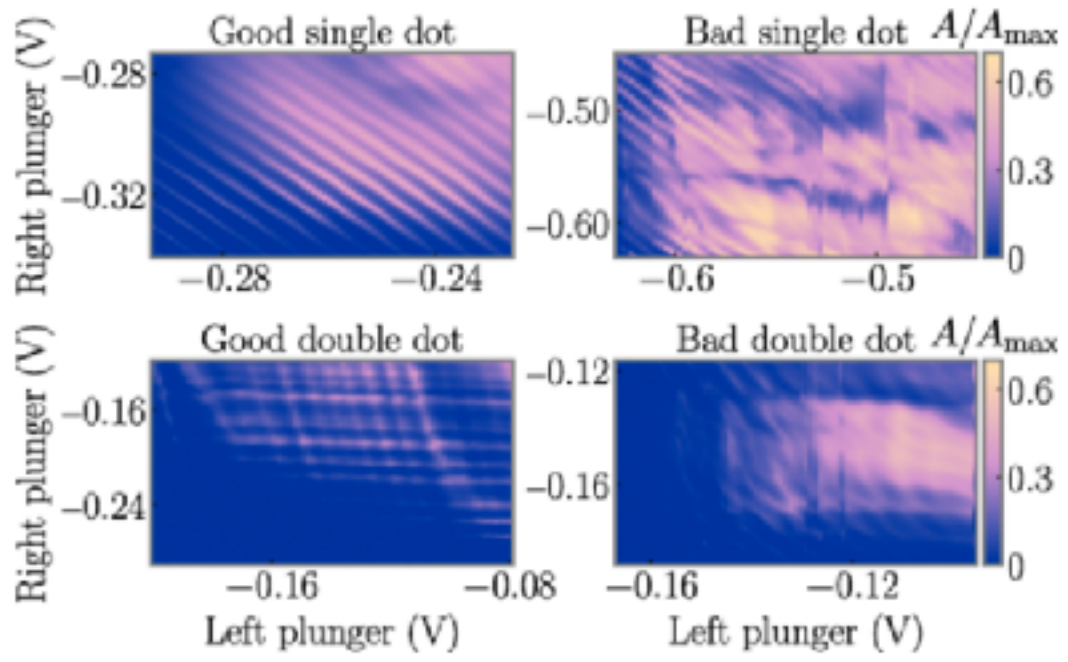
Quantum Dot Configuration

Botzem et al, Phys. Rev. Applied 10, 054026 (2018)
van Diepen et al, Appl. Phys. Lett. 113, 033101 (2018)
Teste et al, Appl. Phys. Lett. 114, 133102 (2019)
Mills et al, Appl. Phys. Lett. 115, 113501 (2019)
van Esbroeck et al, New Journal of Physics, 22, 095003 (2020)

Fine Tuning

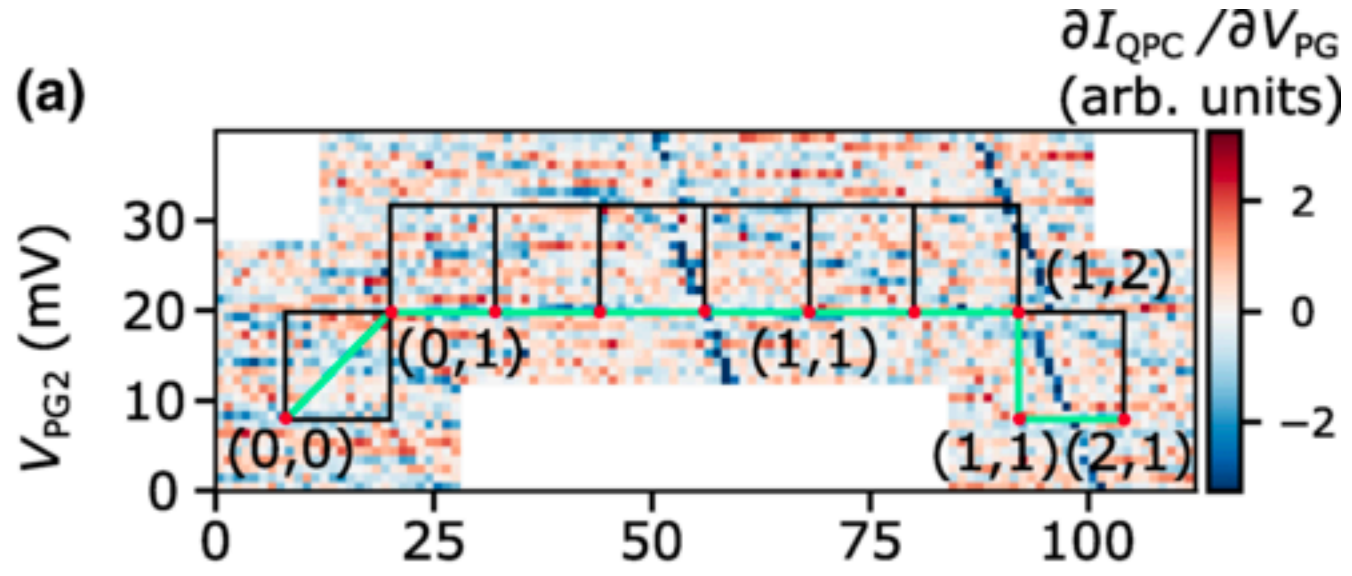
Initial tune-up

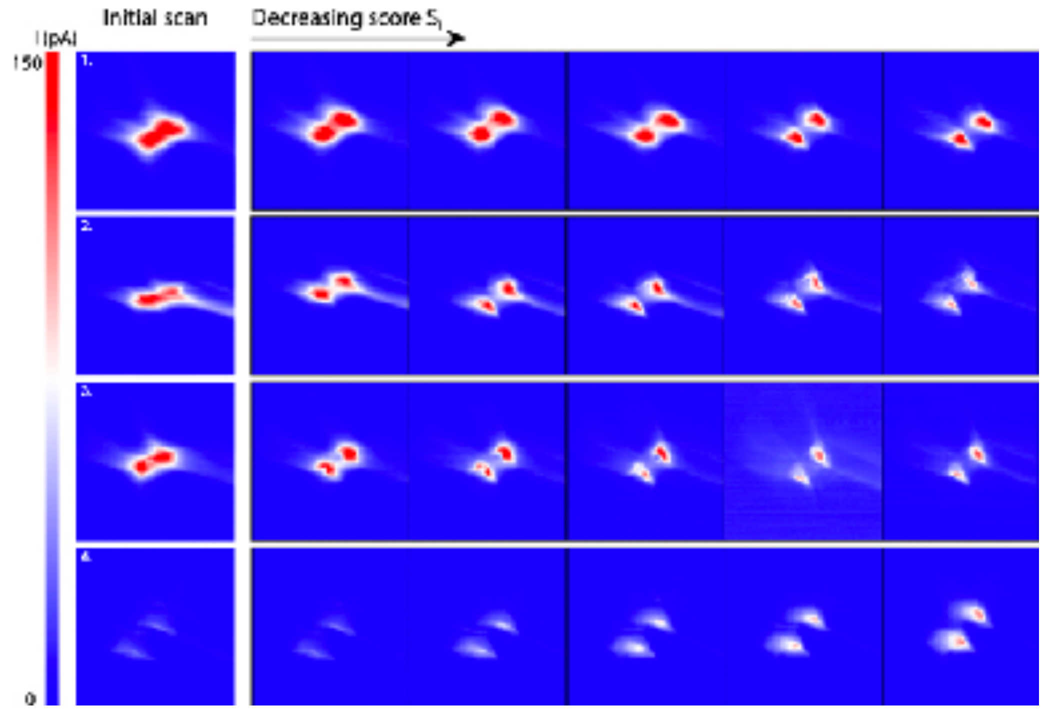
Darulová et al., Phys. Rev. Applied 13, 054005 (2020)
Moon et al., Nature Comms 11, 4161 (2020)



Charge State Calibration

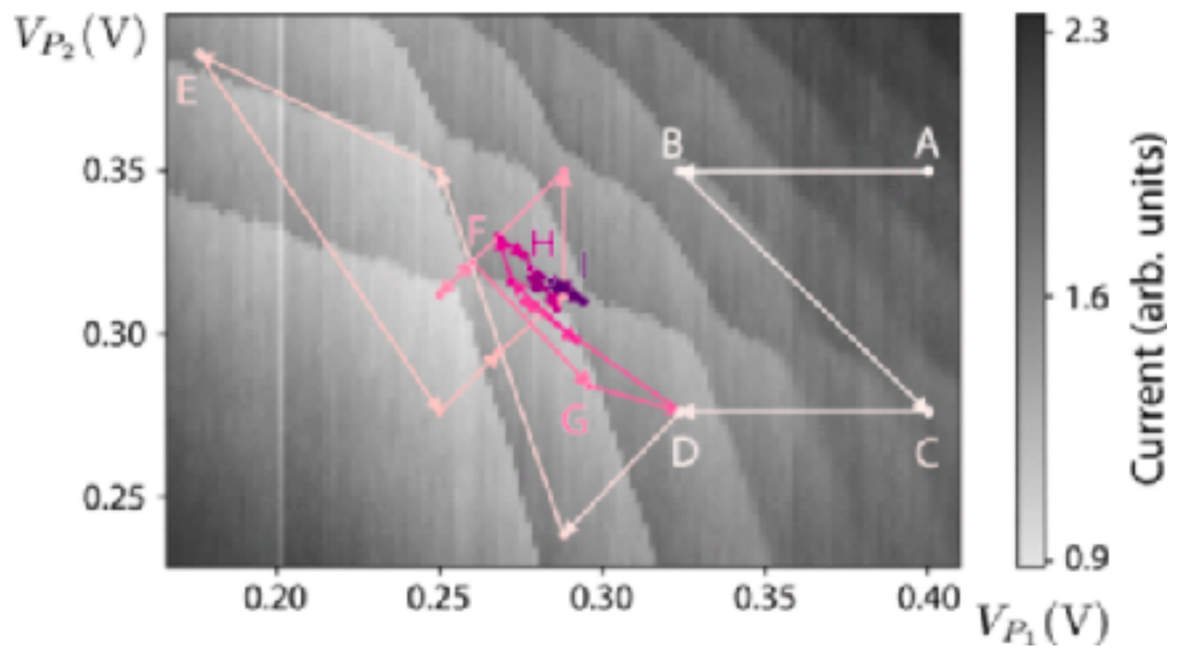
Baart et al., Appl. Phys. Lett. 108, 213104 (2016)
Durrer et al., Phys. Rev. Applied 13, 054019 (2020)





Auto-Tuning

QMAI | TU Delft

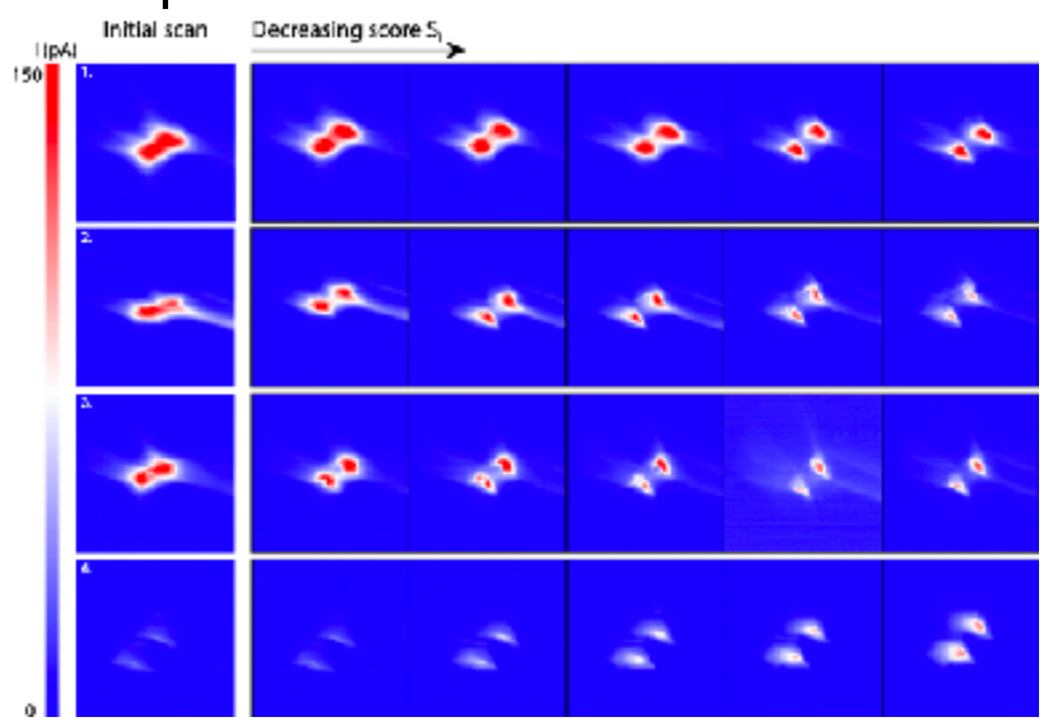


Kalentre et al., npj QI 5, 6 (2019)
Zwolak et al., Phys. Rev. Applied 13, 034075 (2020)

Quantum Dot Configuration

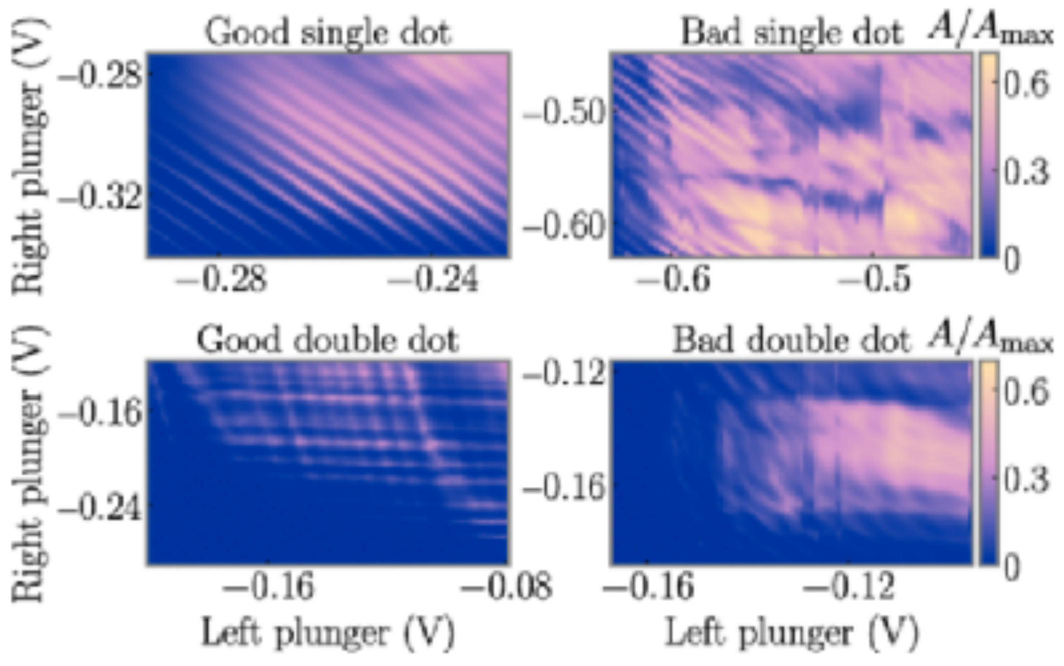
Botzem et al, Phys. Rev. Applied 10, 054026 (2018)
van Diepen et al, Appl. Phys. Lett. 113, 033101 (2018)
Teste et al, Appl. Phys. Lett. 114, 133102 (2019)
Mills et al, Appl. Phys. Lett. 115, 113501 (2019)
van Esbroeck et al, New Journal of Physics, 22, 095003 (2020)

Fine Tuning



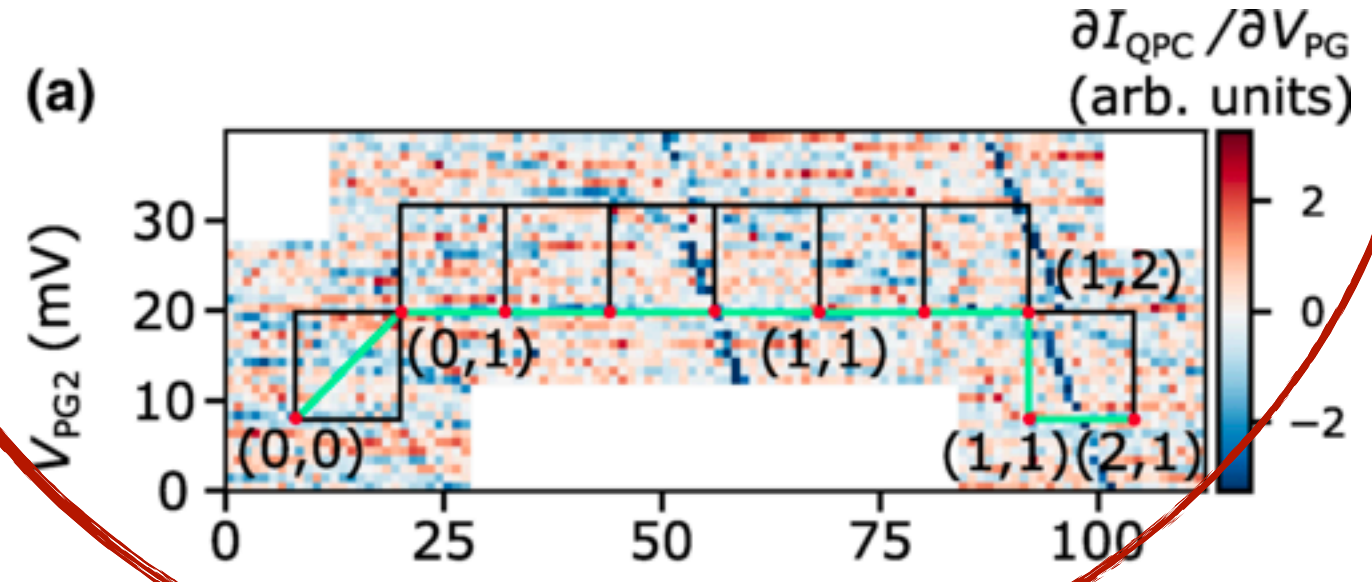
Initial tune-up

Darulová et al., Phys. Rev. Applied 13, 054005 (2020)
Moon et al., Nature Comms 11, 4161 (2020)



Charge State Calibration

Baart et al., Appl. Phys. Lett. 108, 213104 (2016)
Durrer et al., Phys. Rev. Applied 13, 054019 (2020)



Final exercise

STEP 1: Make masked groups of 4-5

STEP 2: Discuss and try to answer the following questions:

(1) What are some important practices in machine learning for:

(A) preparing data
(B) building networks
(C) training the model

(2) What are an important properties of the physics problem that call for ML solution

QMAI | TU Delft

Some exciting resources

- Fei-Fei Li, Stanford, CS231n http://cs231n.stanford.edu
- Florian Marquardt, Machine Learning for Physicists, https://machine-learning-for-physicists.org
- Kenny Choo, Eliska Greplova, Michael Denner, Mark H. Fischer, Titus Neupert: https://ml-lectures.org/
- Michael Nielsen: Neural Networks and Deep Learning http://neuralnetworksanddeeplearning.com
- just find a problem and try to solve it with ML!

QMAI | TU Delft

Would you like to give an online talk about your work?

You can register at Virtual Science Forum Speakers' Corner and pair a talk with arXiv submission/thesis defense/…

virtualscienceforum.org