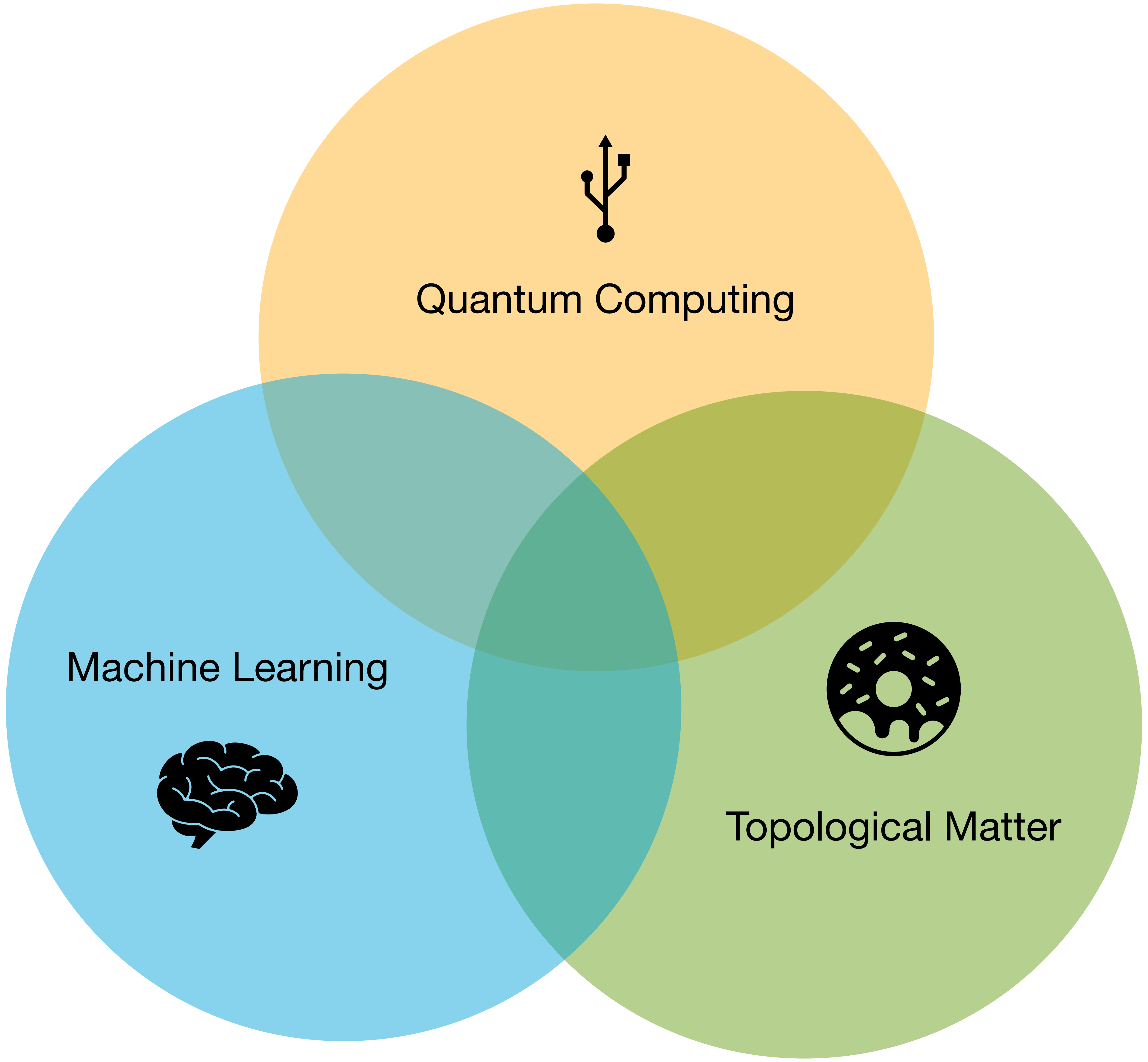
Engineered Topological Matter and Machine Learning for Quantum Applications

Eliška Greplová

QMAI | TU Delft

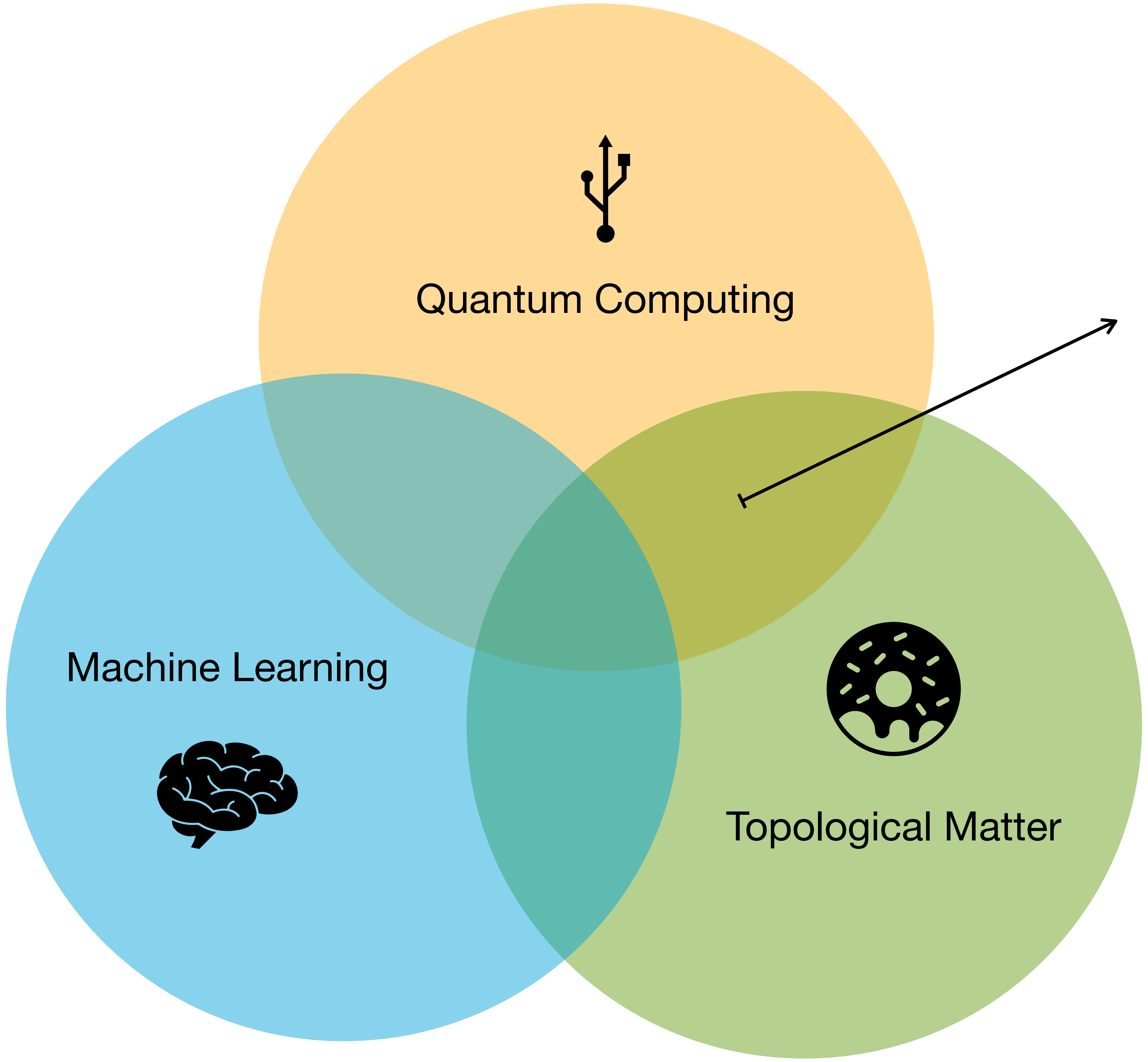
Topological Matter School '22

QMAI | TU Delft



Topological Matter School ‘22

QMAI | TU Delft

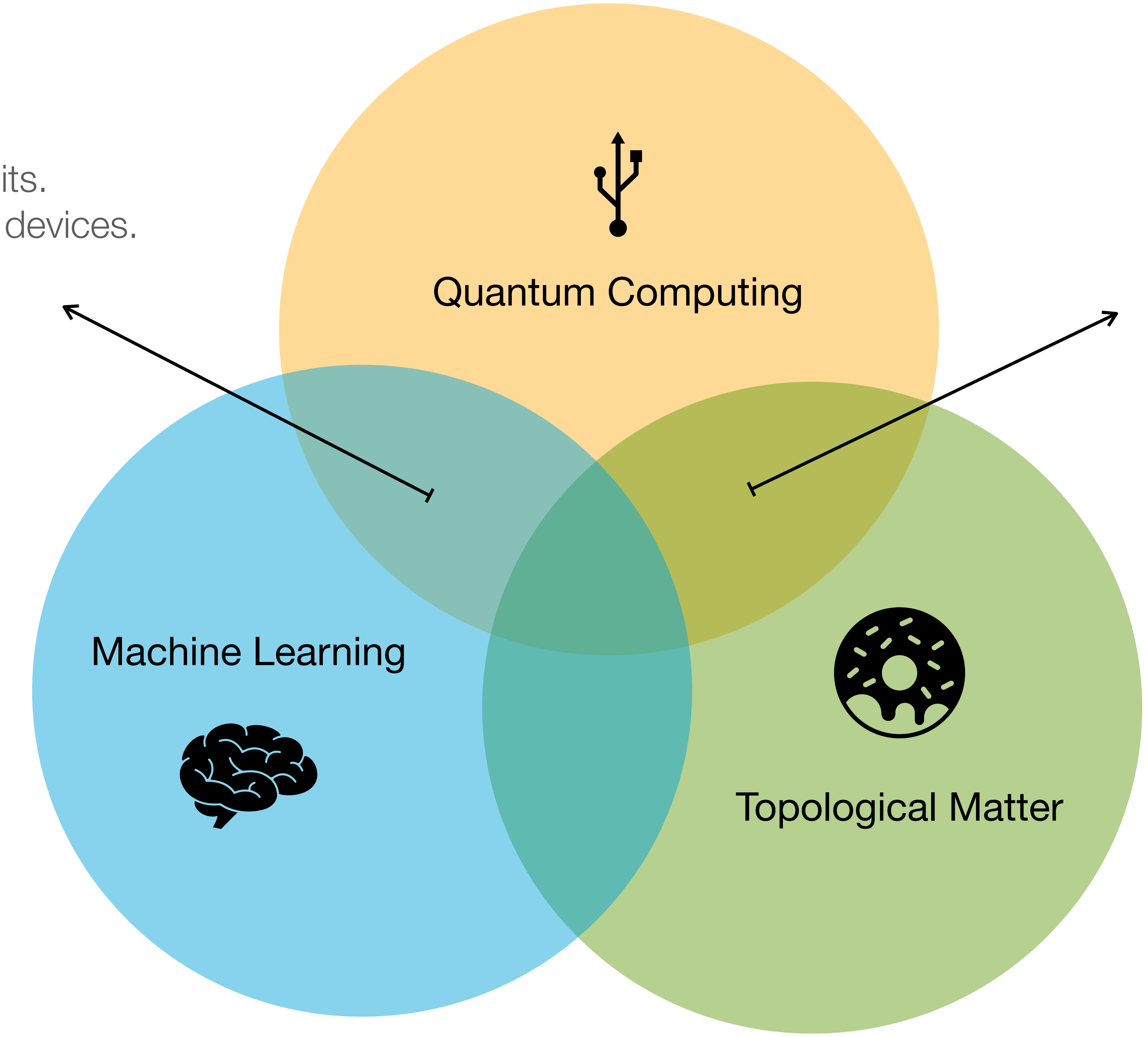


Use topological excitations for quantum computing taking advantage of protection against errors.

QMAI | TU Delft

Topological Matter School '22

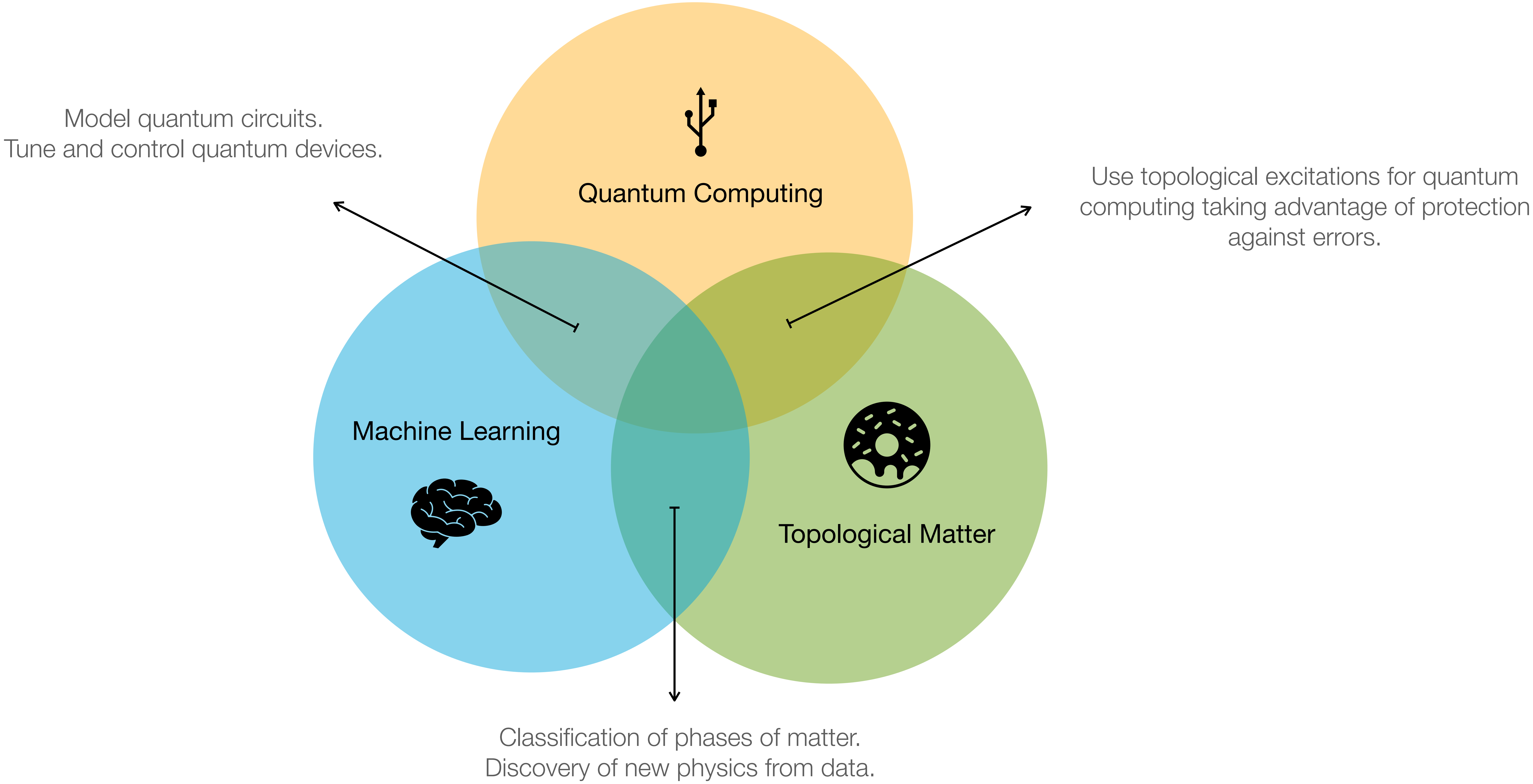
Model quantum circuits.
Tune and control quantum devices.



Use topological excitations for quantum computing taking advantage of protection against errors.

QMAI | TU Delft

Topological Matter School '22

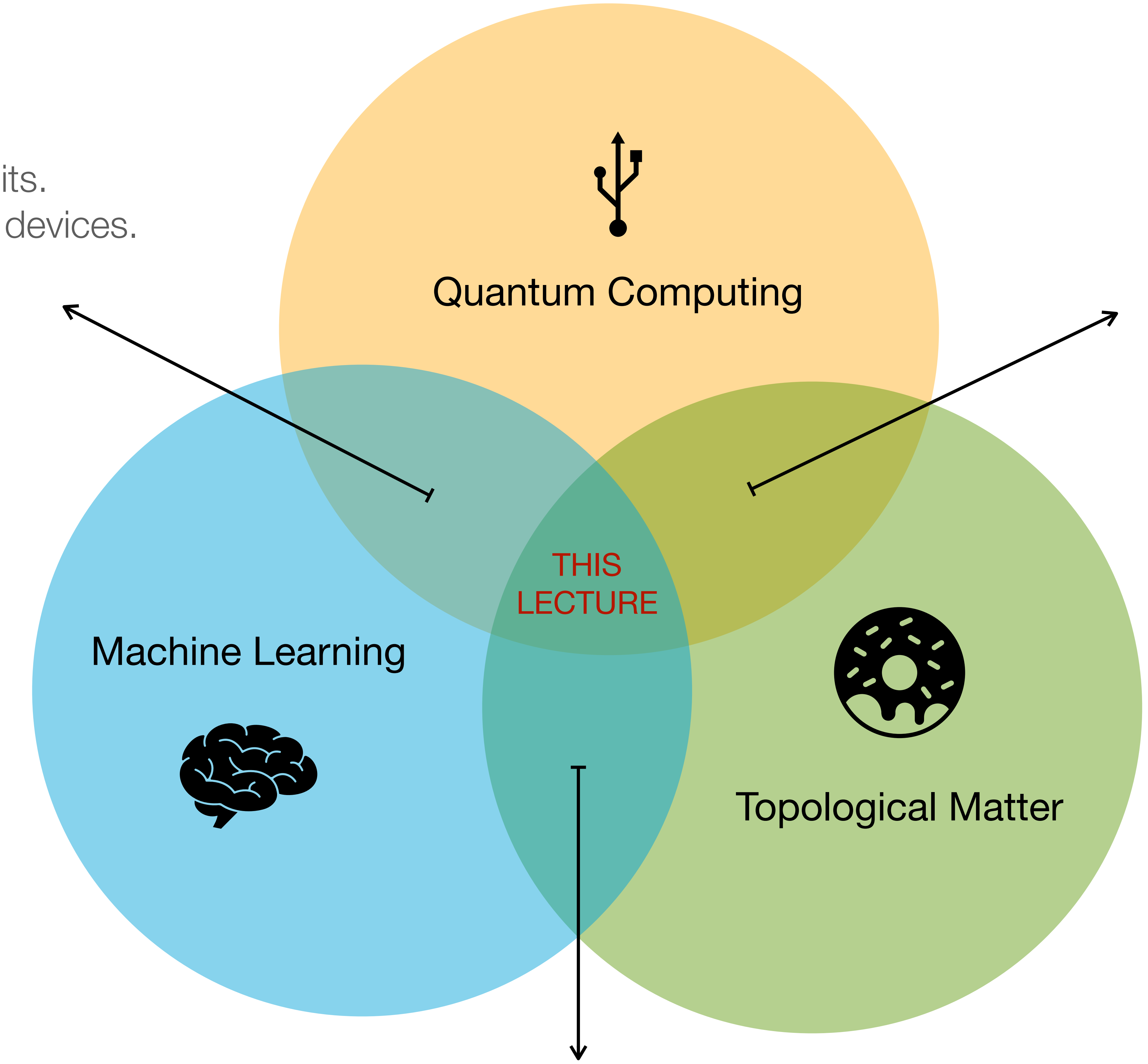


Topological Matter School '22

QMAI | TU Delft

Model quantum circuits.
Tune and control quantum devices.

Use topological excitations for quantum computing taking advantage of protection against errors.

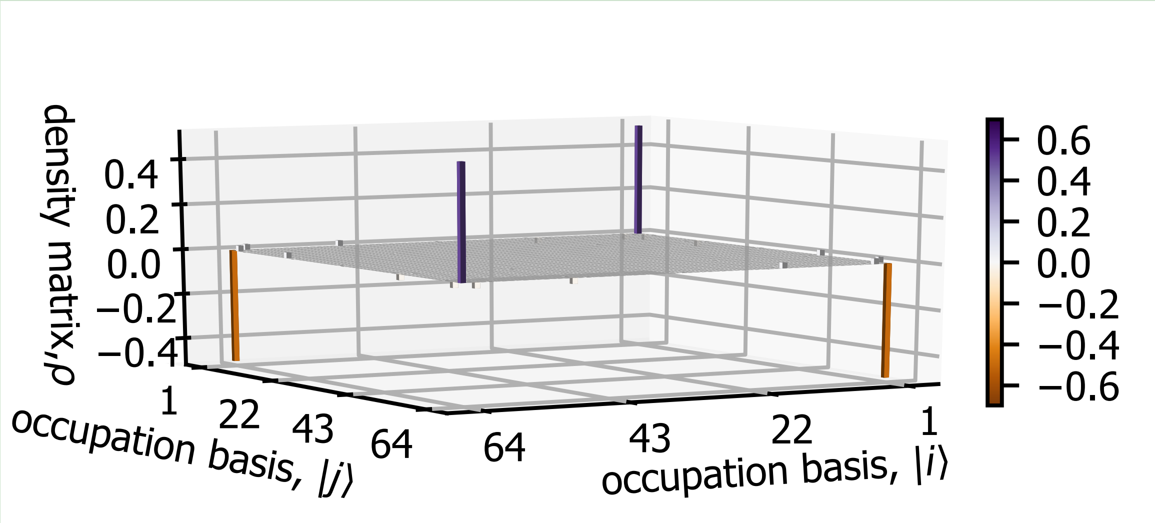


Classification of phases of matter.
Discovery of new physics from data.

QMAI | TU Delft

Motivation: Topological Metamaterials

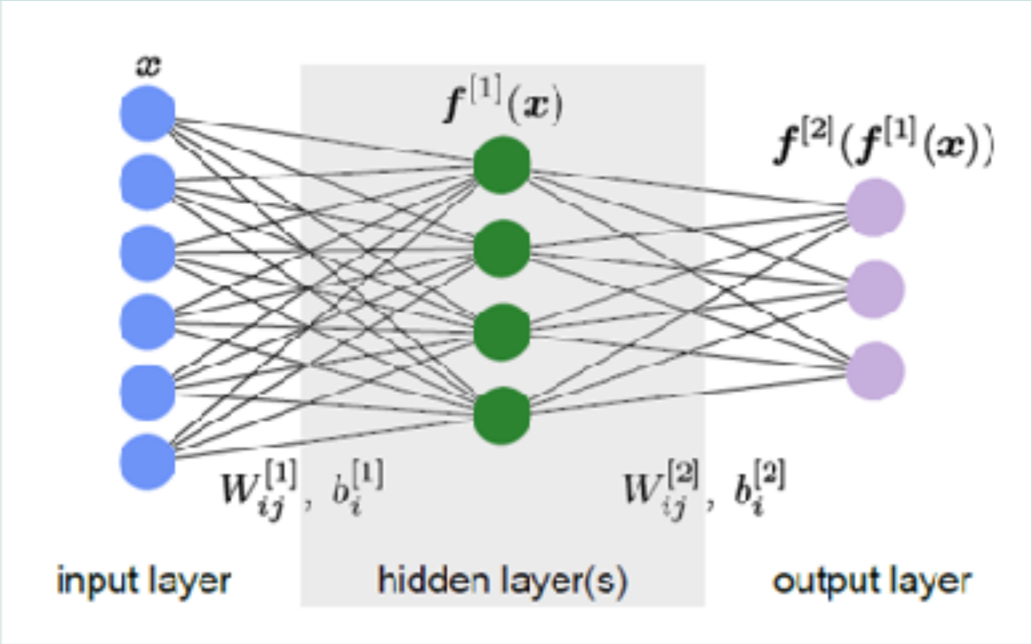
Stabilizing Entanglement in Superconducting Circuits Using Topology



arXiv:2205.09100

ML Primer: Neural Networks Introduction

Learning Physics from Data: Why and How to Get Started



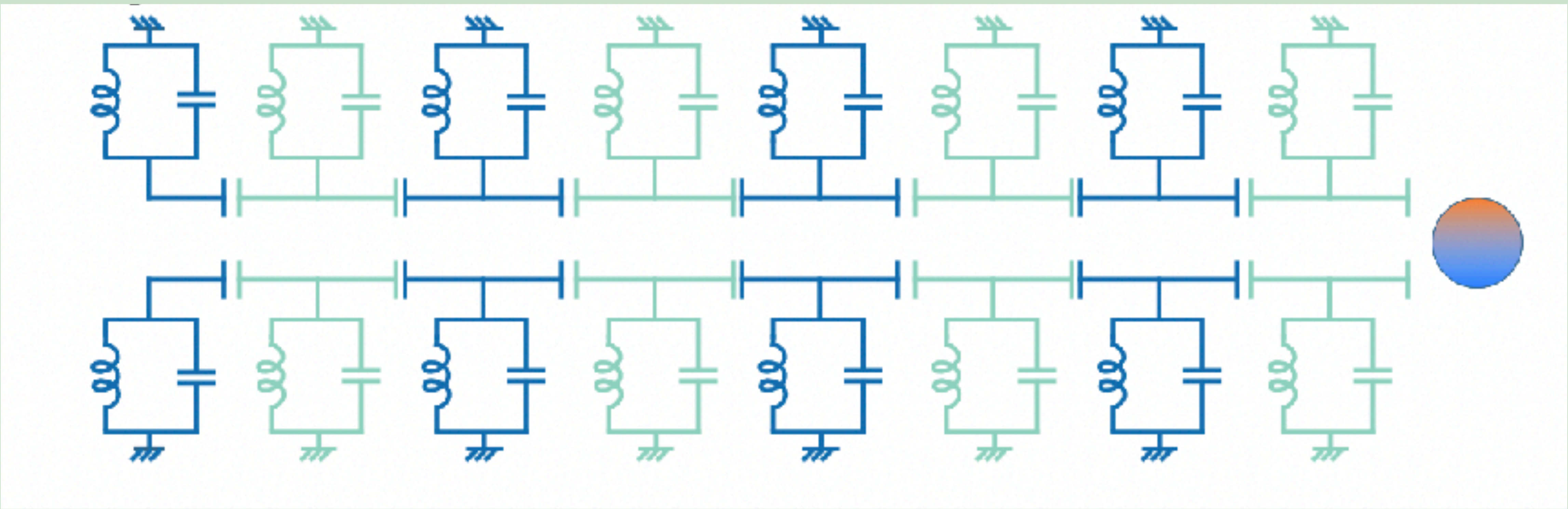
arXiv:2102.04883

Hands-On: ML in Q-Computing Research

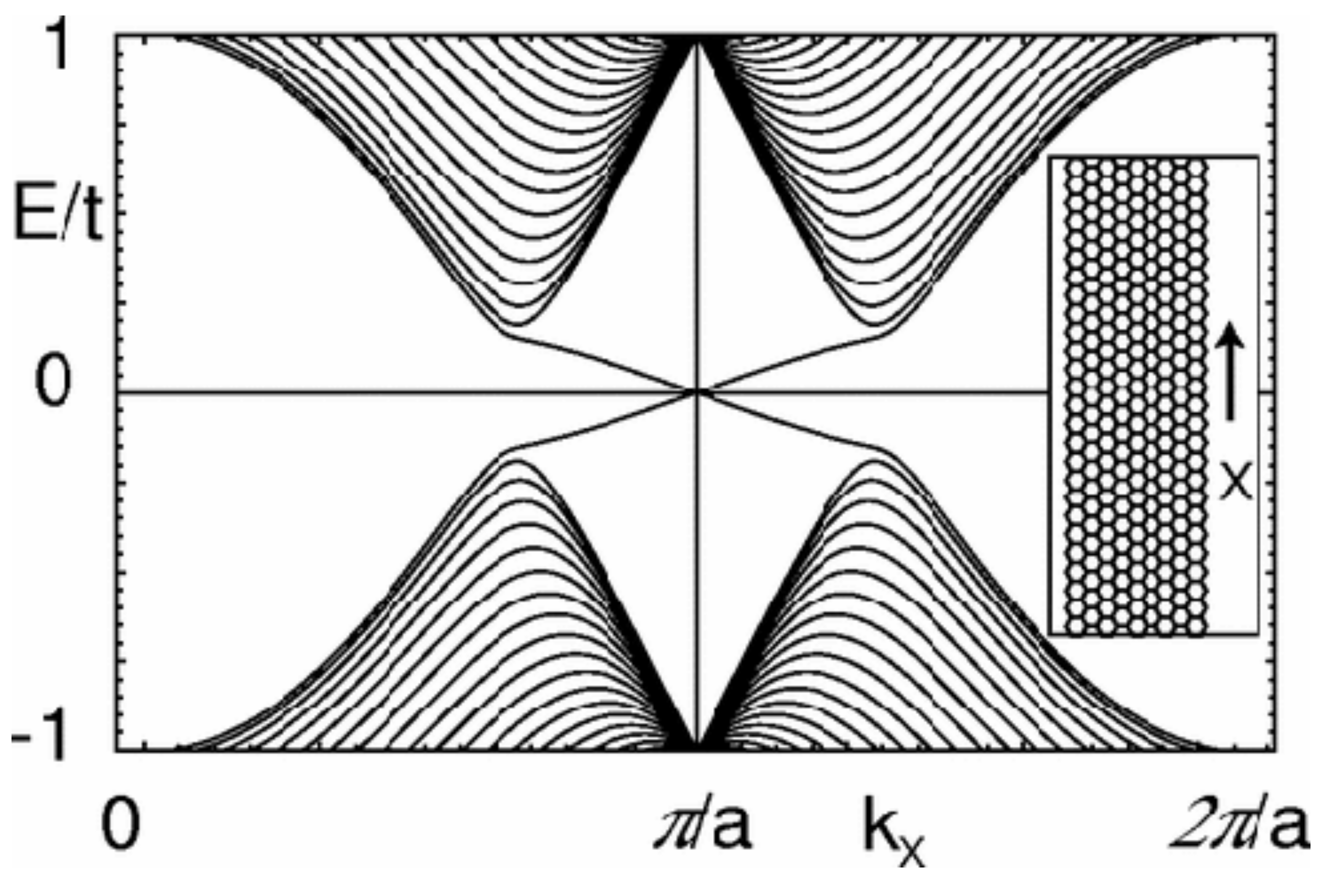
Writing Code to Classify Topological Phases in Superconducting Circuits

PyTorch

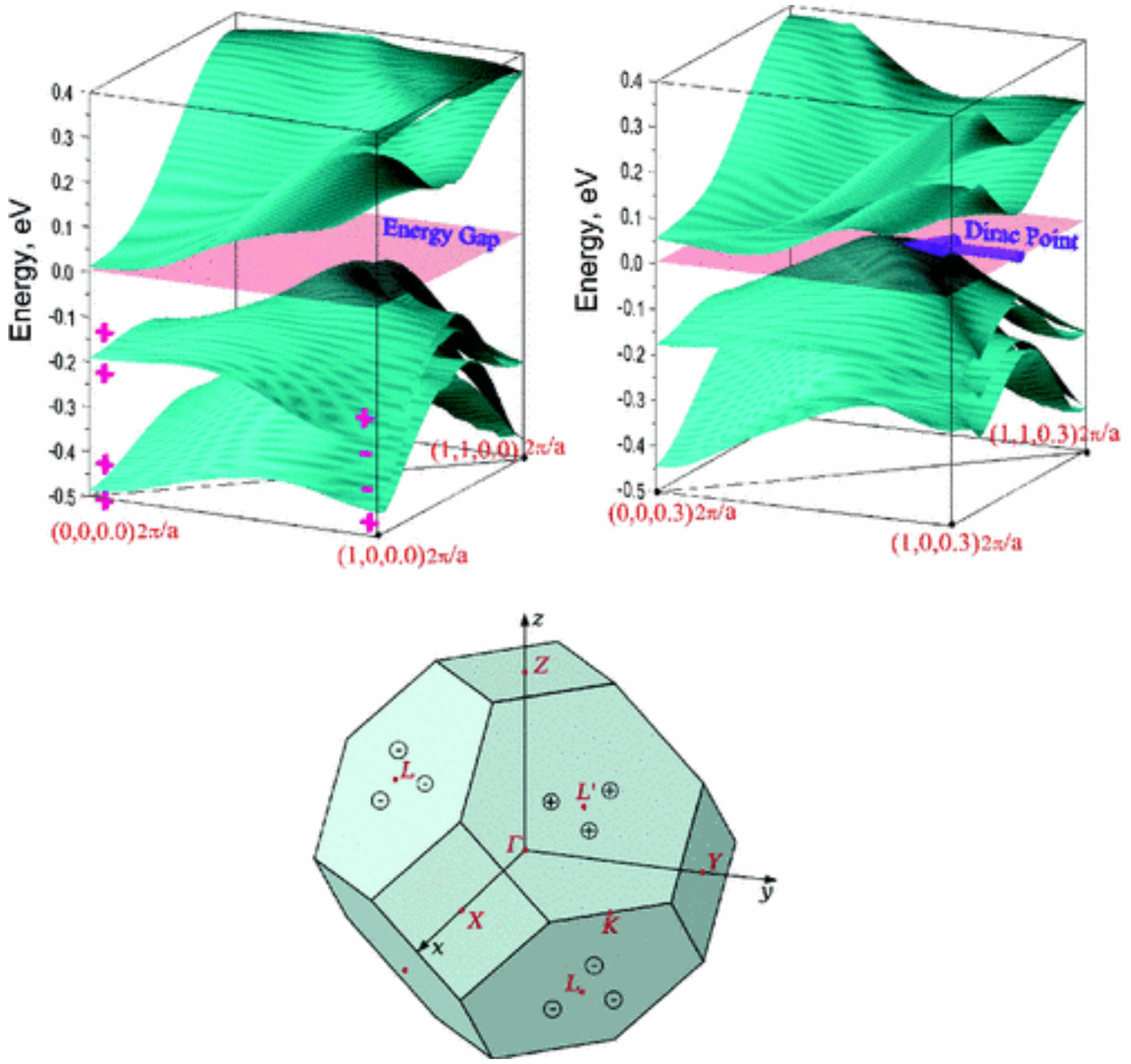
Motivation:
Quantum Topological Metamaterials with Superconducting Circuits



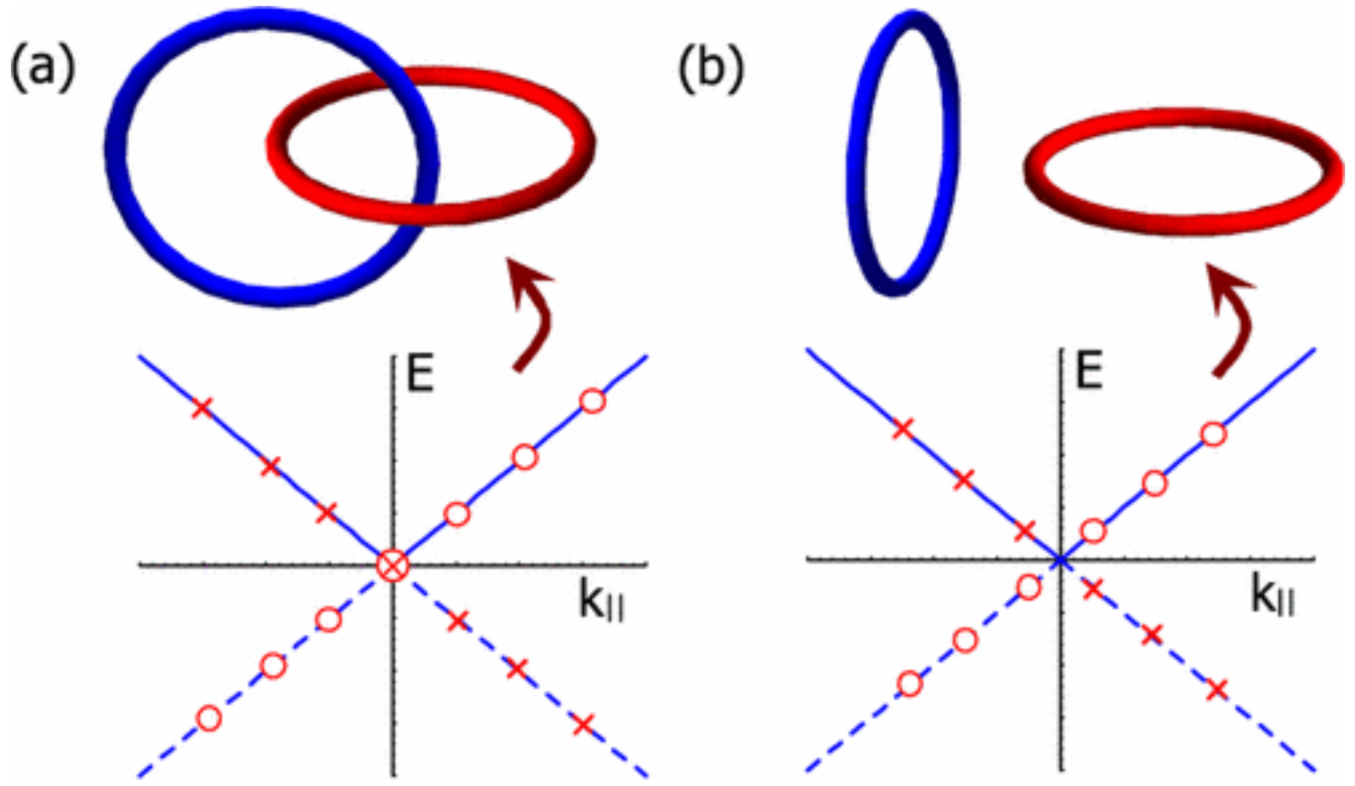
Topology and Condensed Matter



C. L. Kane and E. J. Mele Phys. Rev. Lett. **95**, 226801

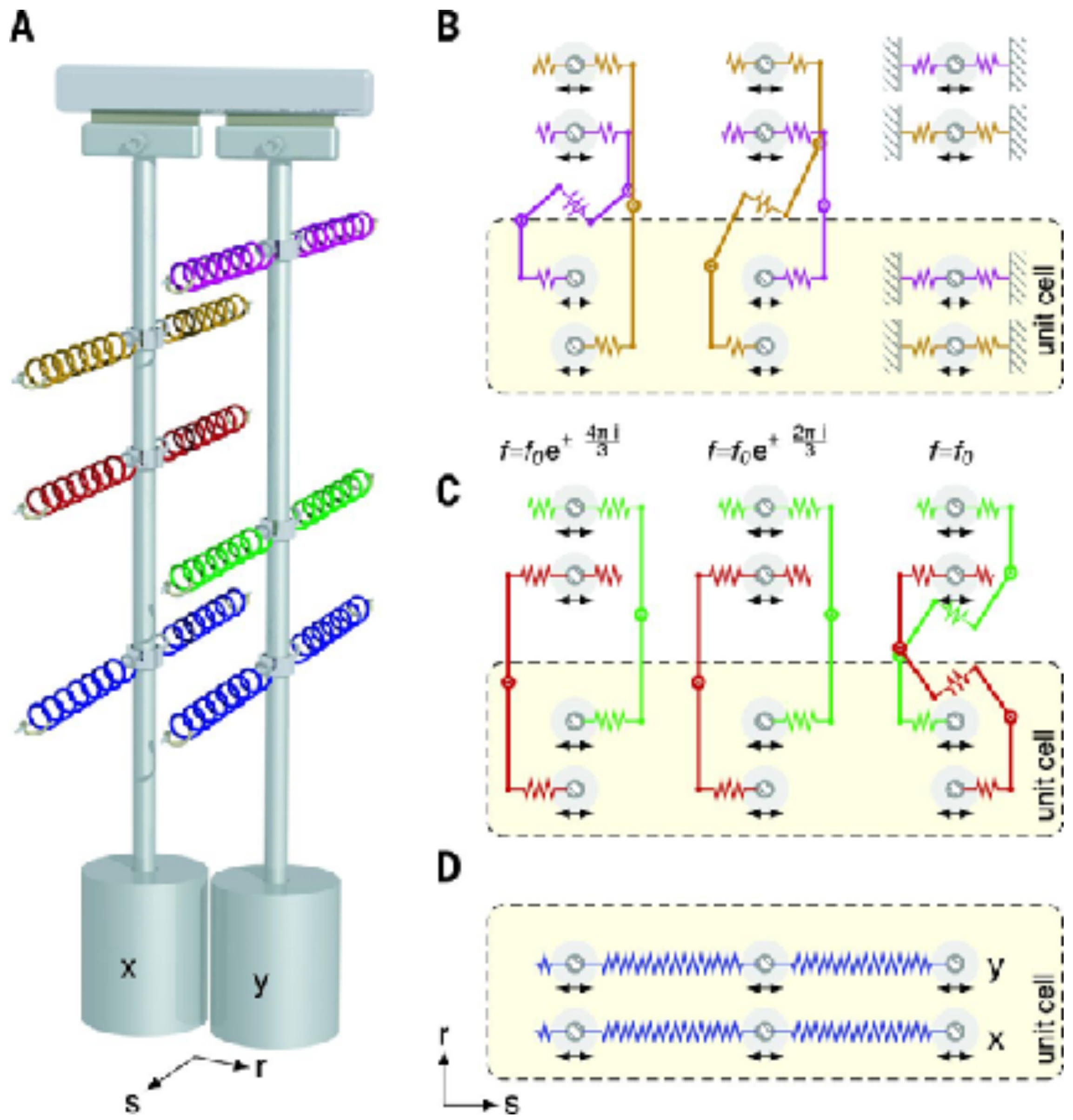


Xiangang Wan, Ari M. Turner, Ashvin Vishwanath, and Sergey Y. Savrasov Phys. Rev. B **83**, 205101

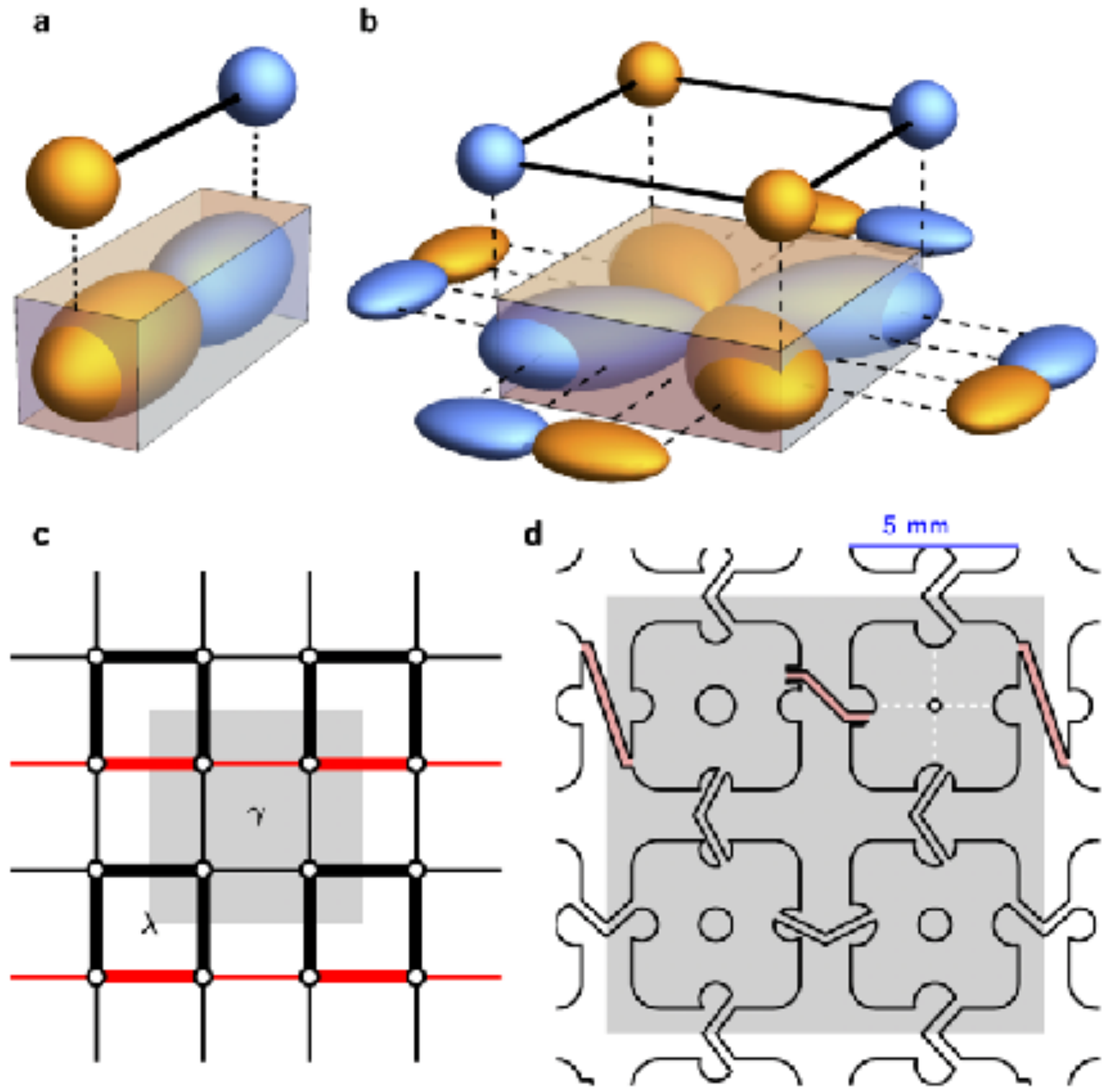


Xiao-Liang Qi, Taylor L. Hughes, S. Raghu, and Shou-Cheng Zhang Phys. Rev. Lett. **102**, 187001

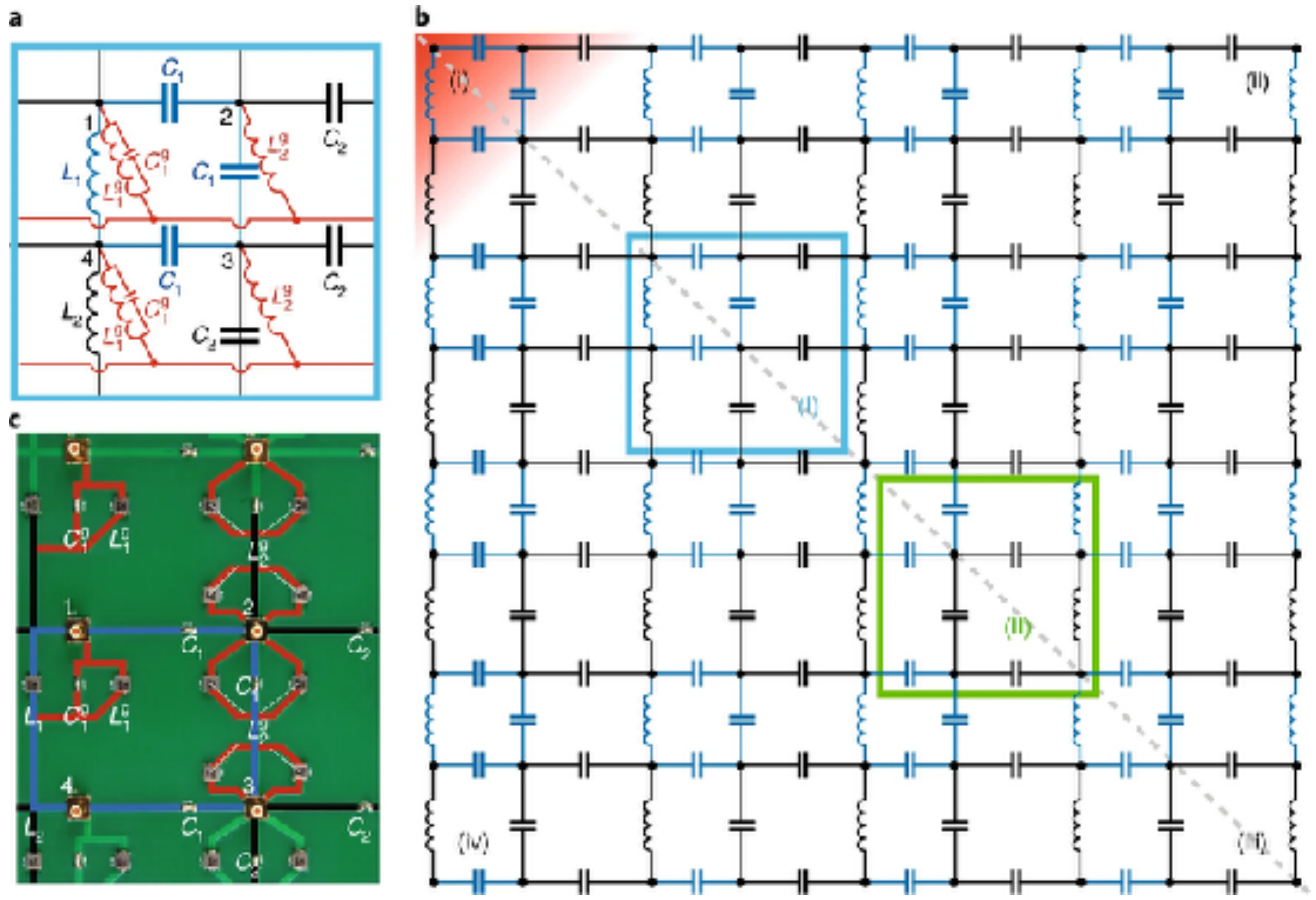
Classical Topological Materials



Süsstrunk, Roman, and Sebastian D. Huber. "Observation of phononic helical edge states in a mechanical topological insulator." *Science* 349.6243 (2015): 47-50.



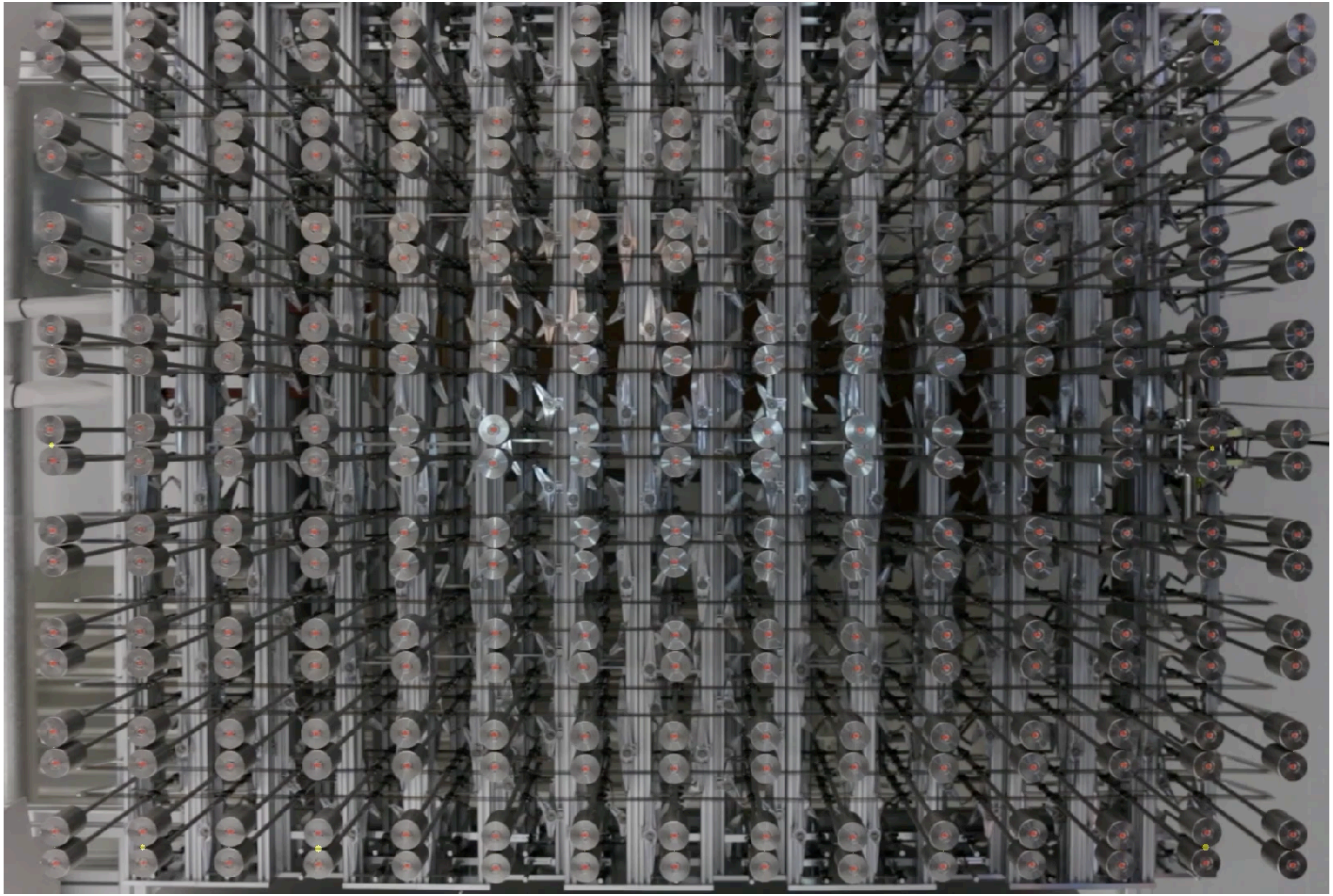
Serra-Garcia, Marc, et al. "Observation of a phononic quadrupole topological insulator." *Nature* 555.7696 (2018): 342-345.



Imhof, Stefan, et al. "Topolectrical-circuit realization of topological corner modes." *Nature Physics* 14.9 (2018): 925-929.

Topological Metamaterials

QMAI | TU Delft



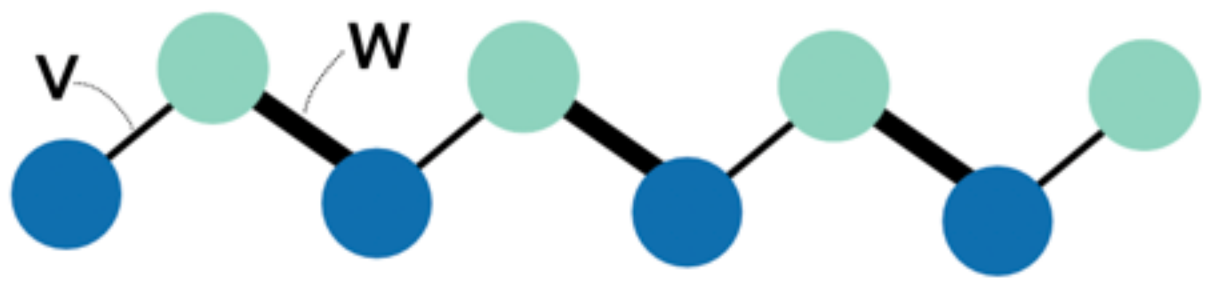
Observation of phononic helical edge states in a mechanical topological insulator

ROMAN SÜSSTRUNK AND SEBASTIAN D. HUBER

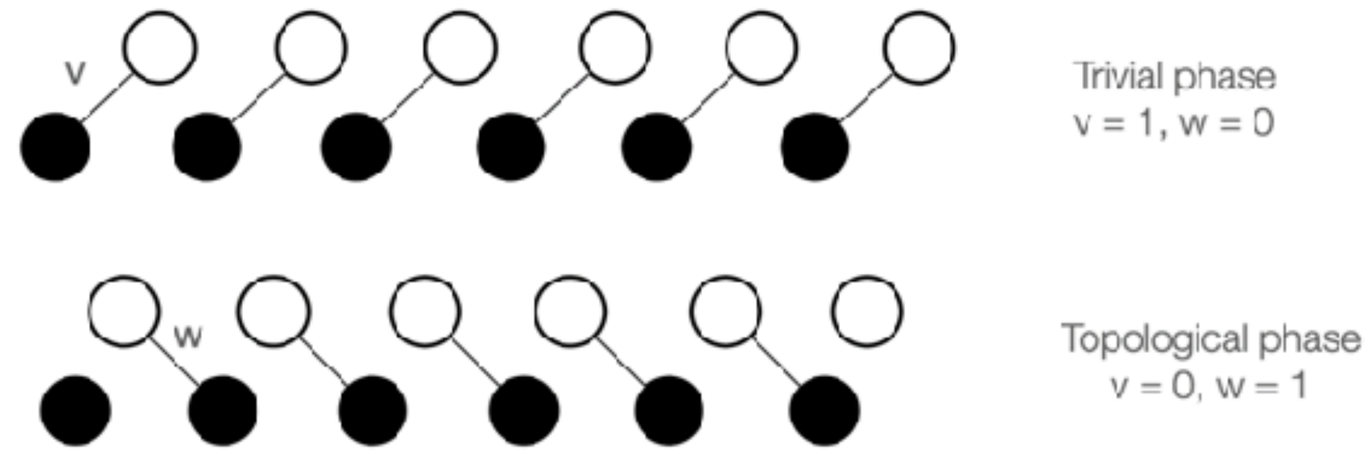
SCIENCE • 3 Jul 2015 • Vol 349, Issue 6243

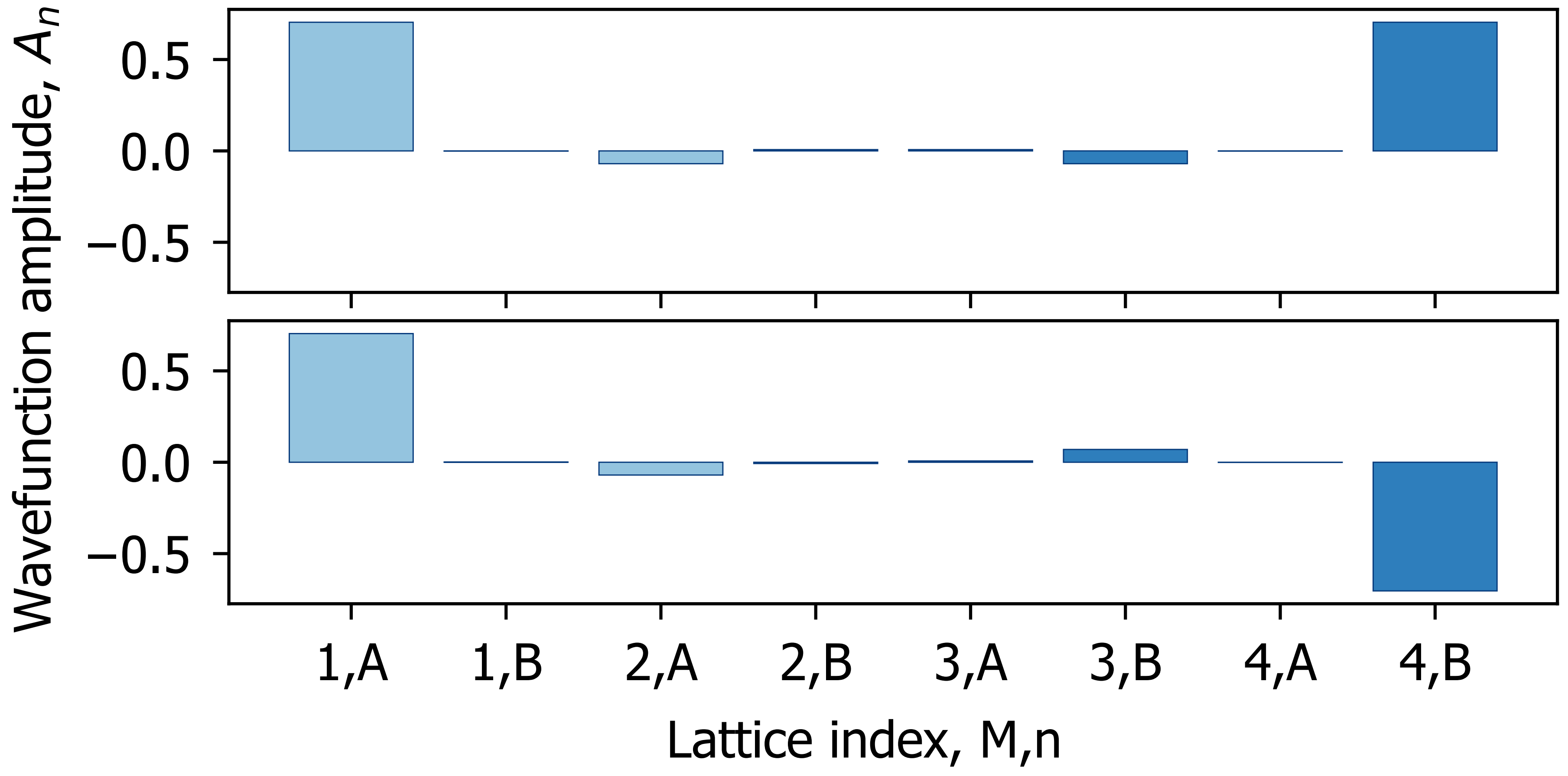
QMAI | TU Delft

Example: SSH model

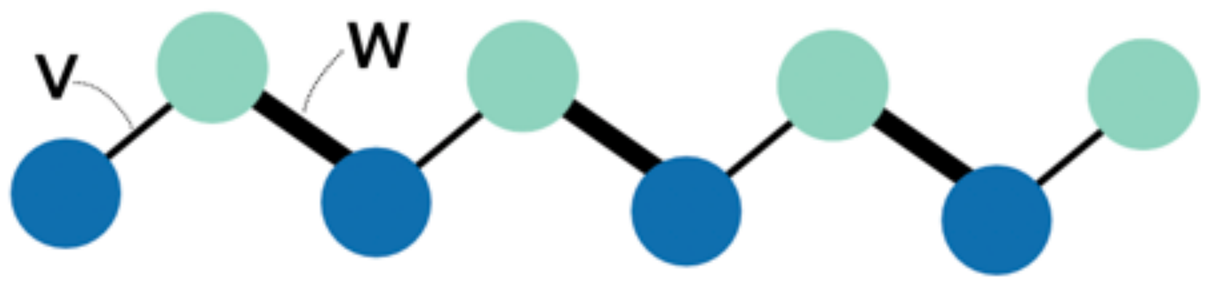


$$H_{SSH} = v\sum_{m=1}^{N}(|m,B\rangle\langle m,A| + h.c.) + w\sum_{m=1}^{N-1}(|m+1,A\rangle\langle m,B| + h.c.),$$

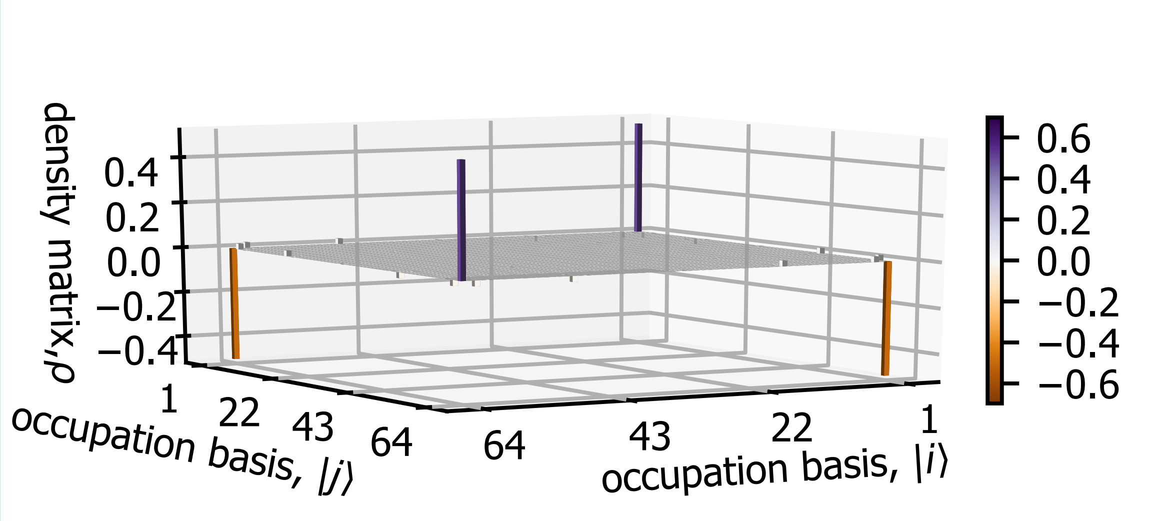




SSH model in this lecture

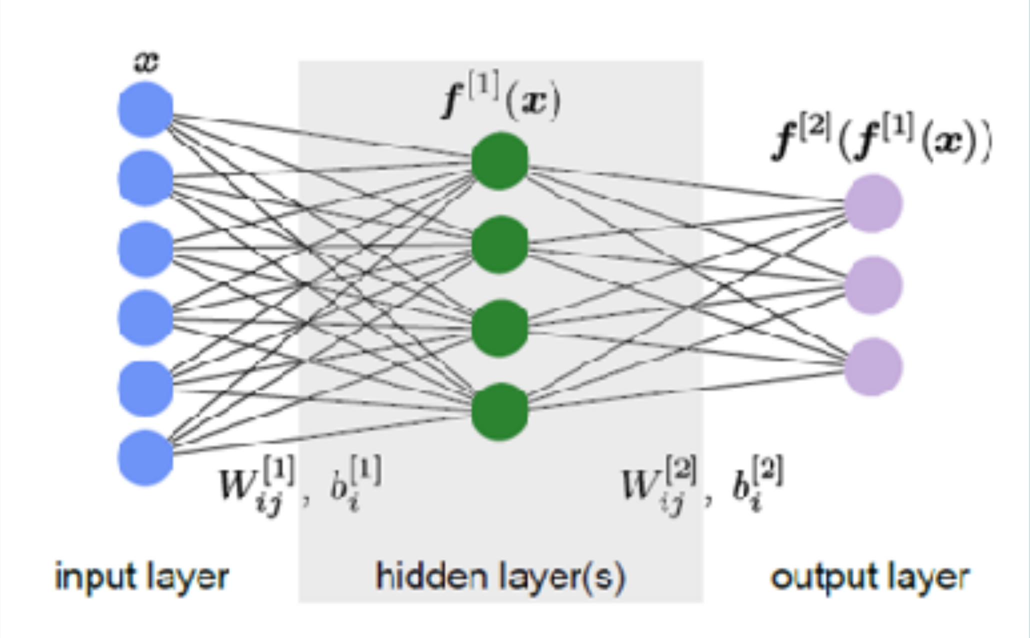


SSH in QMAI research



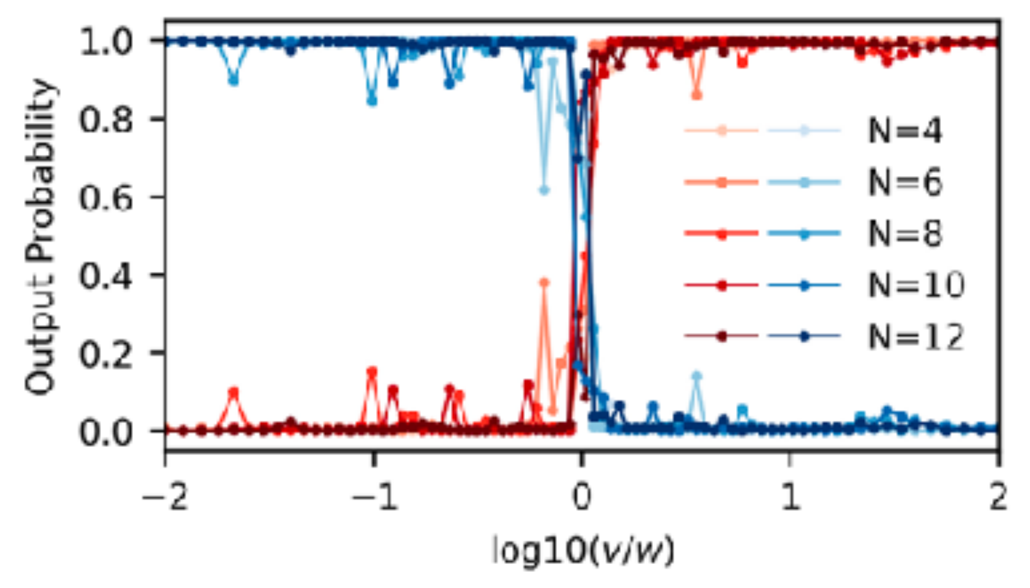
arXiv:2205.09100

ML generic intro

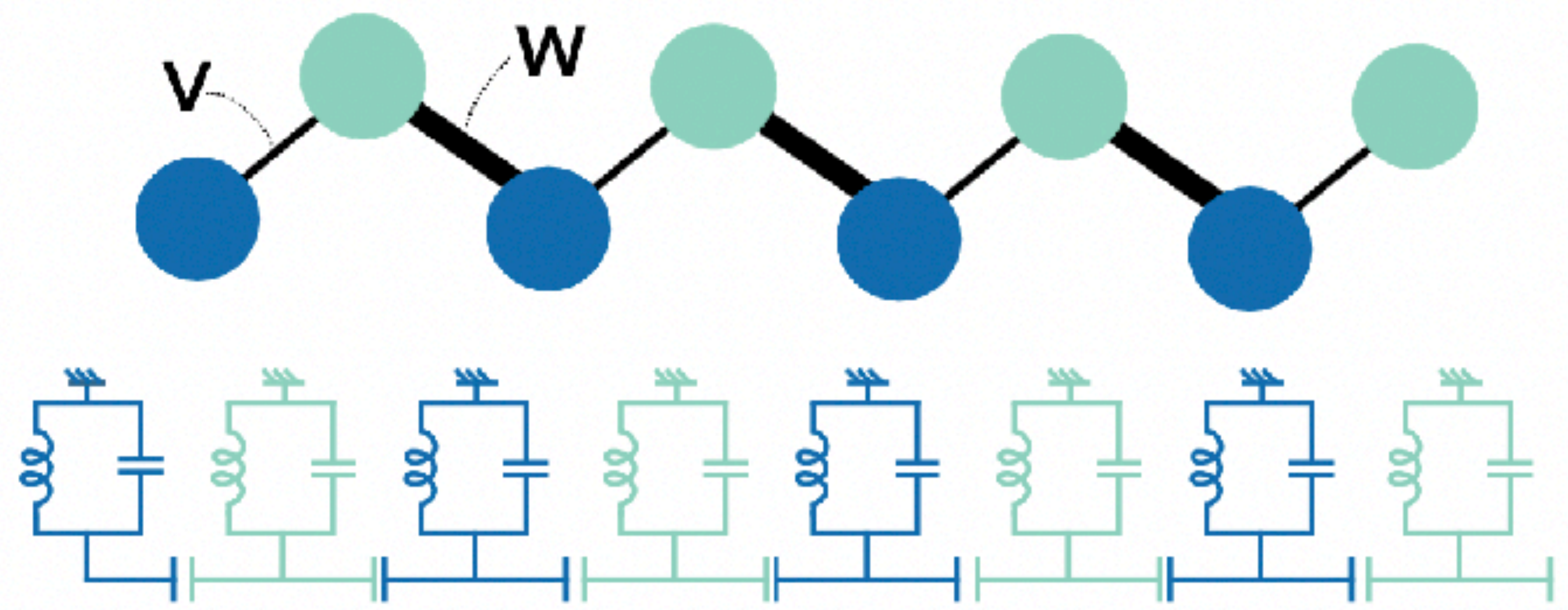


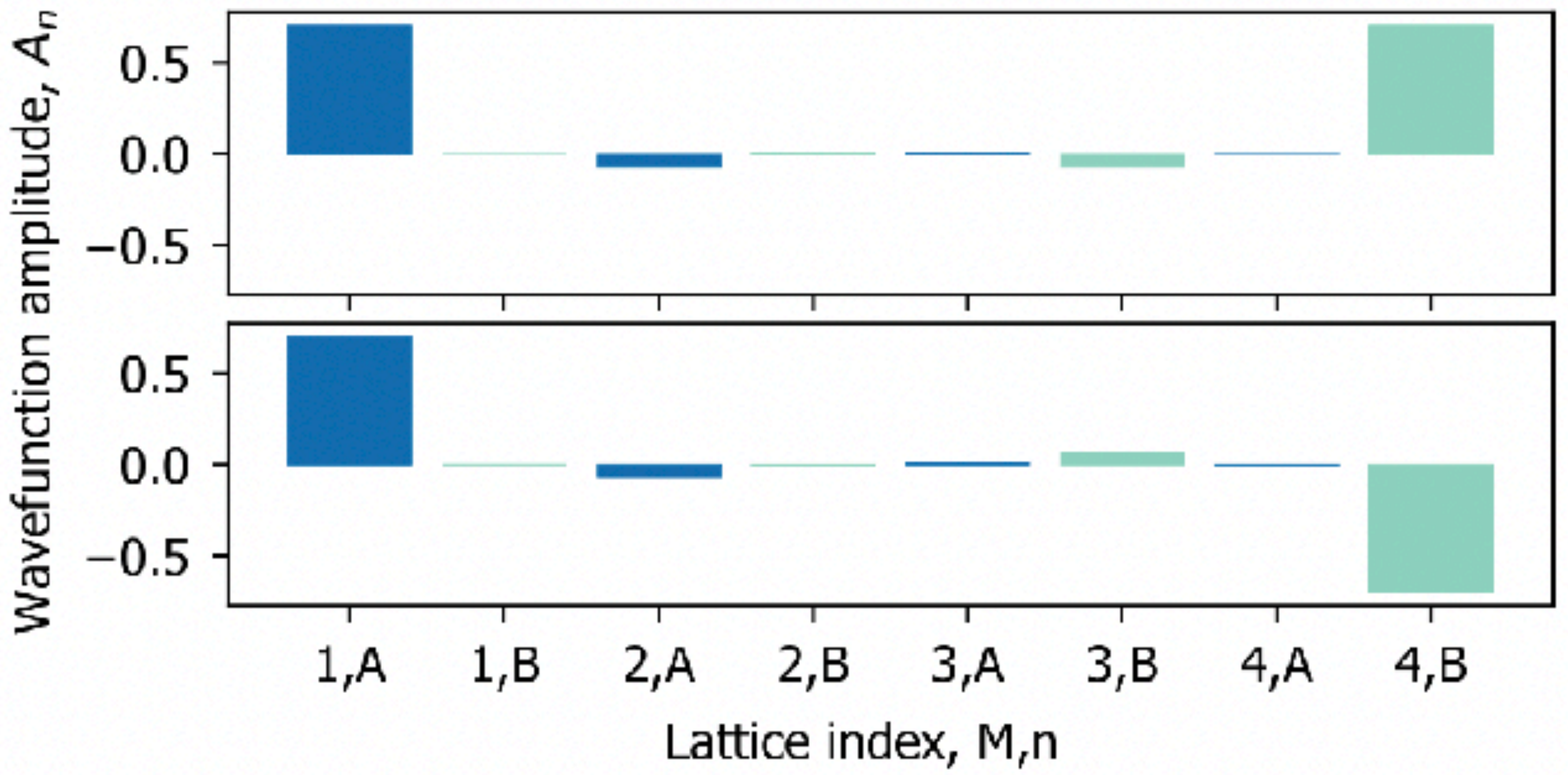
arXiv:2102.04883

ML applied on topological SSH states



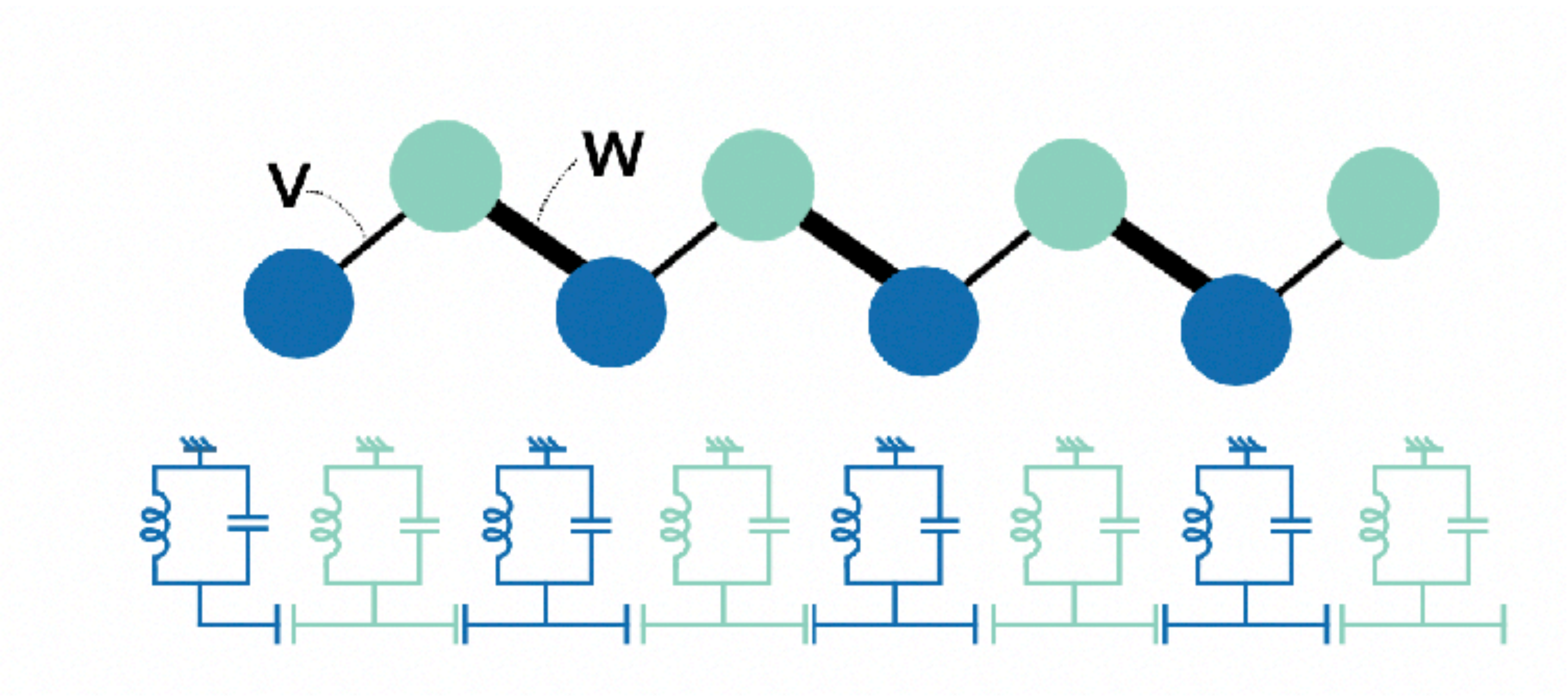
SSH model with superconducting resonators

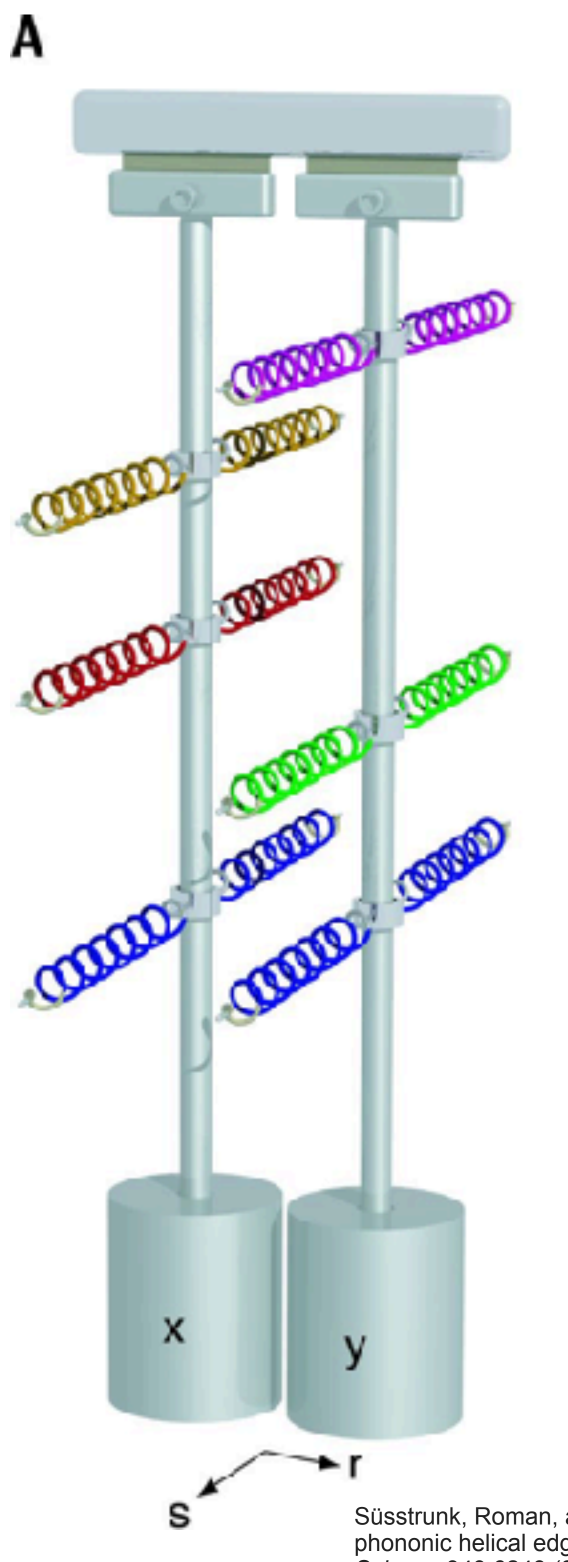




QMAI | TU Delft

Can we explore intrinsically quantum features?

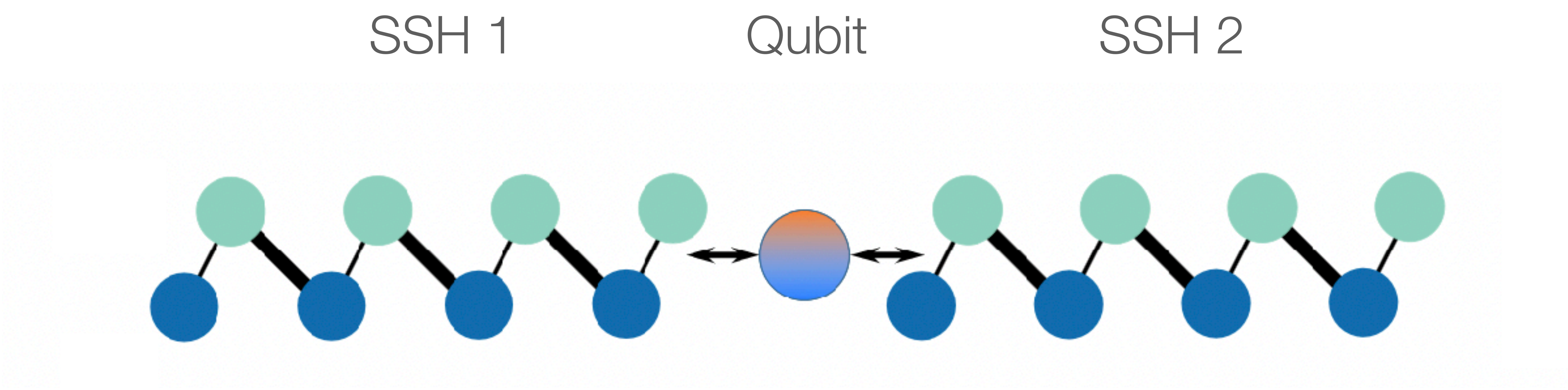




Süsstrunk, Roman, and Sebastian D. Huber. "Observation of phononic helical edge states in a mechanical topological insulator." *Science* 349.6243 (2015): 47-50.

Guliuxin Jin, **EG**: arXiv:2205.09100 (2022)

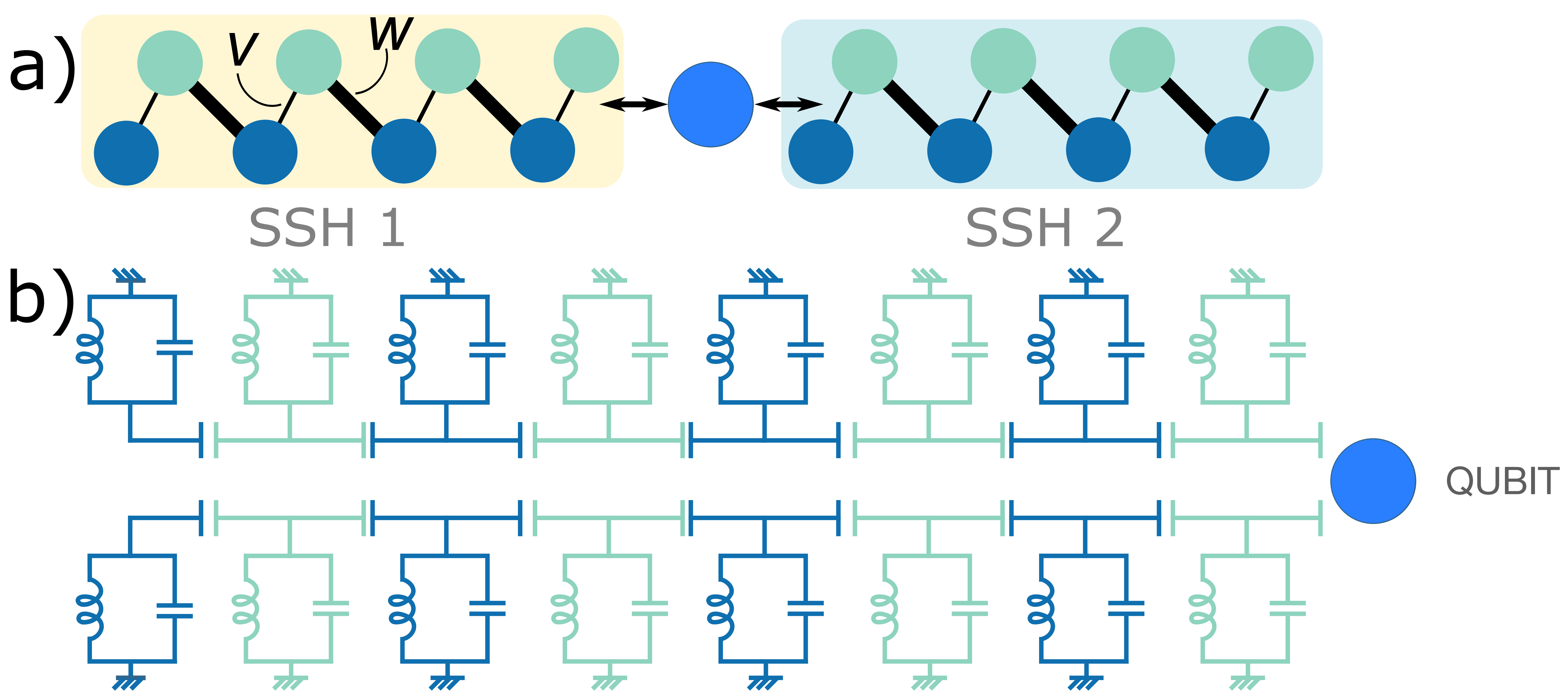
Generating entanglement of the topological modes



Guliuxin Jin, **EG**: arXiv:2205.09100 (2022)

QMAI | TU Delft

Generating entanglement of the topological modes

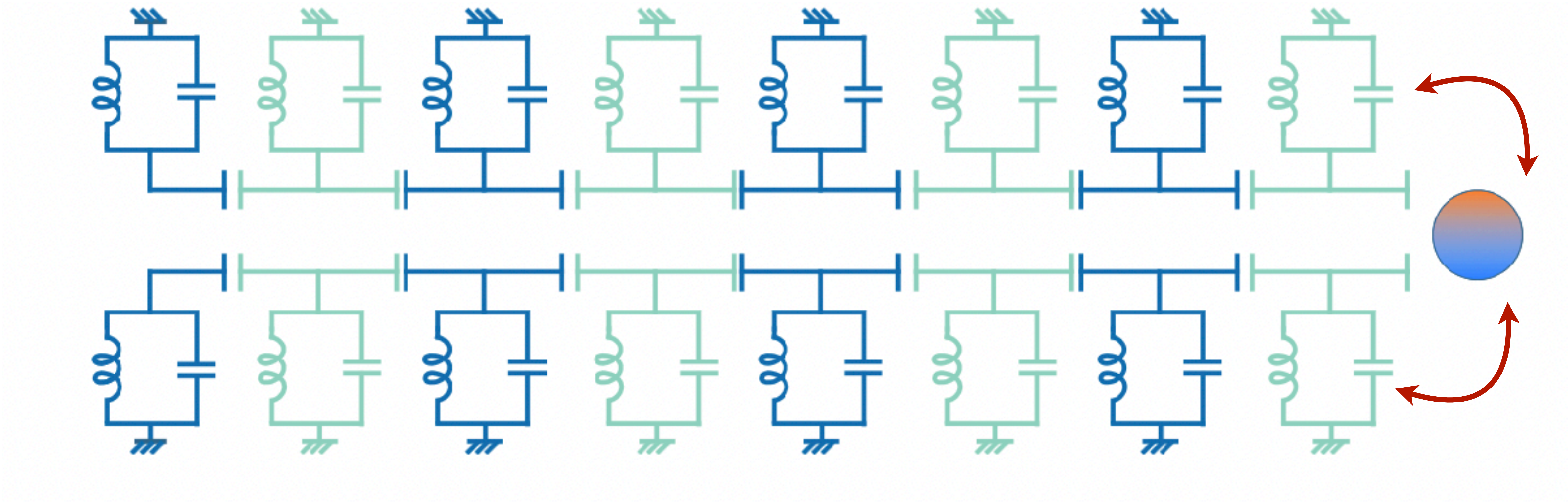


Guliuxin Jin, **EG**: arXiv:2205.09100 (2022)

Entangling topo modes: Protocol

QMAI | TU Delft

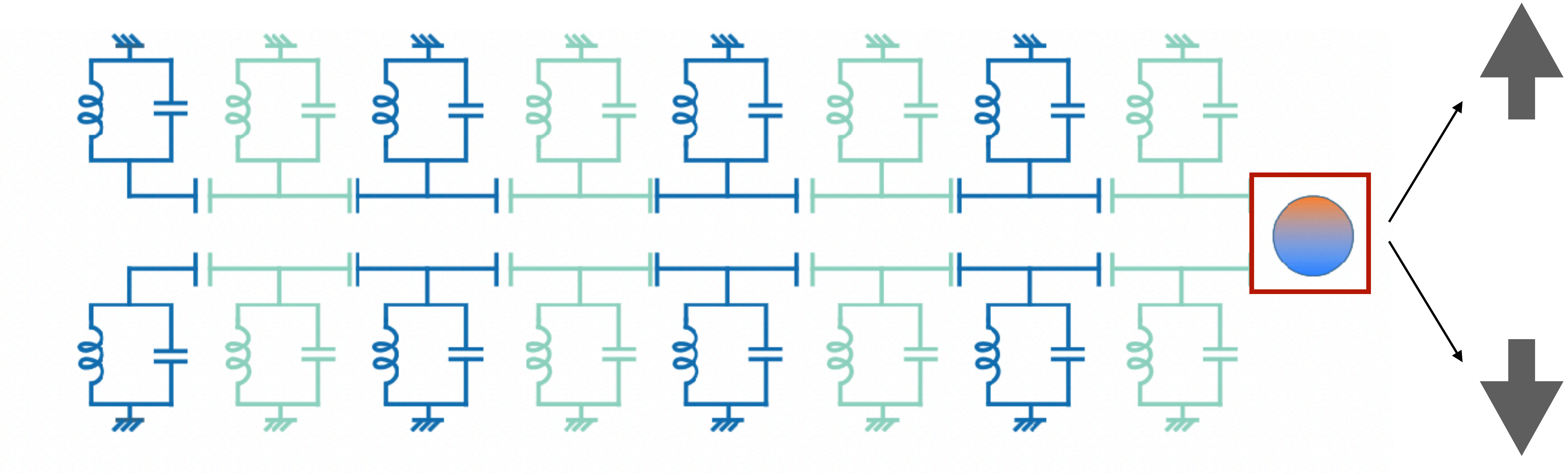
Step 1: Couple



Guliuxin Jin, **EG**: arXiv:2205.09100 (2022)

Entangling topo modes: Protocol

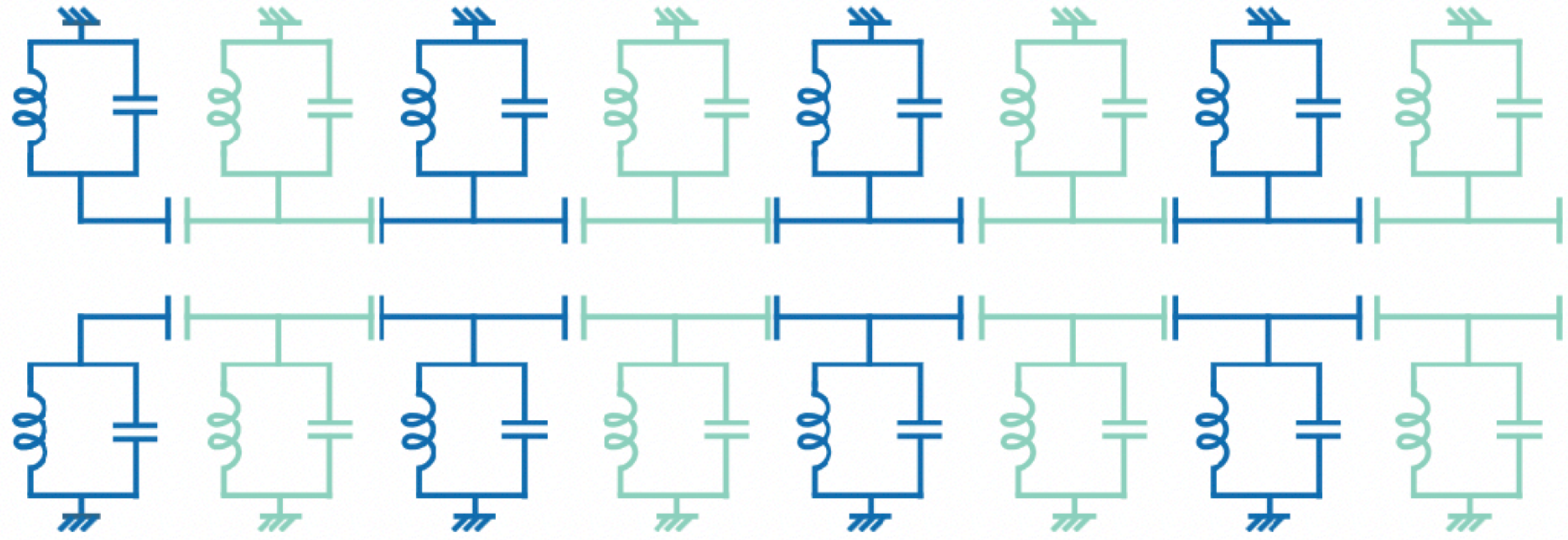
Step 2: Measure Qubit

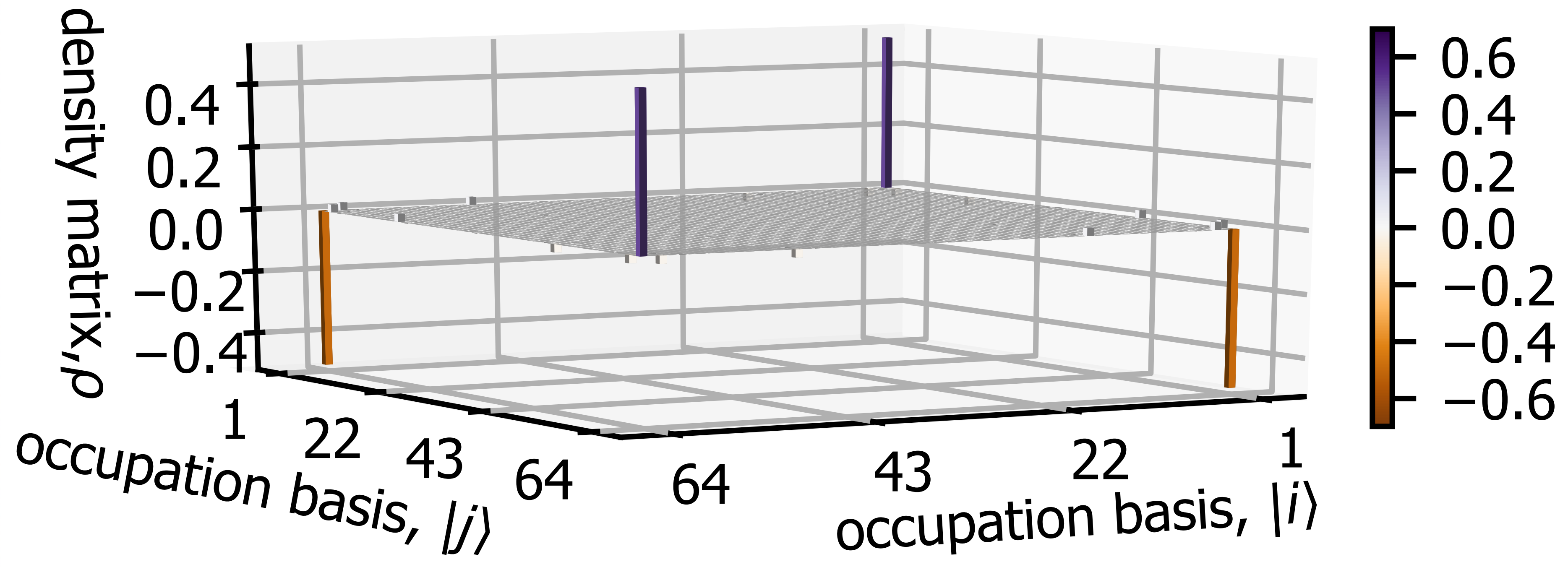


Guliuxin Jin, **EG**: arXiv:2205.09100 (2022)

Entangling topo modes: Protocol

Step 3: Maximally entangled state





Guliuxin Jin, **EG**: arXiv:2205.09100 (2022)

Does topology help with the entanglement stability?

Entanglement Measure: Negativity

$$\mathcal{N}(\rho) = \left| \sum_{\lambda_i < 0} \lambda_i \right| = \sum_i \frac{|\lambda_i| - \lambda_i}{2},$$

Eigenvalues of the partially transposed density matrix

Guliuxin Jin, **EG**: arXiv:2205.09100 (2022)

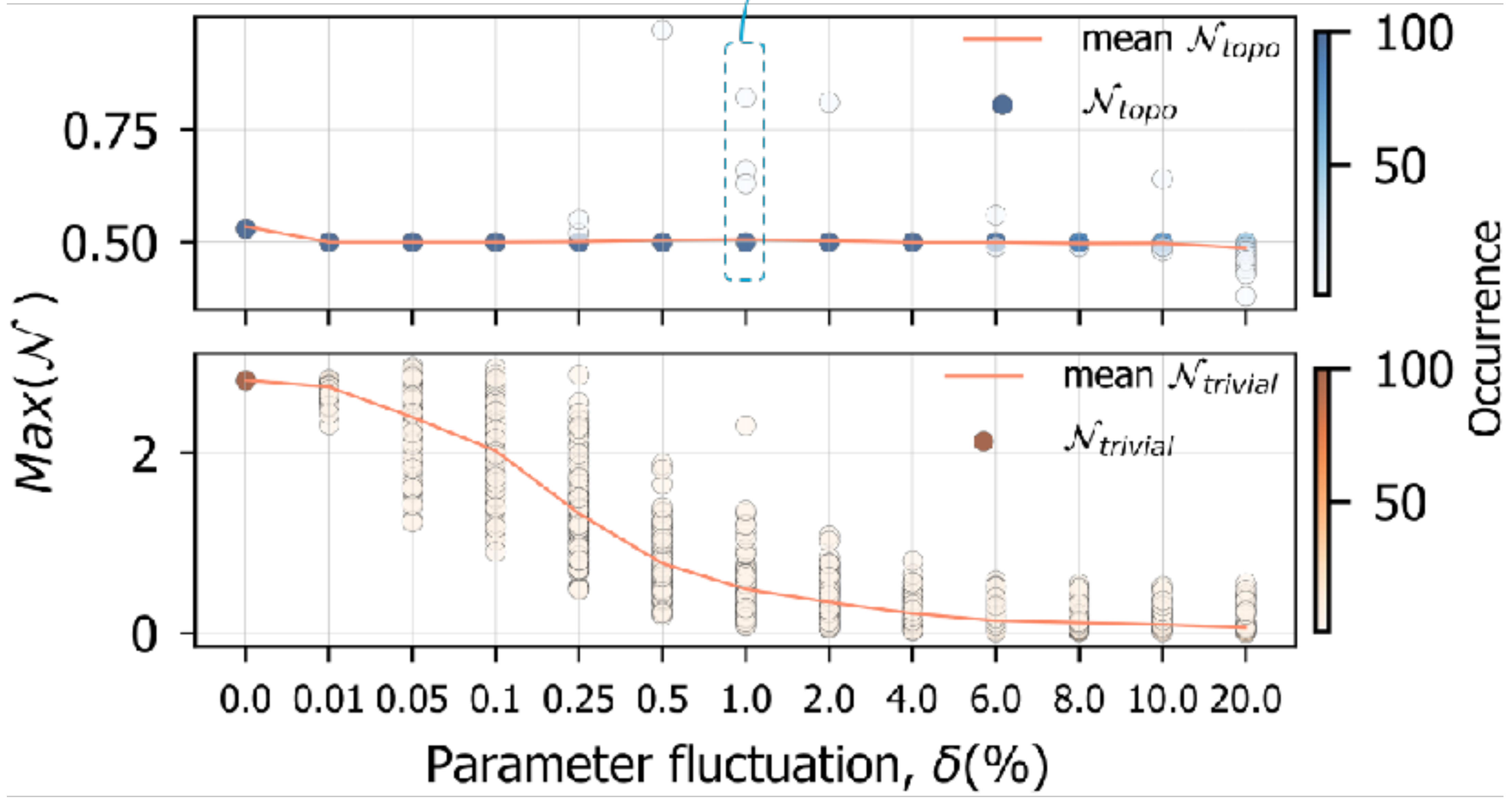
Does topology help with the entanglement stability?

Entanglement Measure: Negativity

$$\mathcal{N}(\rho) = \left| \sum_{\lambda_i<0} \lambda_i \right| = \sum_i \frac{|\lambda_i| - \lambda_i}{2},$$

Eigenvalues of the partially transposed density matrix

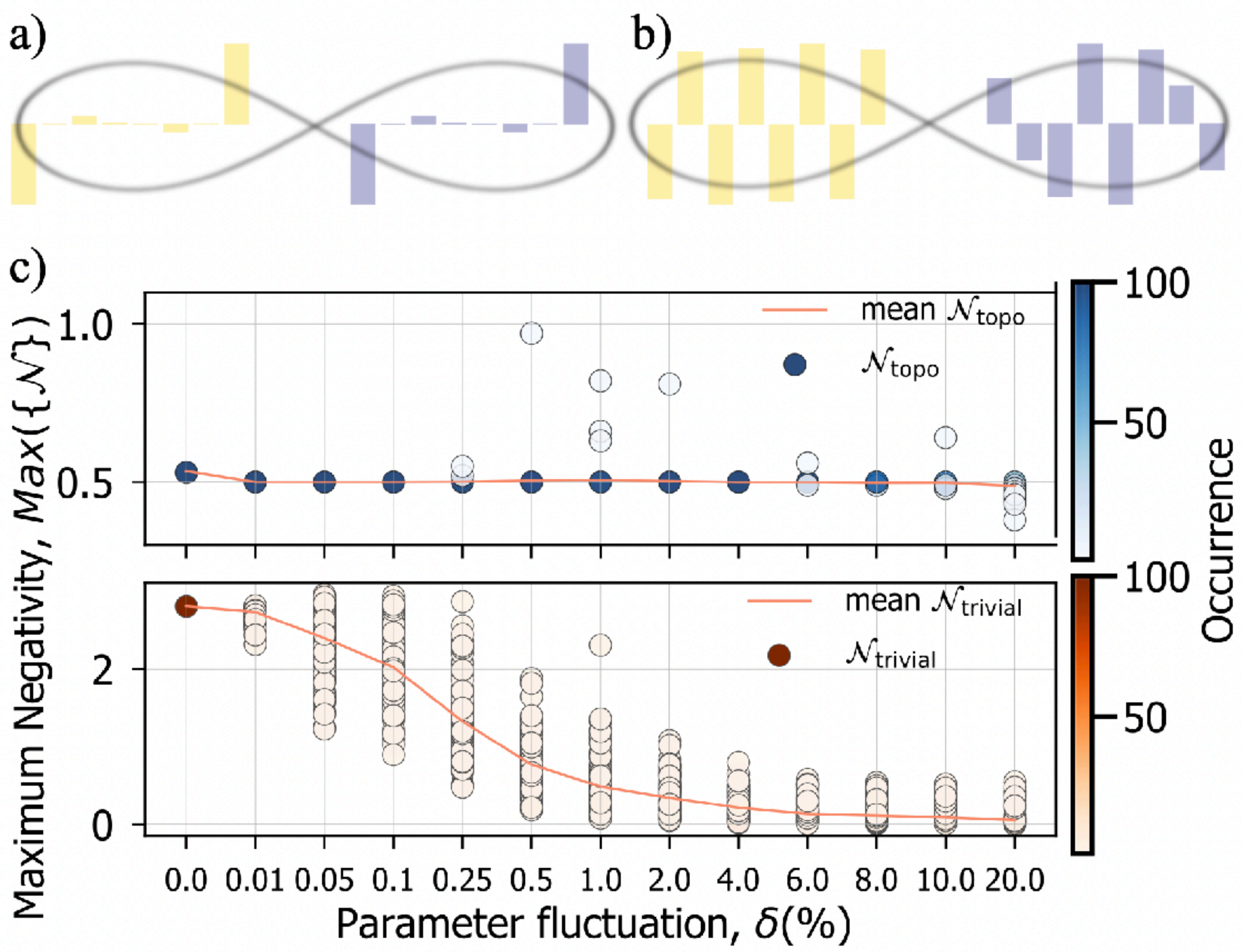
100 random Hamiltonians



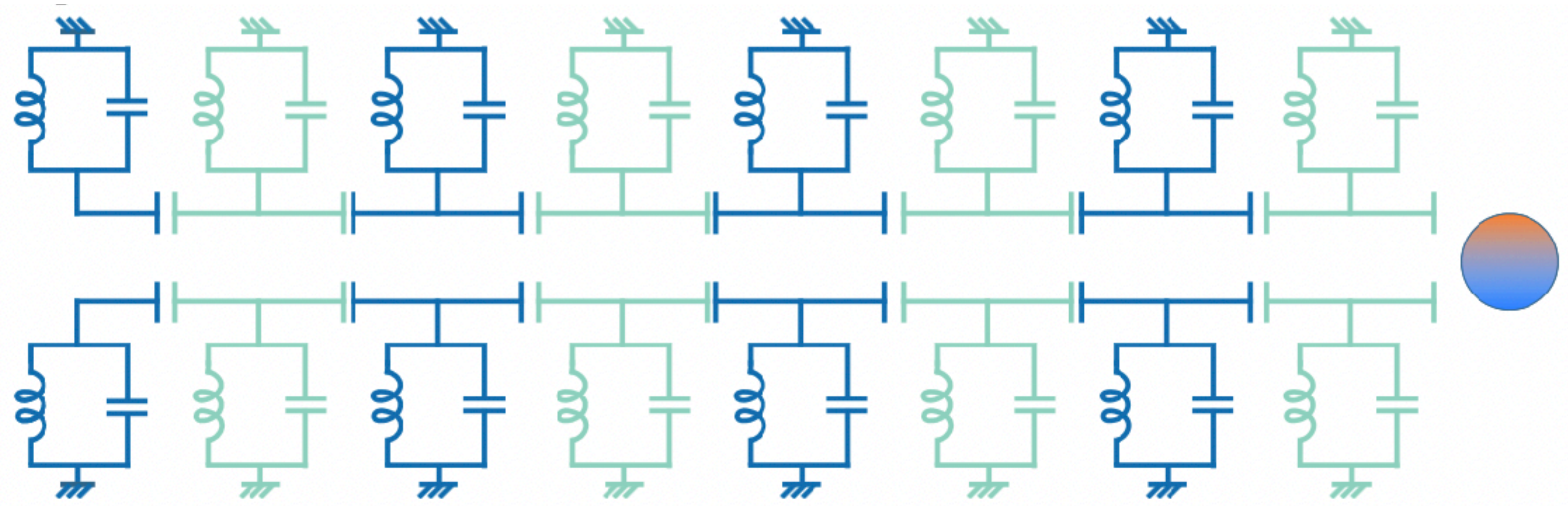
topological
trivial

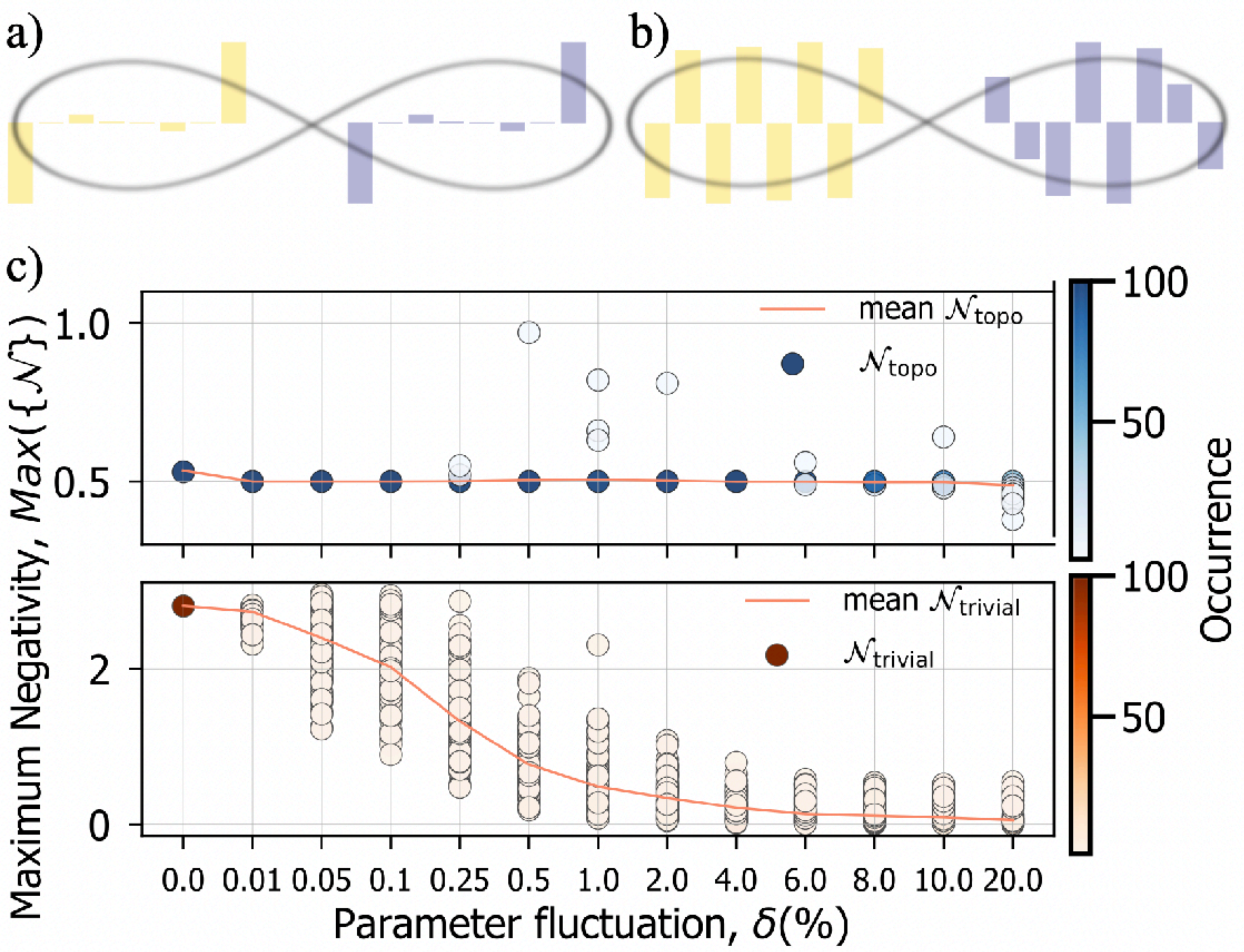
Guliuxin Jin, **EG**: arXiv:2205.09100 (2022)

QMAI | TUDelft



Simple topological model + on-chip engineering =
practical entanglement stabilization





Guliuxin Jin, **EG**: arXiv:2205.09100 (2022)

QMAI | TU Delft

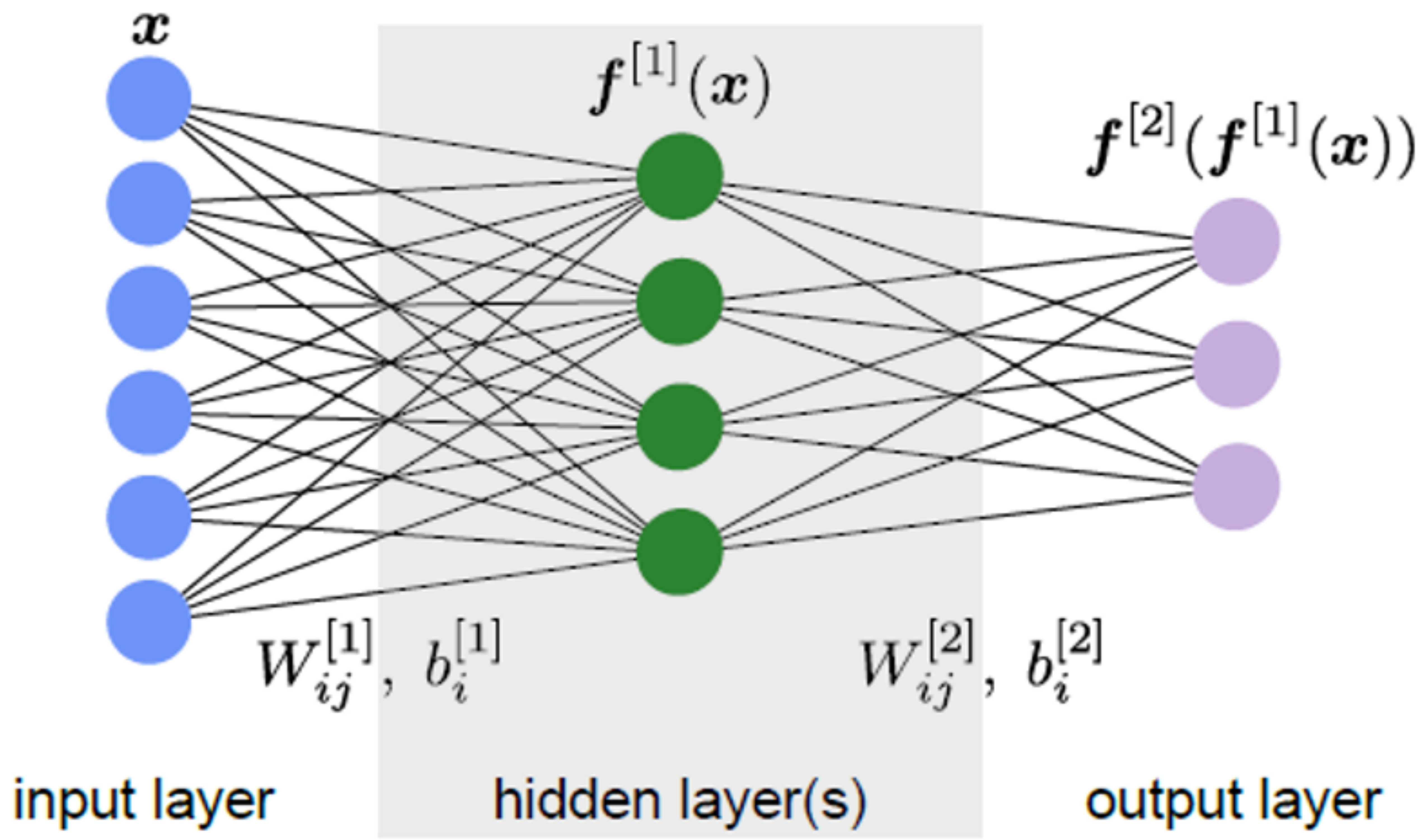
BREAK 10 mins

QMAI | TU Delft

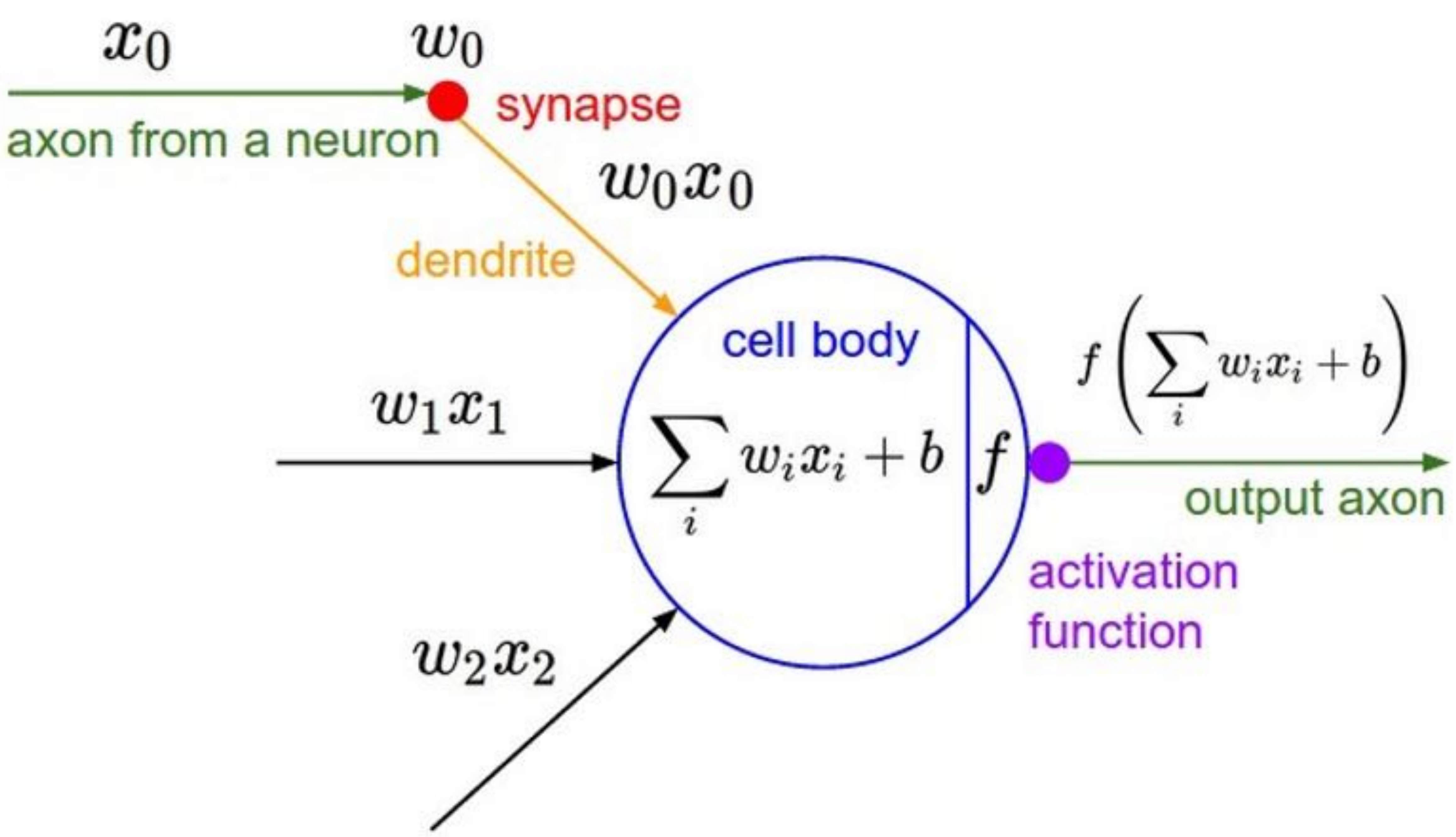
Machine Learning Primer: Neural Network Introduction

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

Neural Networks 101: Supervised learning

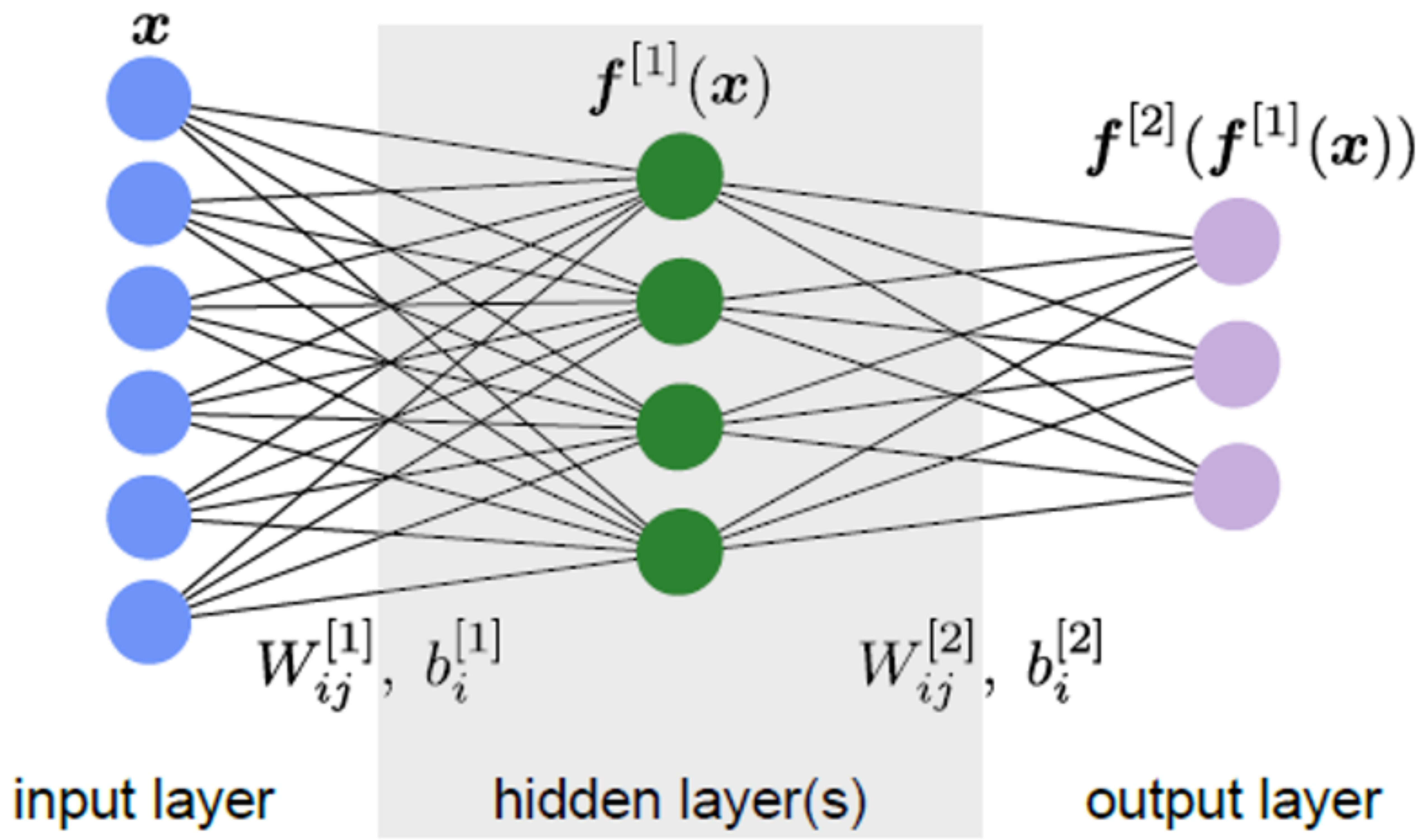


What happens in a neuron?

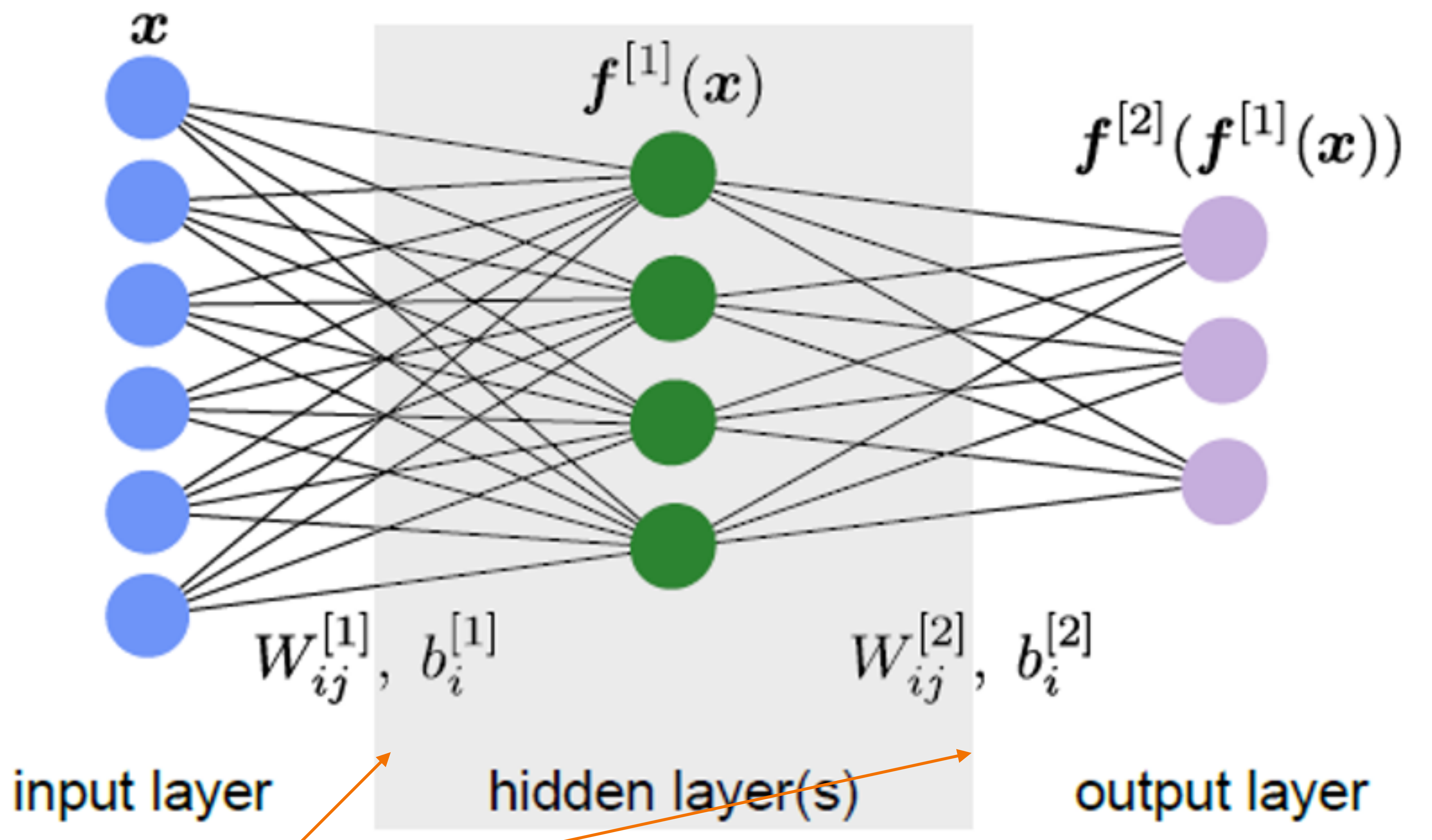


From https://cs231n.github.io

Putting neurons in the networks

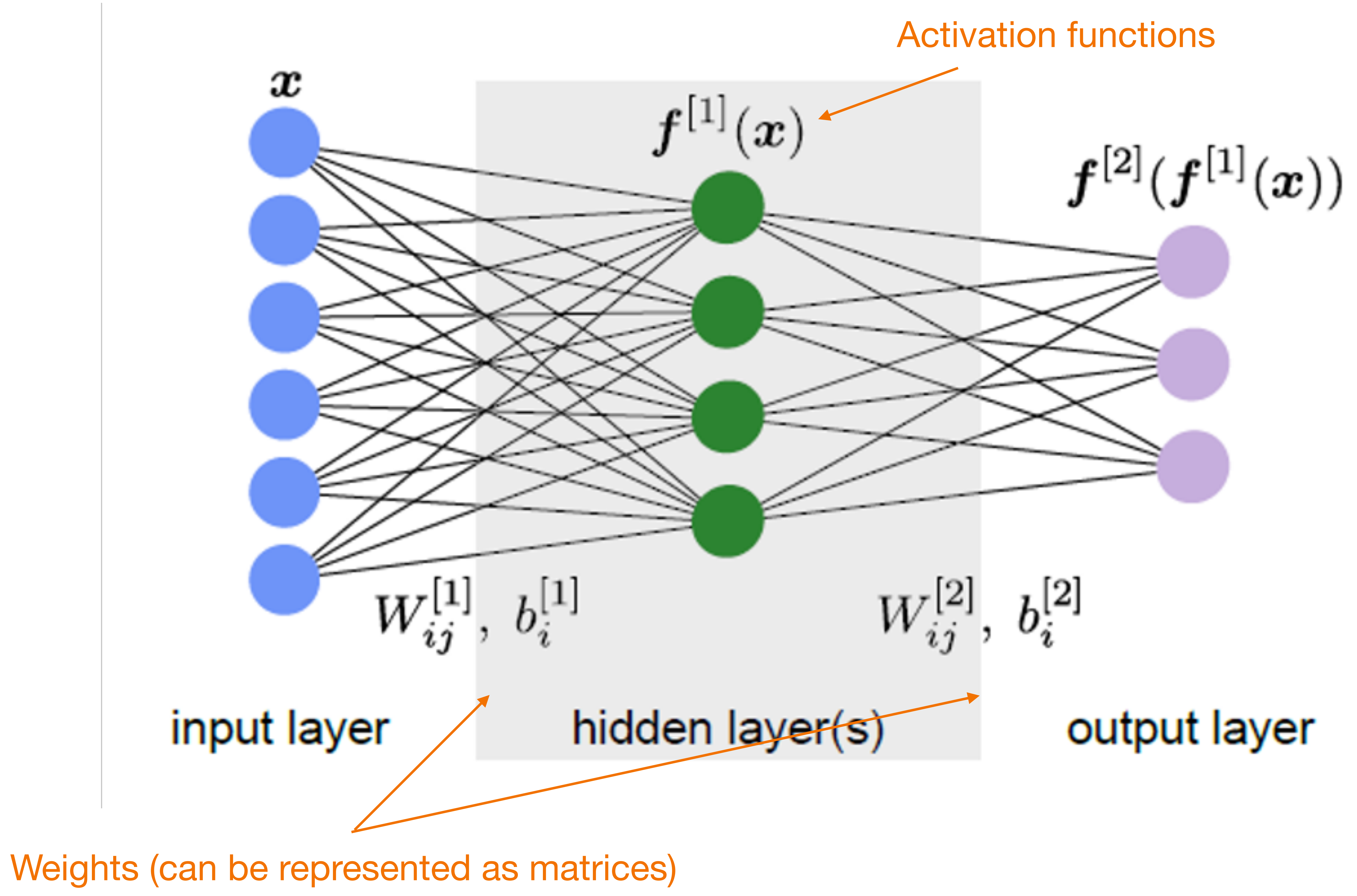


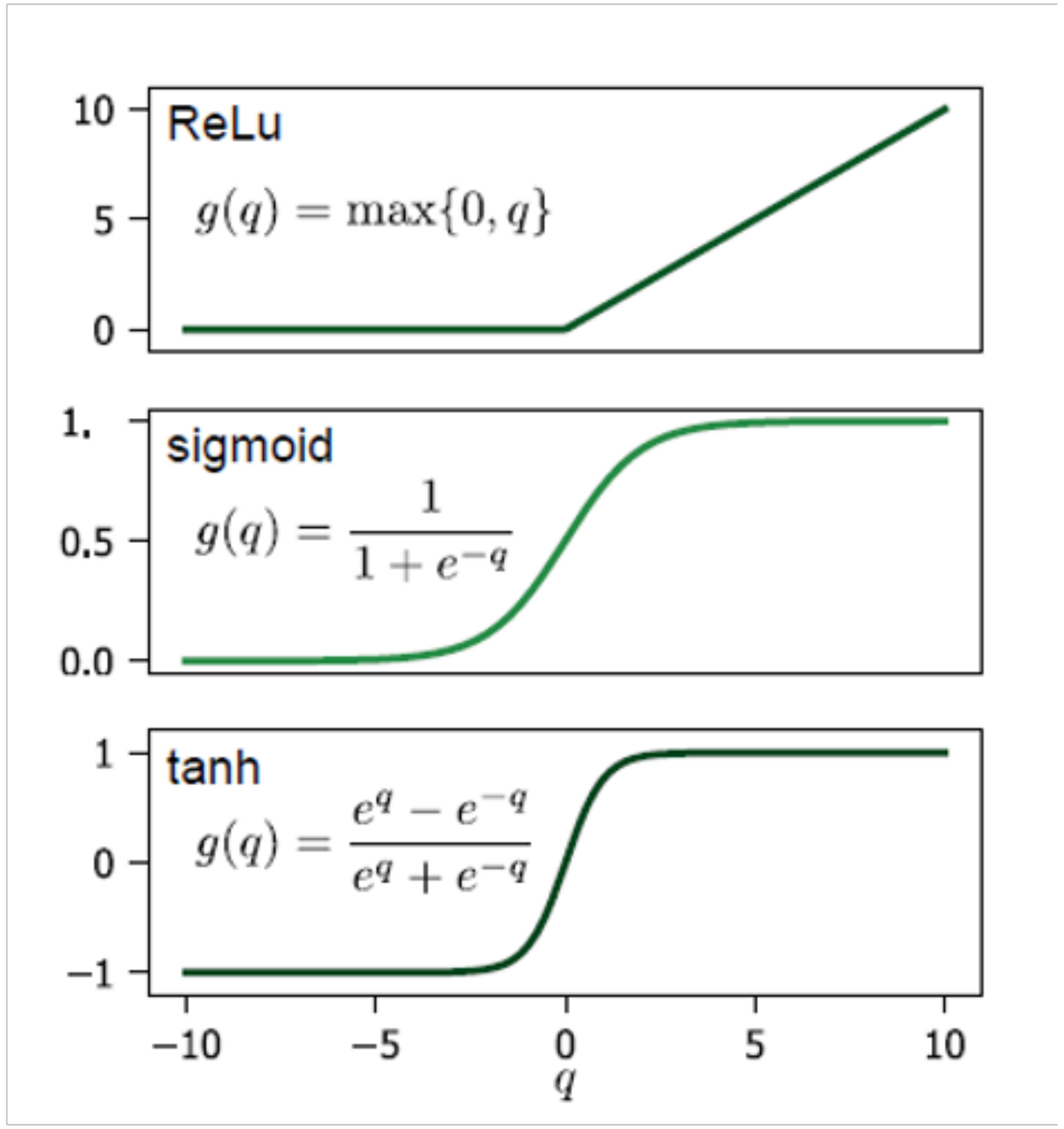
Putting neurons in the networks



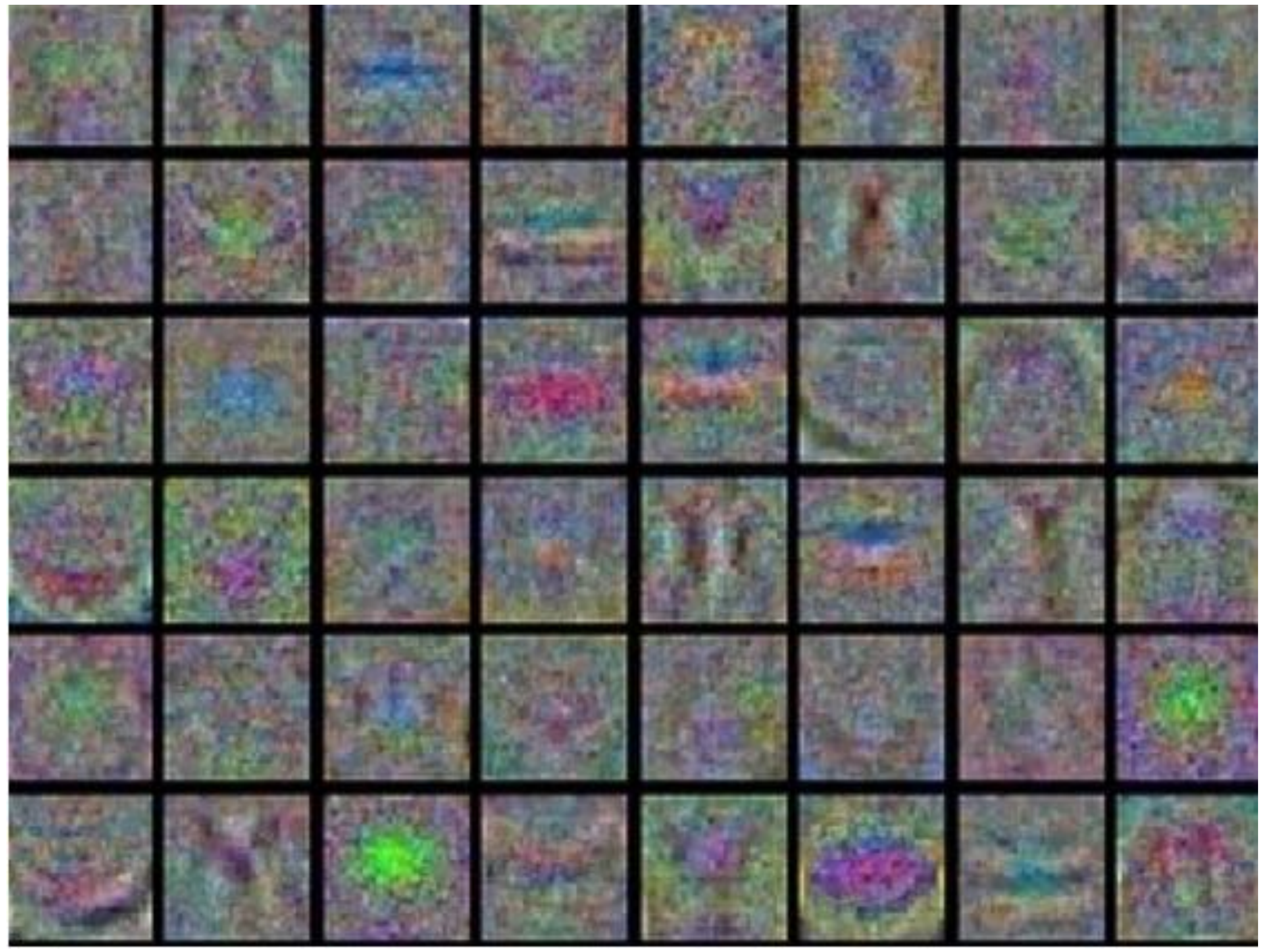
Weights (can be represented as matrices)

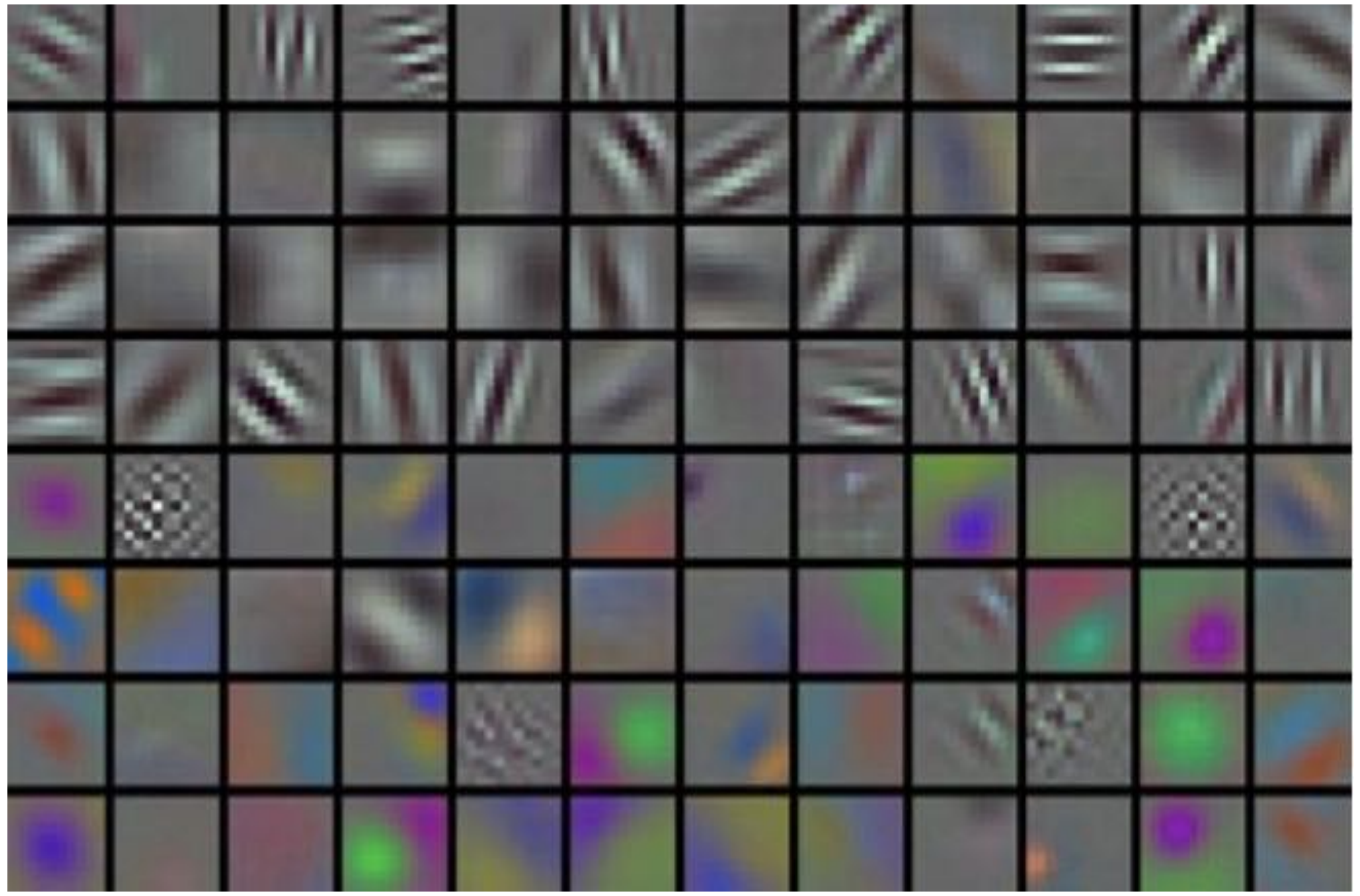
Putting neurons in the networks





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

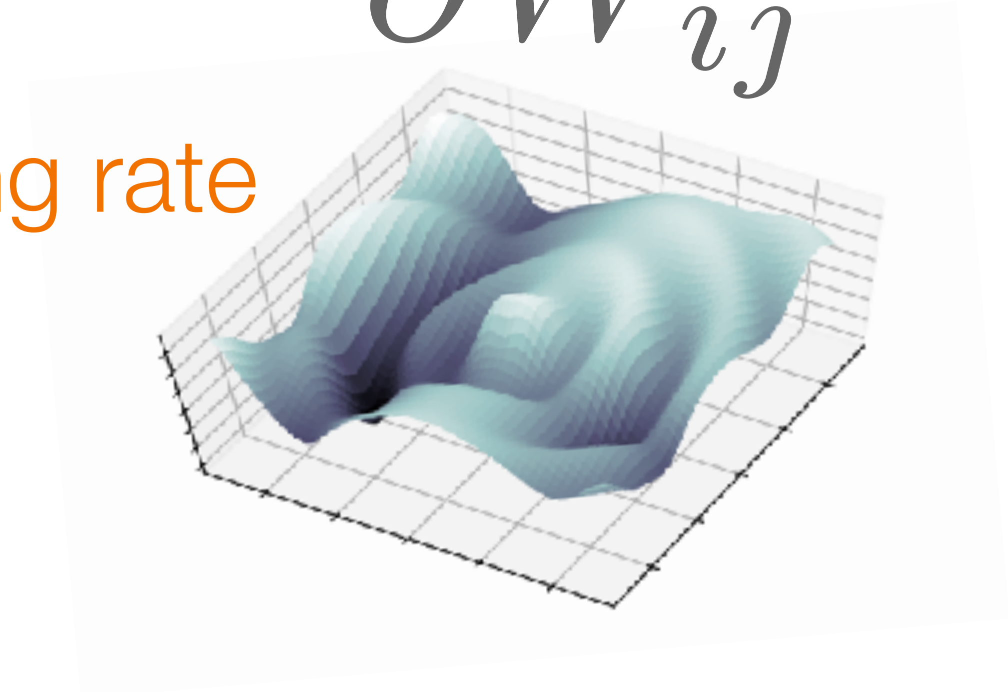
$$W_{ij} \to W_{ij} - \epsilon \frac{\partial L}{\partial W_{ij}}$$

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

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

learning rate



How do we calculate the derivative?

- CHAIN RULE REMINDER:

How do we calculate the derivative?

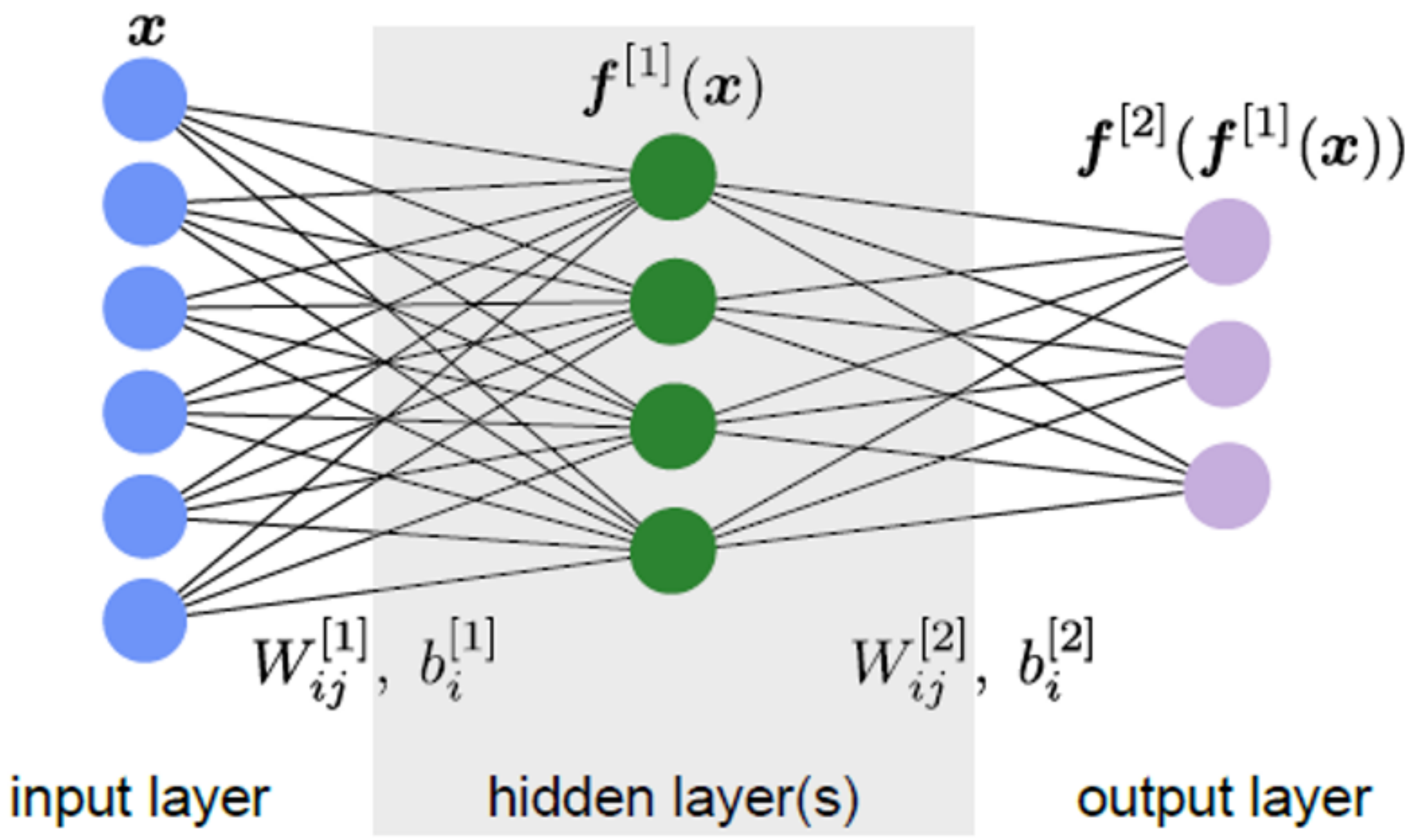
- CHAIN RULE REMINDER:

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

How do we calculate the derivative?

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

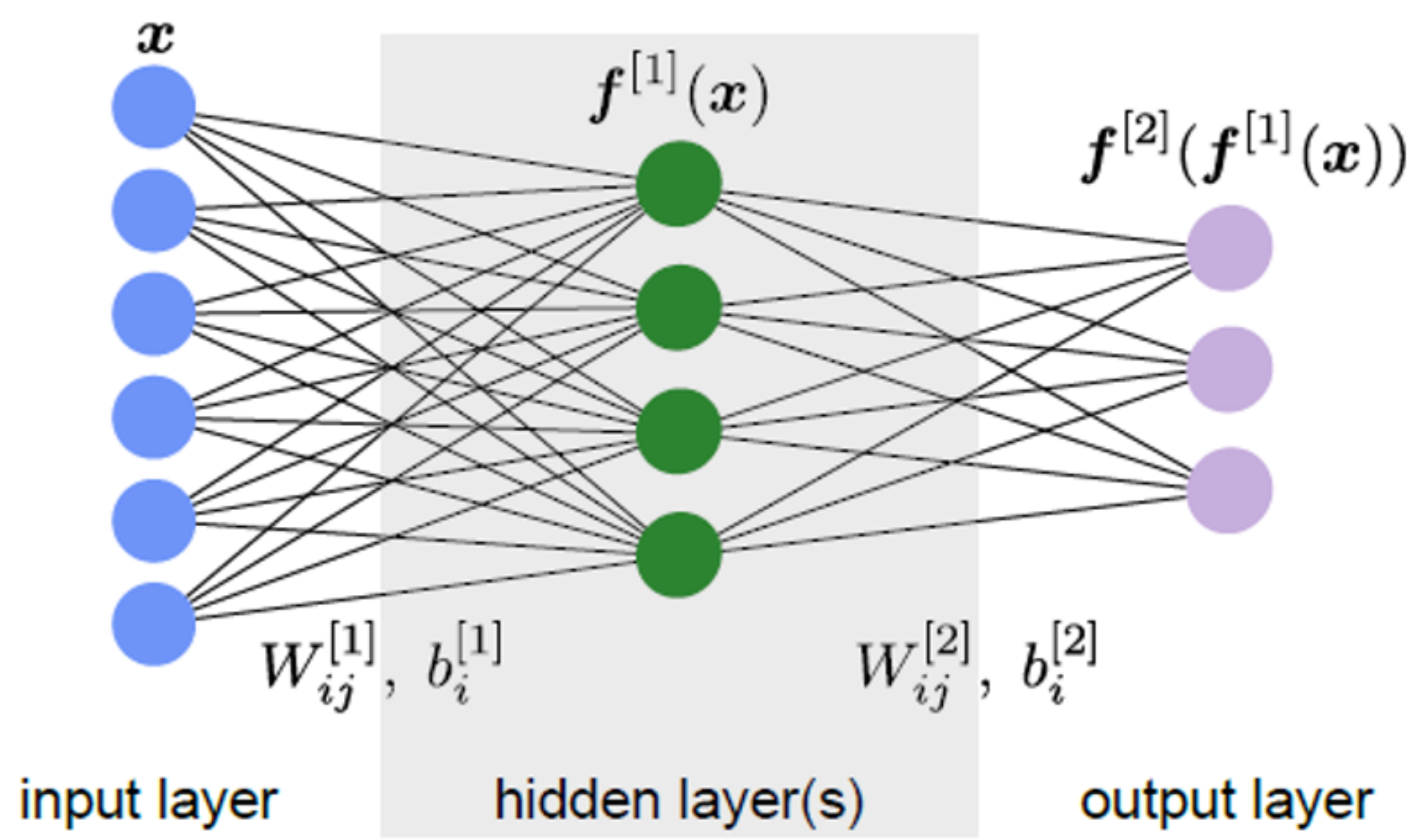
- APPLY TO NEURAL NET:



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:

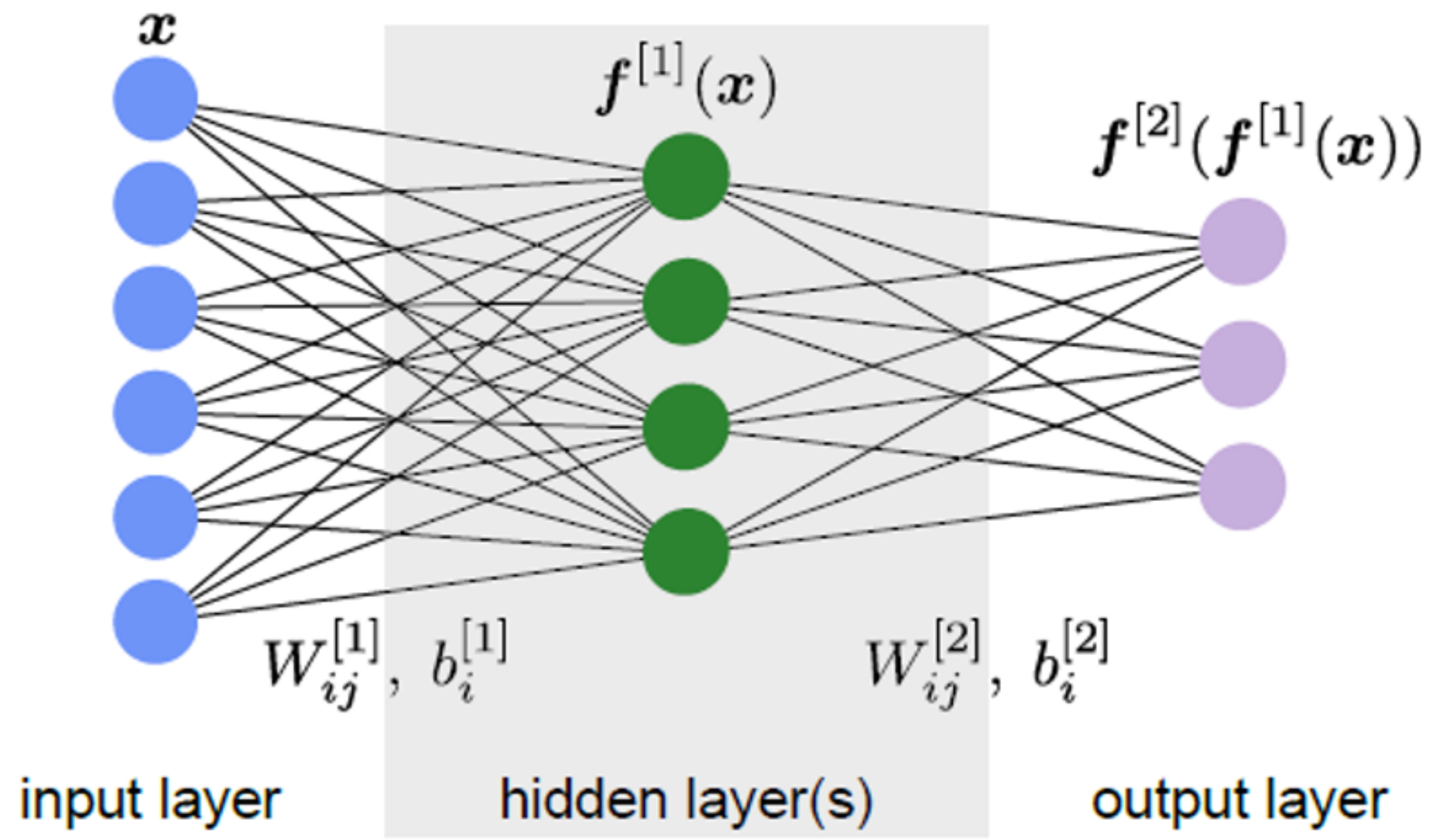


How do we calculate the derivative?

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

First calculate this using
the network's output

- APPLY TO NEURAL NET:



QMAI | TU Delft

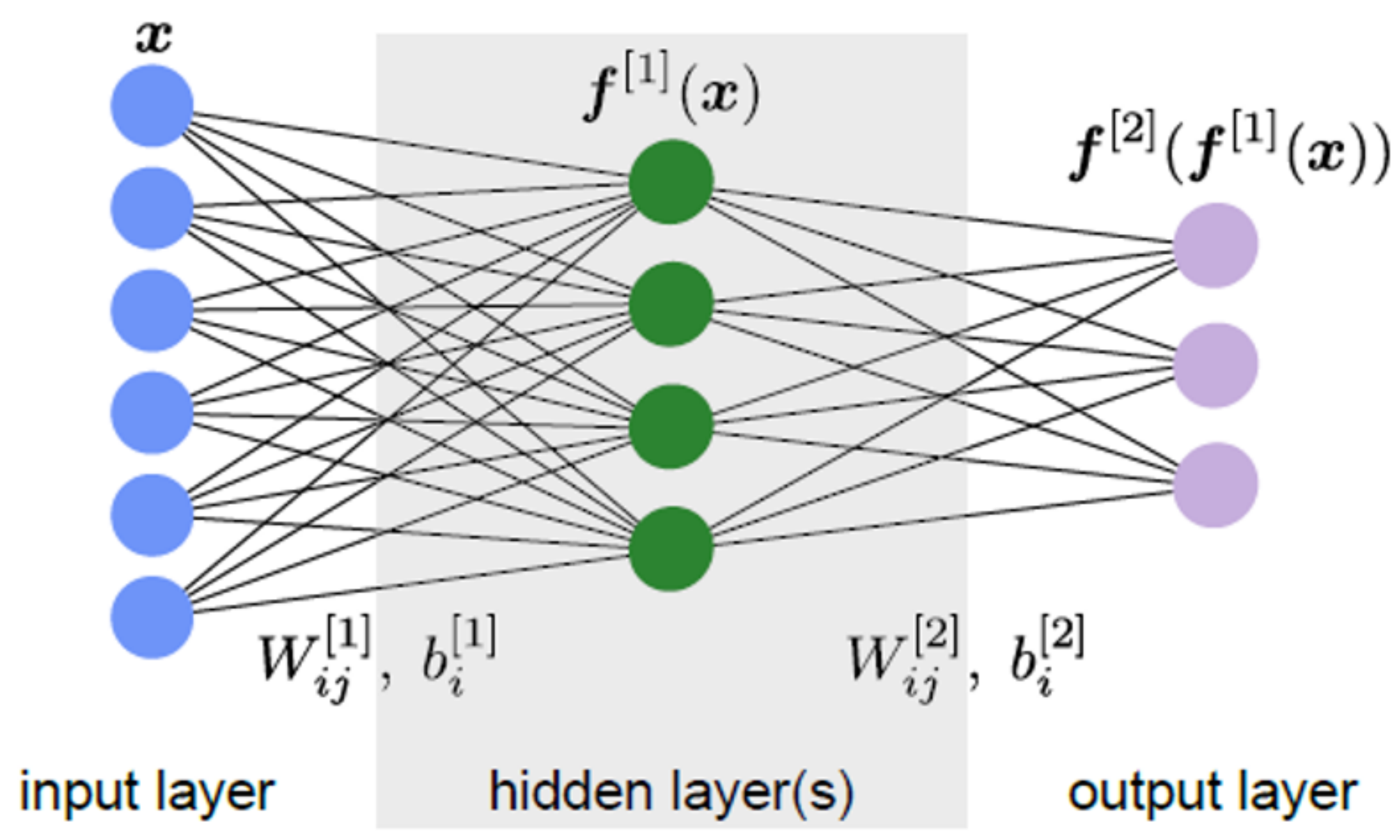
How do we calculate the derivative?

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

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

First calculate this using the network's output

- APPLY TO NEURAL NET:



How do we calculate the derivative?

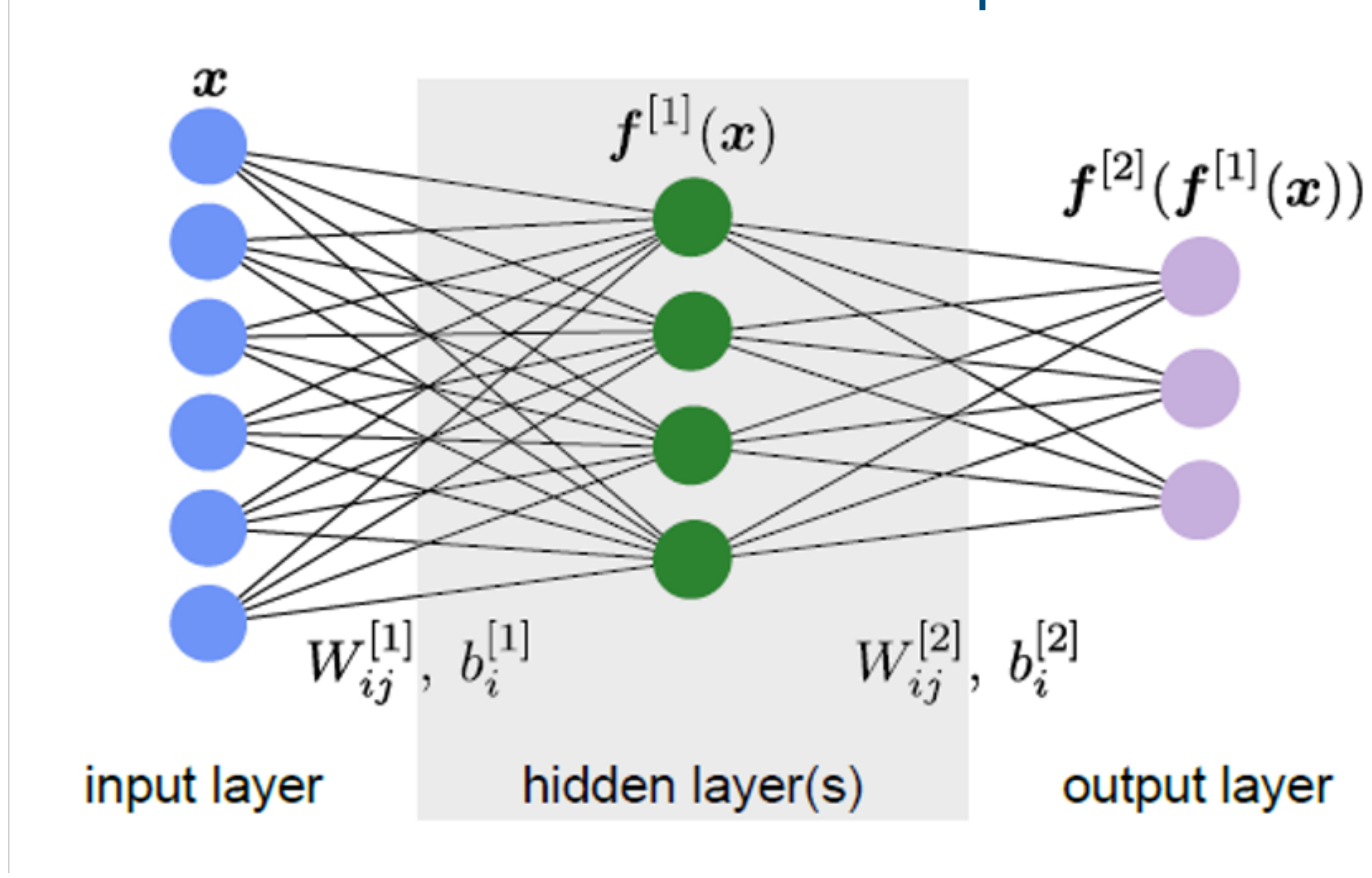
Calculate this using estimate of f[2] from previous step

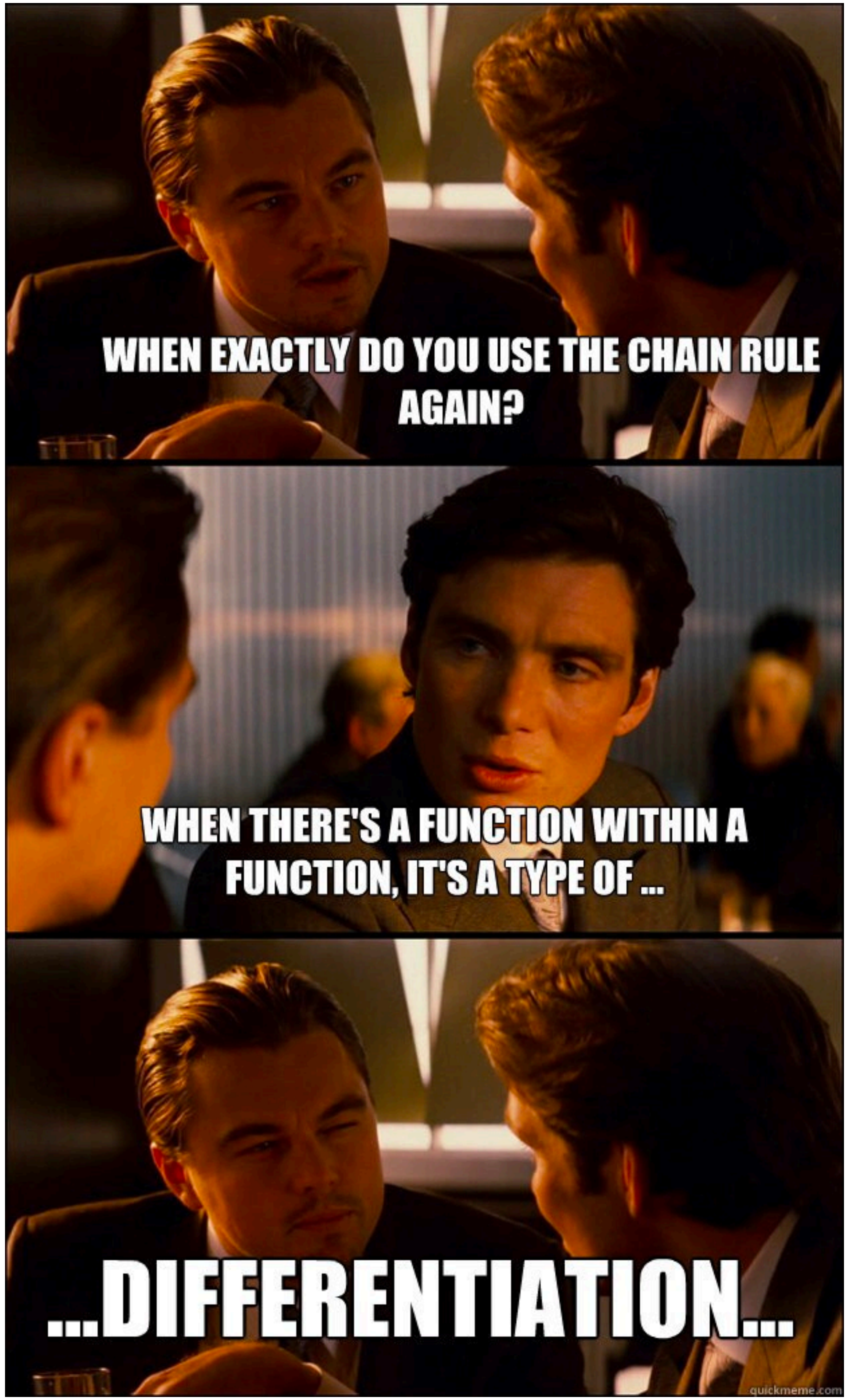
$$\frac{dL(f^{[2]})}{dW_{ij}^{[1]}} = \frac{dL}{df^{[2]}}\frac{df^{[2]}}{df^{[1]}}\frac{df^{[1]}}{dW^{[1]}}$$

First calculate this using the network's output

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

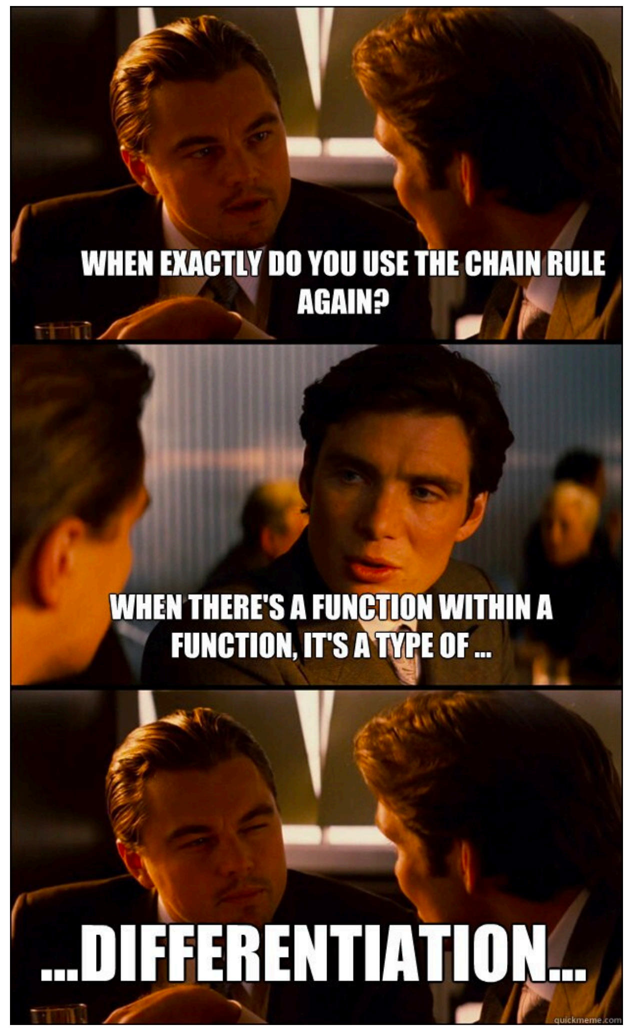
- APPLY TO NEURAL NET:

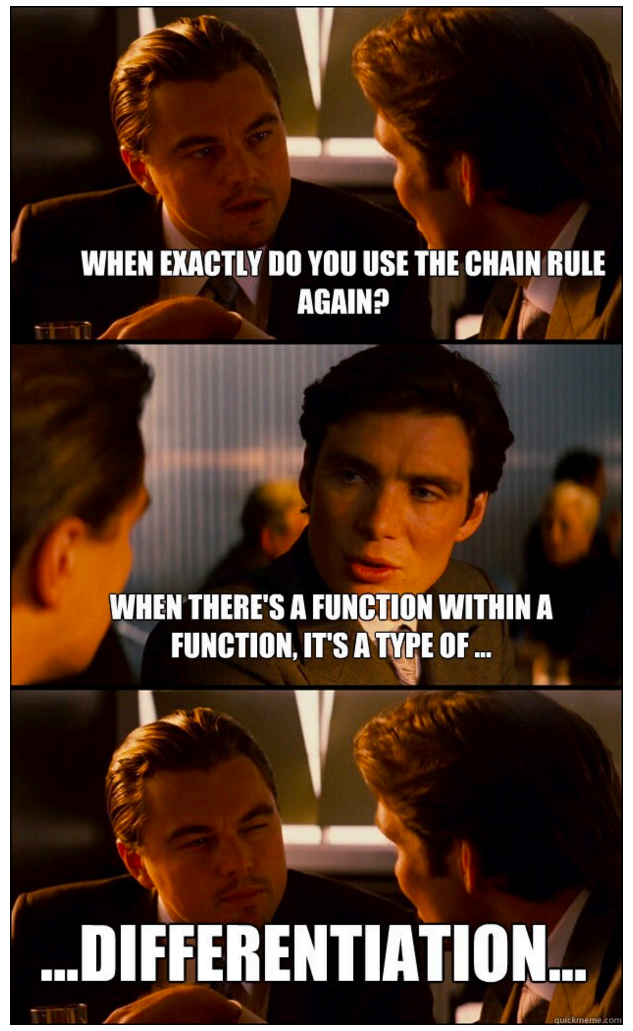


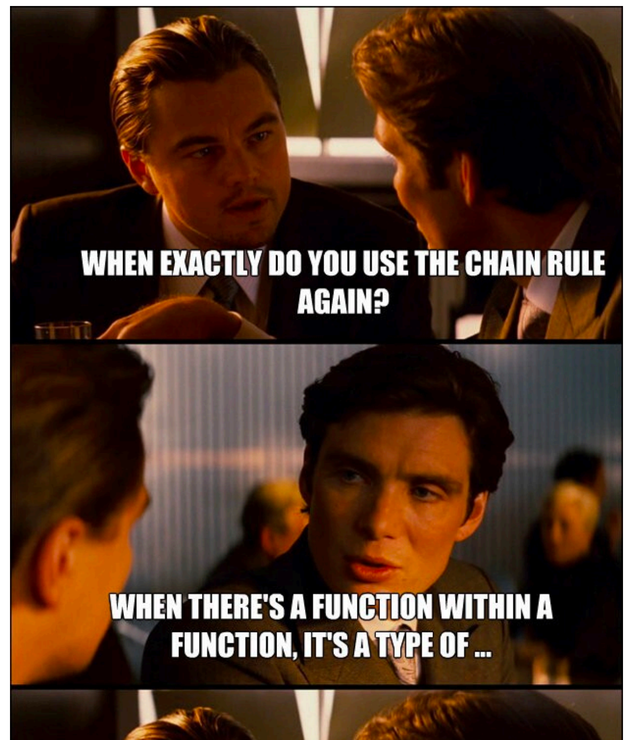


QMAI | TU Delft

Backpropagation algorithm

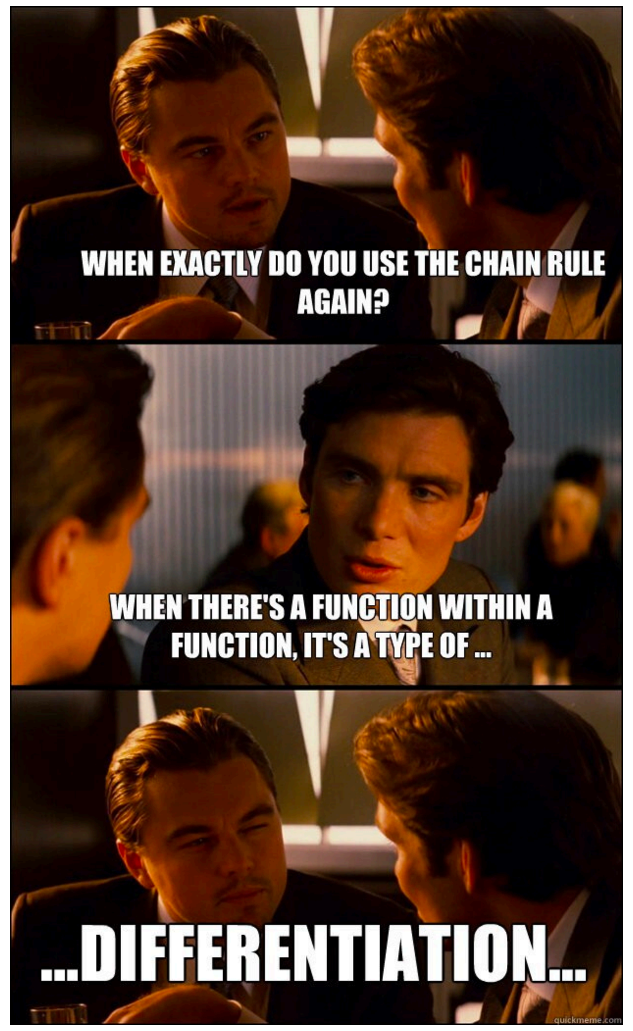


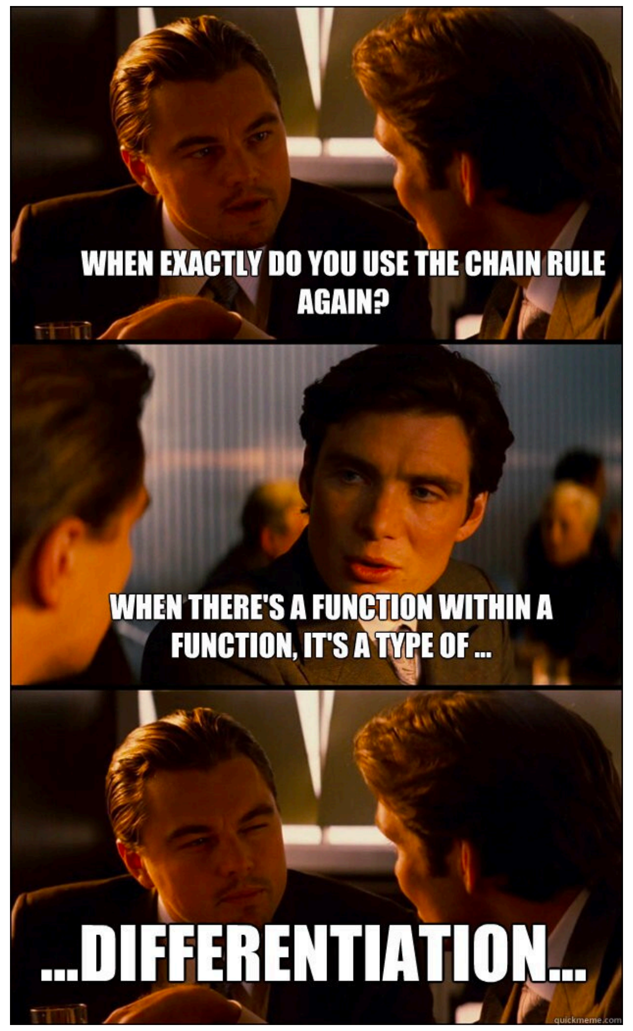




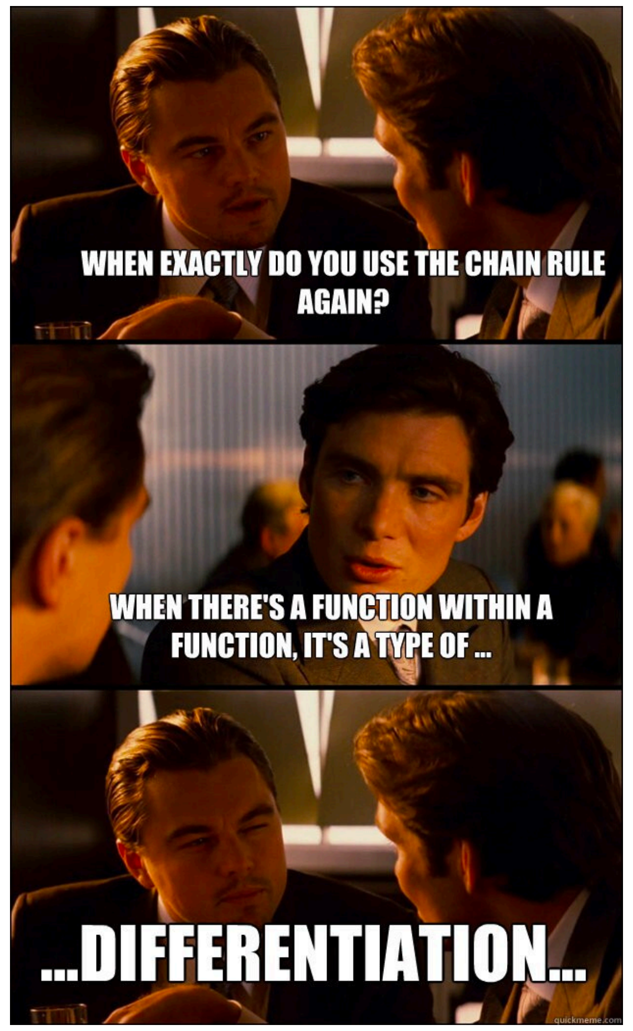
Backpropagation algorithm

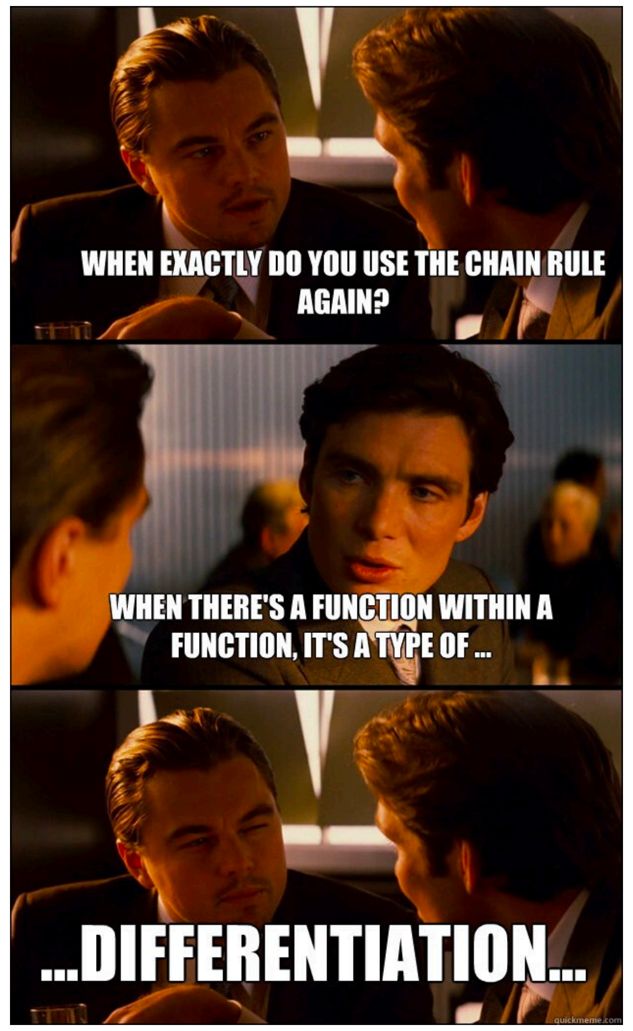
1. Calculate forward propagation of the network
2. Calculate the backward phase
   a: Estimate the error in the final layer
   b: Propagate that error into the previous layer
   c: Evaluate the derivative of each parameter in the network
3. Combine all partial gradients into the final gradient
4. Update the weights using the calculated gradients to minimise the loss

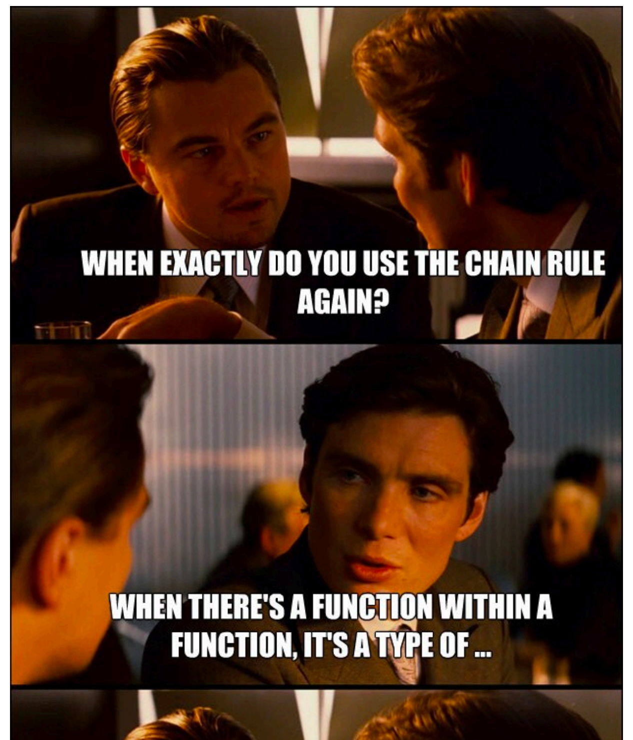




QMAI | TU Delft

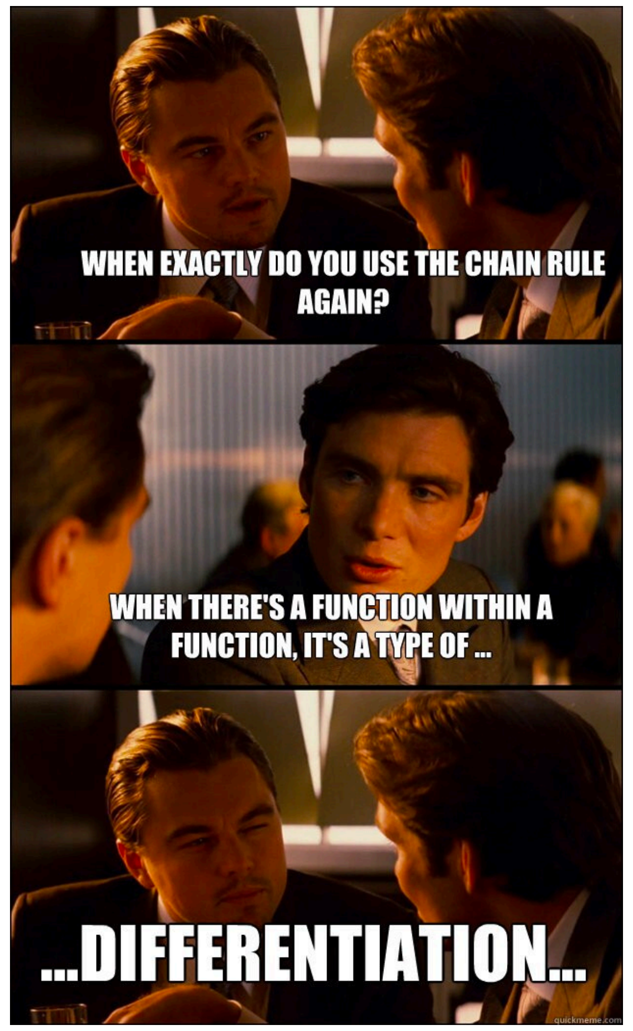


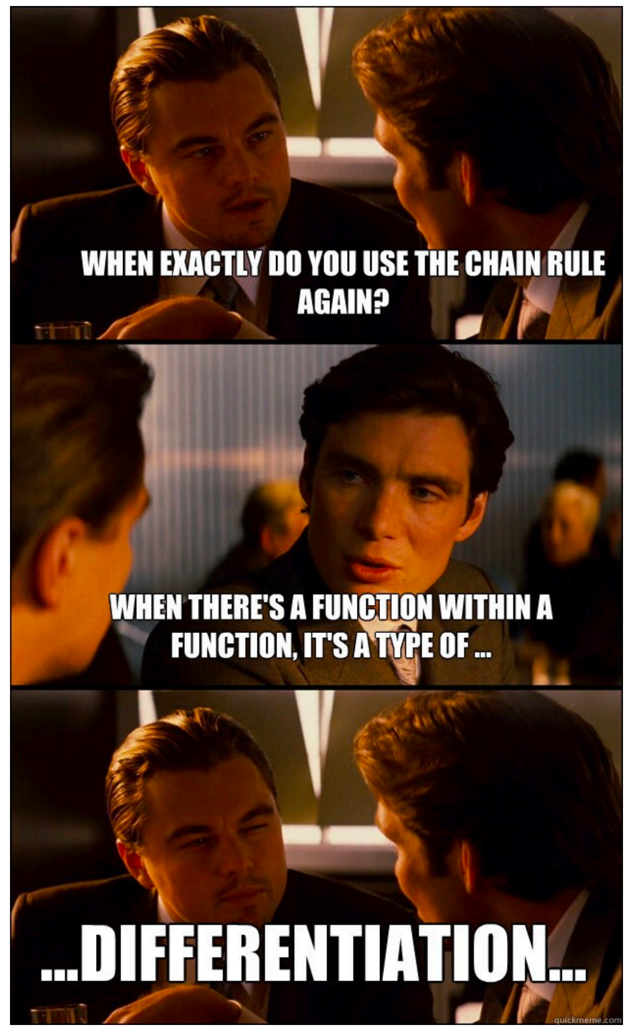




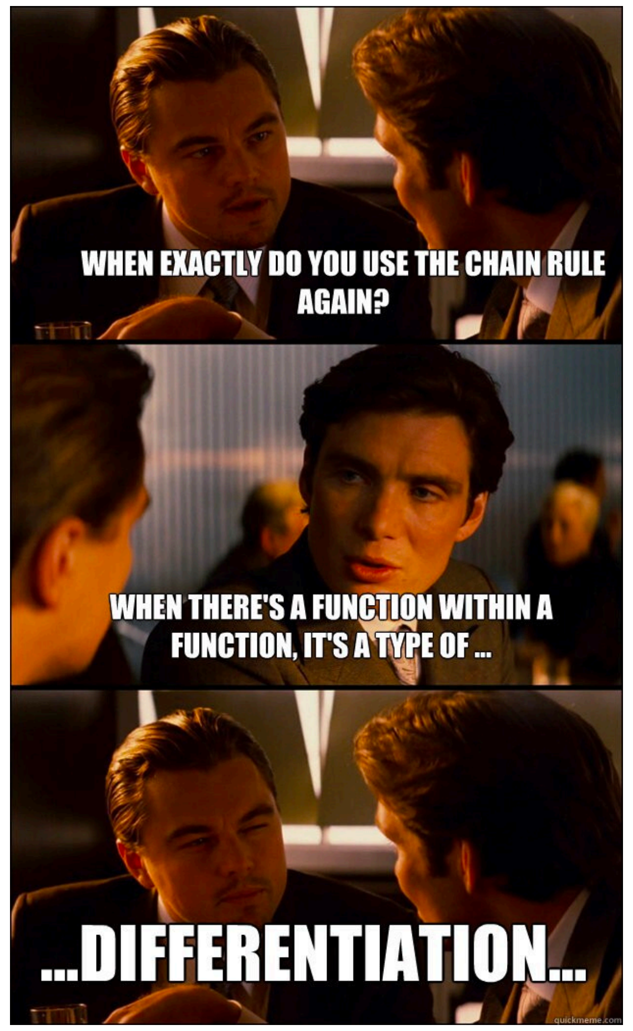
Backpropagation algorithm

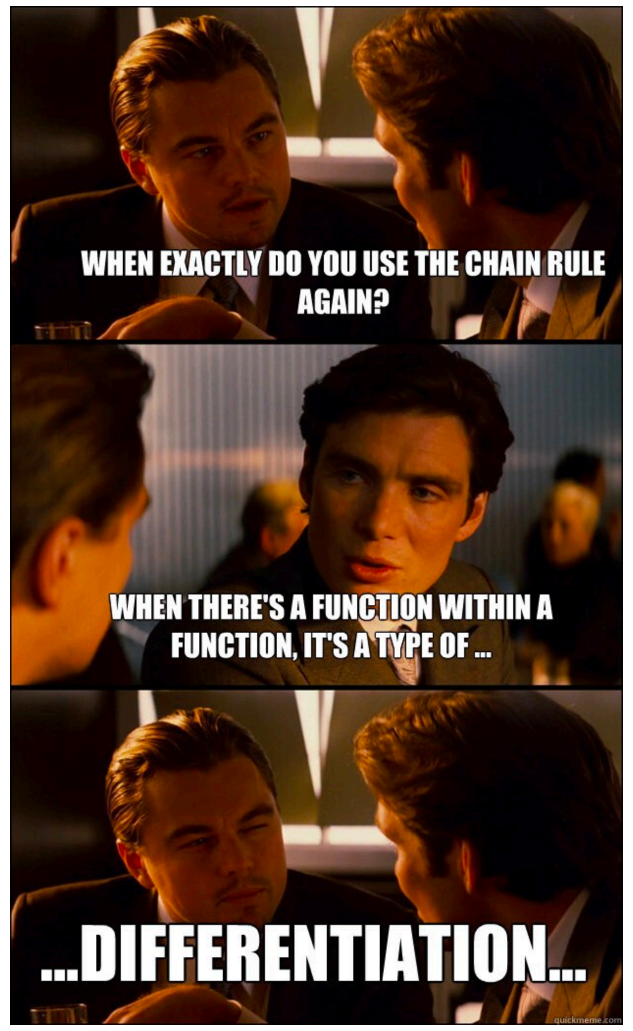
1. Calculate forward propagation of the network
2. Calculate the backward phase
   a: Estimate the error in the final layer
   b: Propagate that error into the previous layer
   c: Evaluate the derivative of each parameter in the network
3. Combine all partial gradients into the final gradient
4. Update the weights using the calculated gradients to minimise the loss

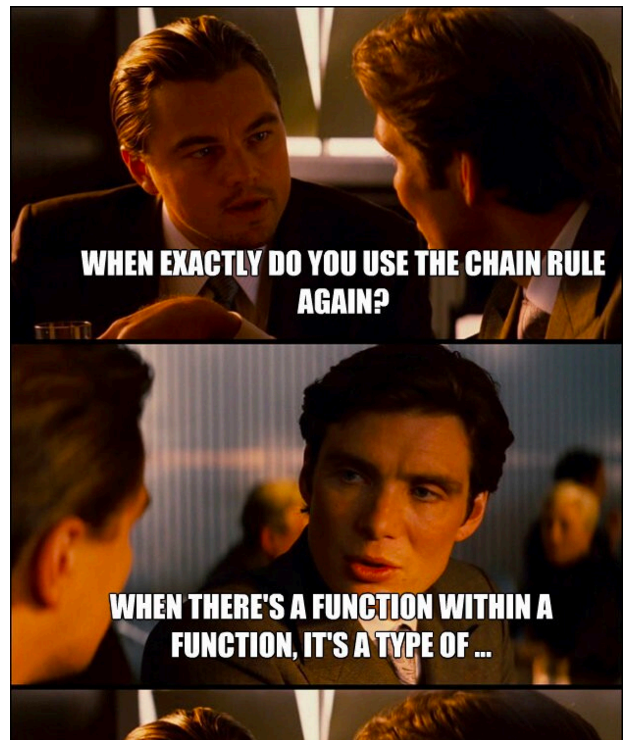




Backpropagation algorithm





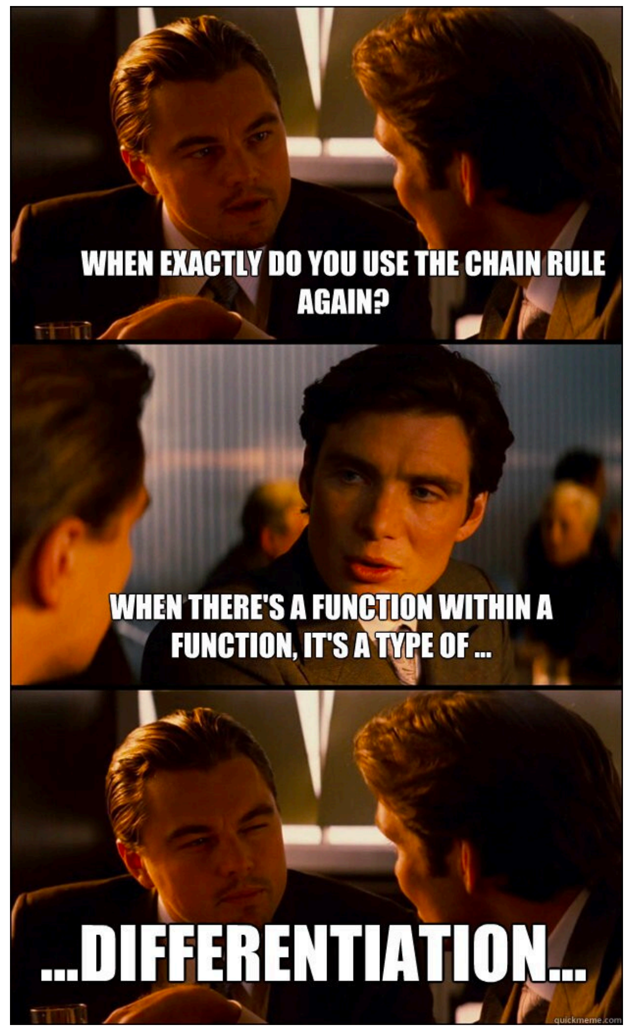


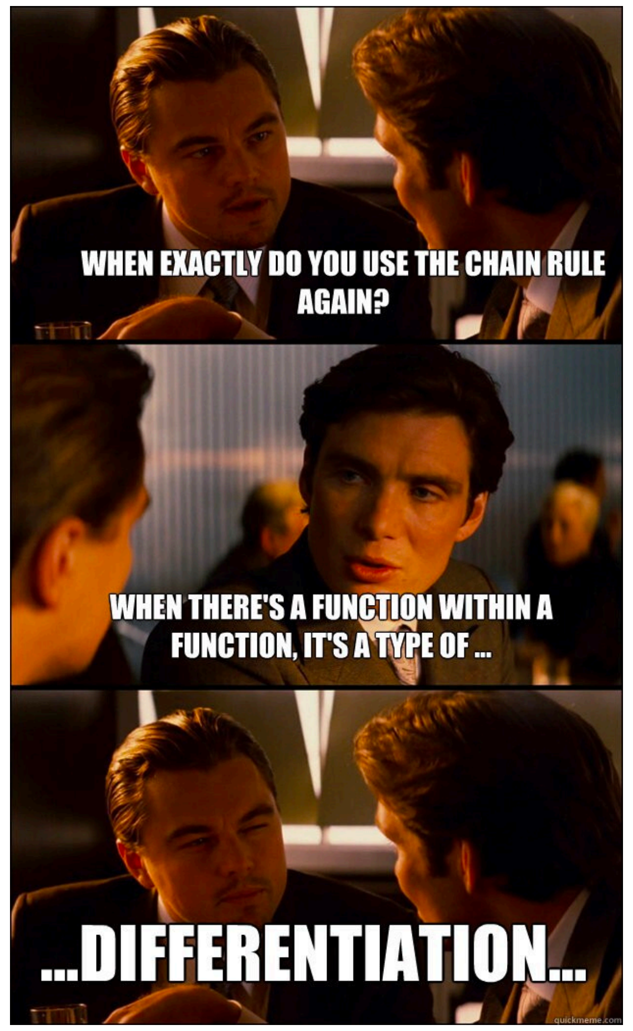
1. Calculate forward propagation of the network
2. Calculate the backward phase

   a: Estimate the error in the final layer

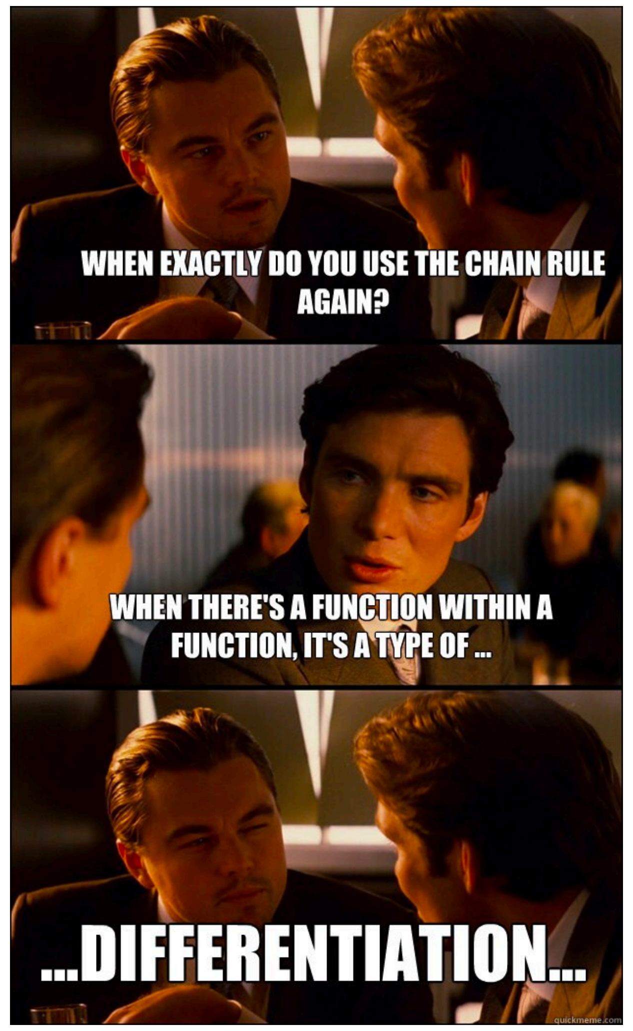
   b: Propagate that error into the previous layer

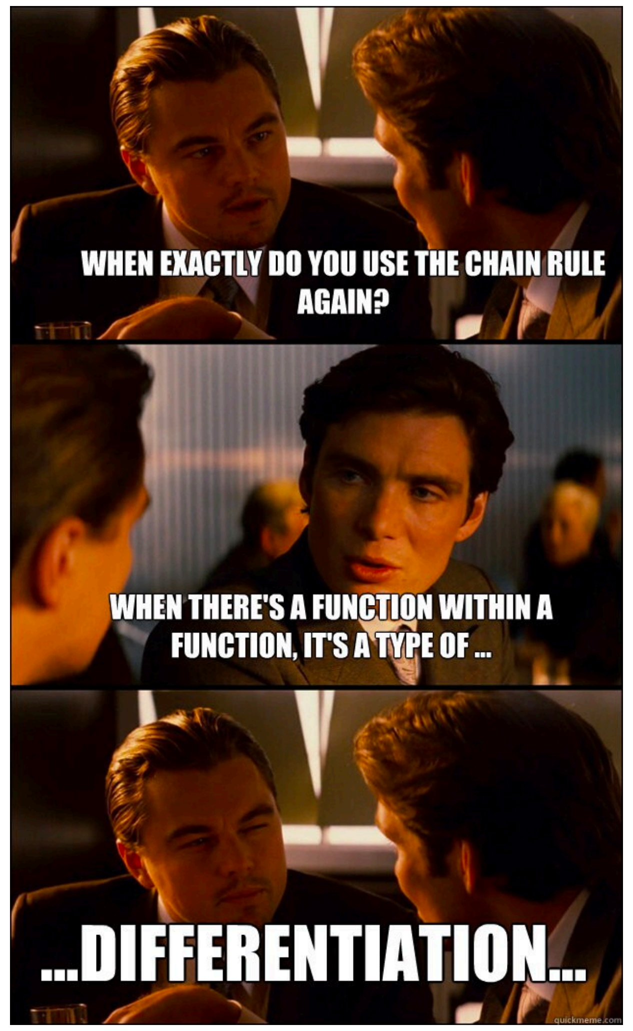
   c: Evaluate the derivative of each parameter in the network
3. Combine all partial gradients into the final gradient
4. Update the weights using the calculated gradients to minimise the loss

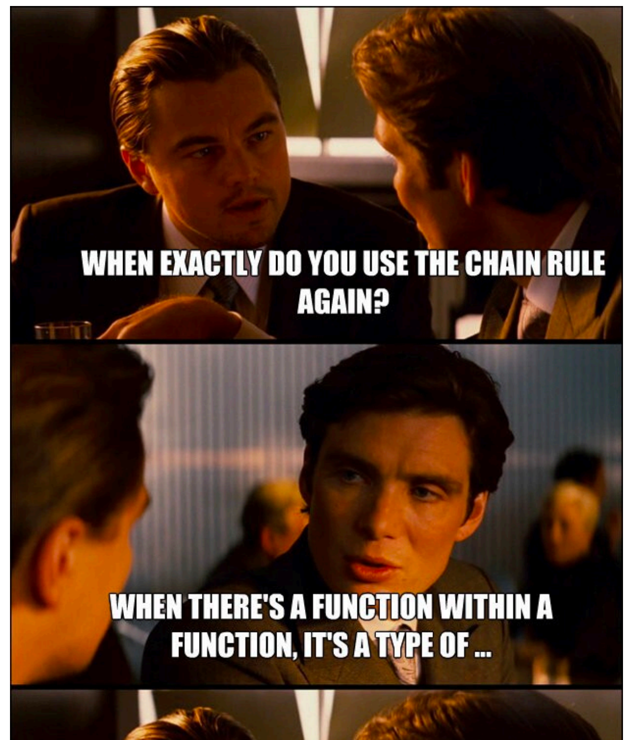




Backpropagation algorithm





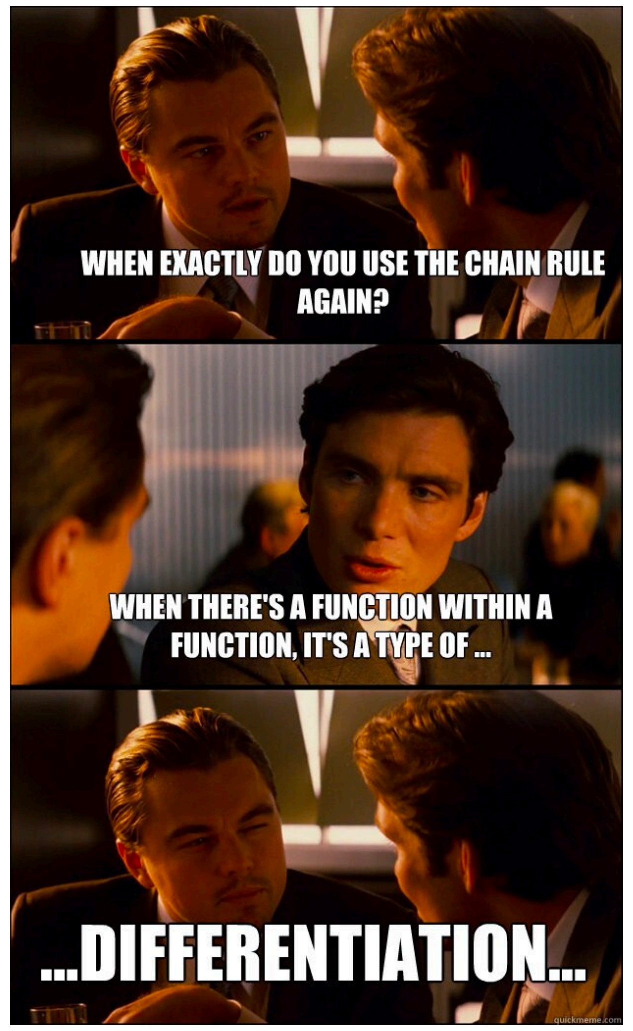


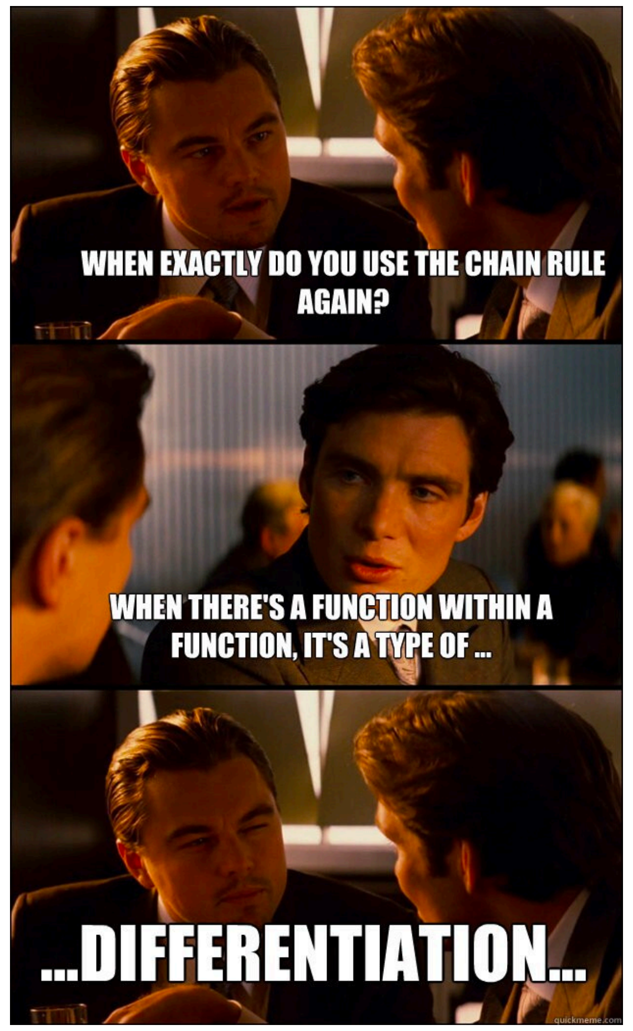
1. Calculate forward propagation of the network
2. Calculate the backward phase

   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





QMAI | TU Delft

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

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

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

QMAI | TU Delft

Classification: a mini example

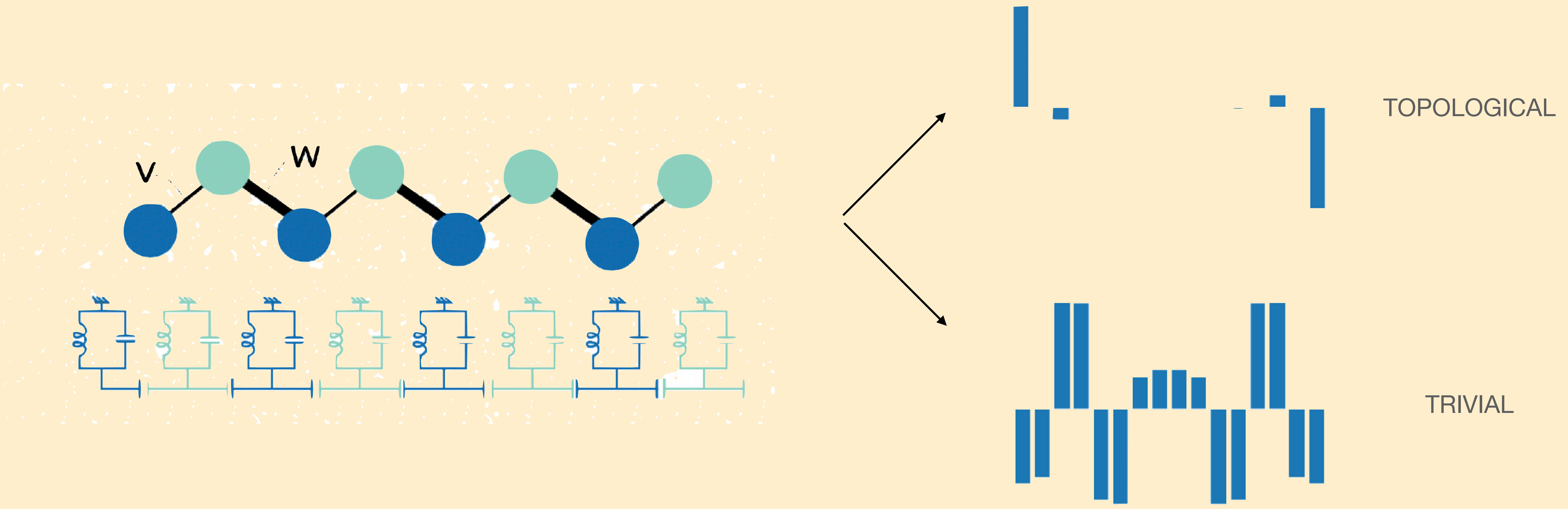
Label: p(x) $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$

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))

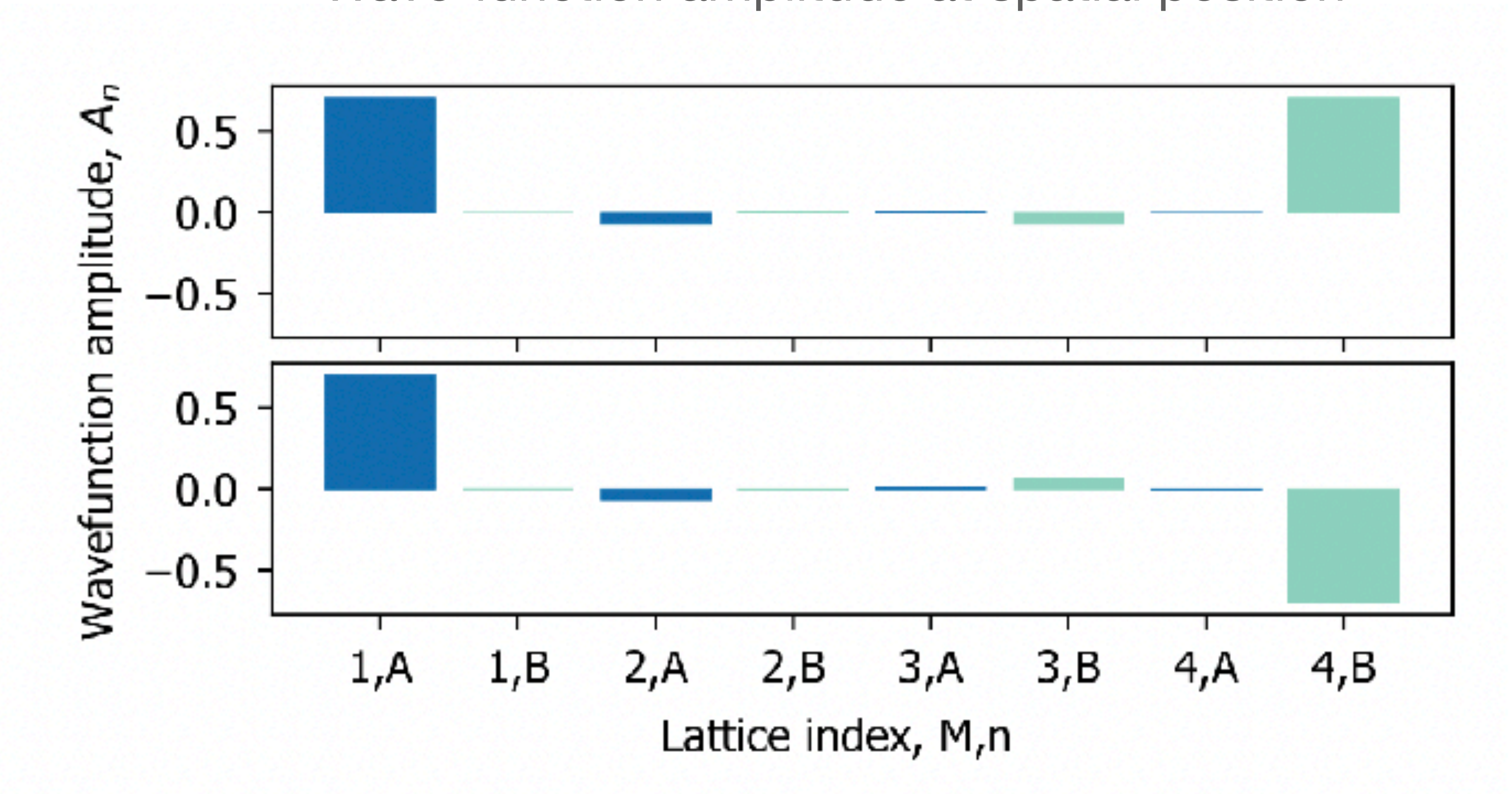
!the loss is minimal if q(x) matches the labels!

Hands-On:
Machine Learning in (Topological) Superconducting Circuits

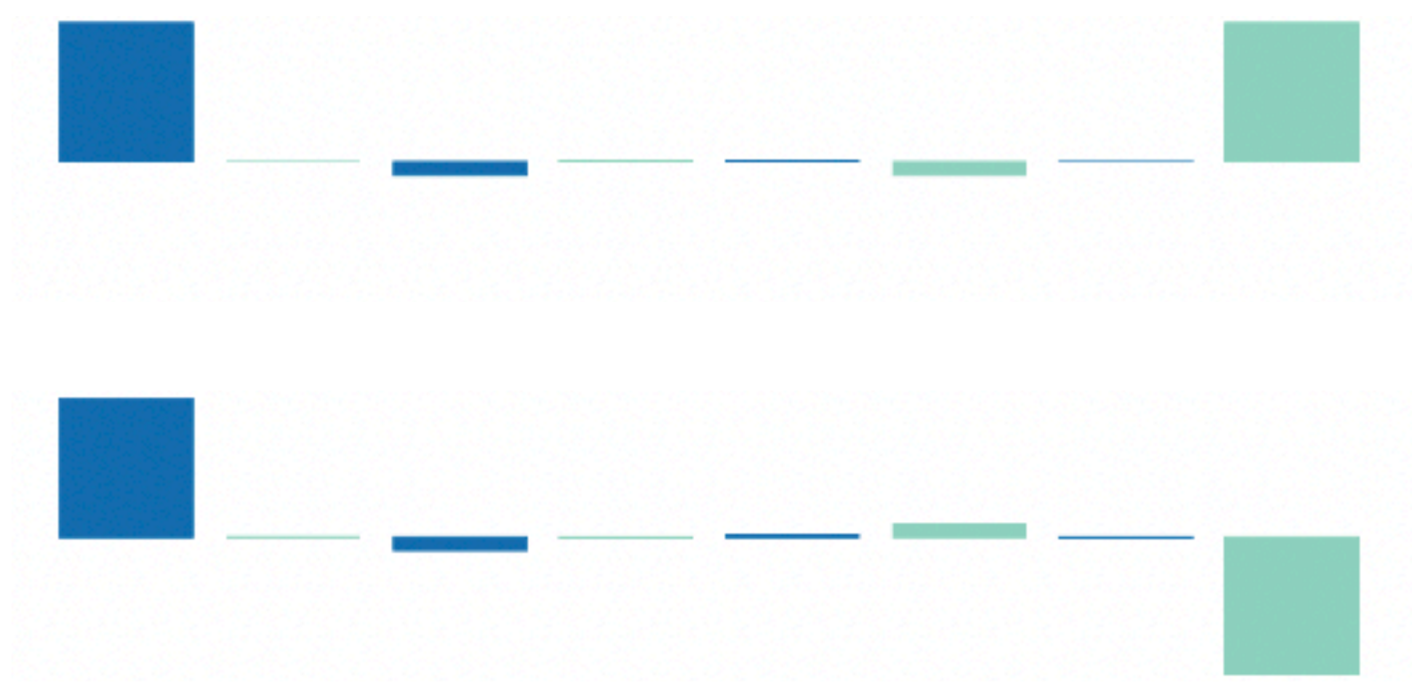


Classification: a mini example

Wave-function amplitude at spatial position



Visual representation



Classification: a mini example

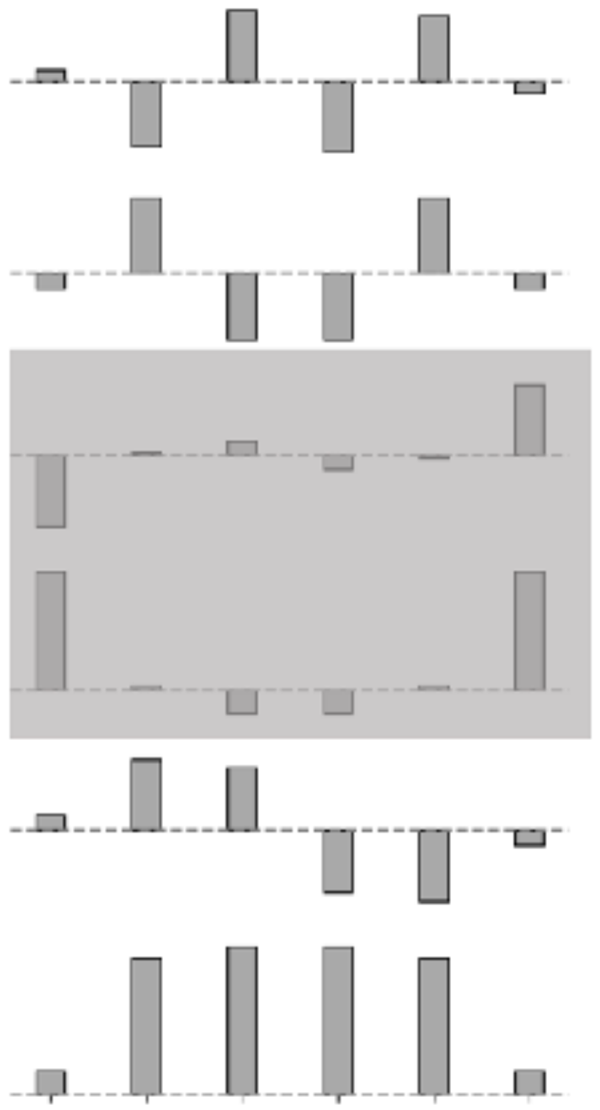
Label: p(x) $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$

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))

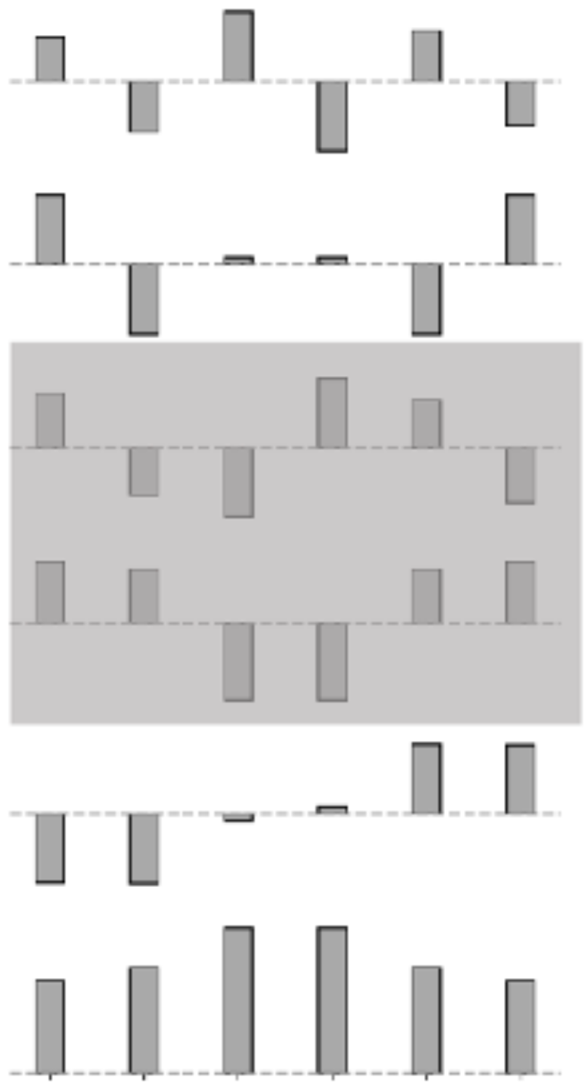
!the loss is minimal if q(x) matches the labels!

Back to physics

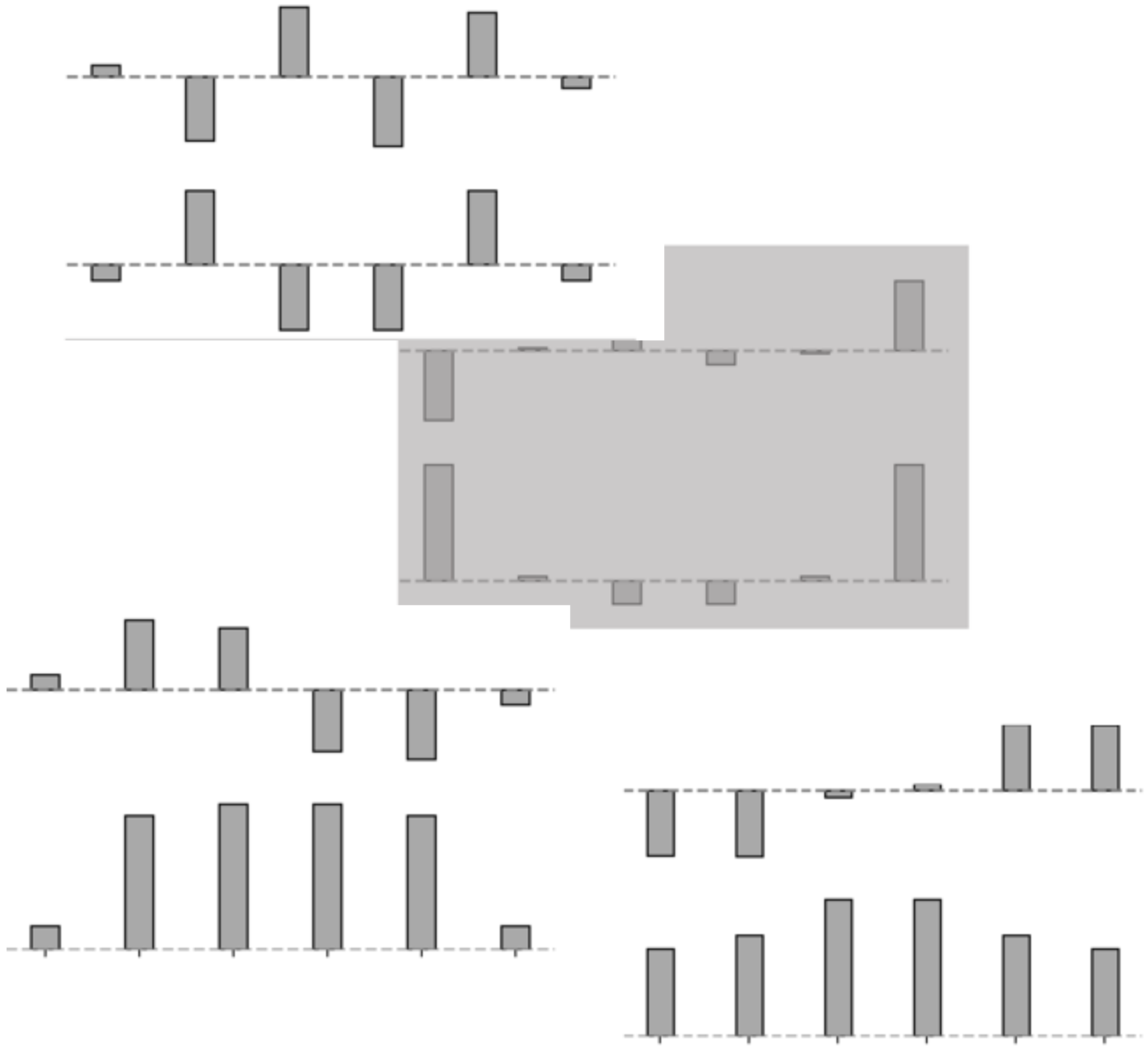


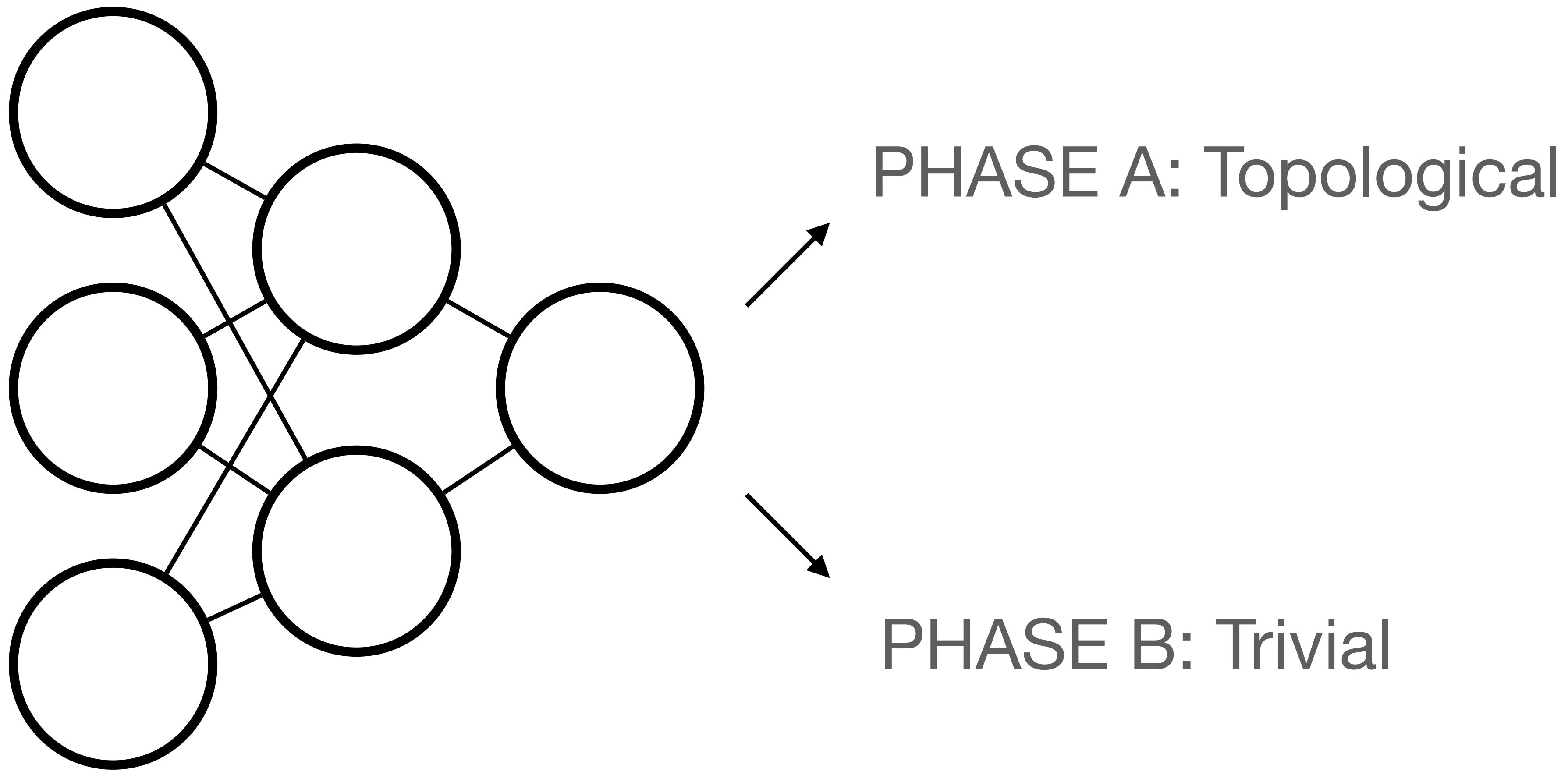
Topological

Trivial



Notebook 1: Supervised learning

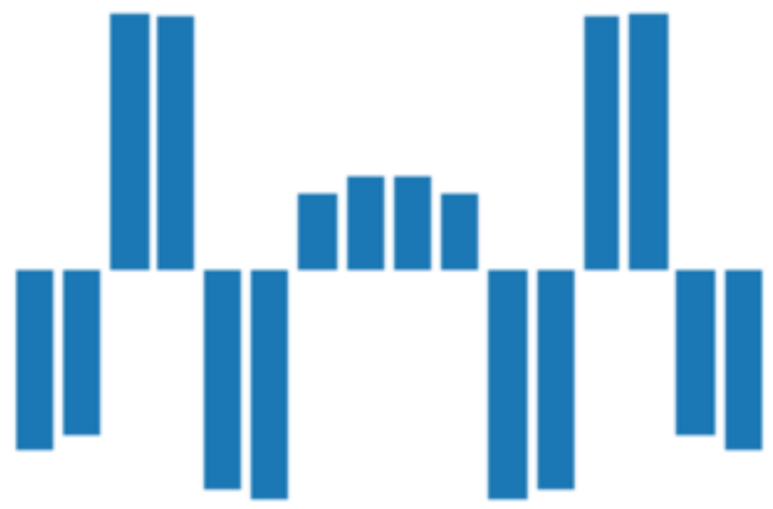




Quiz:SSH Human Classification

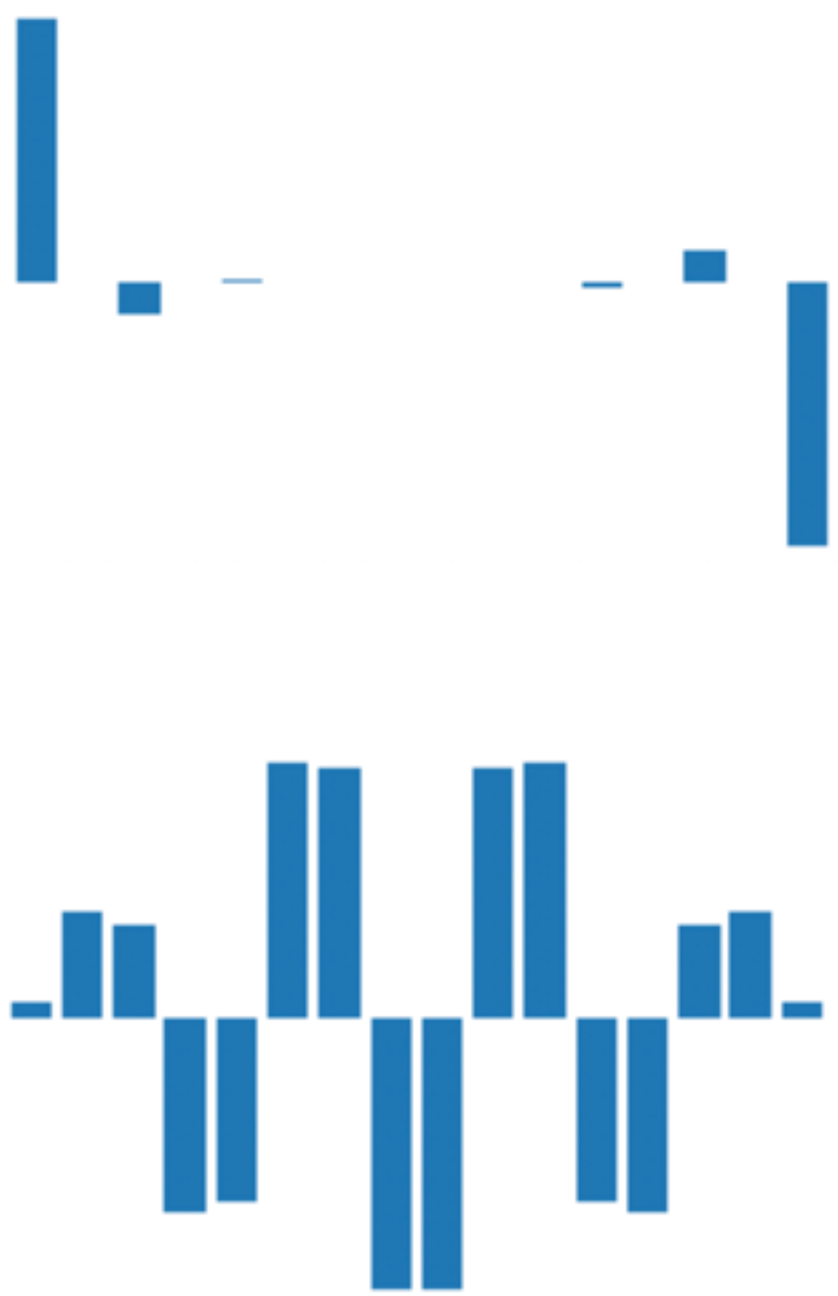
A

B



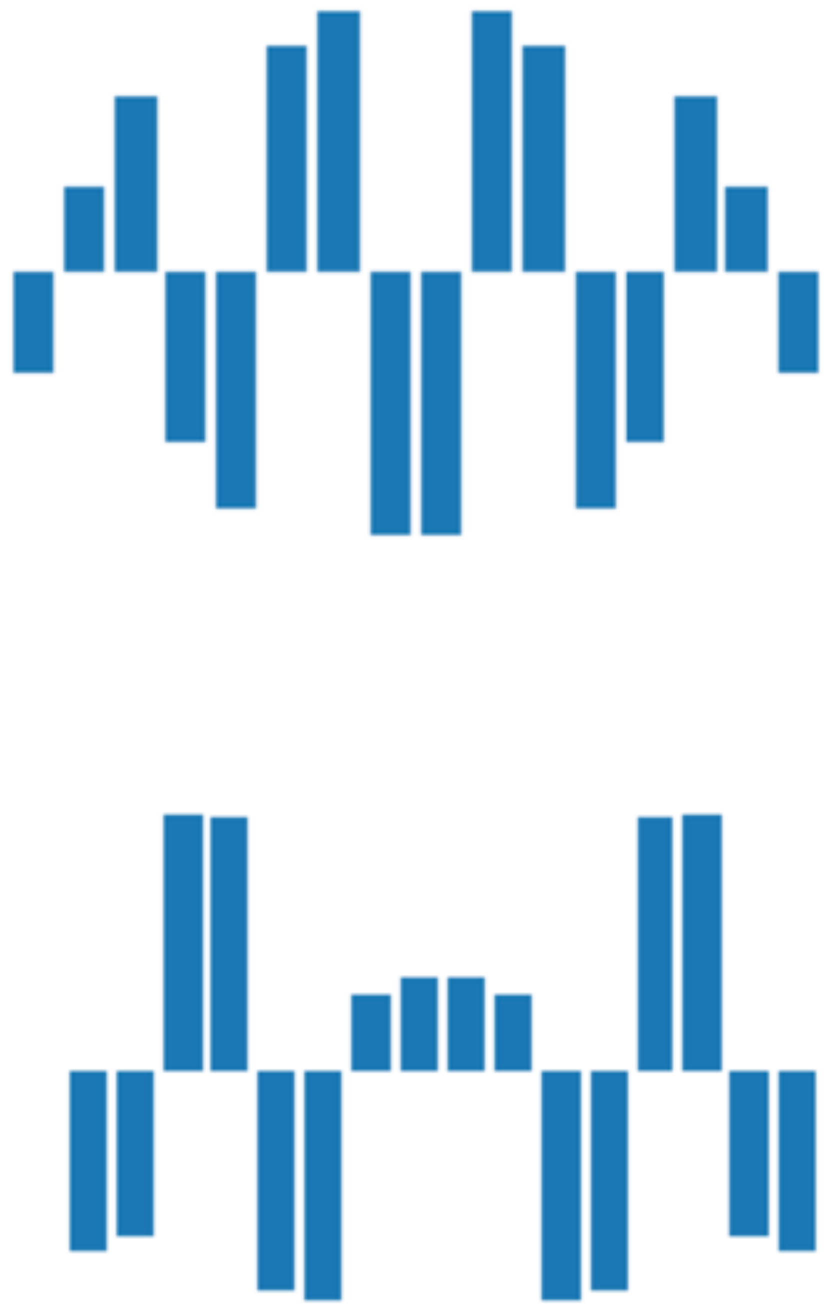
Quiz: SSH Human Classification

A



PHASE A: Topological

B



PHASE B: Trivial

Notebook 1: Supervised learning

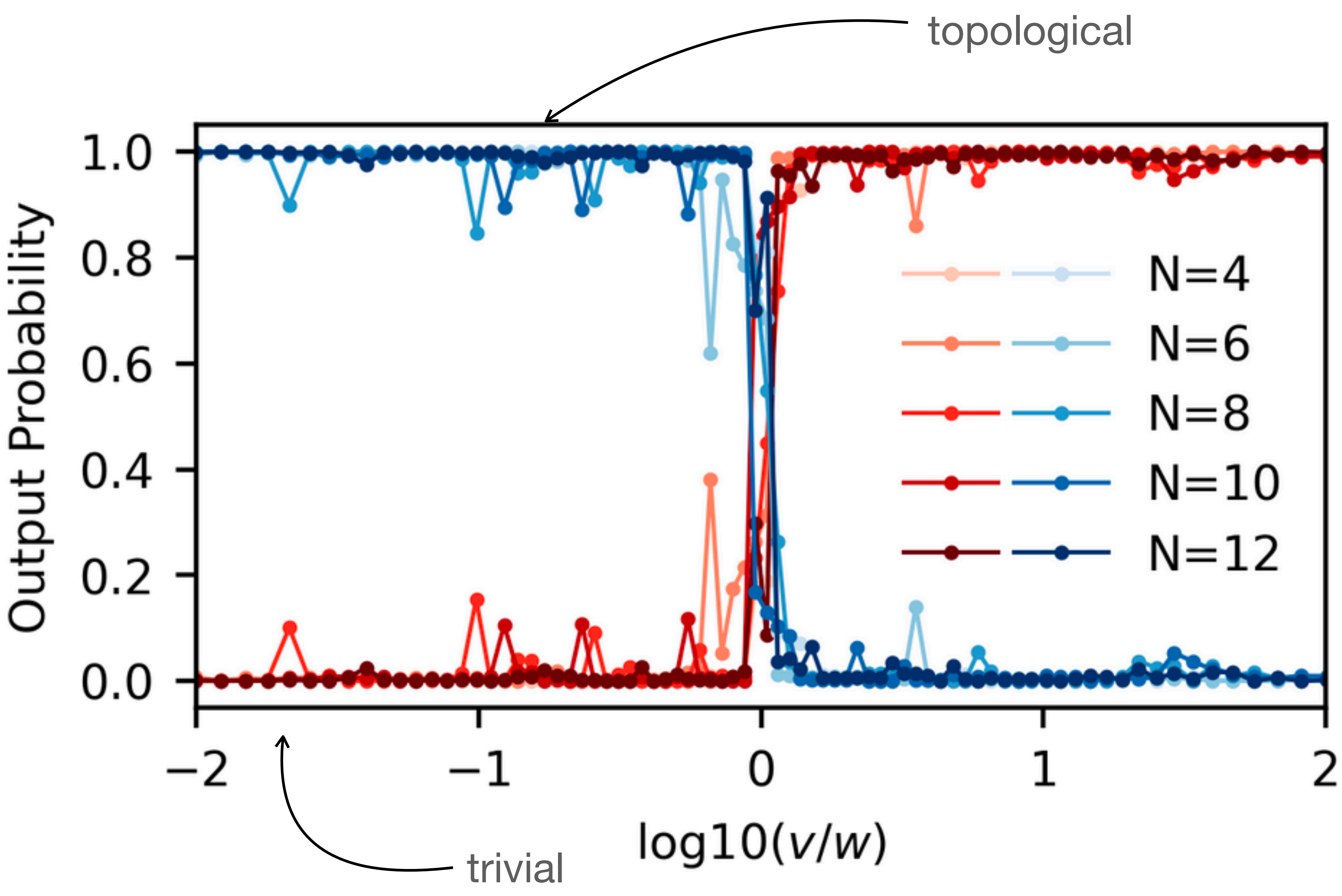
INPUT: 16 sites x 16 eigenstates = 256

```
Using cpu device
NeuralNetwork(
  (flatten): Flatten()
  (linear_relu_stack): Sequential(
    (0): Linear(in_features=256, out_features=32, bias=True)
    (1): ReLU()
    (2): Linear(in_features=32, out_features=16, bias=True)
    (3): ReLU()
    (4): Linear(in_features=16, out_features=2, bias=True)
  )
)
```

2 HIDDEN LAYERS

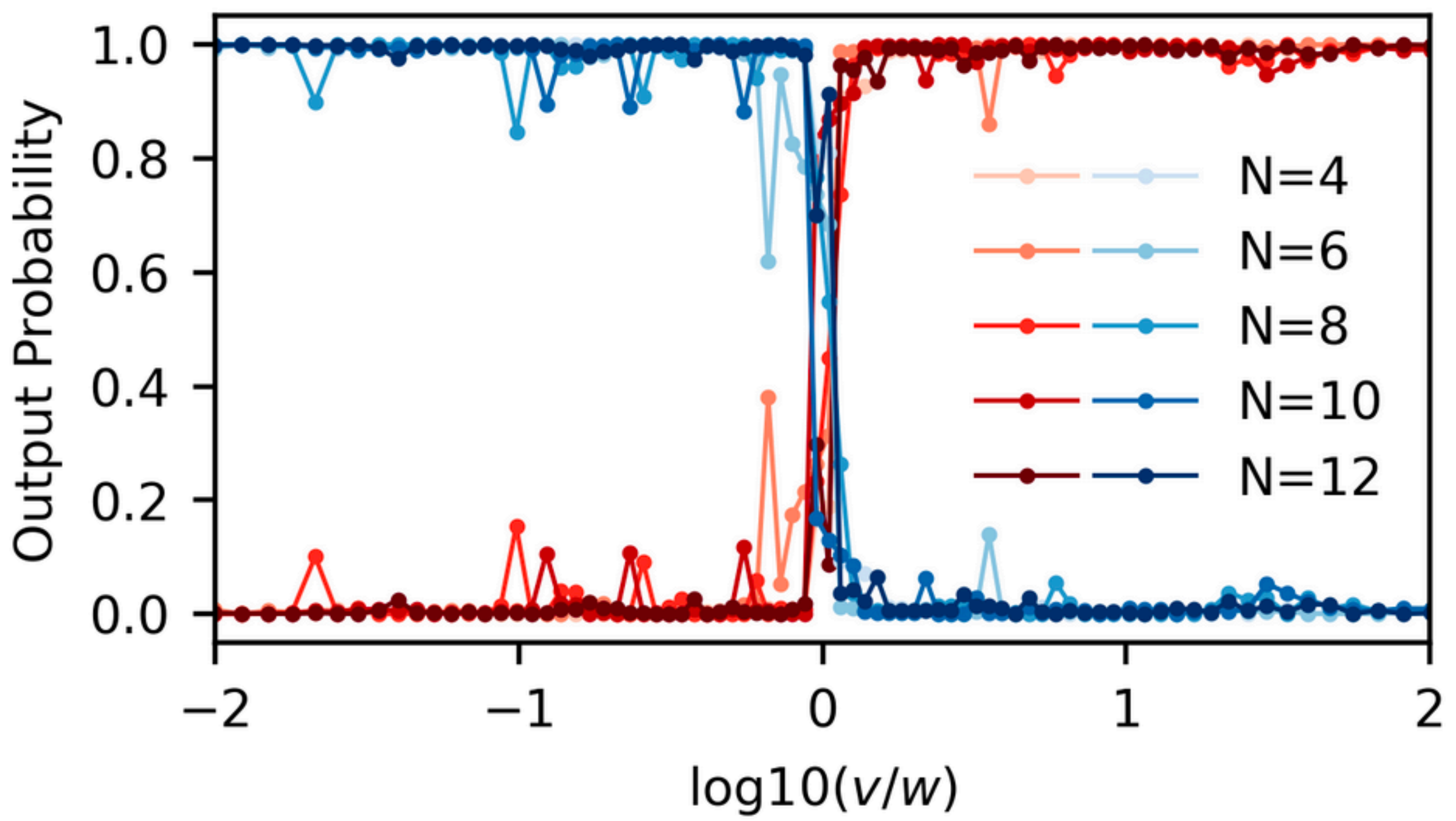
2 OUTPUTS: topo and trivial

How does NN learn the phase transition?

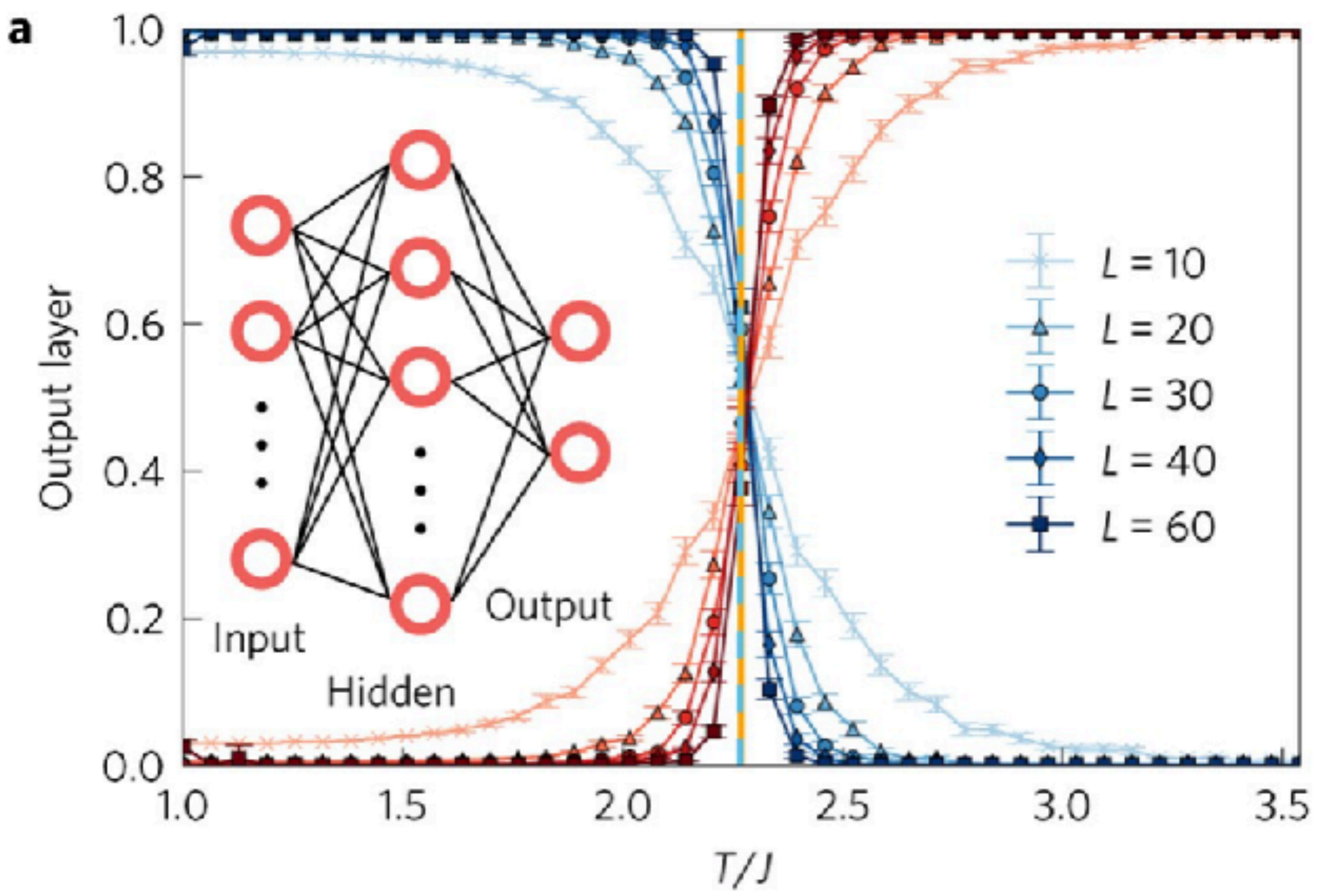


QMAI | TU Delft

How does NN learn the phase transition?



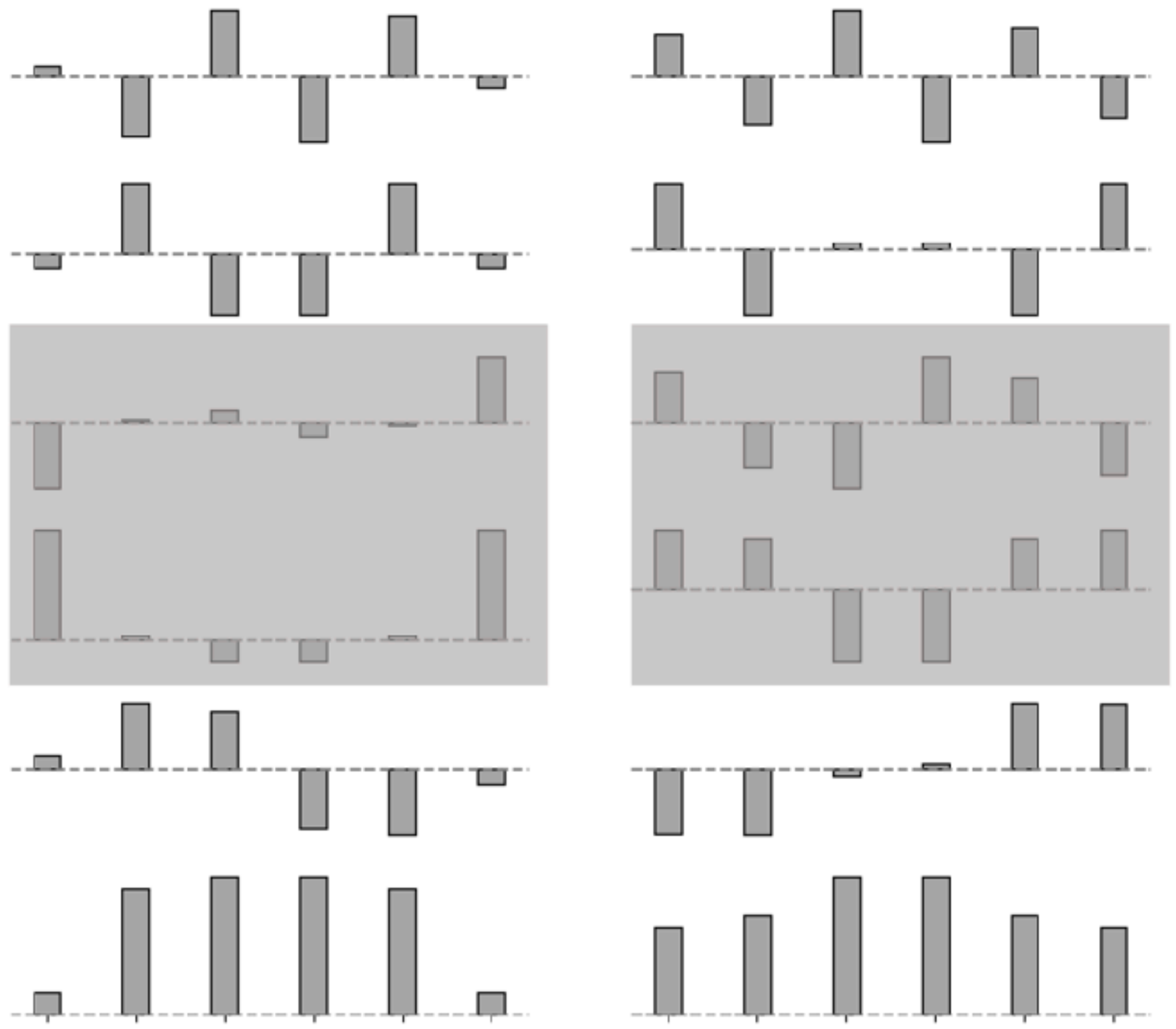
From your exercise notebook



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

Let's make the classification more challenging

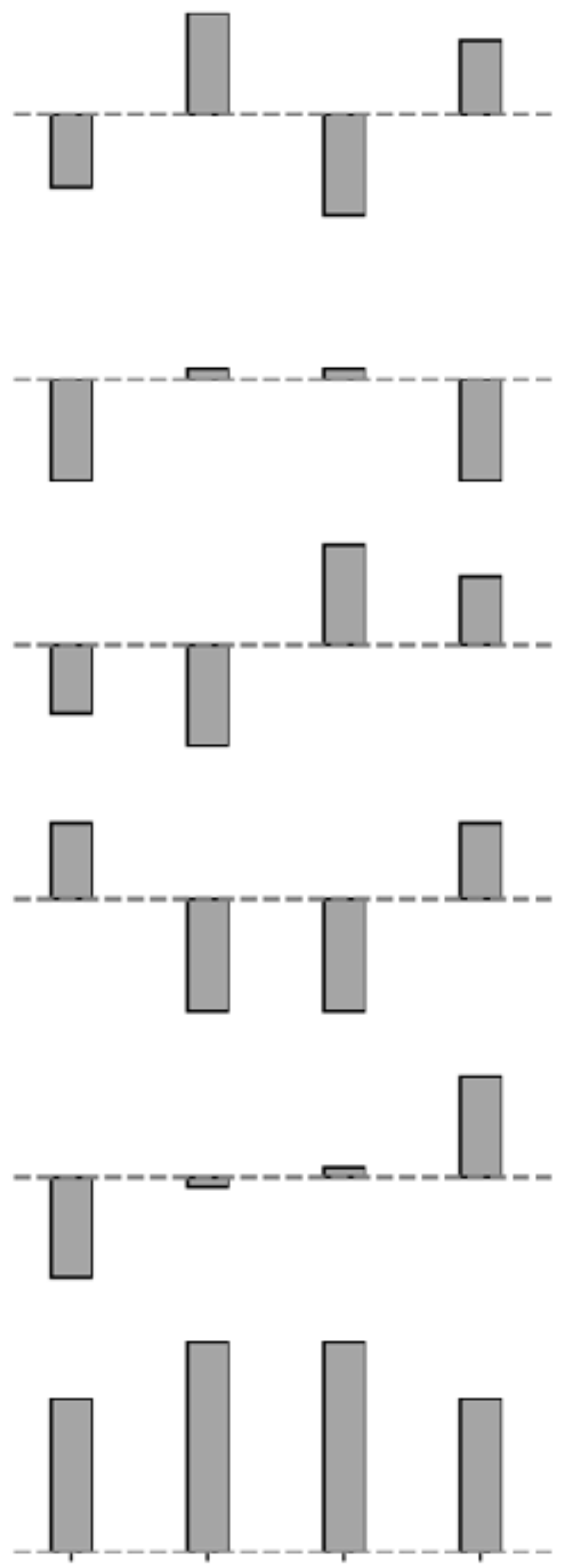
What about the wave function allows you to immediately determine the topology?

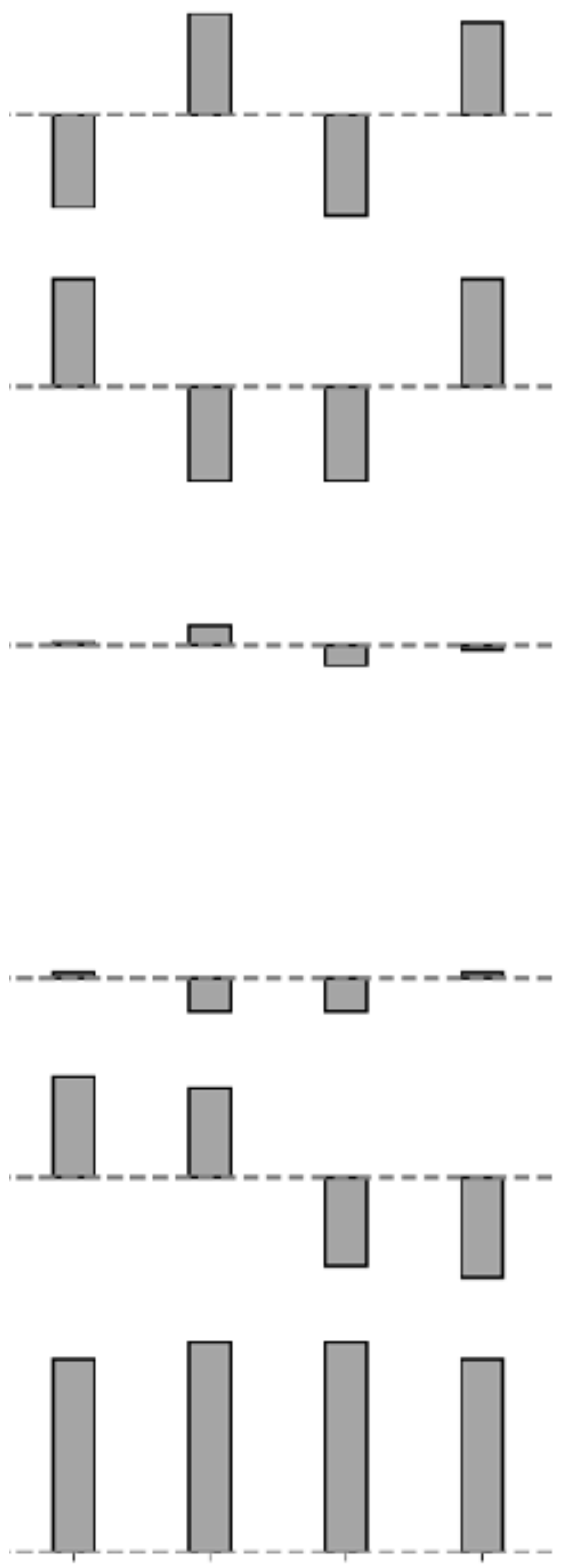


PHASE A: Topological PHASE B: Trivial

QMAI | TU Delft

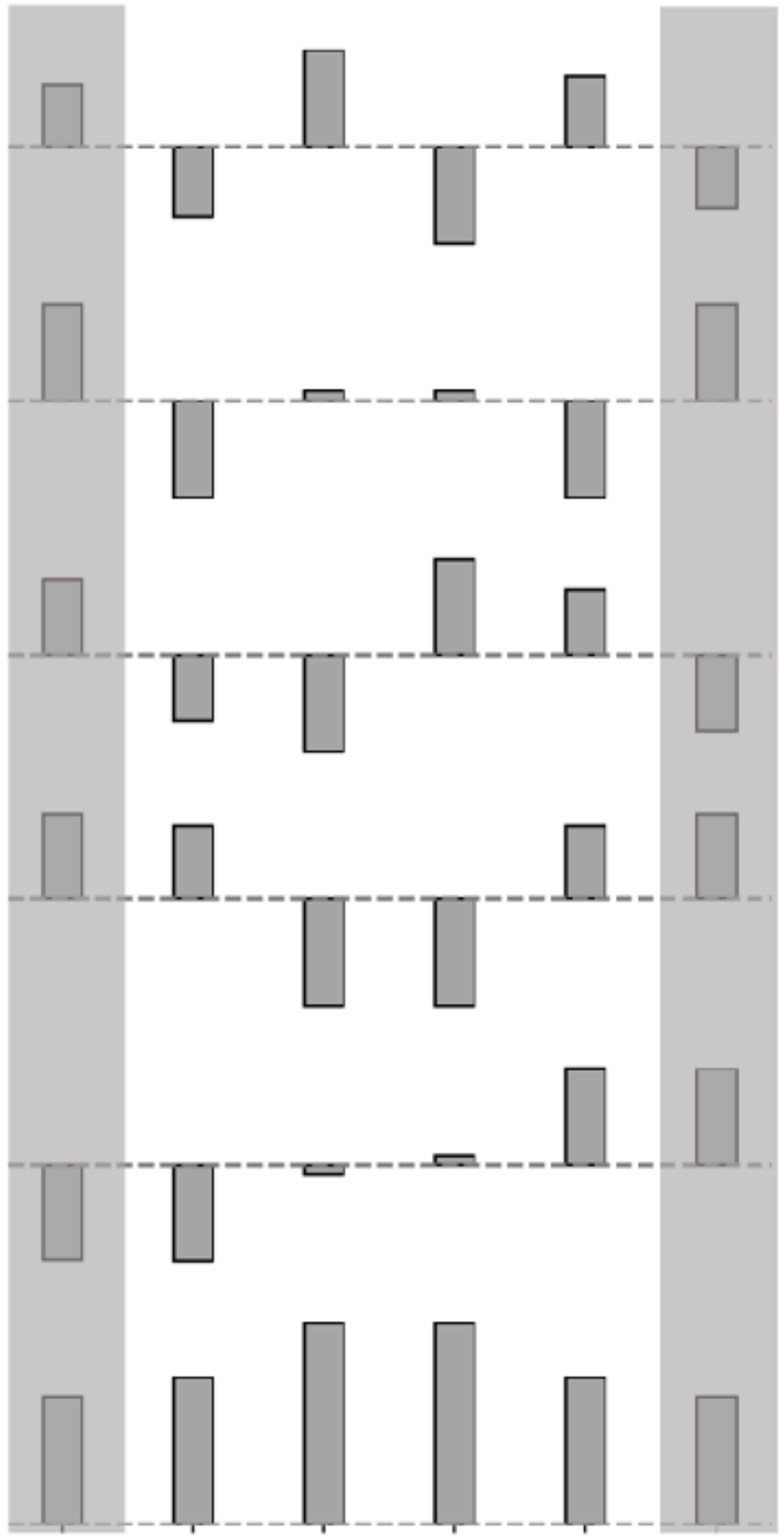
Quiz 2:SSH Human Classification



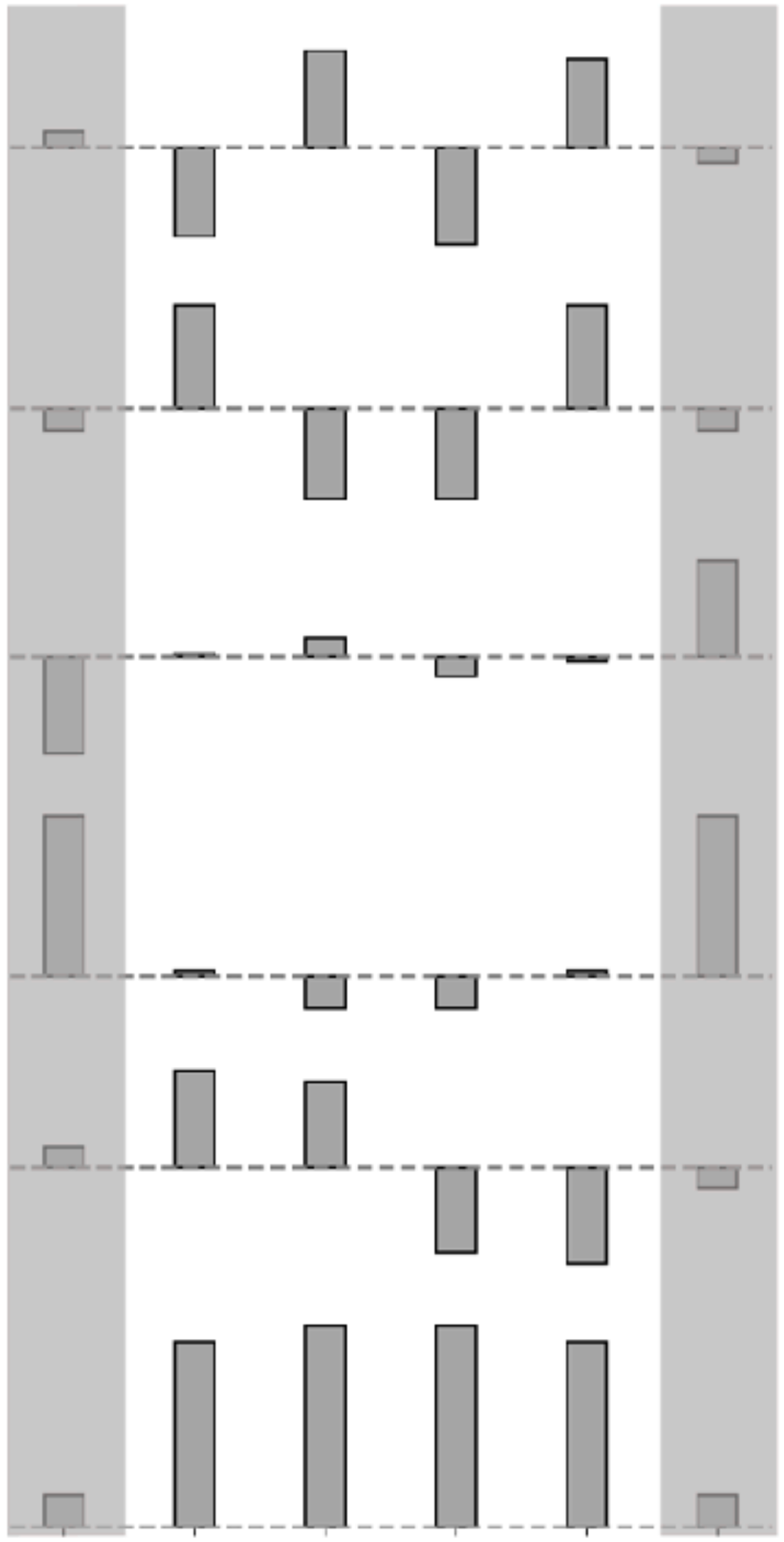


Quiz 2:SSH Human Classification

QMAI | TU Delft

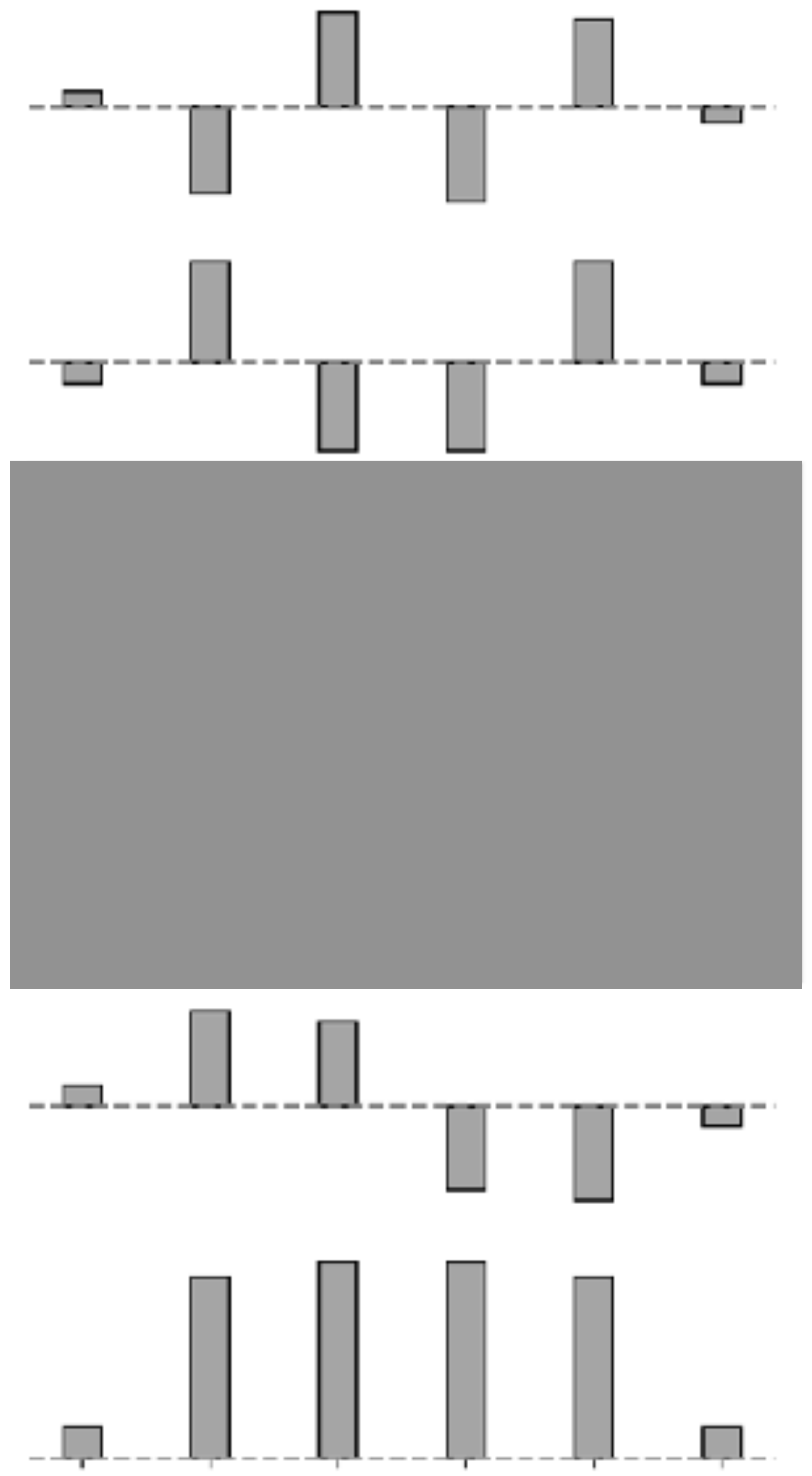


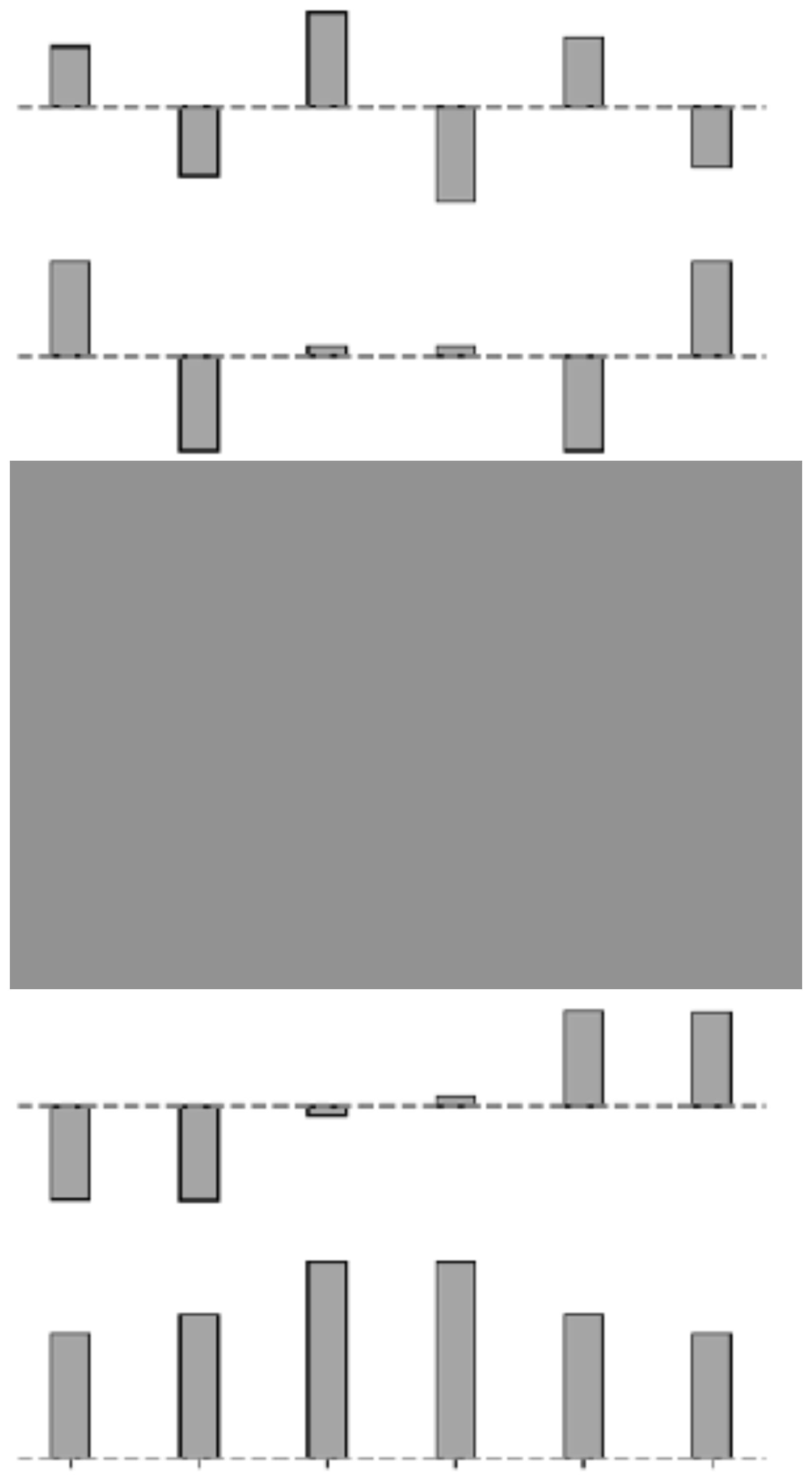
PHASE B: Trivial



PHASE A: Topological

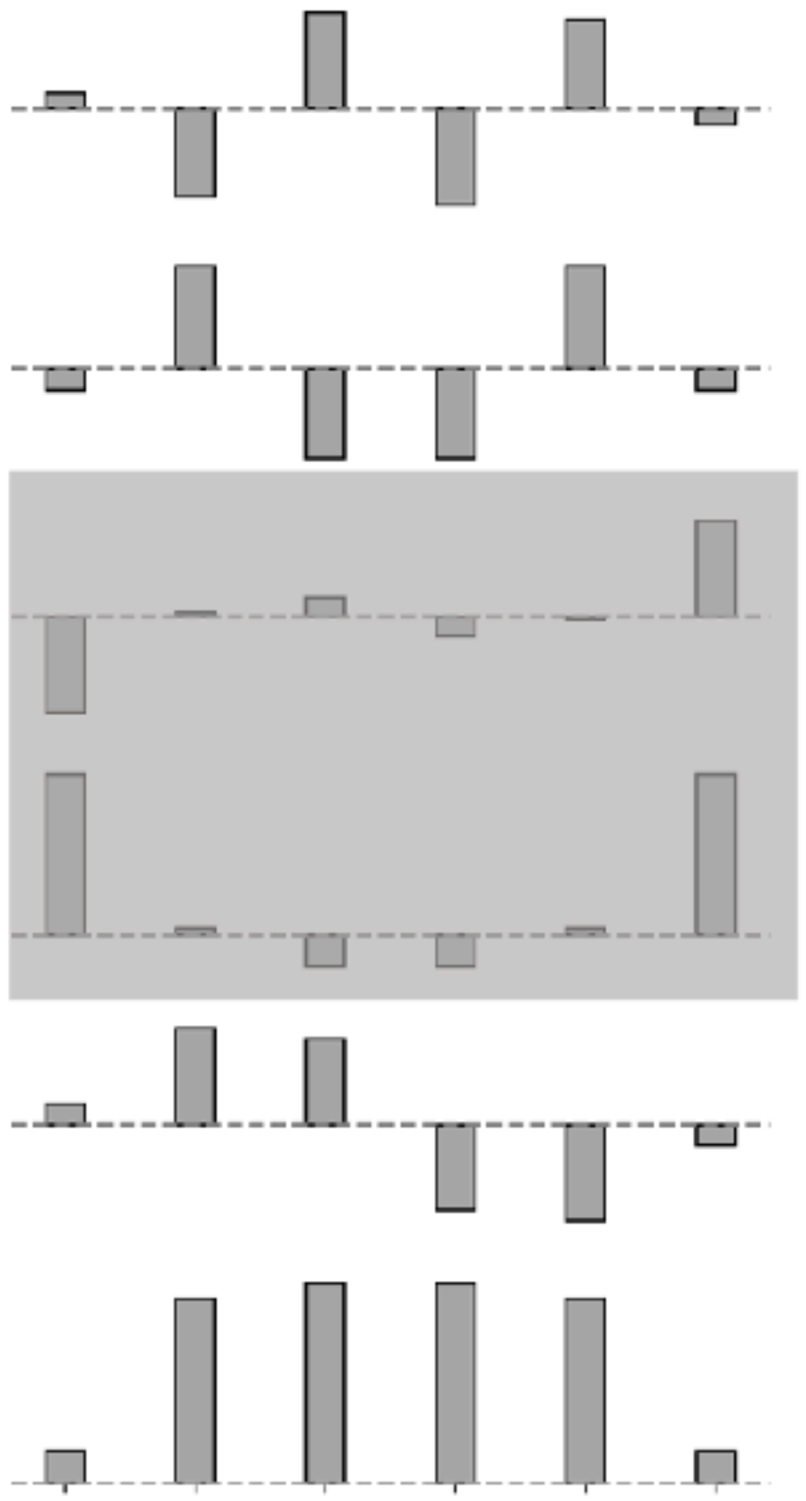
Quiz 3:SSH Human Classification



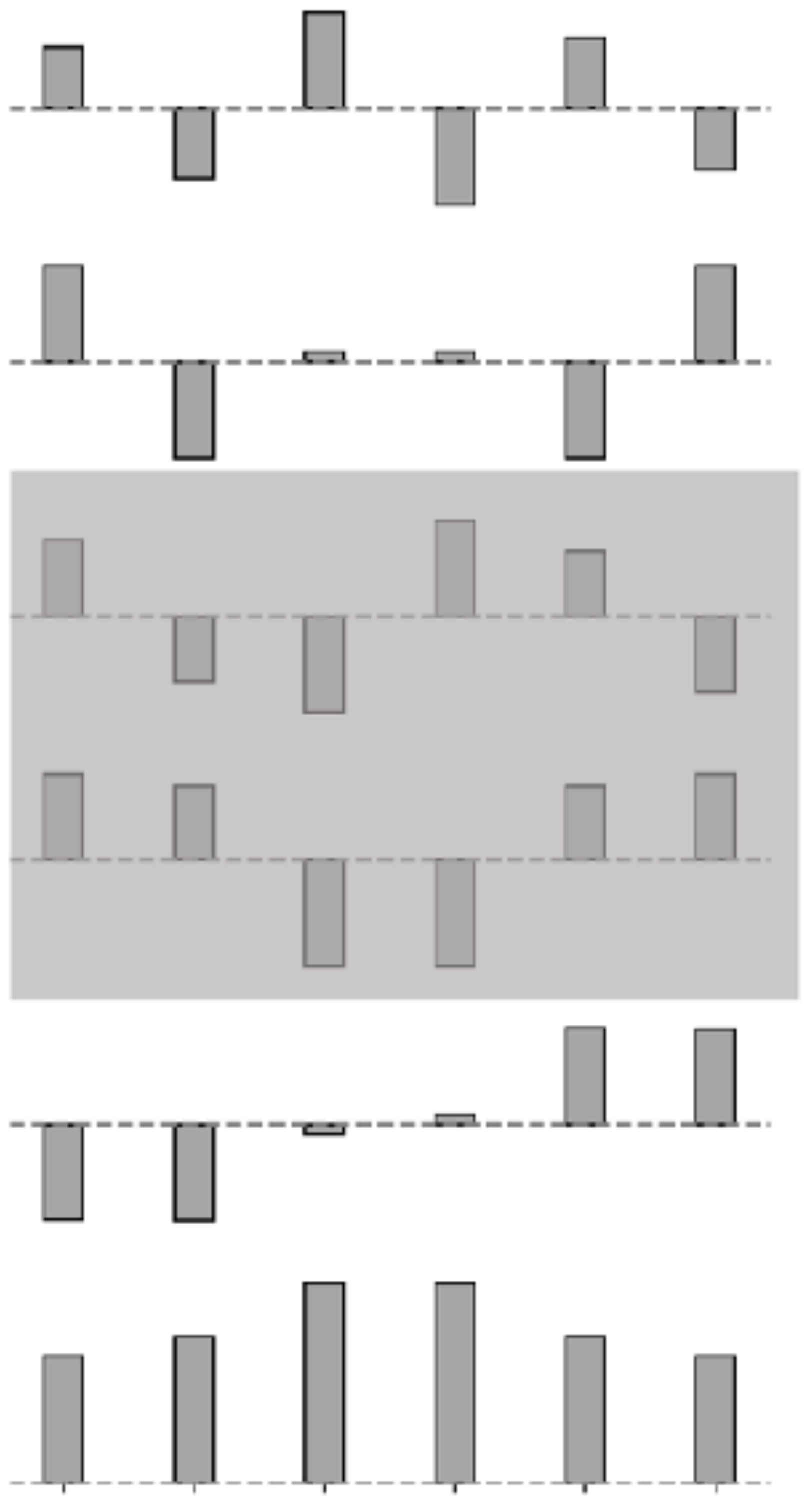


QMAI | TU Delft

Quiz 3:SSH Human Classification



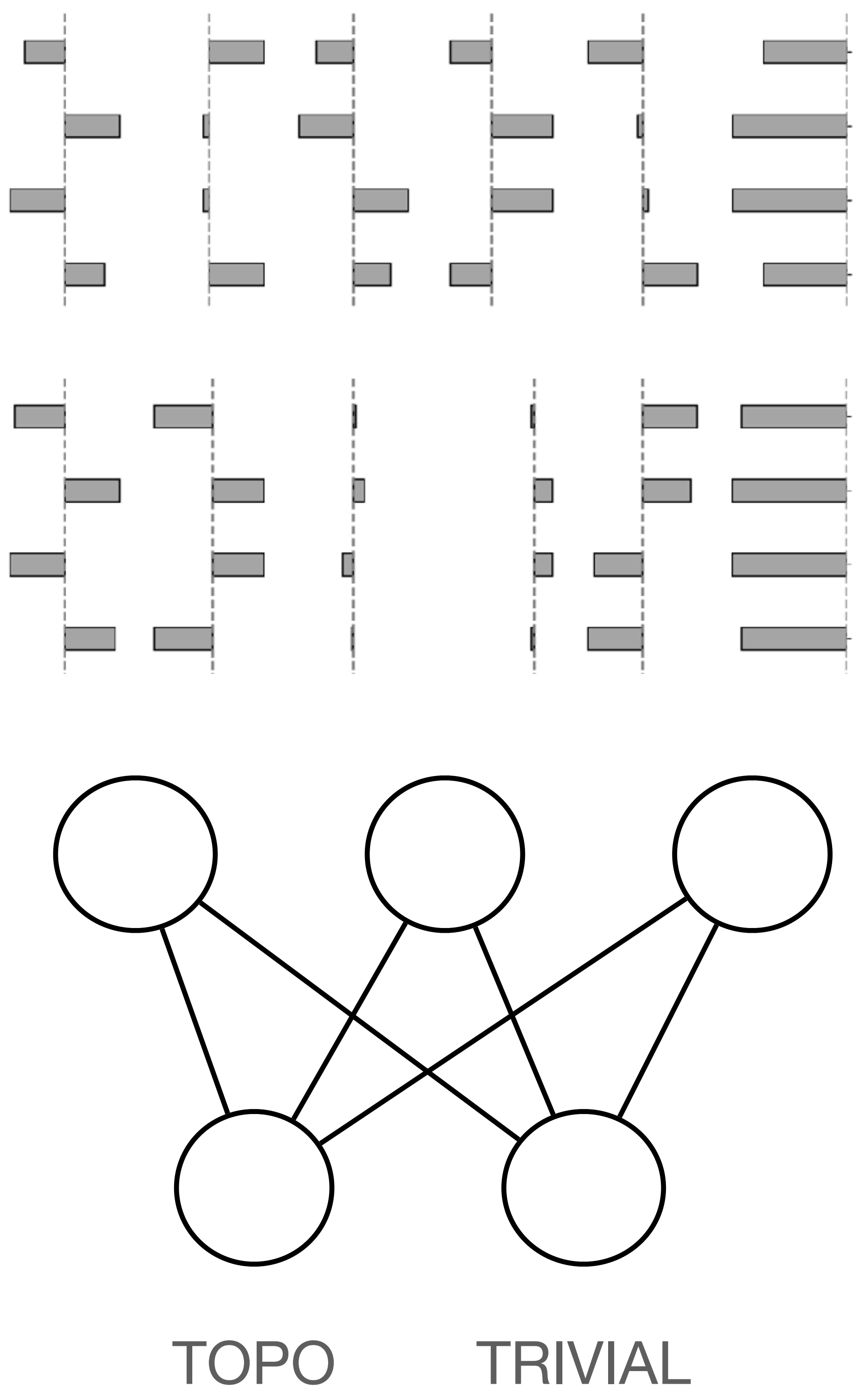
PHASE A: Topological



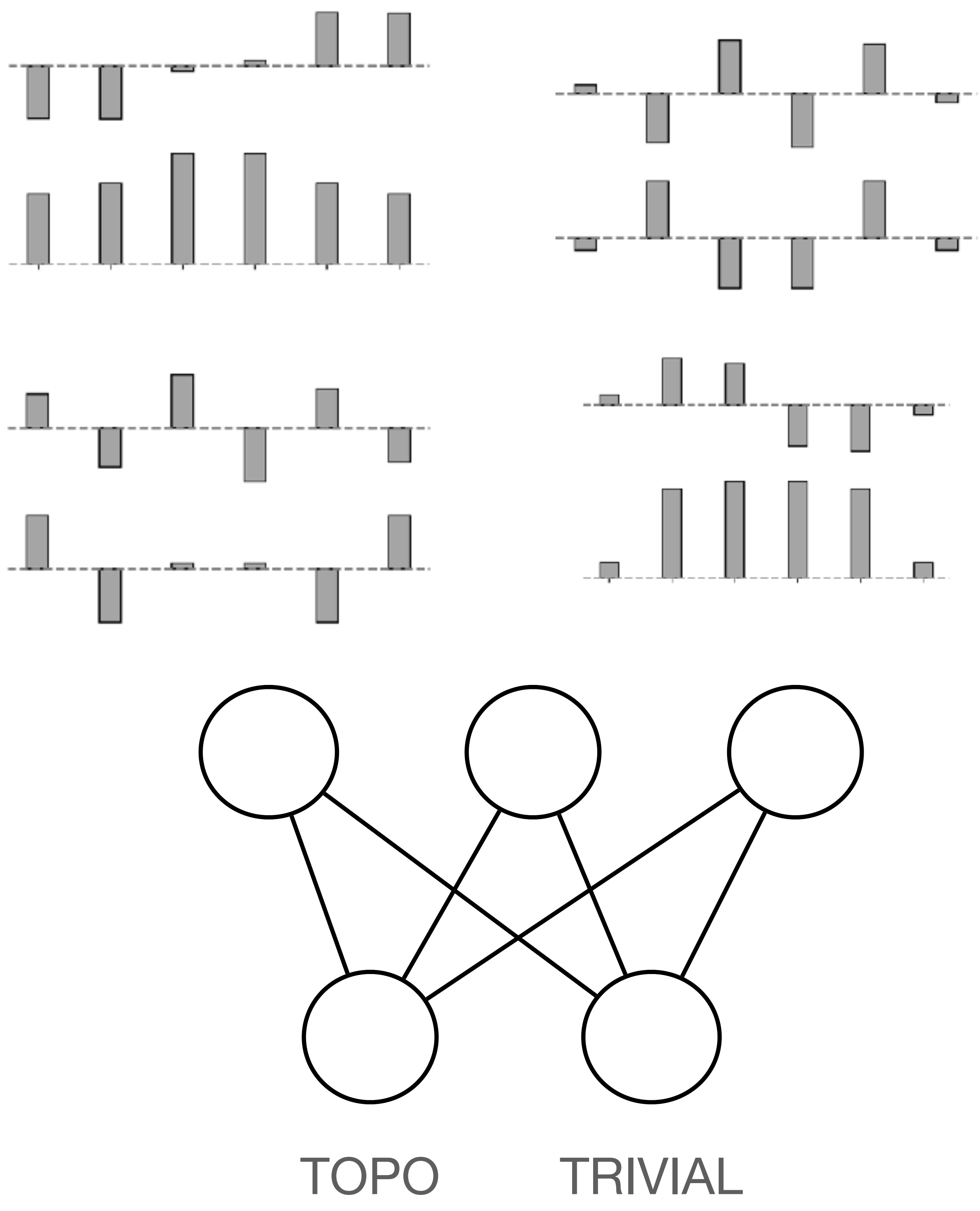
PHASE B: Trivial

Notebook 2: Classification with limited data

Data without edge sites



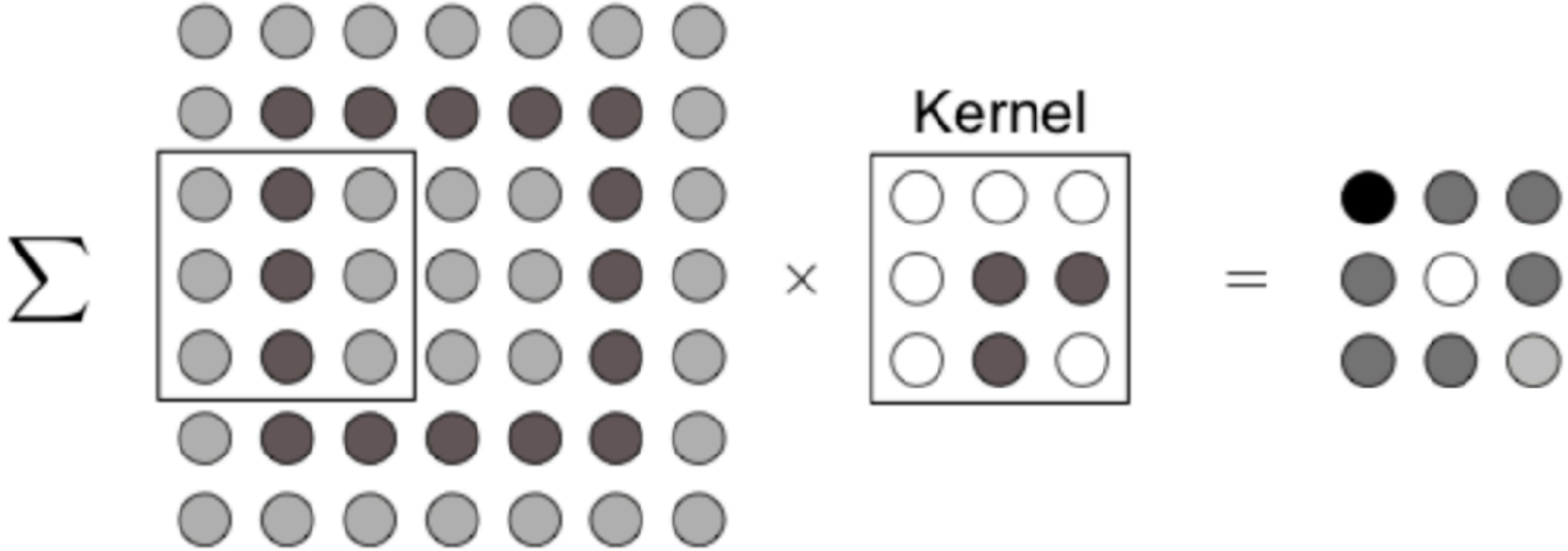
Data without edge modes



Fine-tuning your network

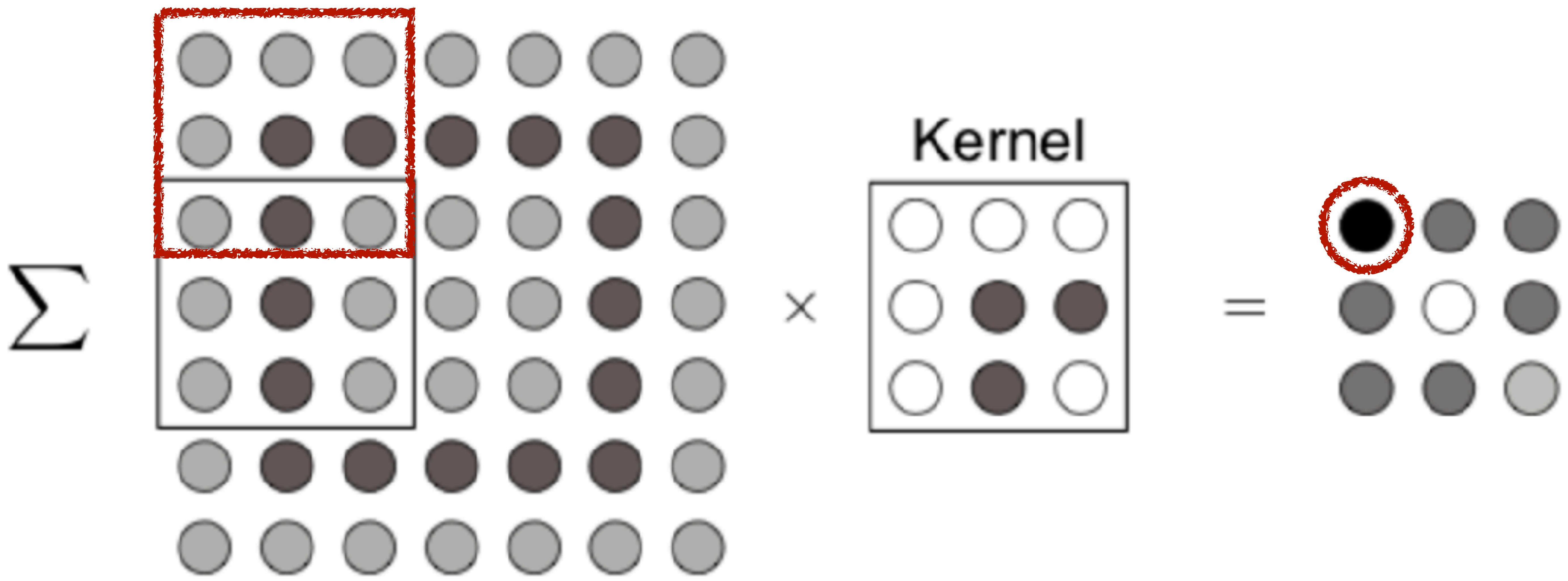
- Add different types of layers

CONVOLUTIONs:



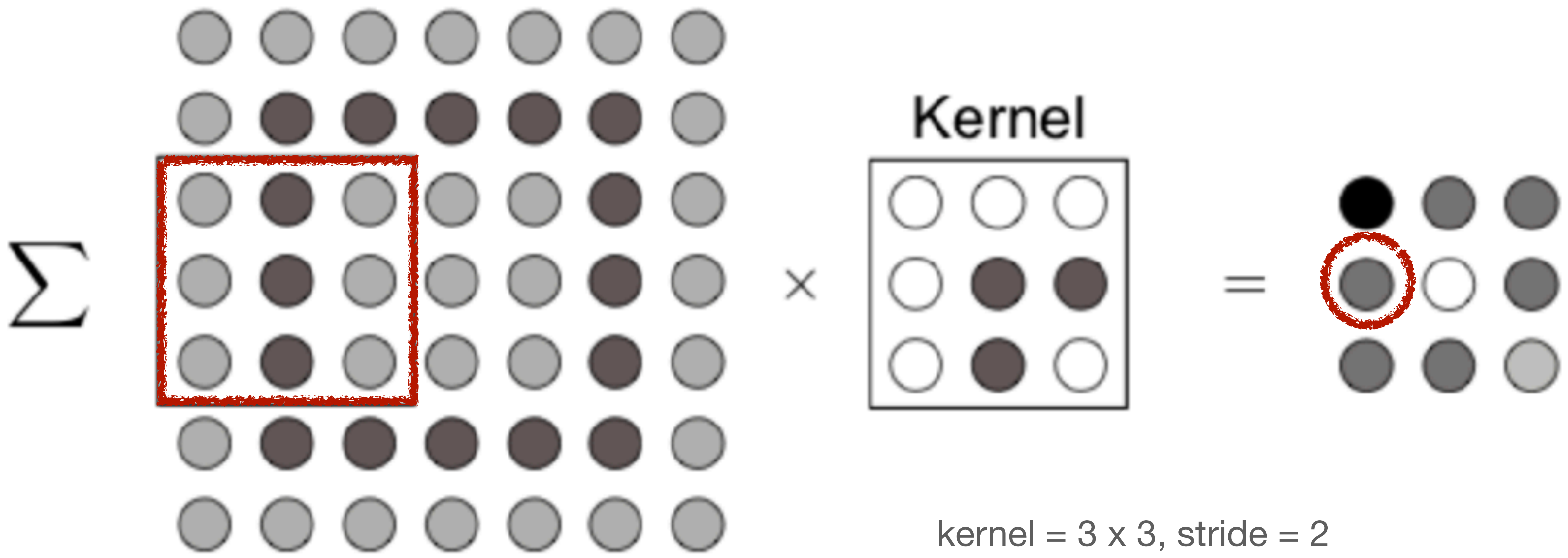
Fine-tuning your network

- Add different types of layers



Fine-tuning your network

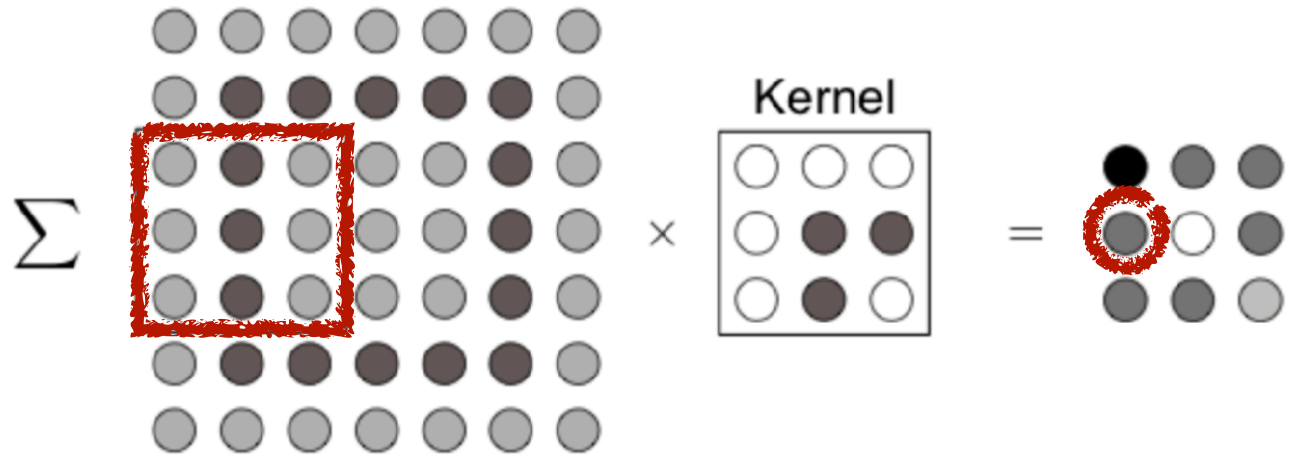
- Add different types of layers



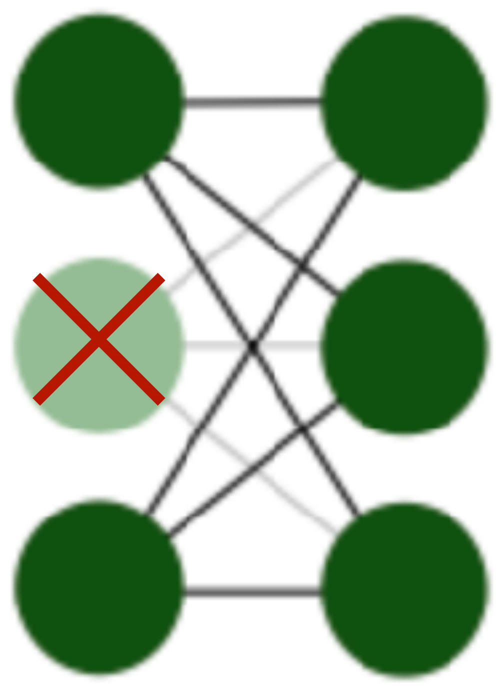
kernel = 3 x 3, stride = 2

Fine-tuning your network

- Add different types of layers

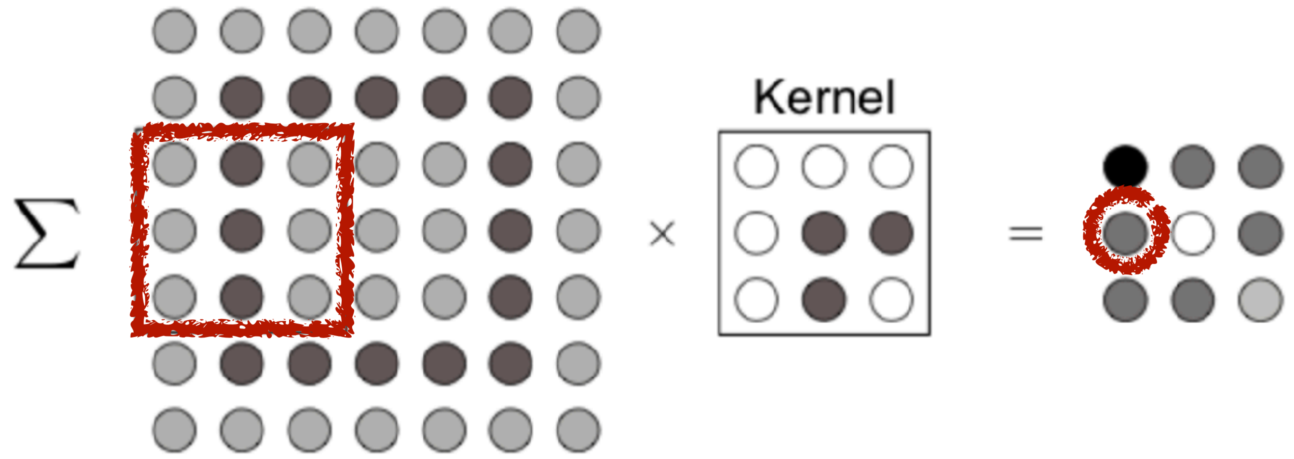


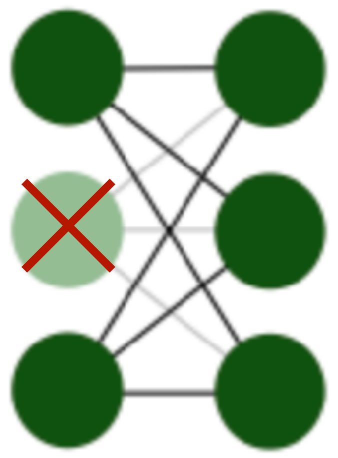
DROPOUT:



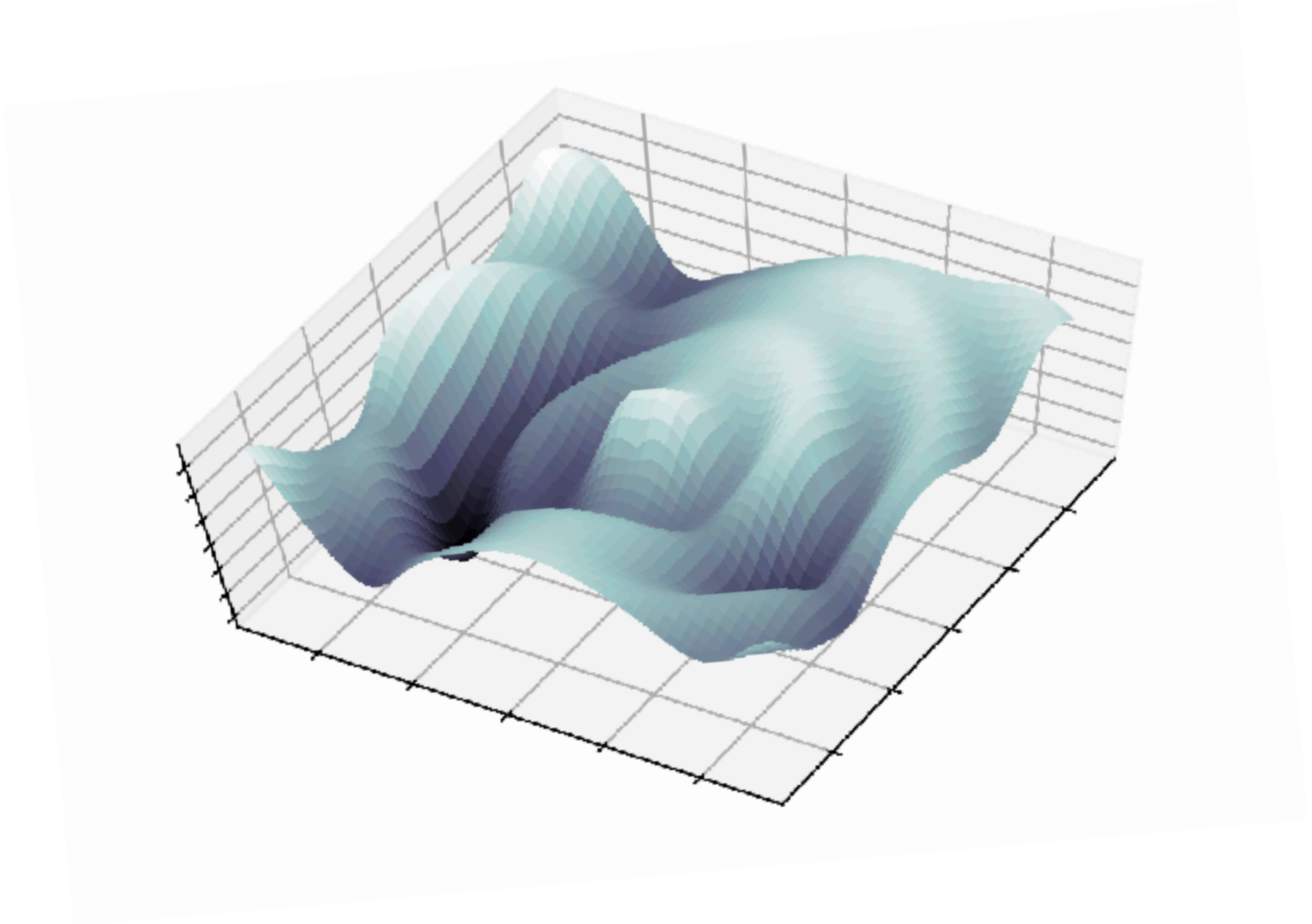
Fine-tuning your network

- Add different types of layers



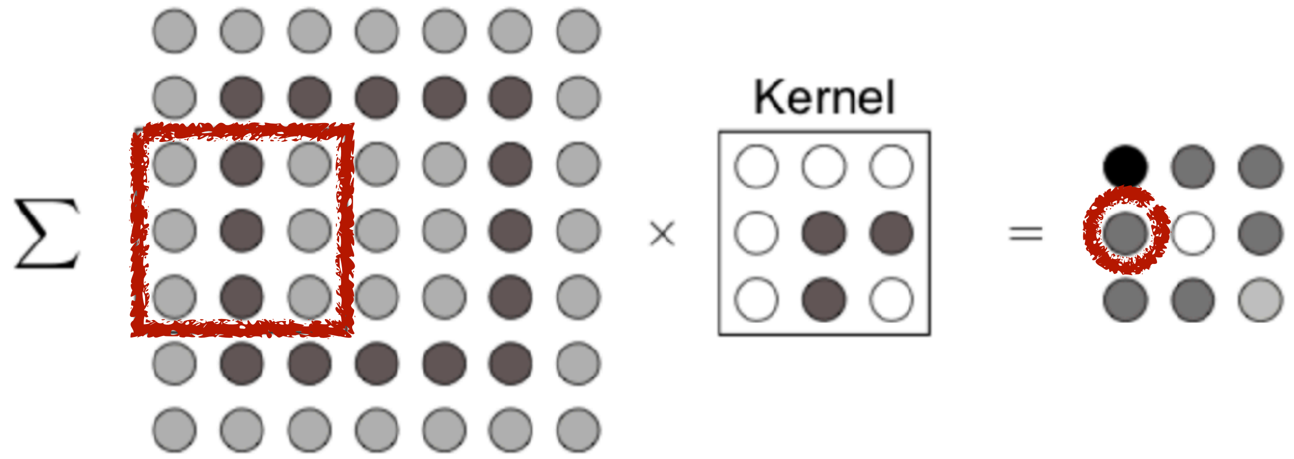


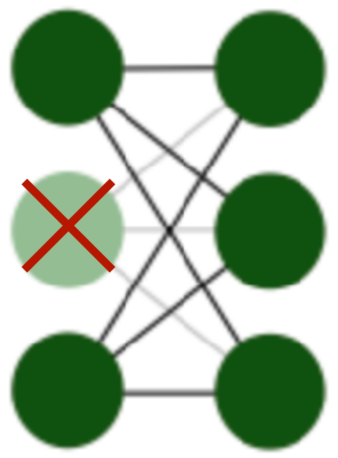
- Vary learning rate



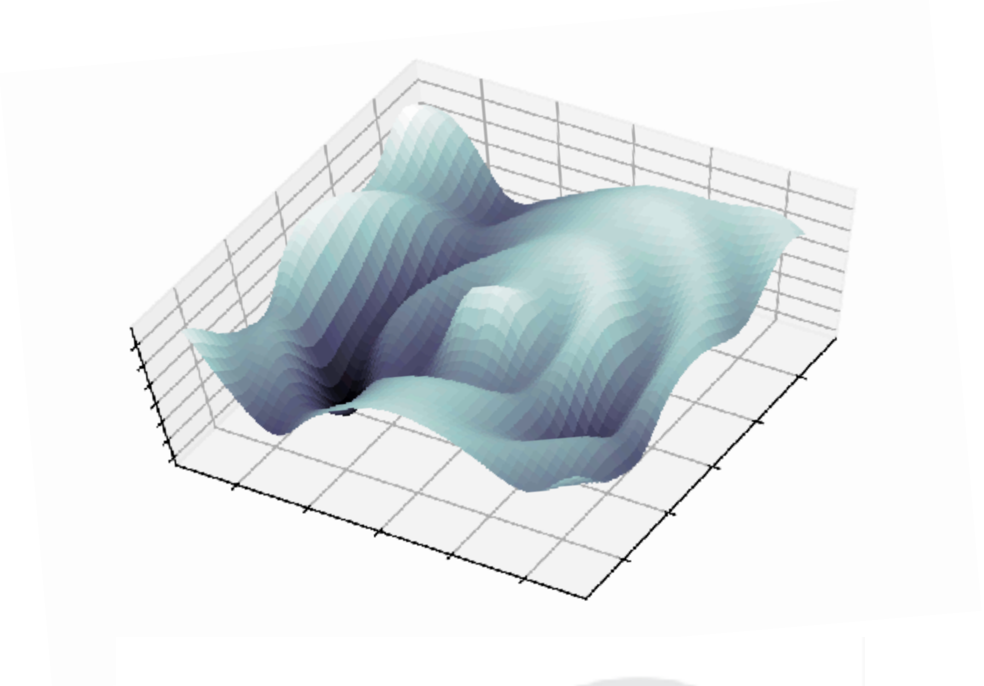
Fine-tuning your network

- Add different types of layers

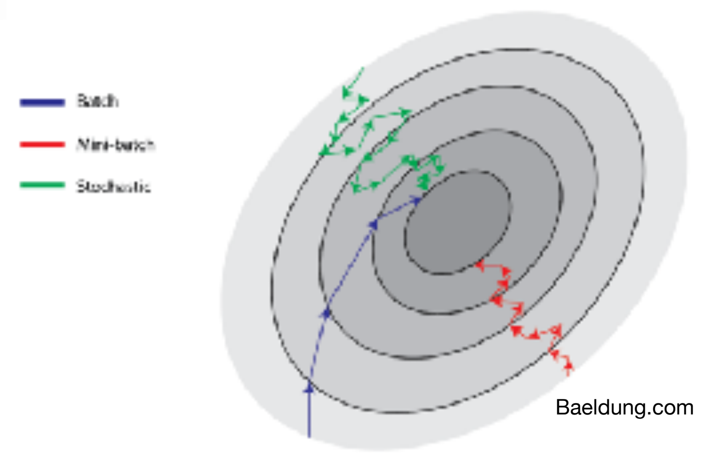




- Vary learning rate



- Vary batch size



More tips and links at the end of Notebook 1!

QMAI | TU Delft

Fine-tuning your network

https://tms-dipc.org/hands-on-information/

Topological Matter School DIPC 2022: Notebook 1

In the lecture we learned how topology can be used to stabilize entaglement in the quantum computing context. In this notebook we will get hands-on experience diagonalizing Su-Schrieffer-Heeger model and learn to classify its phases: topological and trivial one.

Authors: Eliska Greplova, Guliuxin Jin, Naoual El Yazidi

Structurally based on this PyTorch tutorial

```
[ ] # First we load the libraries
    import numpy as np
    import scipy as sp
    import matplotlib.pyplot as plt
    import matplotlib
    from sklearn.utils import shuffle

    ## Machine learning related libraries:
    import torch
    import torch.nn as nn             # base class used to develop all neural network models
    from torch.utils.data import DataLoader # easy and organized data loading to the ML model
    from torch.utils.data import Dataset #for nice loadable dataset creation
```

BREAK 20 mins

PyTorch

QMAI | TUDelft

Exercise Notebooks

https://tms-dipc.org/hands-on-information/

Notebook 1

- supervised learning topological and trivial SSH
- data preparation and loading
- build your own neural network and train it!!

Notebook 2

- supervised learning topological and trivial SSH with limited data
- data slicing info

You will work in team: MAKE GROUPS OF THREE PLEASE!