MACHINE LEARNING FOR QUANTUM MATTER

2023 Perimeter-SAIFR Journeys into Theoretical Physics

Juan Felipe Carrasquilla Álvarez, July 17th, 2023
Vector Institute

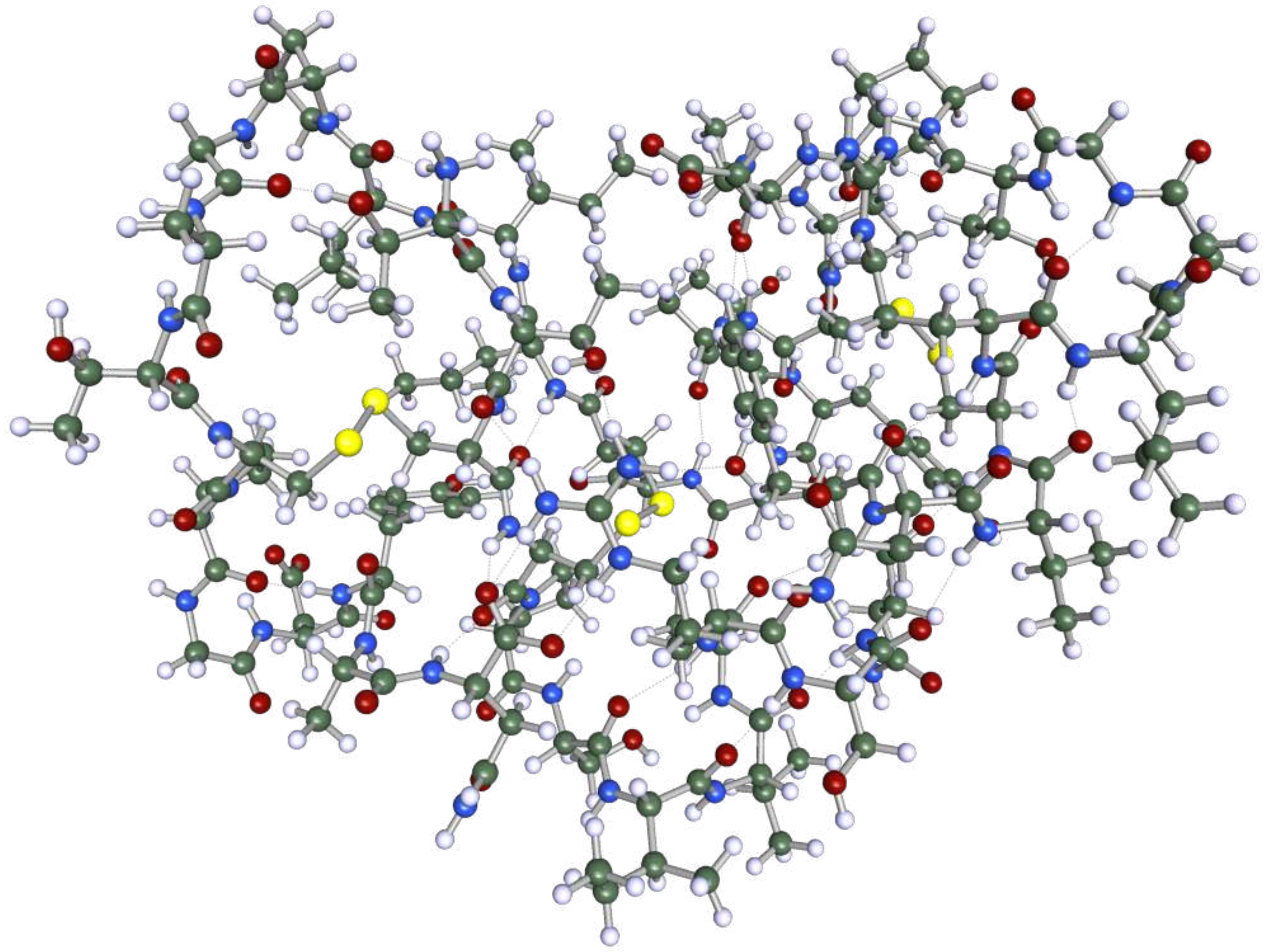
The many-body problem in quantum mechanics

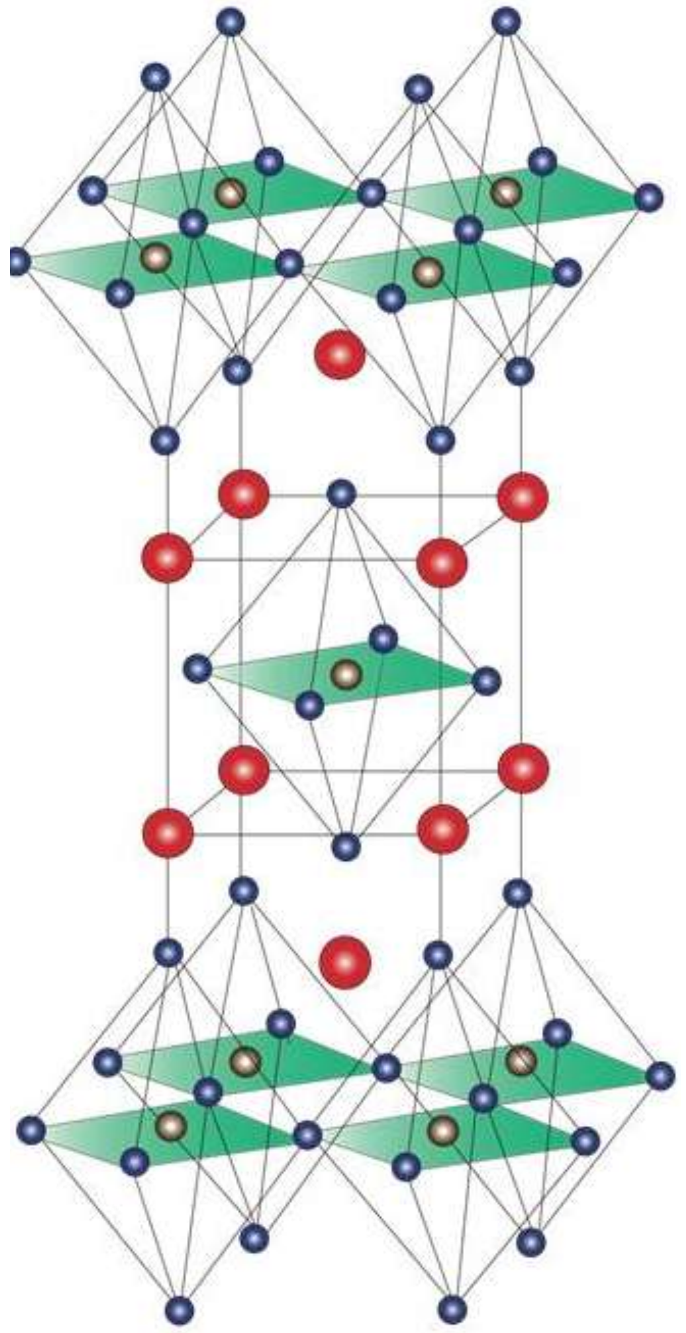
- Generic specification of a quantum state requires resources exponentially large in the number of degrees of freedom N

- Today's best supercomputers can solve the wave equation **exactly** for systems with a maximum of ~45 spins.

- Yet, technologically relevant problems in chemistry, condensed matter physics, and quantum computing are much larger than 45

$|\Psi\rangle$ vector with 2^N





There is still hope for classical algorithms

- Nature is sometimes compassionate: amount of information smaller than the maximum capacity — problems have structure and we exploit it

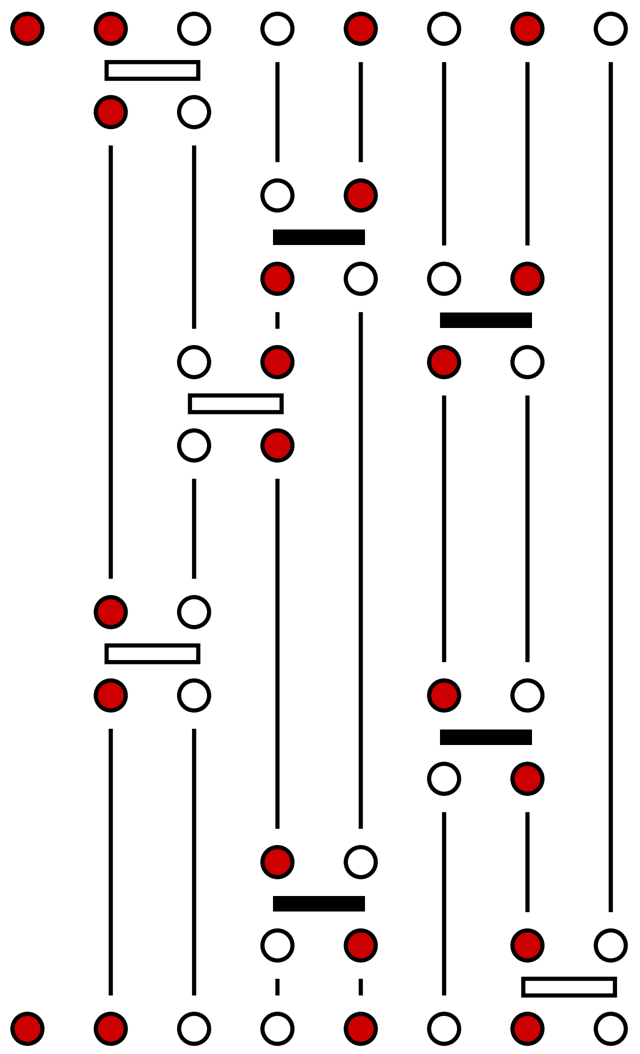
- Quantum Monte Carlo: stochastic exploration of most important regions of the gigantic state space.

- Tensor Networks: Exploit the fact that quantum states realized in nature have little entanglement

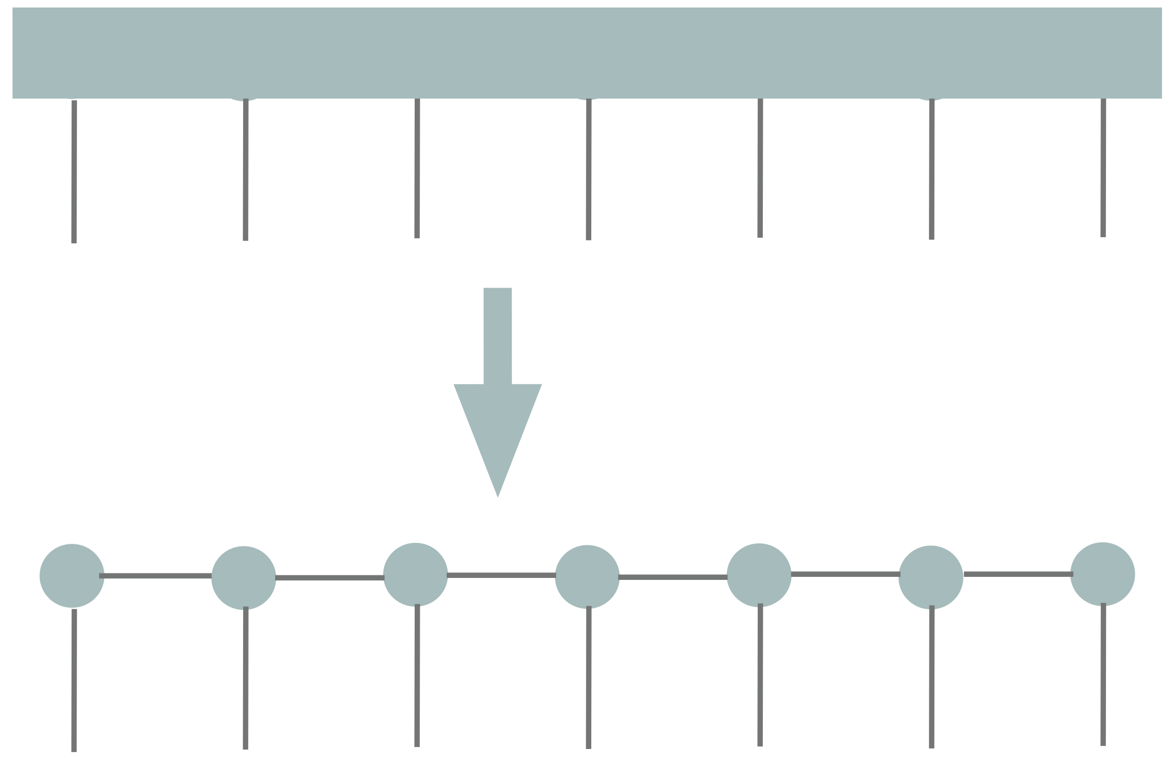
- Both techniques have led to profound implications to our understanding of condensed matter systems

$|\Psi\rangle$ vector with 2^N

QMC



Low entanglement: MPS and other TN



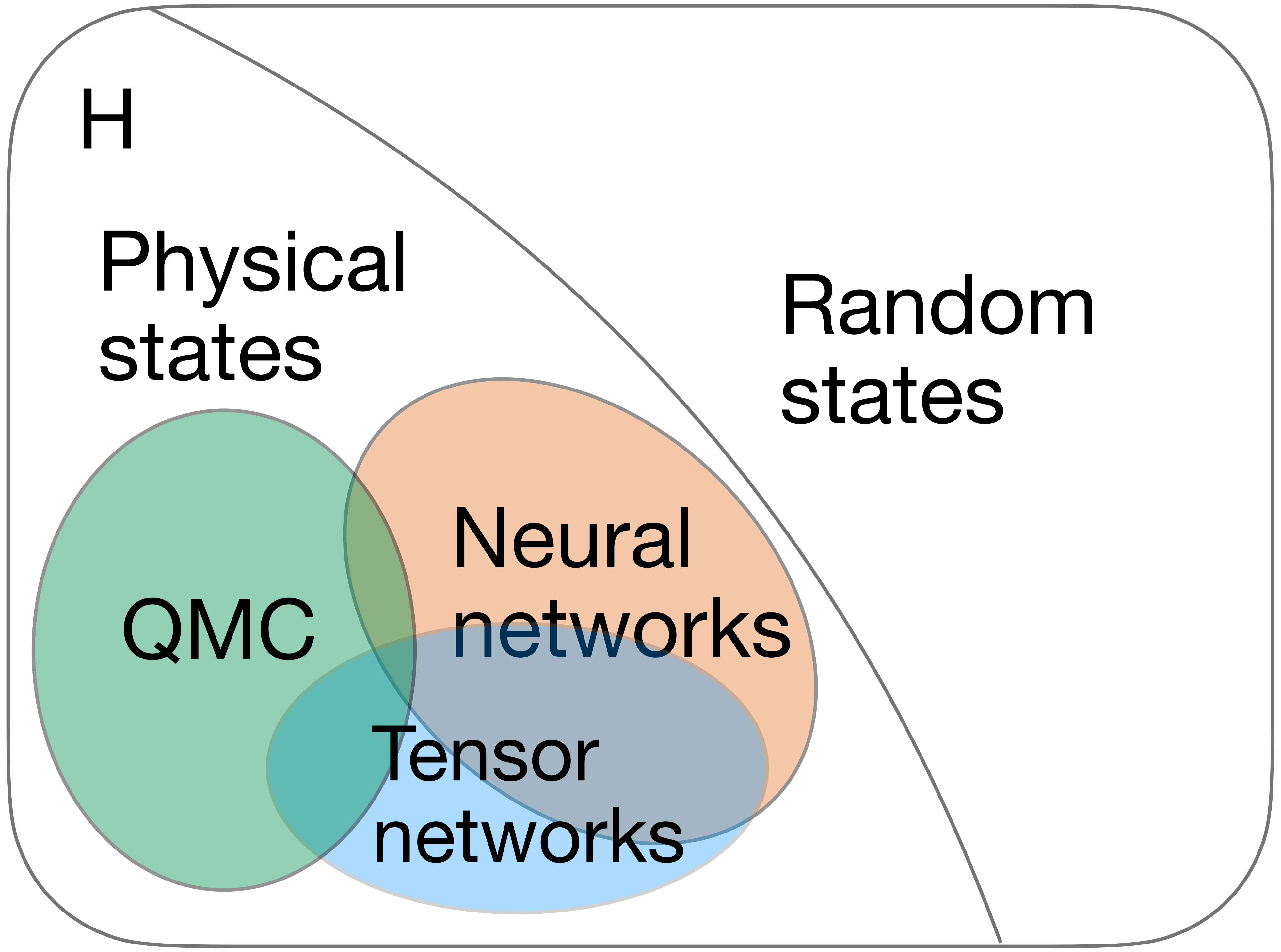
There is still hope for classical algorithms

- Enter Machine learning (ML): ML community deals with highly structured problems arising in natural datasets.

- Insight: both quantum and ML problems have a lot of **shared** structure and symmetry.

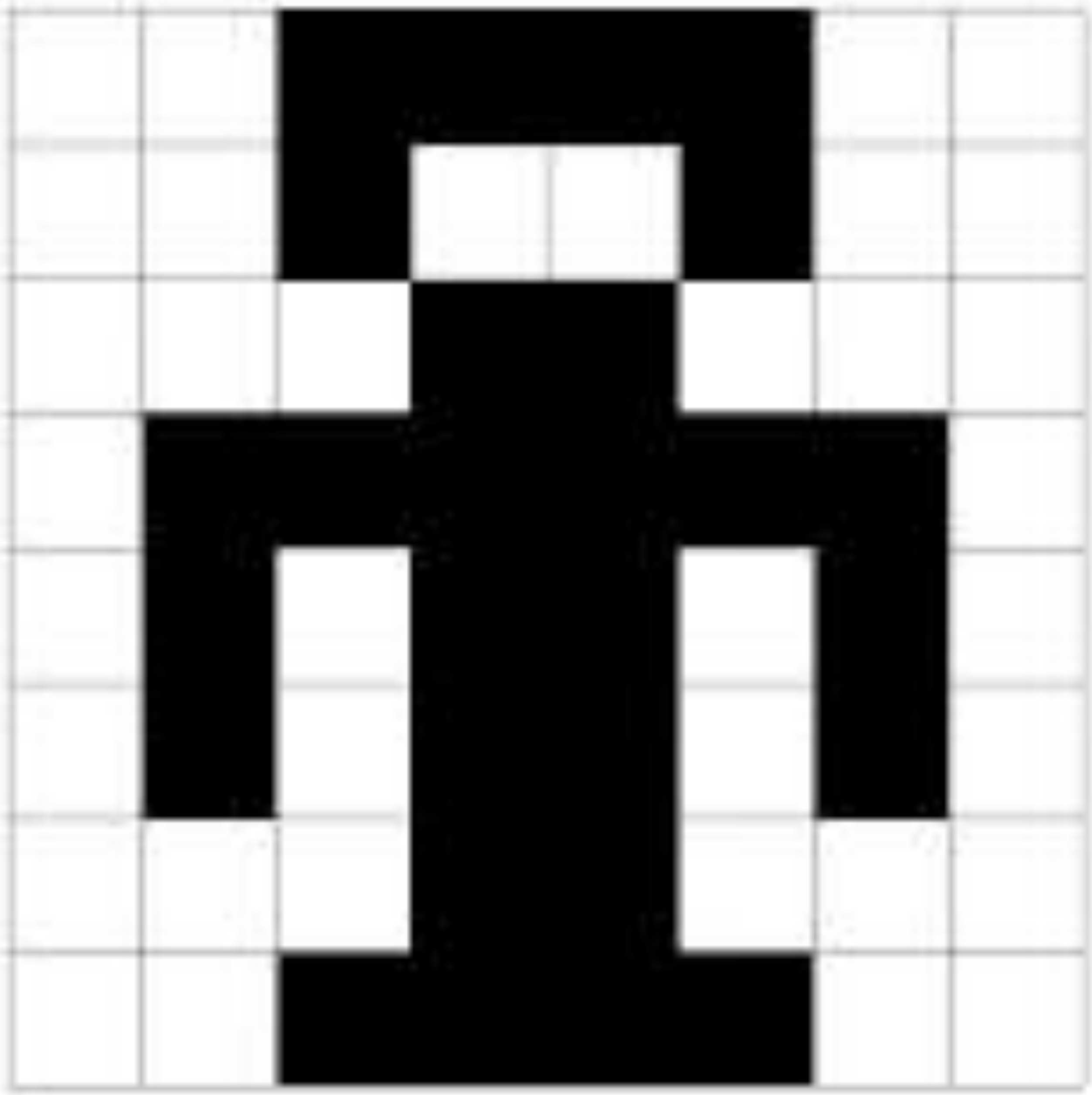
- What are these commonalities and are they important beyond mere resemblance?

$|\Psi\rangle$ vector with 2^N



High dimensionality

- Imagine you have a camera. How many different pictures can a camera take?
- Consider a simple camera with L x L pixels. The camera is take only take binary images — Each pixel can be only 1 or 0



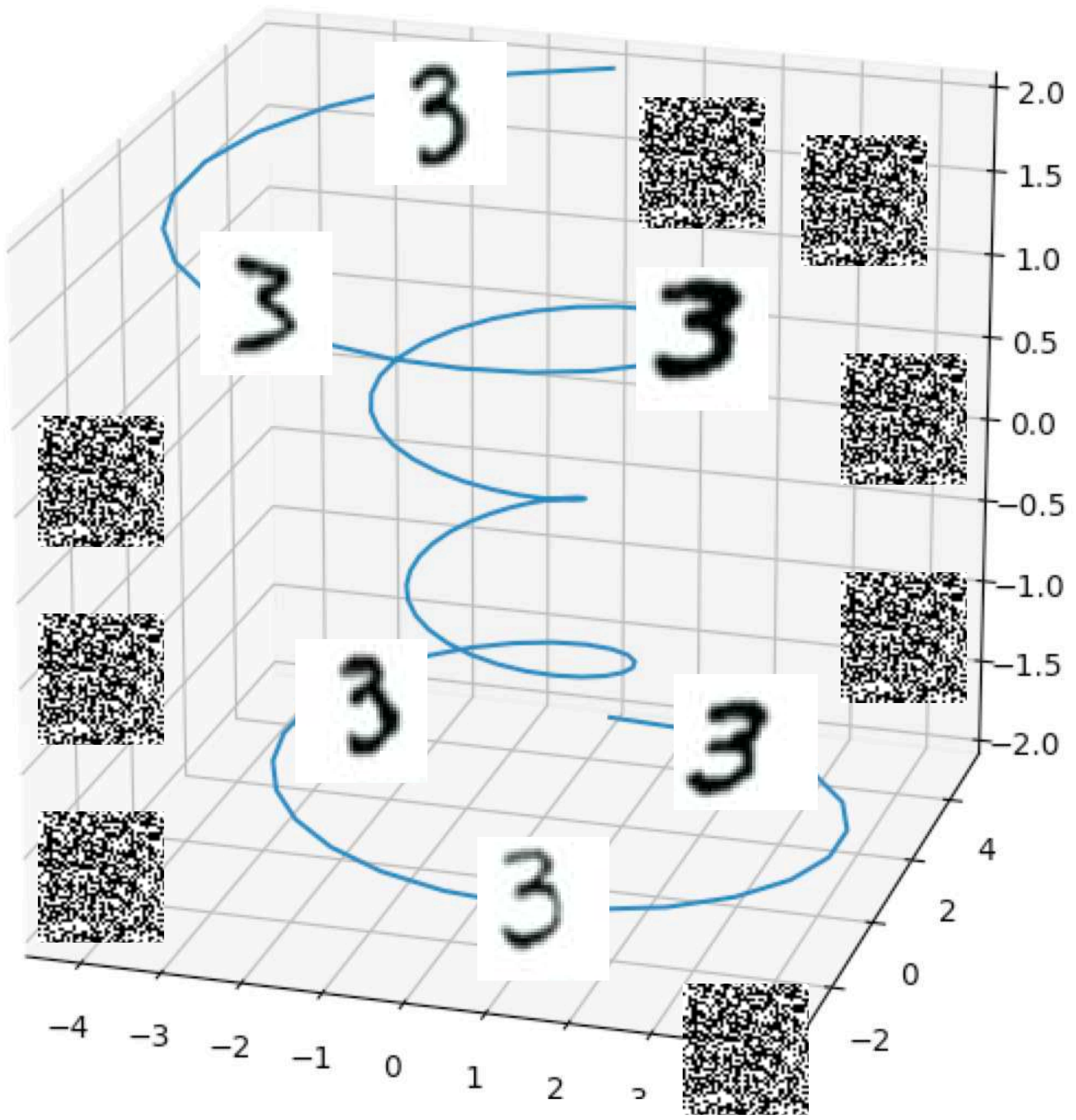
0	0	1	1	1	1	0	0
0	0	1	0	0	1	0	0
0	0	0	1	1	0	0	0
0	1	1	1	1	1	1	0
0	1	0	1	1	0	1	0
0	1	0	1	1	0	1	0
0	0	0	1	1	0	0	0
0	0	1	1	1	1	0	0

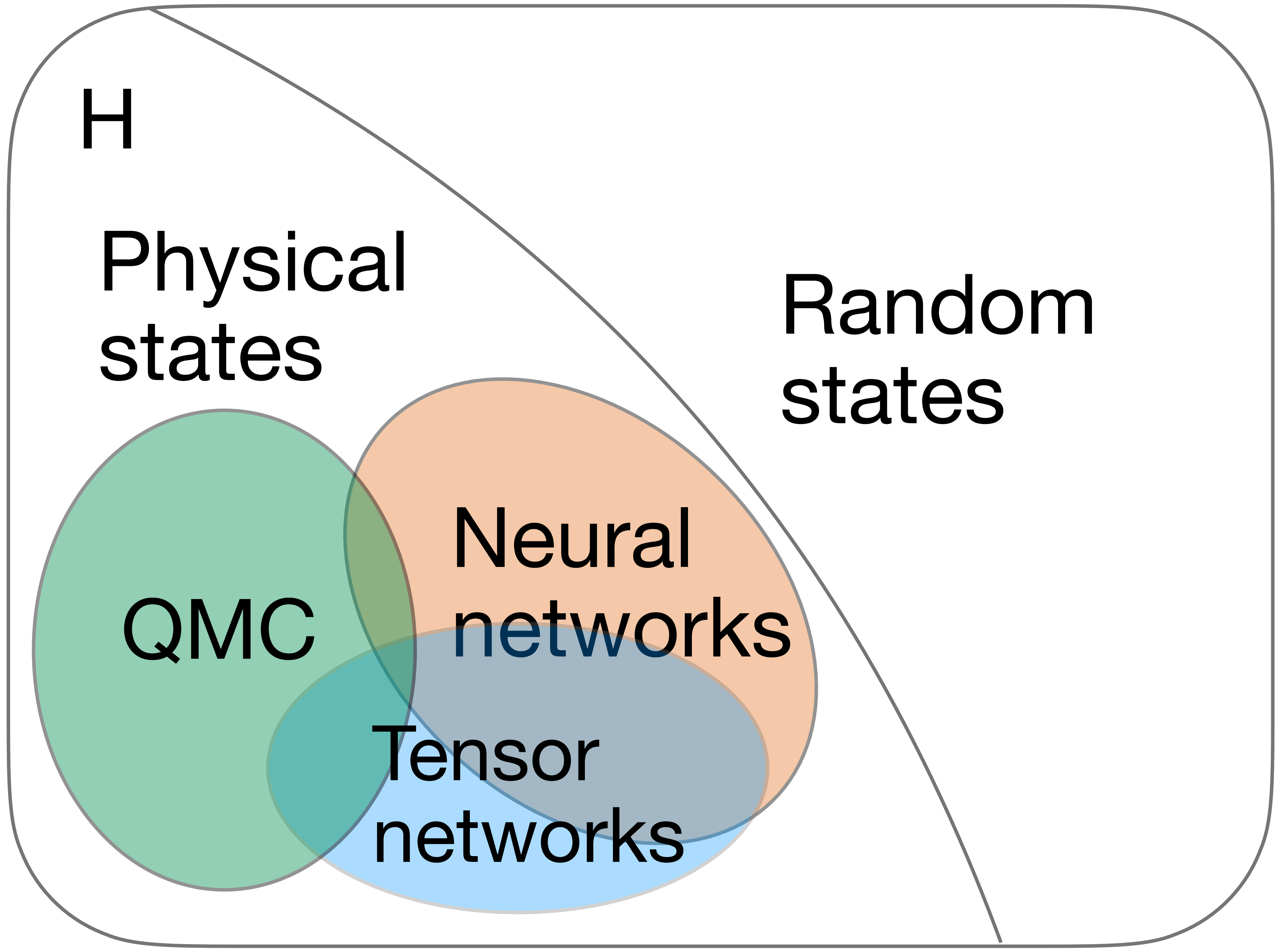
High dimensionality: building a camera pixel by pixel

- The first pixel can be 1 or 0 —> that is 2 values
- The second pixel can be 1 or 0 —> the combinations of the two first pixels is $4 = 2^2$: 0 0, 1 0 , 0 1 , 11.
- Now add a third pixel: 0 0 0,   0 0 1, 0 1 0, 1 0 0, 1 1 0, 0 1 1, 1 0 1, 1 1 1. That is $8 = 2^3$ possible combinations
- 4 pixels —> 0 0 0 0, 0 0 0 1, 0 0 1 0, 0 1 0 0, 1 0 0 0 . .. . there are $16 = 2^4$ possible combinations
- For all of those $L \times L$ pixels we get $N = 2^{L \times L}$
- The space of all possible images is exponentially big—same as quantum states or the Boltzmann distribution in stat mech.

Take 28 x 28 binary images

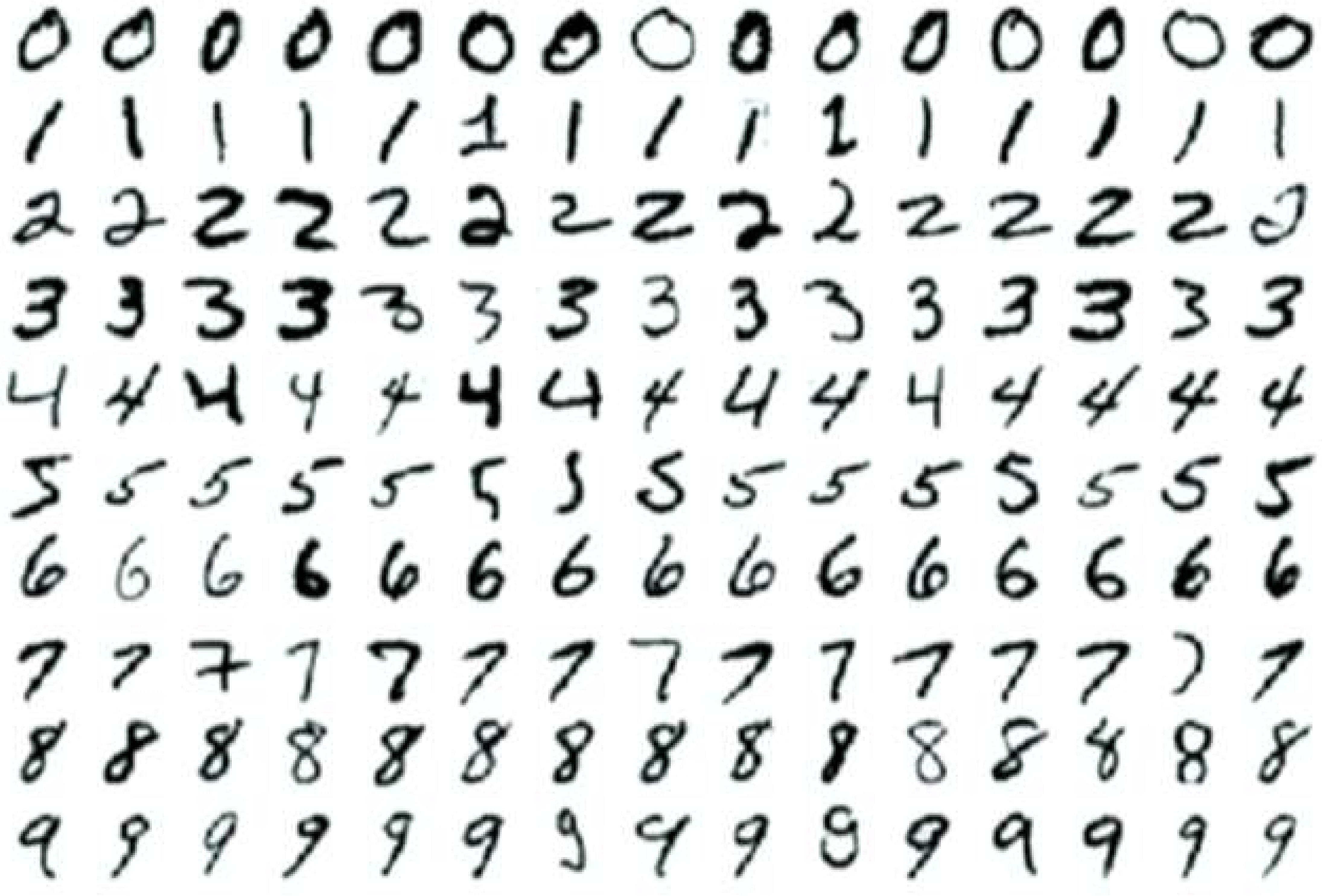
- Size of state space: $2^{28\times 28} = 1.017458 \times 10^{236}$
- Bigger than the number of atoms in the known universe. Most images are noise —> Probability distributions over the images our brain understands live in low-dimensional subspace of these big spaces.

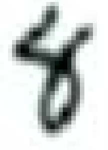
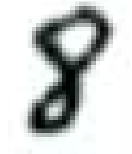




Commonality in some of the mathematical objects

- In unsupervised learning researchers are interested, e.g. in understanding the underlying probability of a dataset. For instance images of handwritten digits

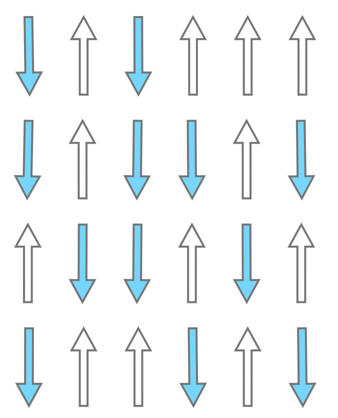
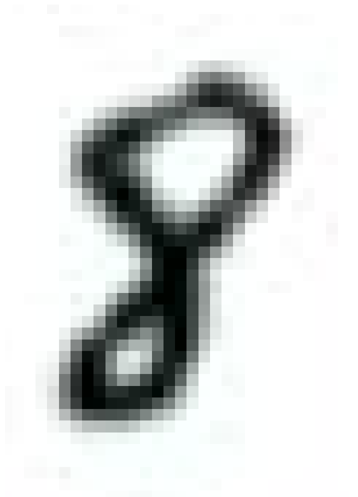


- What is the probability P() or what is the P()

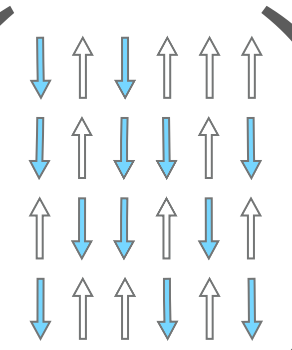
In ML people study P(image) and in stat mech…

- ➤ Boltzmann distribution

$$P(E) = \frac{e^{-E/k_B T}}{Z} \qquad E = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j$$

P() ⟷ P()

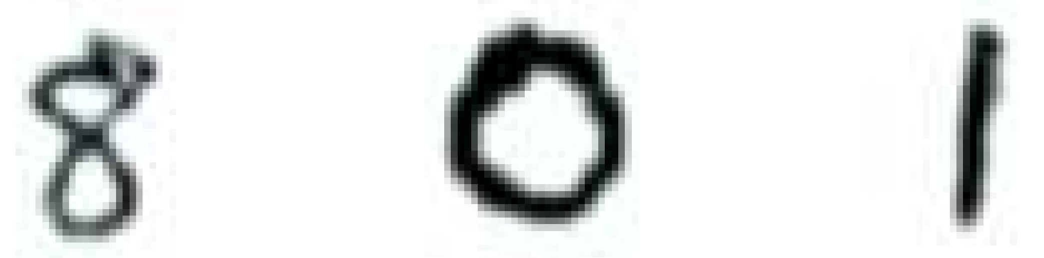
- ➤ Wavefunction

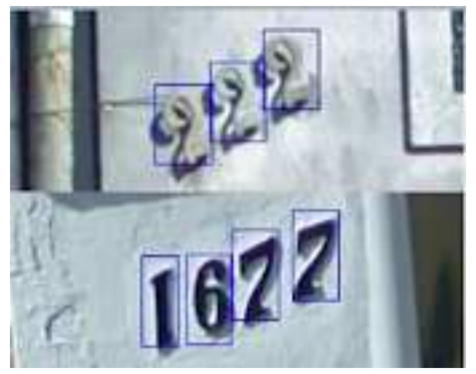
Ψ()

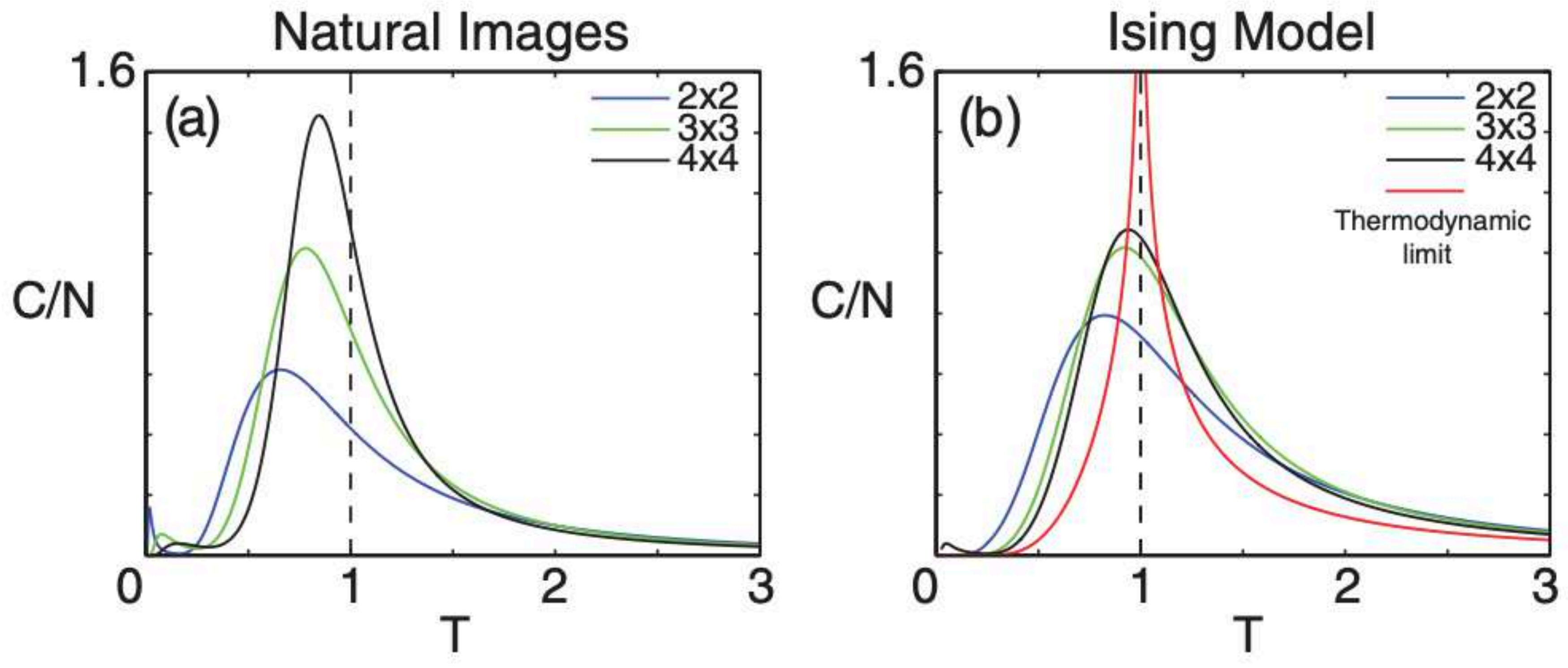
ML and statistical (and quantum) physics are interested in similar high dimensional distributions and wavefunctions

Correlations and symmetries with strikingly similar structure

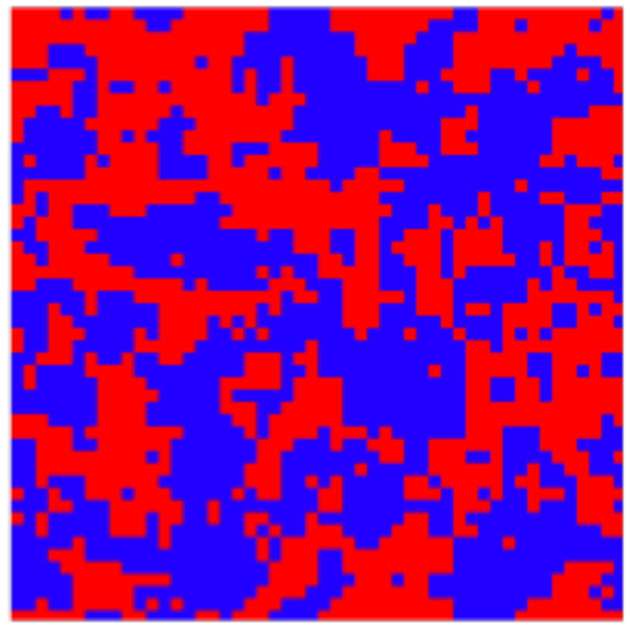
- ➤ Critical correlations:
- ➤ Natural language and natural images
- ➤ Music
- ➤ Flocks of animals
- ➤ All exhibit power-law decaying correlations identical to a (classical or quantum) at a critical point
- ➤ Translational, rotational, reflection, and other symmetries— rich







Statistical Thermodynamics of Natural Images
PRL 110, 018701 (2013)



Scale-free correlations in starling flocks.
PNAS 107 (26) 11865-11870

What’s learning?

- What is learning? ”The activity or process of gaining knowledge or skill by studying, practicing, being taught, or experiencing something.” Merriam Webster dictionary

- “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.” Tom Mitchell

What is machine learning?

- How can we solve a specific problem?
- We can write a program that encodes a set of rules that are useful to solve the problem
- However, In many cases it is very difficult to specify those rules
    - Some tasks (vision, speech, NLP) are too complicated to code.
    - Some systems need to adapt.
    - Handle noise.
    - Solve a differential equation, eg, Schrodinger equation.
- Instead of explicitly writing a program to solve a specific problem, we typically use examples (training data) to train the computer to perform this task (to generalize). Alternatively we can use the differential equation in the solution.

What is machine learning?

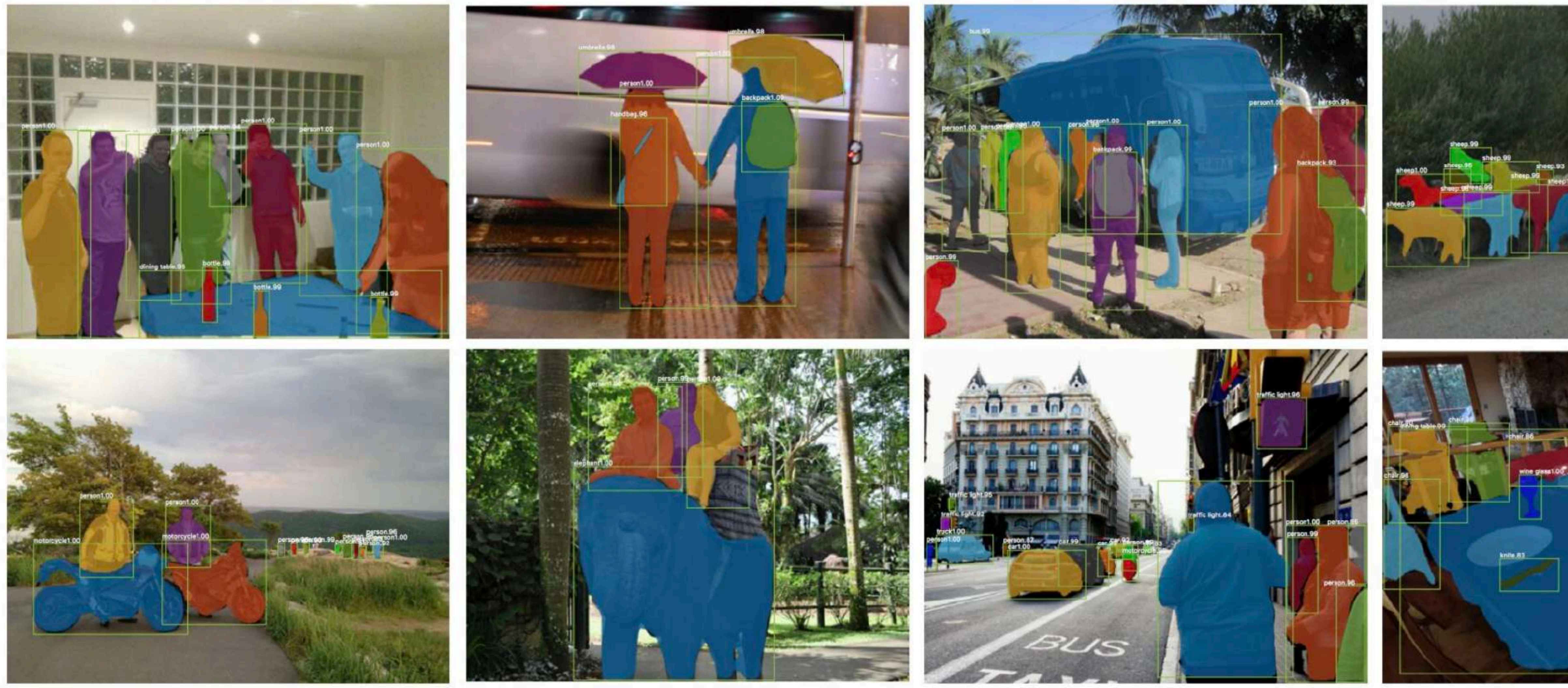
- Learning systems are not directly programmed to solve a problem, instead develop own program based on:
  - Examples of how they should behave
  - From trial-and-error experience trying to solve the problem
- Different than standard CS:
  - Want to implement unknown function, only have access e.g., to sample input-output pairs (training examples)
- Learning simply means incorporating information from the training examples into the system

What is machine learning?

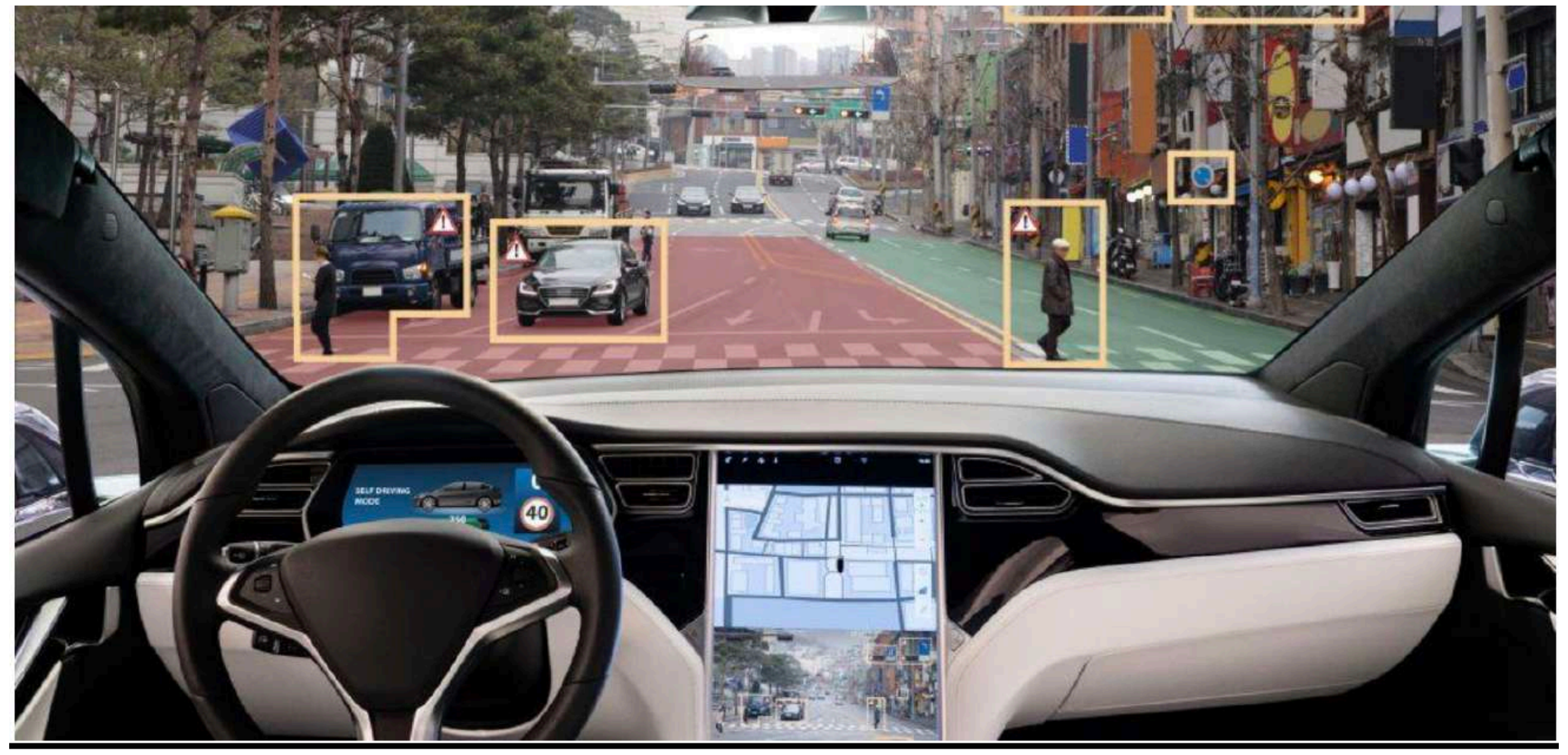
- For many problems, it’s difficult to program the correct behaviour by hand:
    - recognizing people and objects
    - understanding human speech
- Machine learning approach: program an algorithm to automatically learn from data, or from experience
- Why might you want to use a learning algorithm?
  - hard to code up a solution by hand (e.g. vision, speech)
  - system needs to adapt to a changing environment (e.g. spam detection)
  - Want the system to perform better than the human programmers
  - privacy/fairness (e.g. ranking search results)

Examples
Computer vision

- Object detection, semantic segmentation, pose estimation
- Autonomous vehicles
- Analysis of Medical images
- Precision agriculture
- Face recognition
- Robotics



arXiv:2001.05566



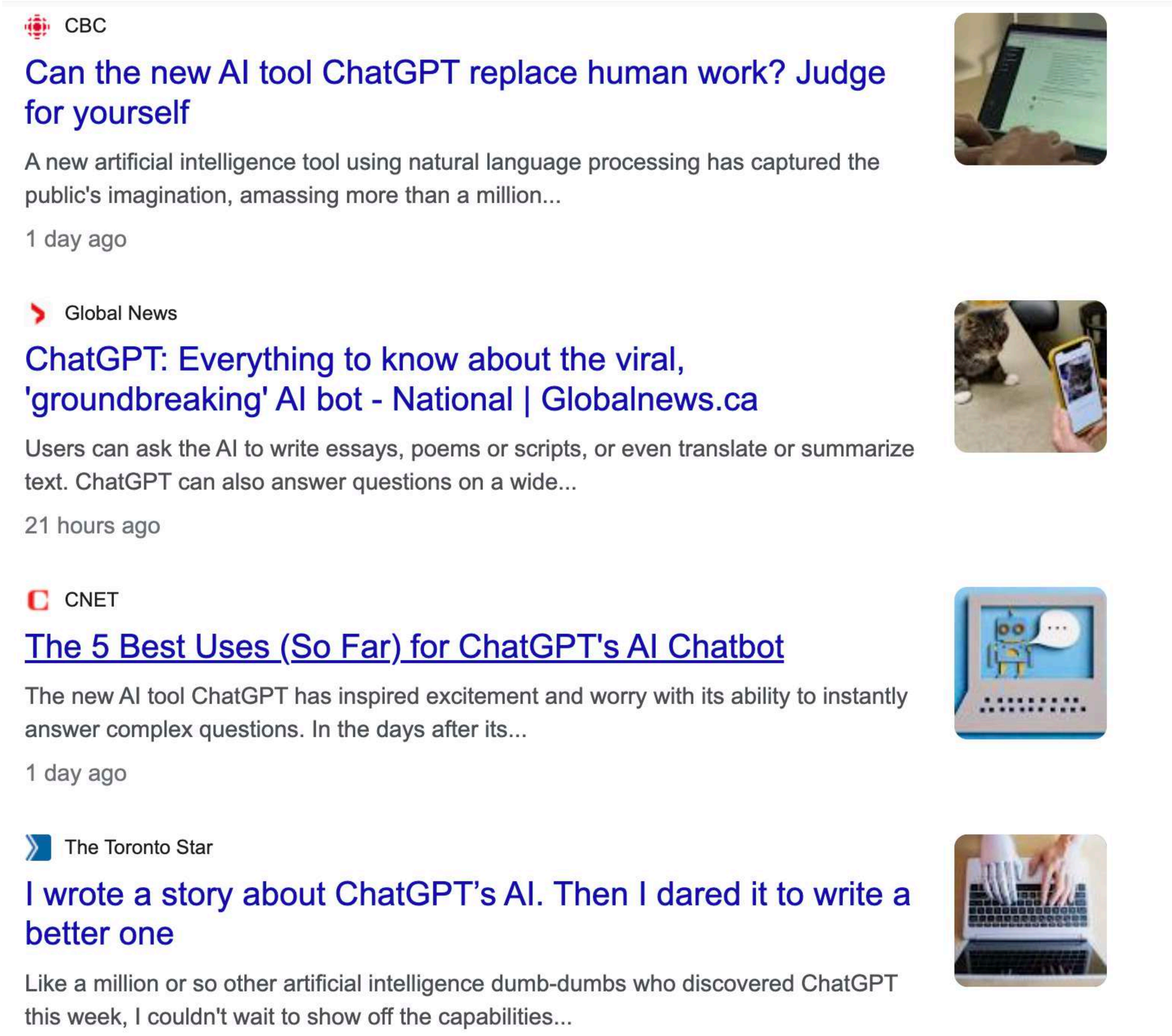
https://trid.trb.org/view/1678741

Natural language processing

- Branch of computer science, linguistics, and machine learning concerned with giving computers the ability to process text and spoken words in a similar way humans do it.
- Machine translation
- Speech recognition
- Sentiment analysis
- Automatic summarization of text
- Text to image/video generation

ChatGPT

- ChatGPT is a machine learning model which interacts in a conversational way
- Dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.



https://openai.com/blog/chatgpt/

ChatGPT

- People now believe these models will likely have deep technological, educational, and societal implications.

Daily Mail

ChatGPT can pass the United States Medical Licensing Exam and the Bar Exam

The rise of the machines? ChatGPT CAN pass US Medical Licensing Exam and the Bar, experts warn - after the AI chatbot received B grade on...

4 hours ago

Firstpost

OpenAI's ChatGPT passes Wharton's MBA Exam, while also qualifying for a medical licence in the US

Medical researchers are actually considering using ChatGPT in actual, clinical cases to assist real doctors. OpenAI's chatbot ChatGPT has been...

5 hours ago

02-21-23 | 9:59 AM

A science fiction magazine closed submissions after being bombarded with stories written by ChatGPT

In a case of life (or something) imitating art, an award-winning publisher of science fiction says it's being overrun with AI-generated work.

https://openai.com/blog/chatgpt/

ChatGPT

write a poem about ICTP-SAIFR and the Journeys into Theoretical Physics program

In São Paulo, where knowledge blooms,
A center of brilliance, ICTP-SAIFR looms.
Within its walls, the curious minds convene,
Embarking on journeys into theoretical physics unseen.

At ICTP-SAIFR, the air crackles with intellectual fire,
Where scholars, young and seasoned, aspire
To unravel the secrets of the universe profound,
In the realm where theoretical physics is crowned.

The Journeys into Theoretical Physics program,
A gateway to knowledge, like a celestial anthem,
Draws minds from across the world's expanse,
Uniting them in the pursuit of cosmic dance.

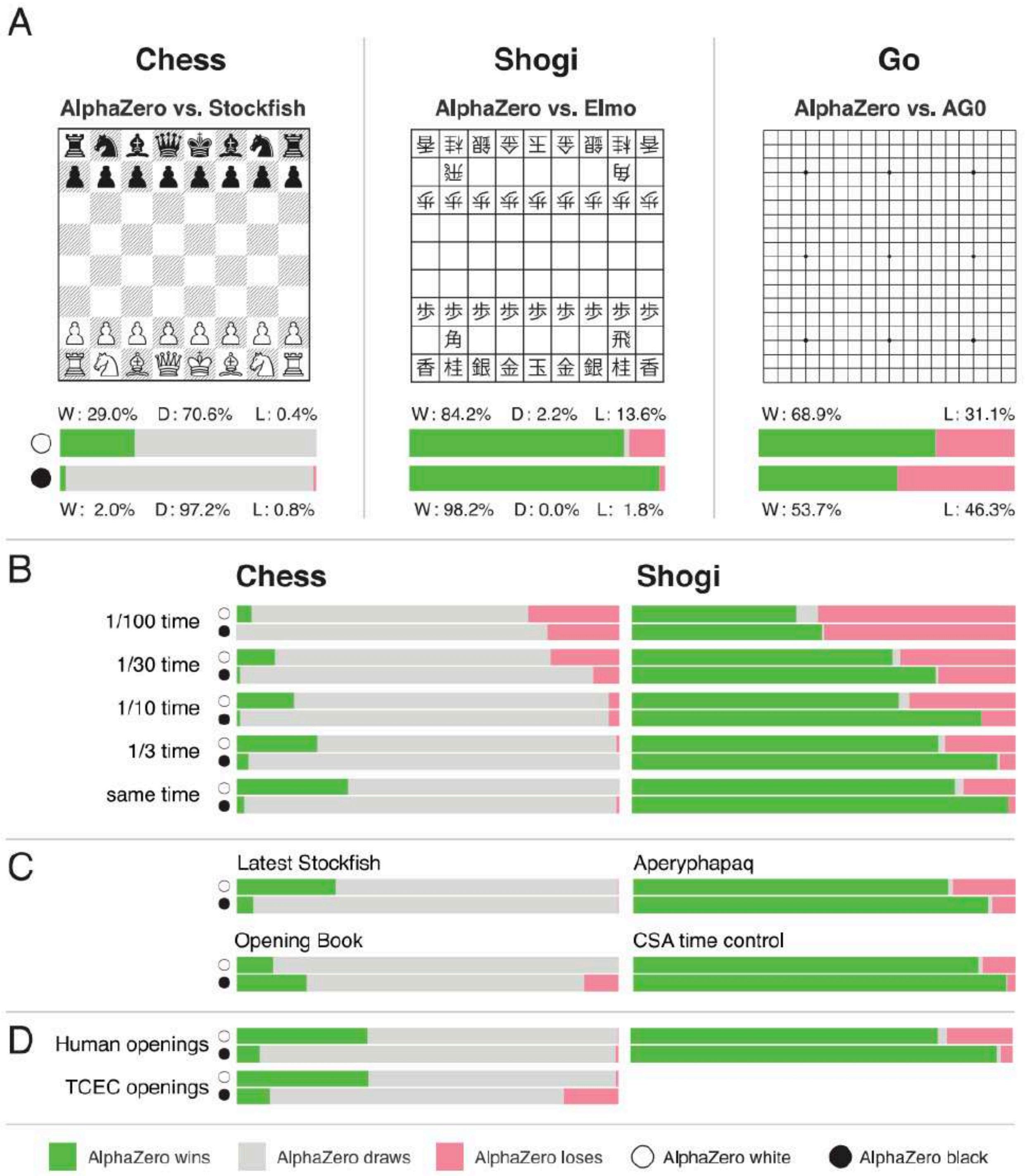
The scholars gather, their spirits alight,
Immersed in equations that illuminate the night,
Seeking answers to questions, profound and deep,
Exploring the mysteries the universe does keep.

In lecture halls adorned with brilliance and light,
Equations dance, filling the rooms with insight.
Blackboards are filled with symbols and signs,
As the beauty of theoretical physics unwinds.

https://openai.com/blog/chatgpt/

Examples

Game play

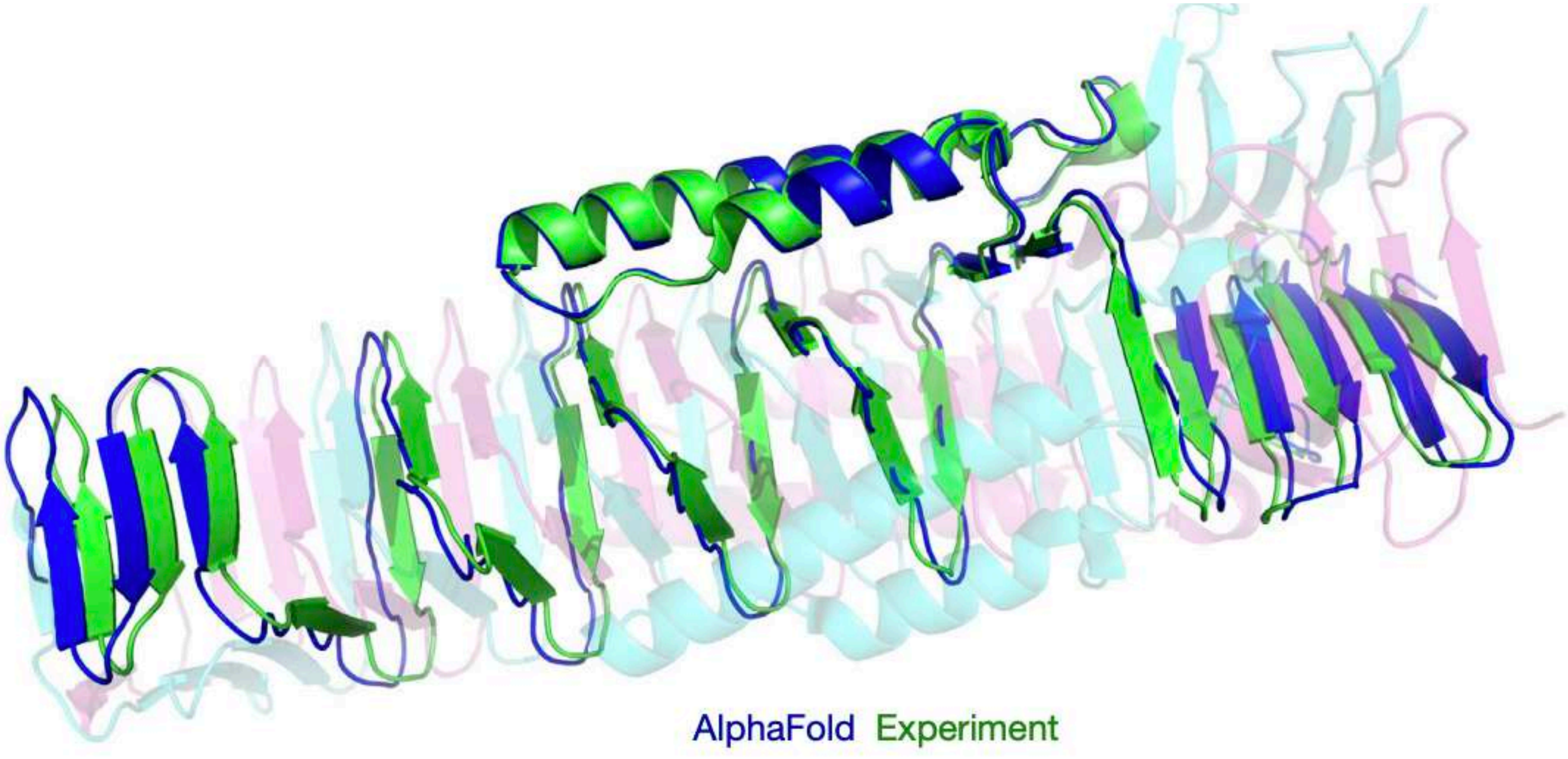


Go is known as the most challenging classical game for artificial intelligence because of its complexity. https://deepmind.com/research/case-studies/alphago-the-story-so-far

single system that taught itself from scratch how to master the games of chess, shogi (Japanese chess), and Go, beating a world-champion program in each case.
https://www.science.org/doi/10.1126/science.aar6404
https://deepmind.com/blog/article/alphazero-shedding-new-light-grand-games-chess-shogi-and-go

Examples

Machine learning in sciences



Highly accurate protein structure prediction with AlphaFold. Nature volume 596, pages 583–589 (2021)

ML broad categories

- **Data-based learning**
- We are provided with useful data that allows us to solve the problem.
- Applies when it is possible to collect data about the system.

- **Equation/law-based learning**
- We are provided an equation we need to solve.
- Example: Physics/chemistry-informed neural networks are a neural networks that can embed the knowledge of any physical/chemical laws that govern a given data-set in the learning process, and can be described by differential equations.
- Eigenvalue problems

ML broad categories
Data-driven learning

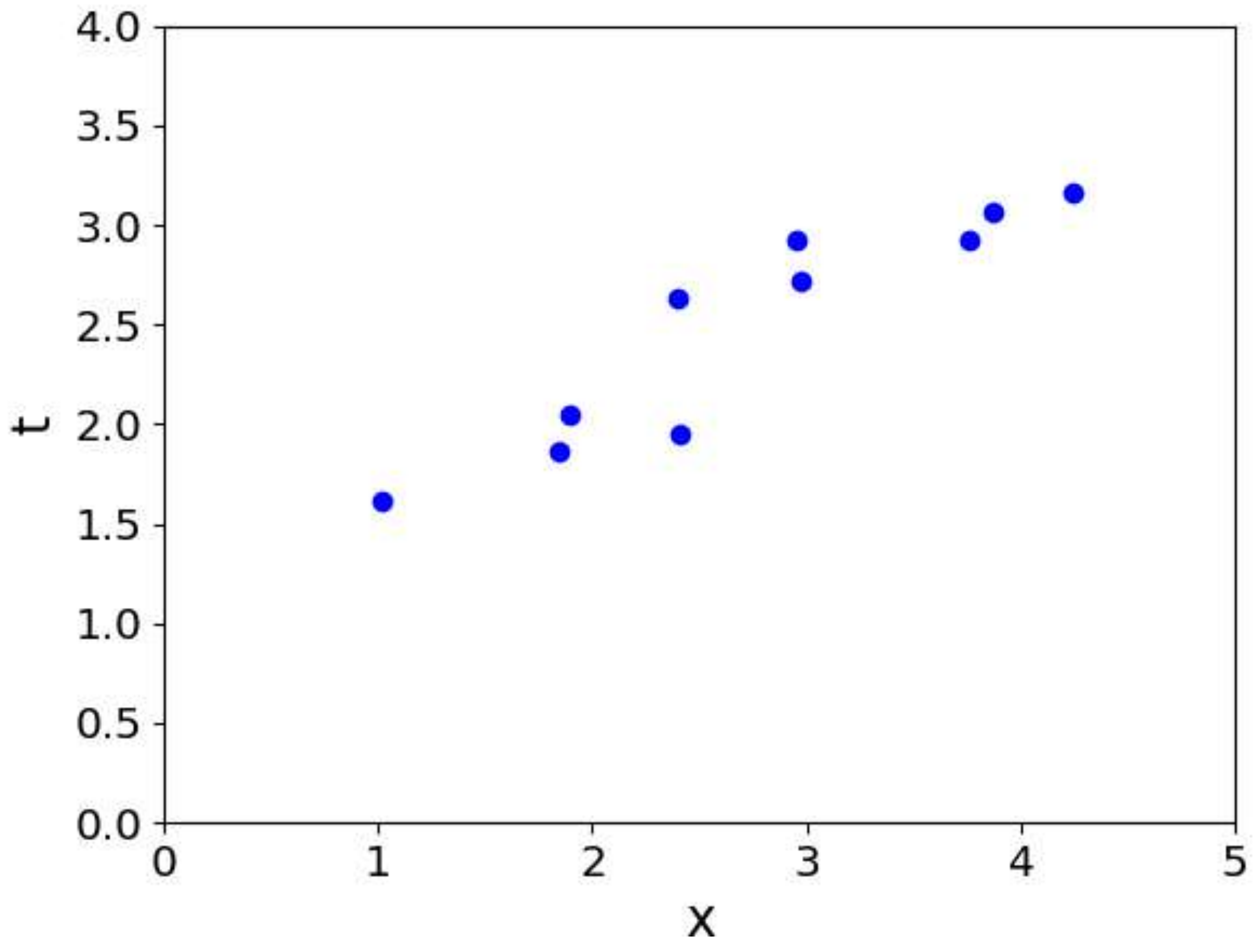
- **Supervised learning** (correct outputs known). Given (x, y) pairs learn a mapping from x to y. Example:
    - Classification: categorical output (object recognition, classifying a phase from measurements)
    - Regression: real-valued output (predicting market prices, customer rating)
- **Unsupervised learning**. Given data points find some structure in the data. Example: Dimensionality reduction, learning a probability distribution. Quantum state tomography.
- **Online learning.** Supervised learning when the data is given sequentially, by an adversary, No separate train/test phases. Example: Spam filtering
- **Reinforcement learning.** Learn actions to maximize future rewards. Delayed playoffs, agent controls what he sees. Example: Flying drones.
- Other categories: active learning, semi-supervised learning.

ML broad categories
Equation/physical law learning

- Eigenvalue/ground state problems/Hamiltonian driven learning
- Approximate dynamics of a classical or quantum system
- Stochastic reaction dynamics in chemistry
- Approximate equilibrium properties of a system in thermal equilibrium
- Approximate steady state of an open quantum system
- For all these systems, it is possible to reformulate the problem in a variational approach.

Supervised learning

Supervised learning setup



NOT

x_0	x_1	t
1	0	1
1	1	0

In supervised learning:

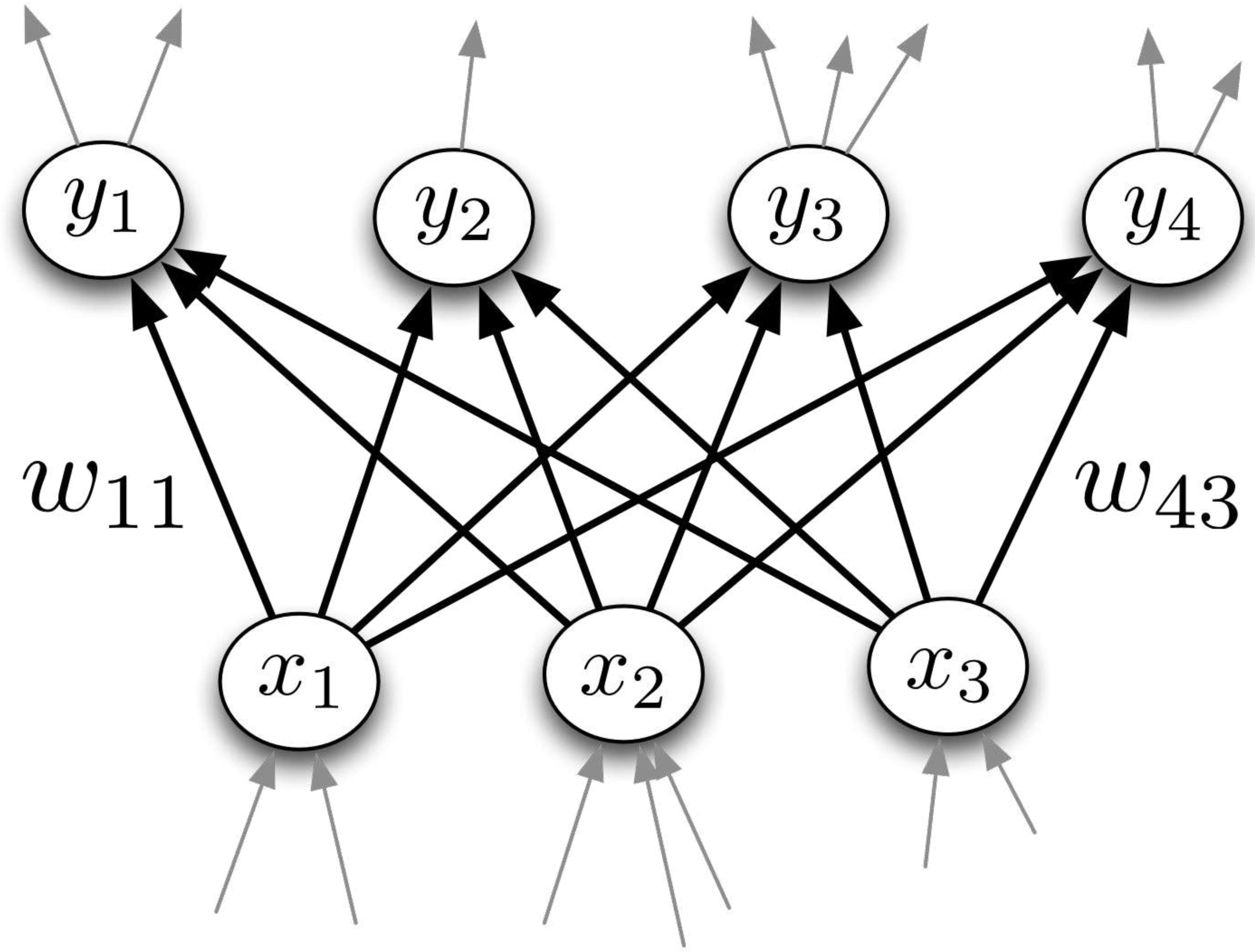
- There is input $\boldsymbol{x} \in \mathscr{X}$, typically a vector of features
- There is target $t \in \mathscr{T}$ (also called response, outcome, output, class)
- Objective is to learn a function $f : \mathscr{X} \rightarrow \mathscr{T}$ such that $t \approx y = f(\boldsymbol{x})$ based on some data $\mathscr{D} = \{(\boldsymbol{x}^{(i)}, t^{(i)})$ for $i = 1,2,\ldots,N\}$

Binary classification

- **Classification**: given a D-dimensional input $\boldsymbol{x} \in \mathbb{R}^D$ predict a discrete-valued target.
- **Binary**: predict a binary target $t \in \{0,1\}$
    - Training examples with $t = 1$ are called positive examples, and training examples with $t = 0$ are called negative examples.
    - $t \in \{0,1\}$ or $t \in \{+1, -1\}$.
    - We will build models $y = f(\boldsymbol{x})$ which predict the targets given some input $\boldsymbol{x}$, ie we want y to match t.
    - The models will have a set of adjustable parameters.

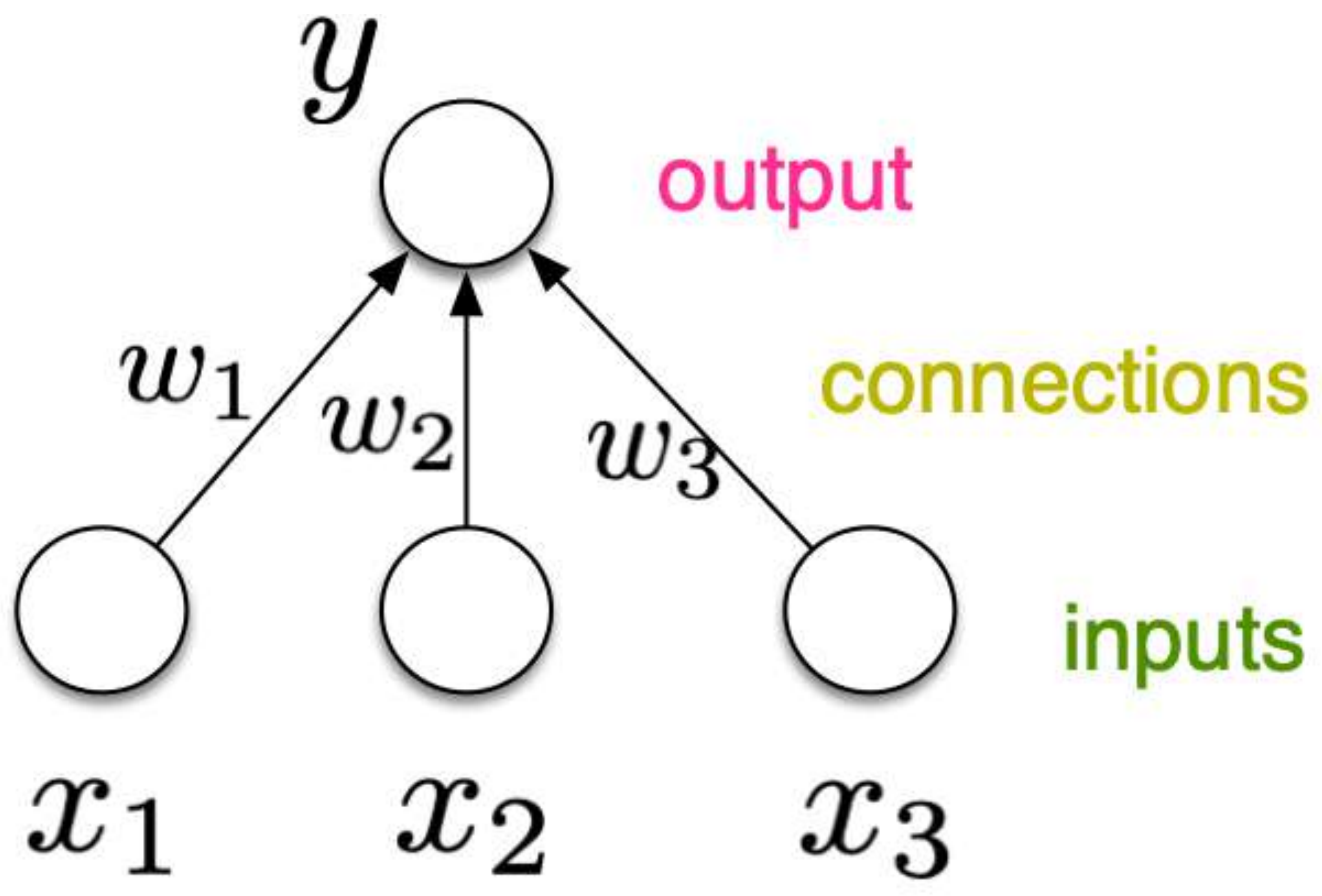
Neural networks

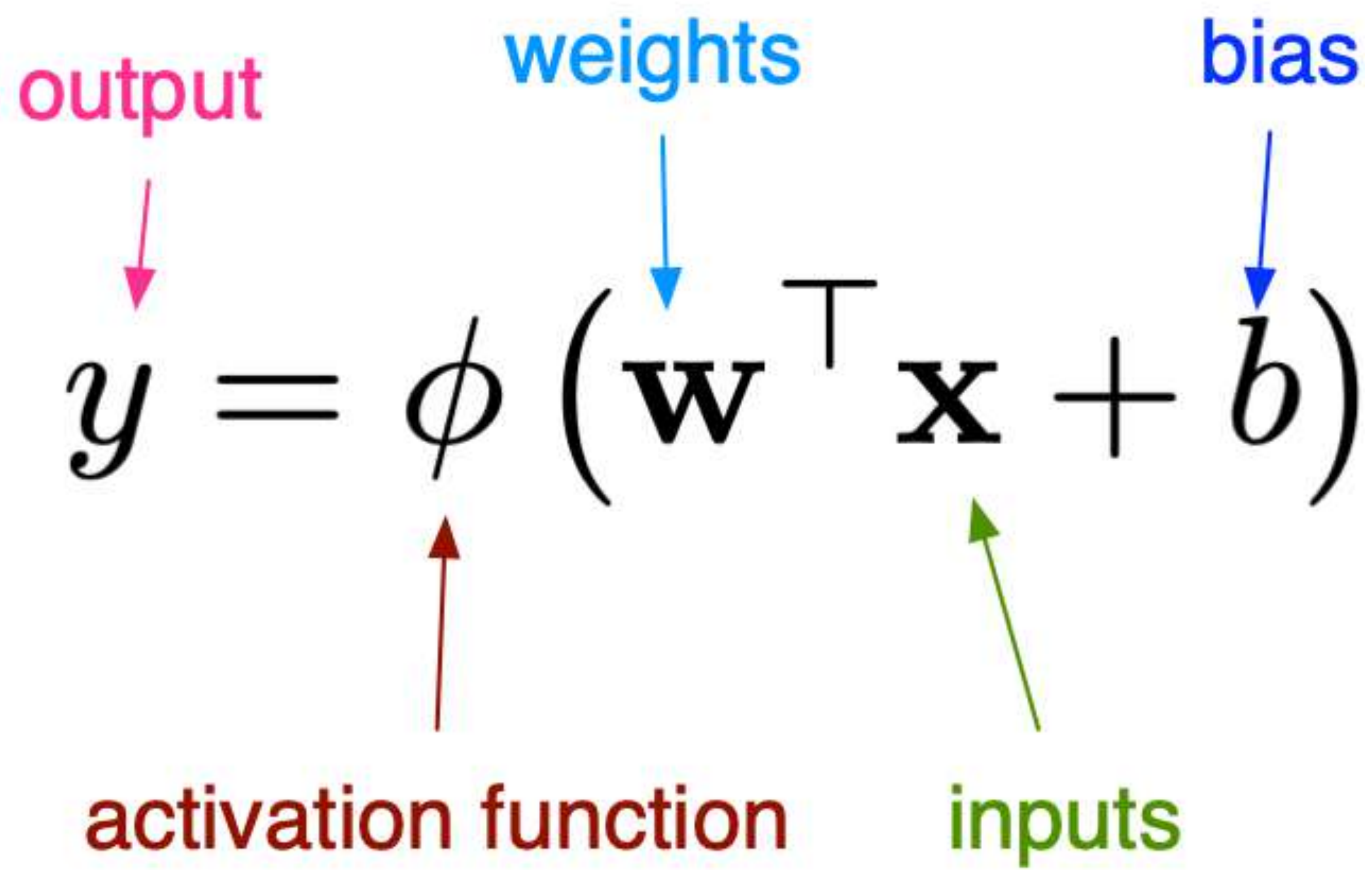
- - -



Basic unit

- For neural nets, we use a neuron, or unit:

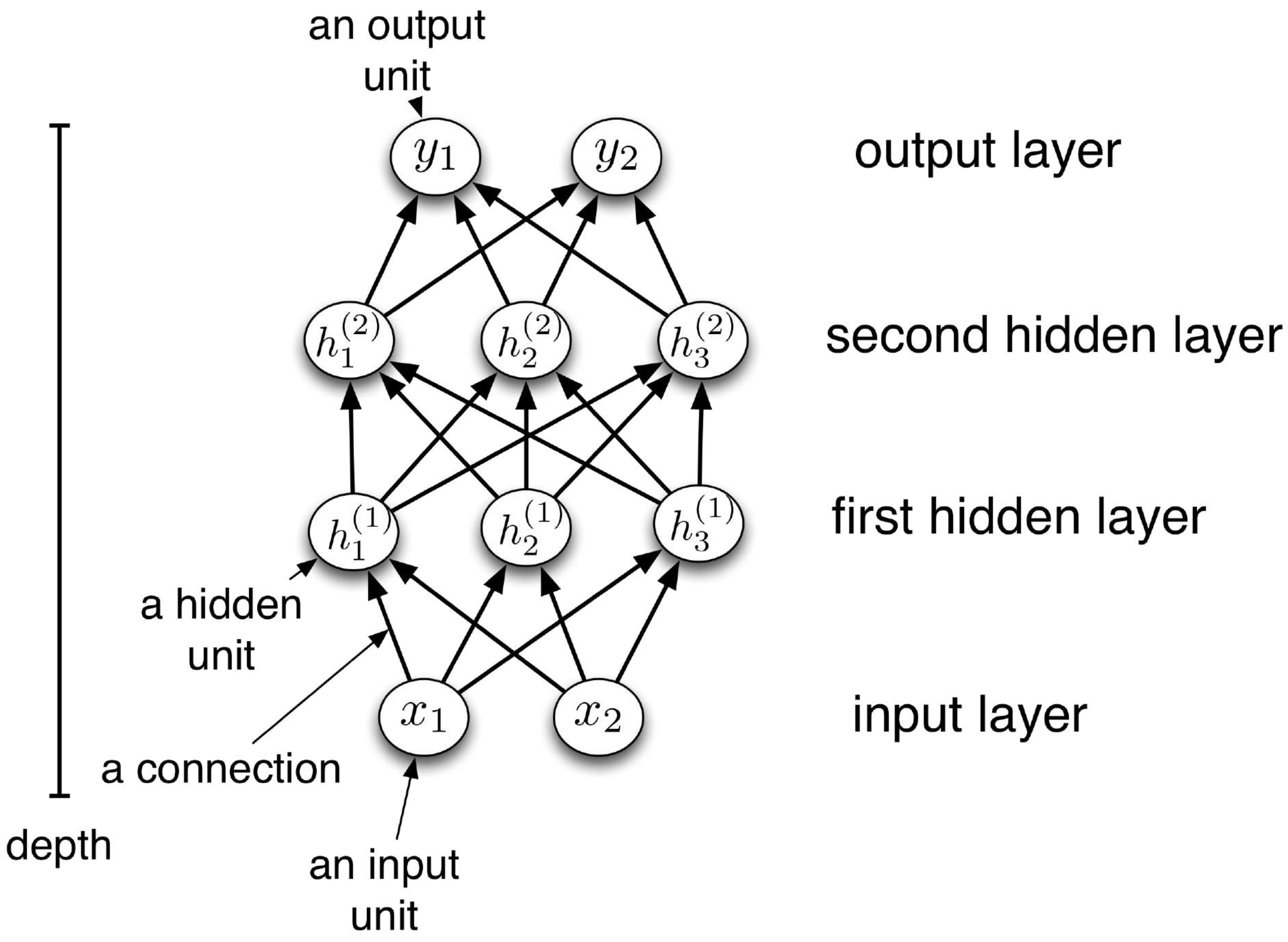




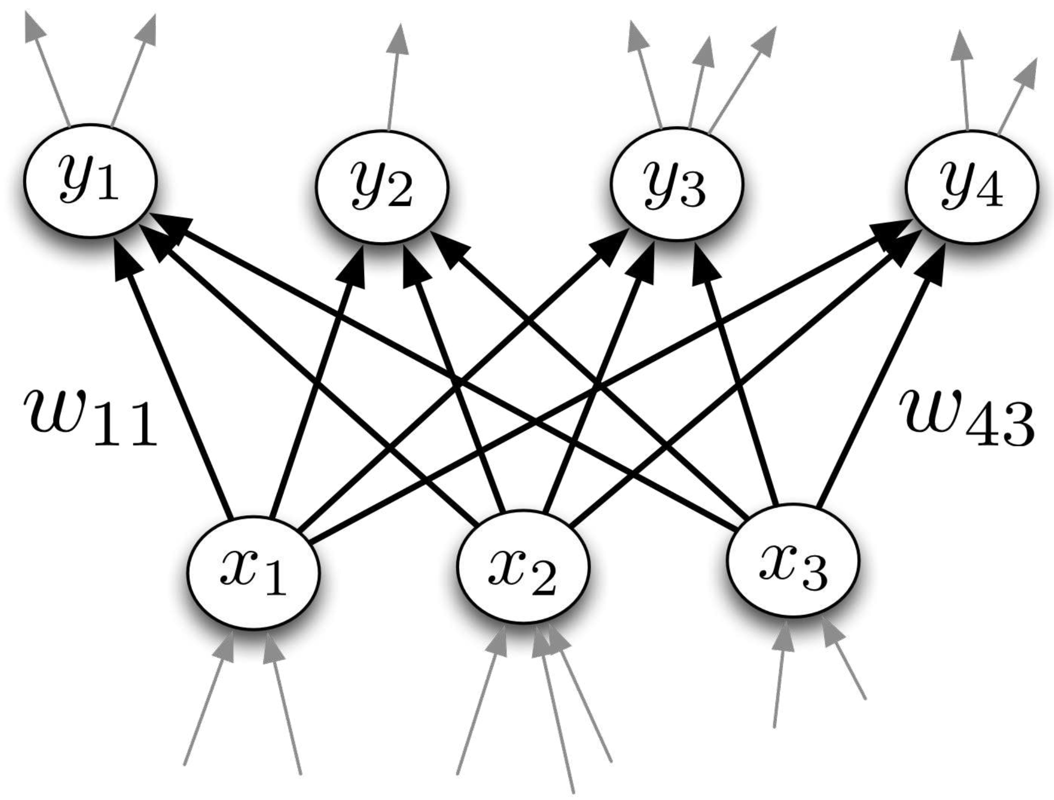
- Typically ϕ is a non-linear function. It can be between 0 and 1 (active/inactive neuron)
- By putting together lots of these incredibly simplistic neuron-like processing units, we can do some powerful computations!

Multilayer Perceptrons

- We can connect lots of units together into a **directed acyclic graph**.
- Typically, units are grouped into **layers**.
- This gives a **feed-forward neural network**.



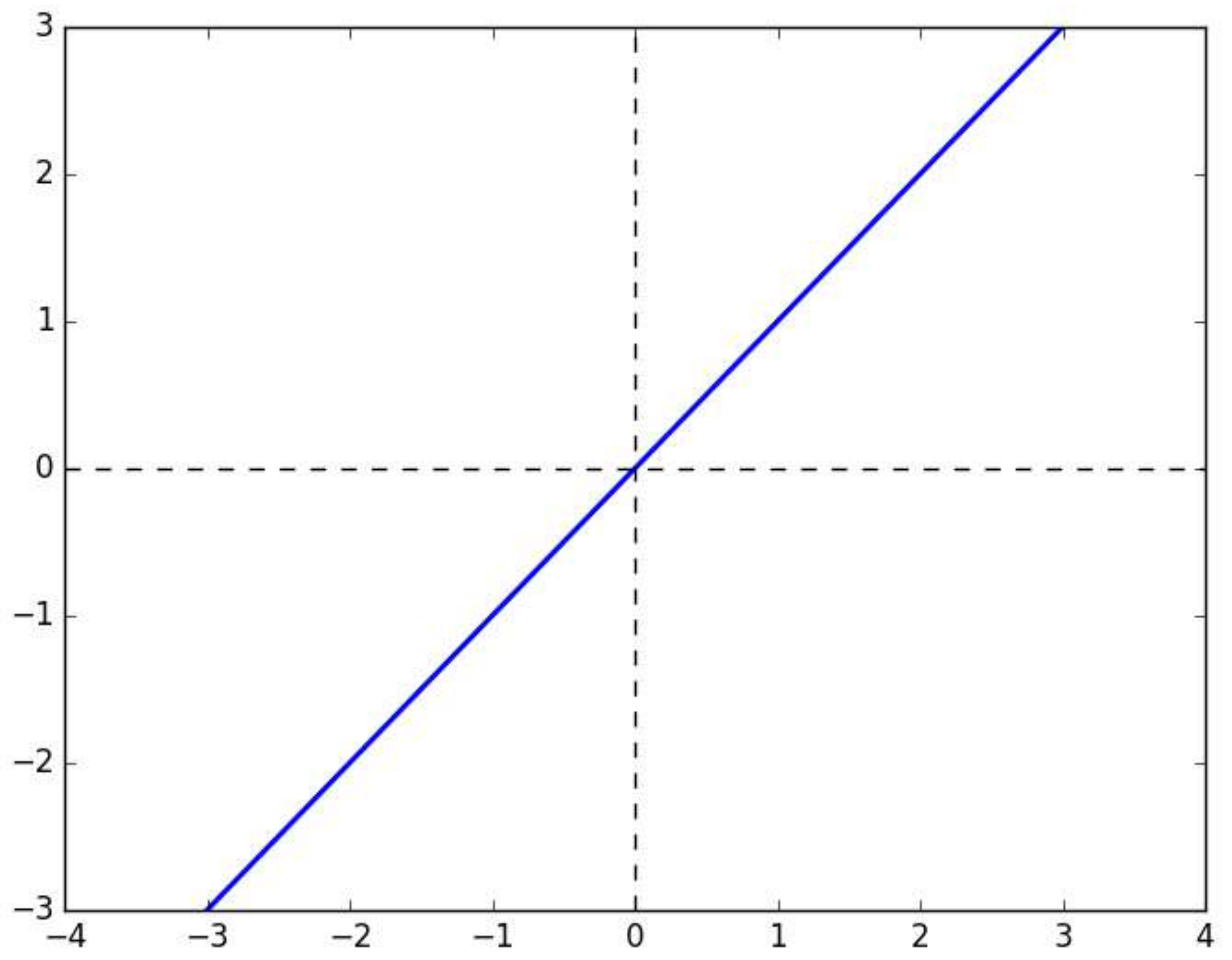
Multilayer perceptrons



- Each hidden layer i connects N_{i-1} input units to N_i output units.
- In a fully connected layer, all input units are connected to all output units.
- Note: the inputs and outputs for a layer are distinct from the inputs and outputs to the network.
- If we need to compute M outputs from N inputs, we can do so using matrix multiplication. This means we'll be using an $M \times N$ matrix.
- The outputs are a function of the input units: $\boldsymbol{y} = f(\boldsymbol{x}) = \phi(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$
- ϕ is typically applied component-wise
- A multilayer network consisting of fully connected layers is called a multilayer perceptron.

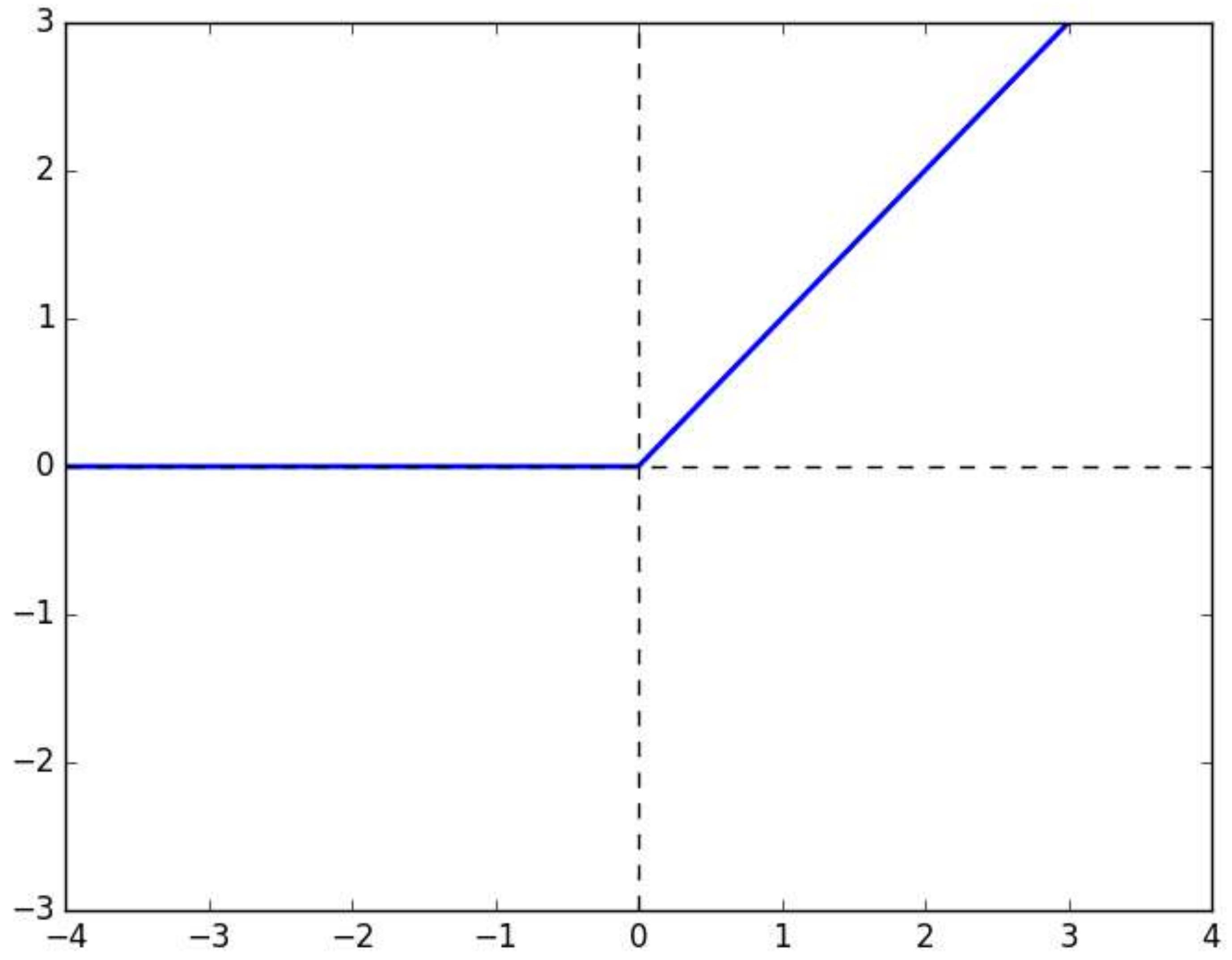
Multilayer Perceptrons

- Common activation functions:



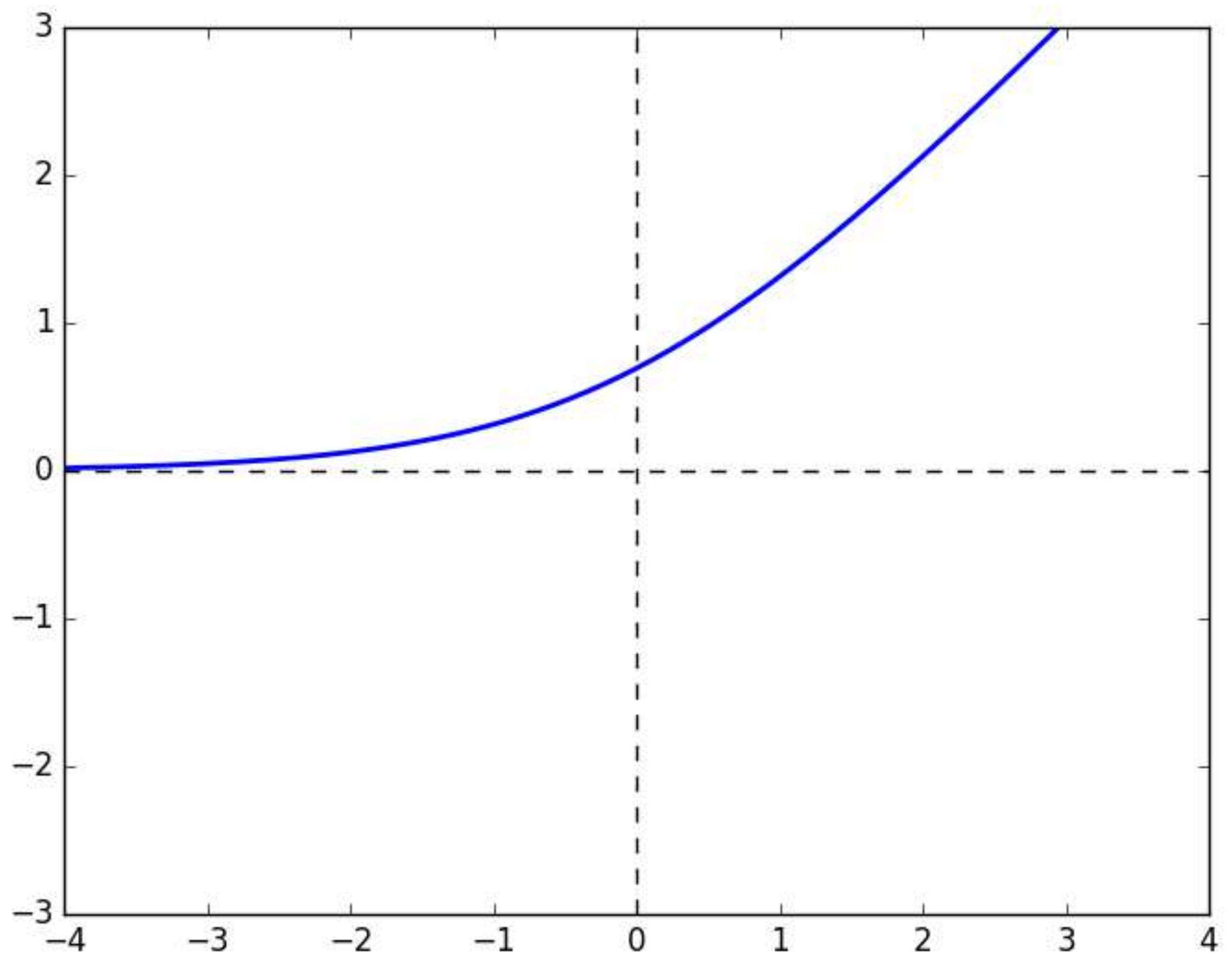
Identity

$y = z$



Rectified linear unit (ReLu)

$y = \max(0, z)$

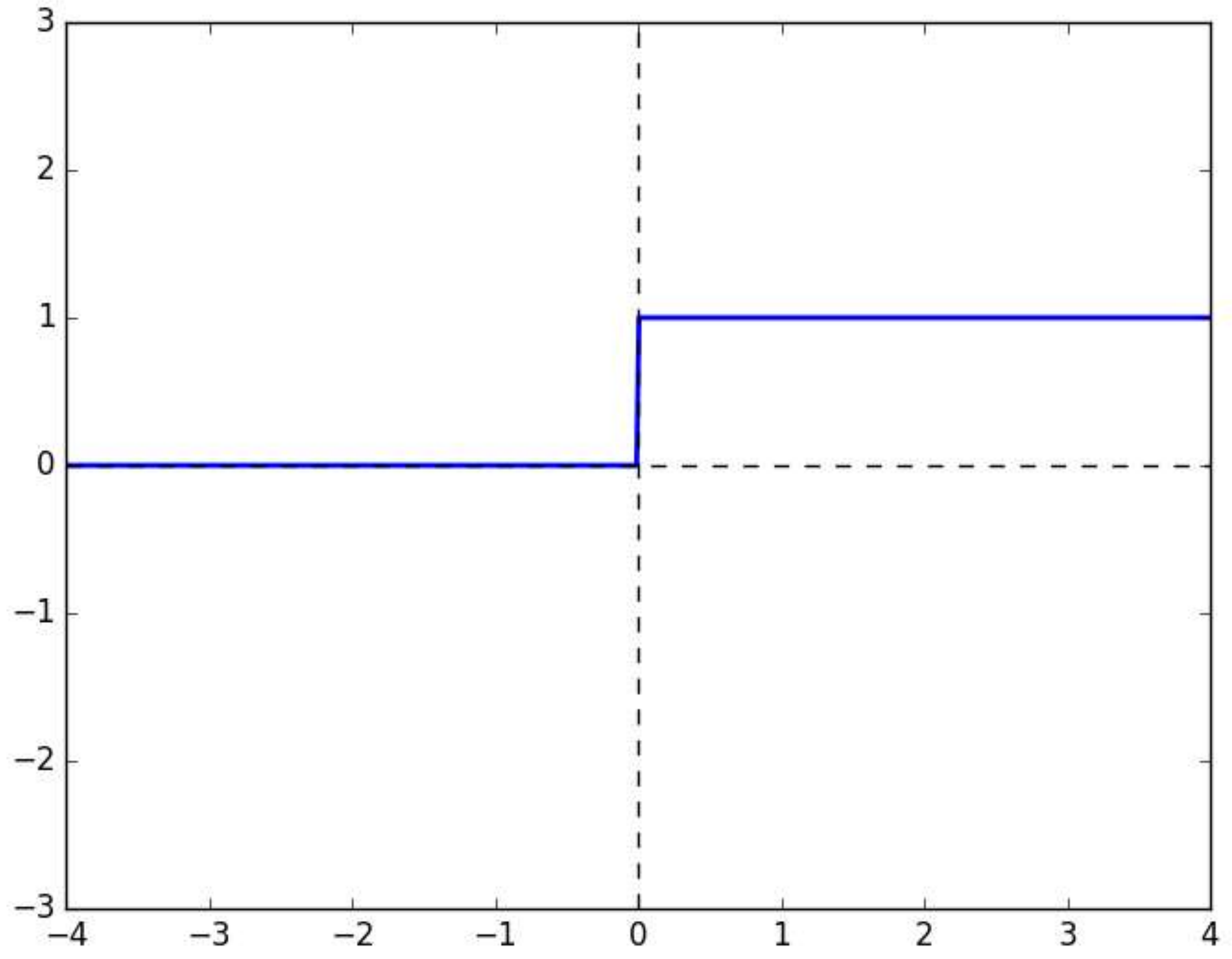


Softplus

$y = \log(1 + e^z)$

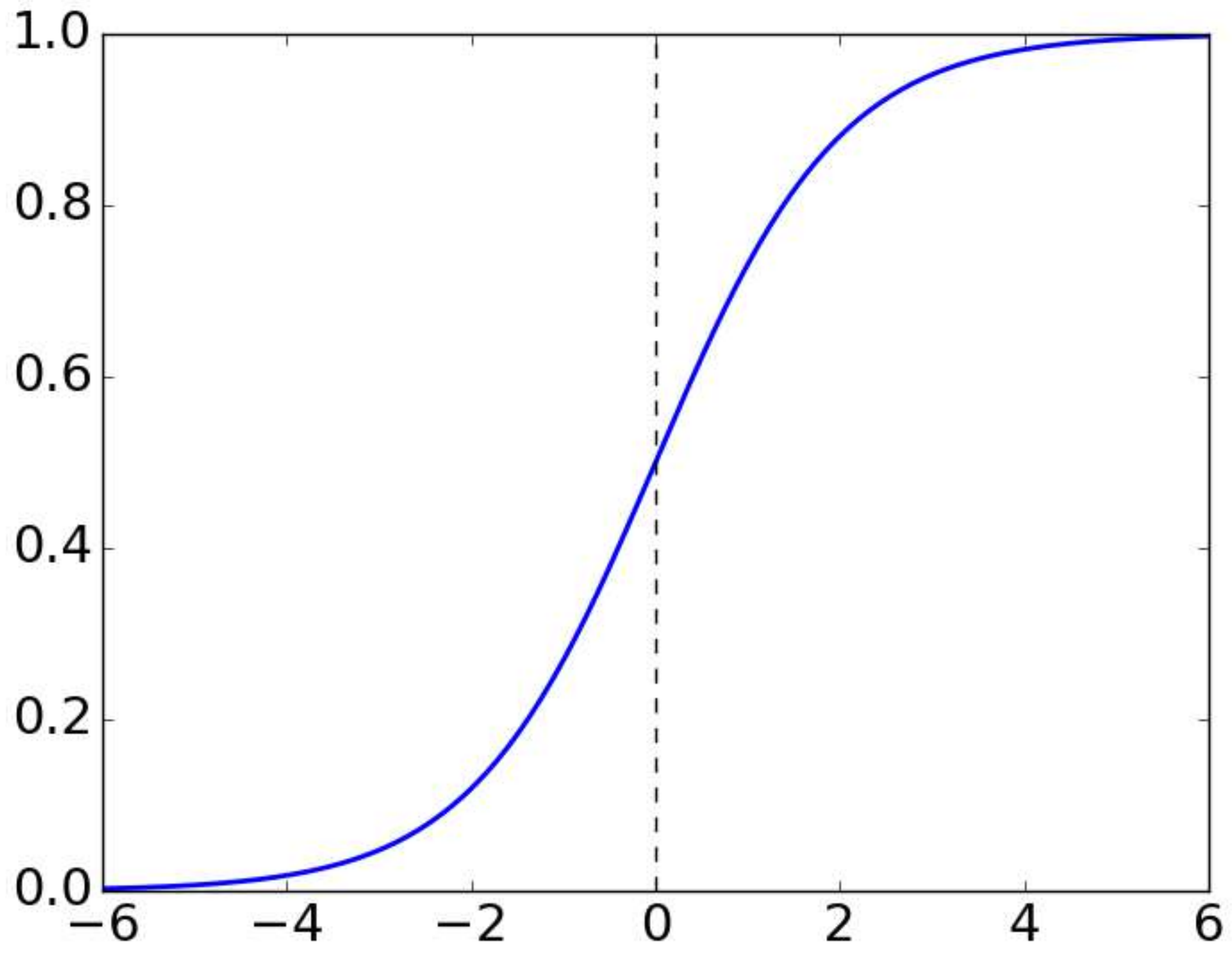
Multilayer Perceptrons

- Common activation functions:



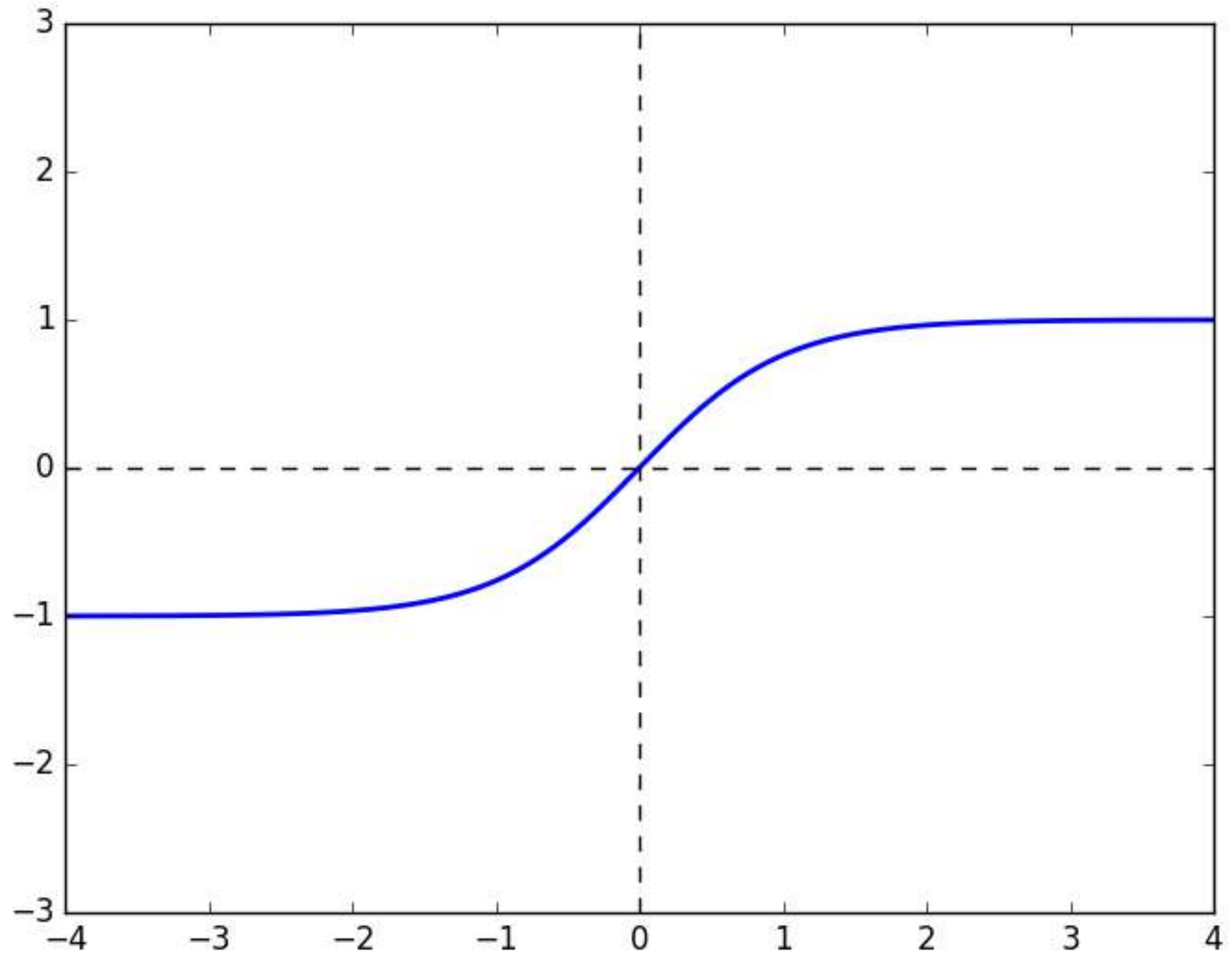
Hard-threshold

$$y = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{if } z \leq 0 \end{cases}$$



Logistic

$$y = \frac{1}{1 + e^{-z}}$$



Hyperbolic tangent

$$y = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

Multilayer perceptrons

- Each layer computes a function, so the network computes a composition of functions:

$$\boldsymbol{h}^{(1)} = f^{(1)}(\boldsymbol{x}) = \phi(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})$$

$$\boldsymbol{h}^{(2)} = f^{(2)}(\boldsymbol{h}^{(1)}) = \phi(\boldsymbol{W}^{(2)}\boldsymbol{h}^{(1)} + \boldsymbol{b}^{(2)})$$

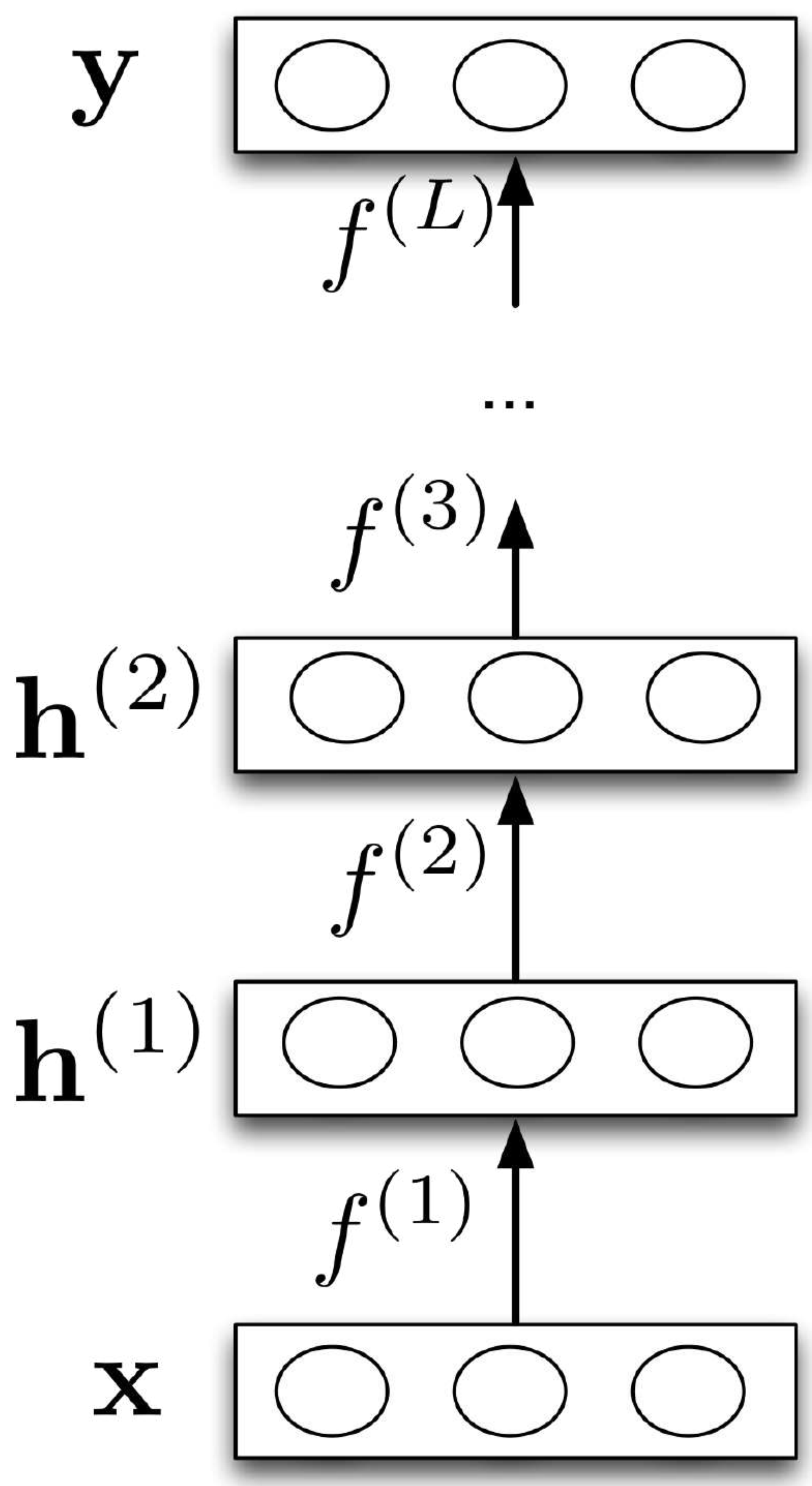
$$\vdots$$

$$\boldsymbol{y} = f^{(L)}(\boldsymbol{h}^{(L-1)})$$

- Or more simply:

$$\boldsymbol{y} = f^{(L)} \circ \cdots \circ f^{(1)}(\boldsymbol{x}) \,.$$

- Neural nets provide modularity: we can implement each layer's computations as a black box.



Loss Function

- A loss function $\mathcal{L}(y, t)$ defines how bad it is if, for some example $\boldsymbol{x}$, the algorithm predicts y, but the target is actually t.
- Squared error loss function: $\mathcal{L}(y, t) = \frac{1}{2}\left(y - t\right)^2$ (but there are many more)
- $y - t$ is the residual, and we want to make this small in magnitude
- The $\frac{1}{2}$ factor is just to make the calculations convenient.
- Cost function: loss function averaged over all training examples

$$\mathcal{J}(\boldsymbol{w}) = \frac{1}{2N}\sum_{i=1}^{N}\left(y^{(i)} - t^{(i)}\right)^2$$

- Terminology varies. Some call "cost" *empirical or average loss.*

Optimization: solving the minimization problem

- We defined a cost function $\mathcal{J}(\boldsymbol{w})$, which we'd like to minimize.
- Recall from calculus: the minimum of a smooth function (if it exists) occurs at a critical point, i.e. point where the derivative is zero.
    - multivariate generalization: set the partial derivatives $\partial \mathcal{J} / \partial w_i$ to zero
    - Equivalently, we can set the gradient to zero. The gradient is the vector of partial derivatives:

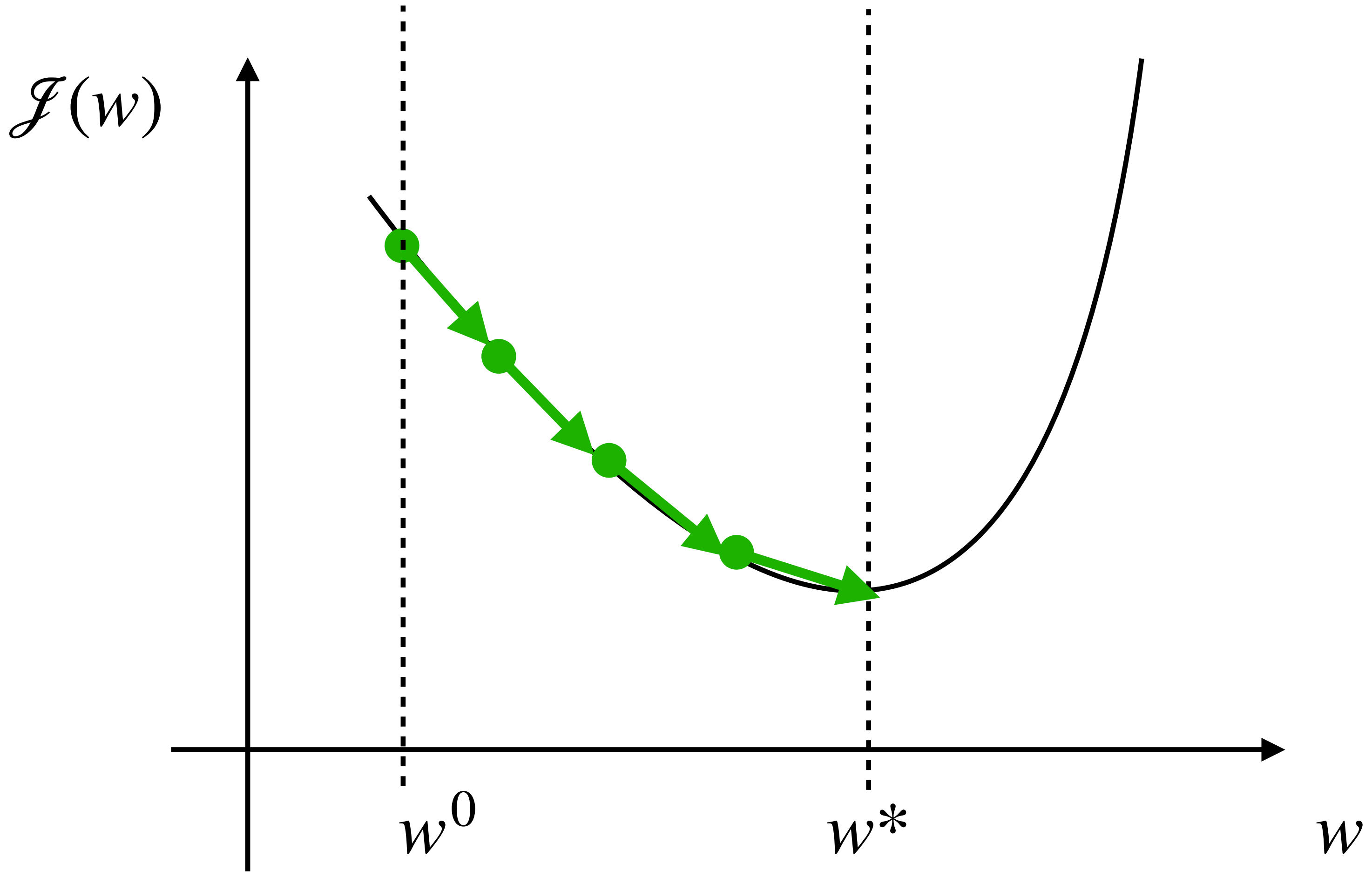
$$\nabla_{\boldsymbol{w}} \mathcal{J} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{w}} = \begin{pmatrix} \frac{\partial \mathcal{J}}{\partial w_1} \\ \vdots \\ \frac{\partial \mathcal{J}}{\partial w_D} \end{pmatrix}$$

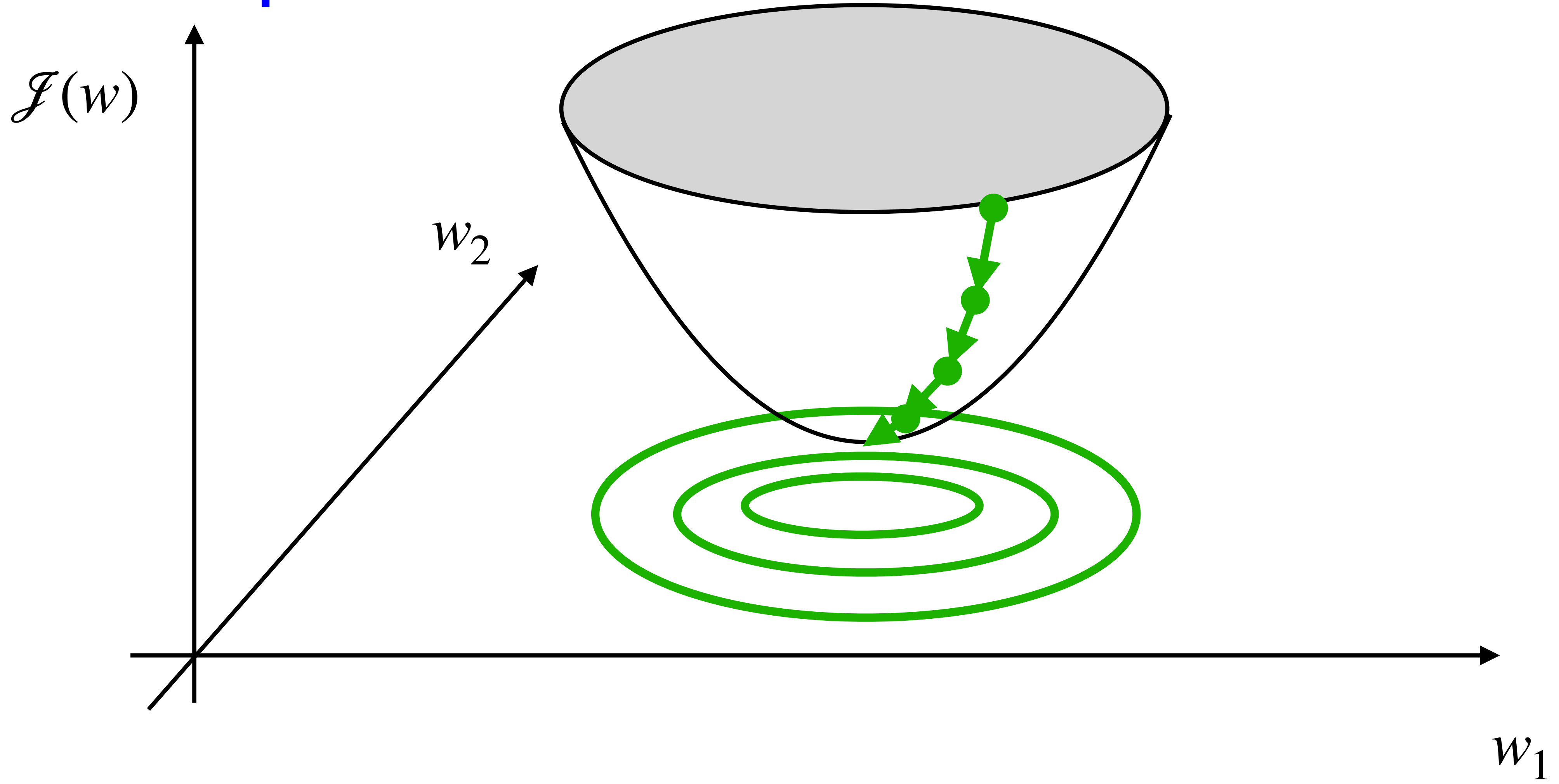
Solutions may be direct or iterative

- Sometimes we can directly find provably optimal parameters (e.g. set the gradient to zero and solve in closed form). We call this a direct solution.
- Iterative solution methods repeatedly apply an update rule that gradually takes us closer to the solution. This applies to neural networks.

Iterative solution: gradient descent

- All optimization problems we cover in these lectures don't have a direct solution.
- To minimize the cost function, we use the broadly applicable gradient descent.
- Gradient descent is an iterative algorithm, which means we apply an update repeatedly until some criterion is met.
- We initialize the weights to something reasonable (e.g. all zeros) and repeatedly adjust them in the direction of steepest descent.





Gradient descent

- Observe:
    - If $\partial \mathcal{J} / \partial w_j > 0$, then increasing w_j increases $\mathcal{J}$.
    - If $\partial \mathcal{J} / \partial w_j < 0$, then increasing w_j decreases $\mathcal{J}$.
- The following update always decreases the cost function for small enough α (unless $\partial \mathcal{J} / \partial w_j = 0$):

$$w_j \leftarrow w_j - \alpha \frac{\partial \mathcal{J}}{\partial w_j}$$

- $\alpha > 0$ is a learning rate (or step size). The larger it is, the faster $\boldsymbol{w}$ changes.
    - We'll see later how to tune the learning rate, but values are typically small, e.g. 0.01 or 0.0001.
    - If cost is the sum of N individual losses rather than their average, smaller learning rate will be needed ($\alpha' = \alpha / N$).

Gradient descent

- This gets its name from the gradient. Recall the definition:

$$\nabla_{\boldsymbol{w}} \mathcal{J} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{w}} = \begin{pmatrix} \frac{\partial \mathcal{J}}{\partial w_1} \\ \vdots \\ \frac{\partial \mathcal{J}}{\partial w_D} \end{pmatrix}$$

    - This is the direction of fastest increase in $\mathcal{J}$.

- Update rule in vector form:

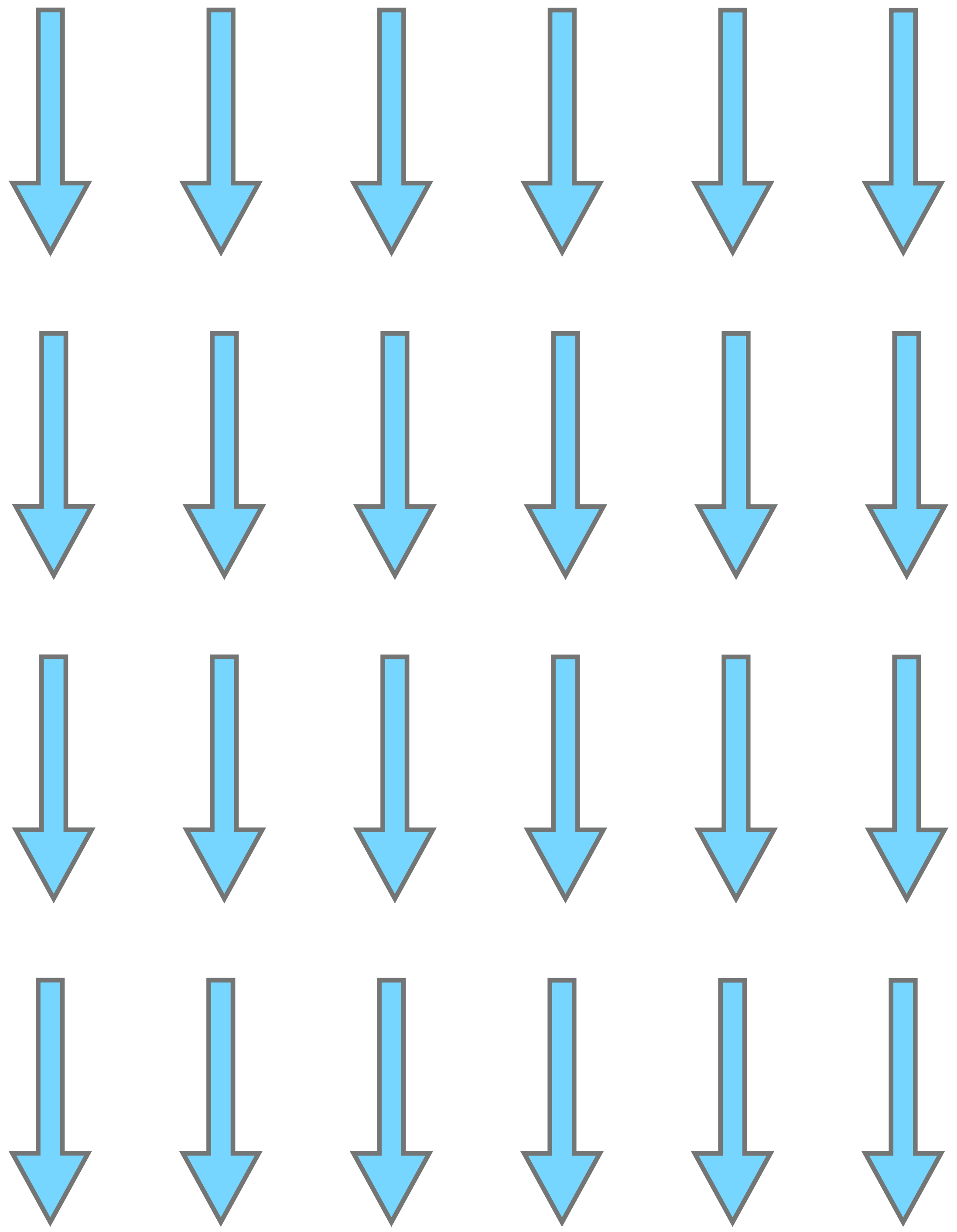
$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha \frac{\partial \mathcal{J}}{\partial \boldsymbol{w}}$$

- So gradient descent updates $\boldsymbol{w}$ in the direction of fastest decrease.
- Observe that once it converges, we get a critical point, i.e., $\frac{\partial \mathcal{J}}{\partial \boldsymbol{w}} = \mathbf{0}$

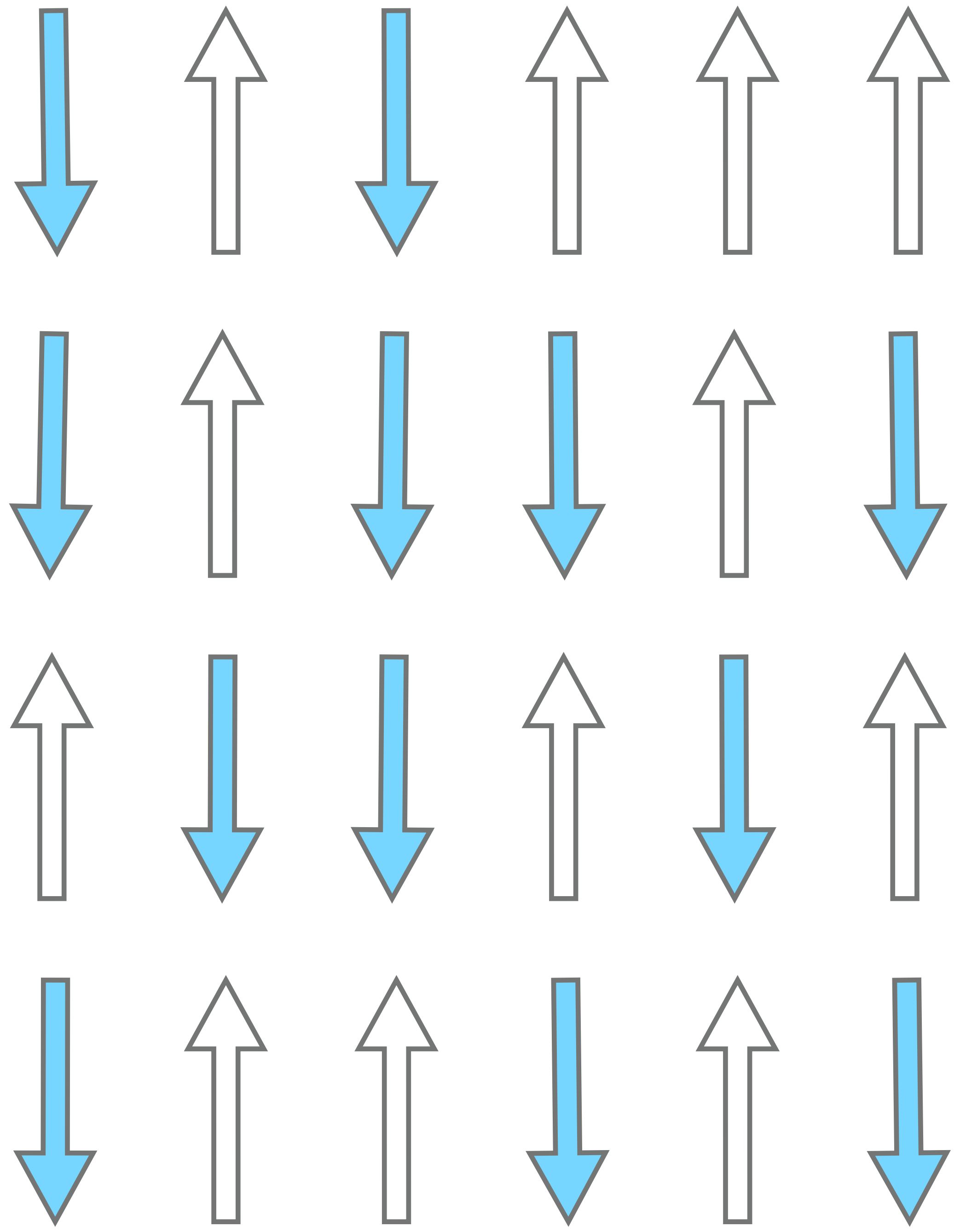
PHASES, PHASE TRANSITIONS, AND THE ORDER PARAMETER

Ising ferromagnet in two dimensions

$$E = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j \qquad \sigma_i = \pm 1$$



Temperature

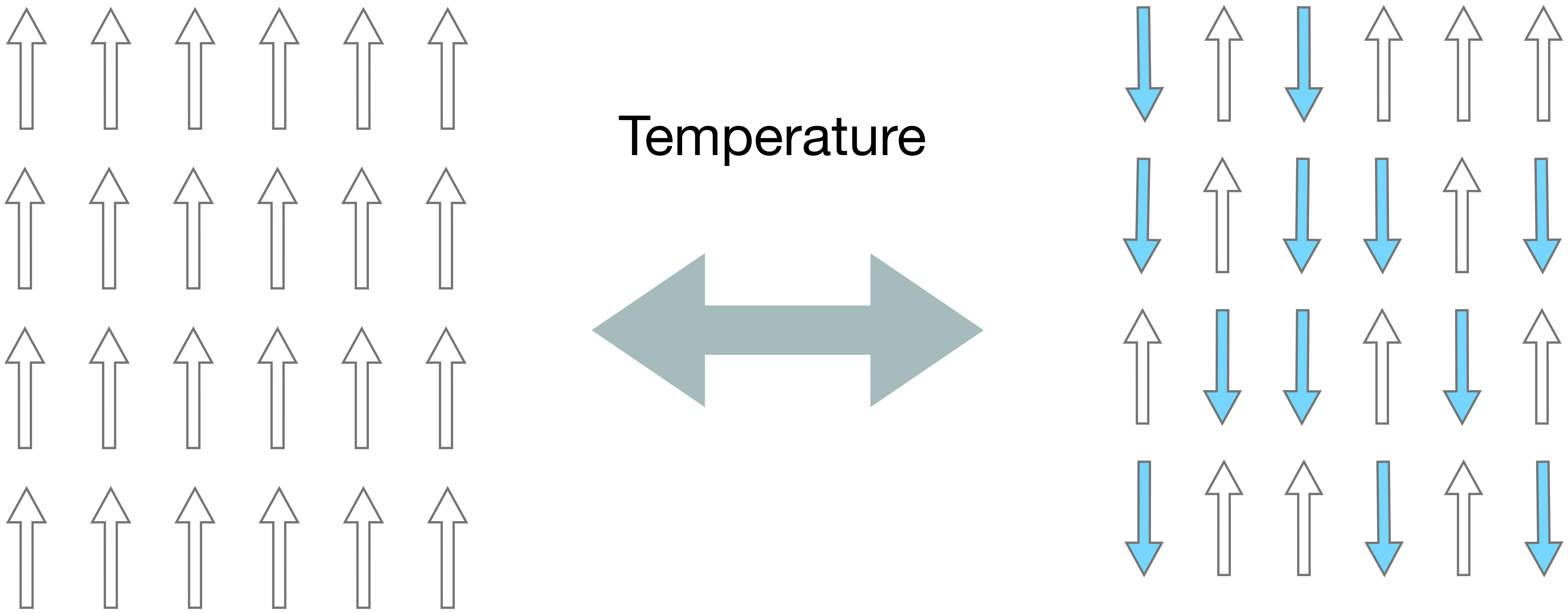


Ferromagnet *Lars Onsager Phys. Rev. 65, 117* **Paramagnet**

PHASES, PHASE TRANSITIONS, AND THE ORDER PARAMETER

Ising ferromagnet in two dimensions

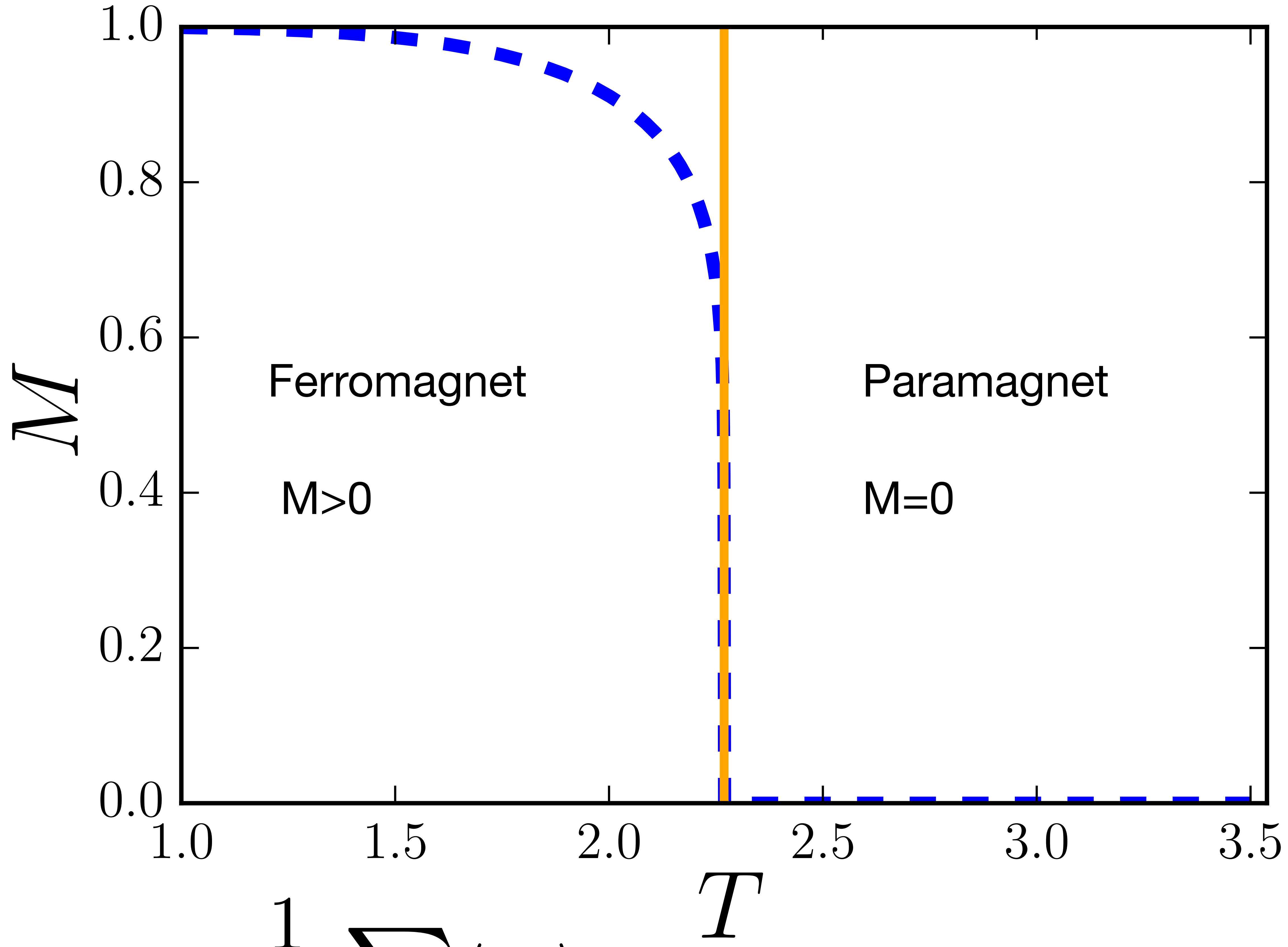
$$E = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j \qquad \sigma_i = \pm 1$$



Ferromagnet *Lars Onsager Phys. Rev. 65, 117* **Paramagnet**

Phases, phase transitions, and the order parameter

Ferromagnetic transition: order parameter

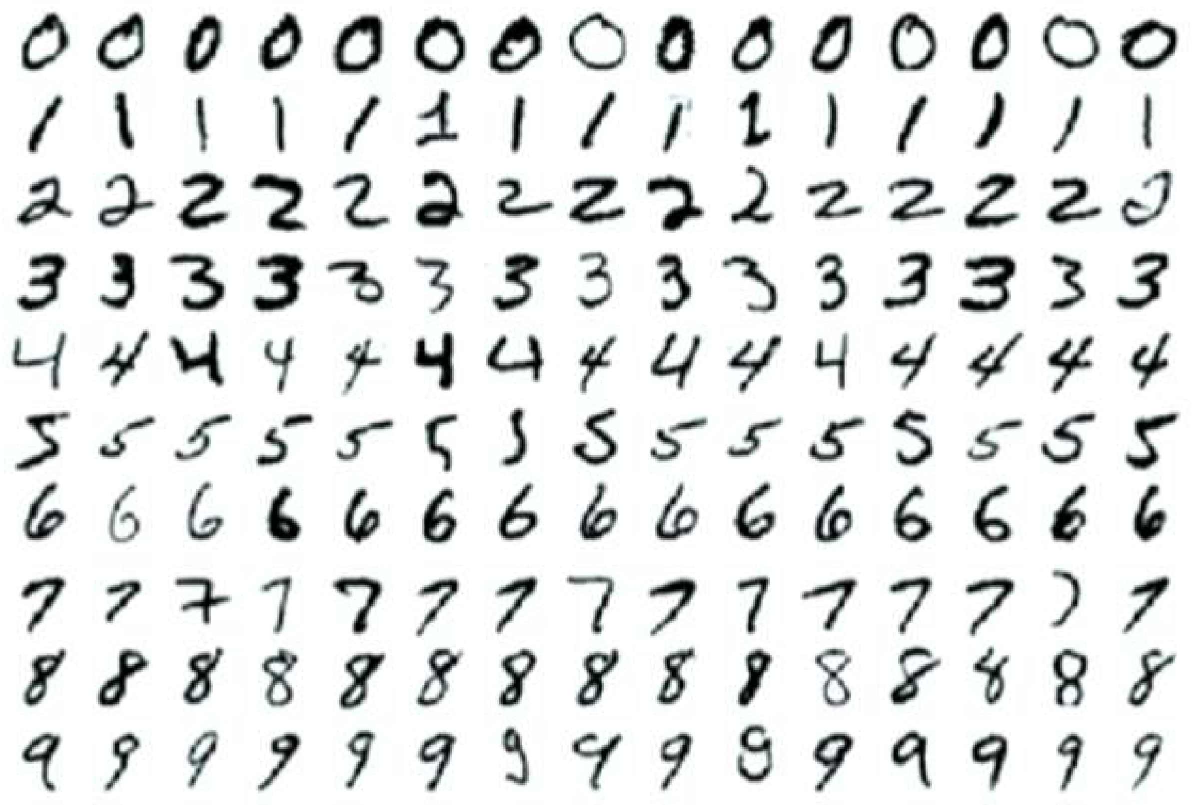


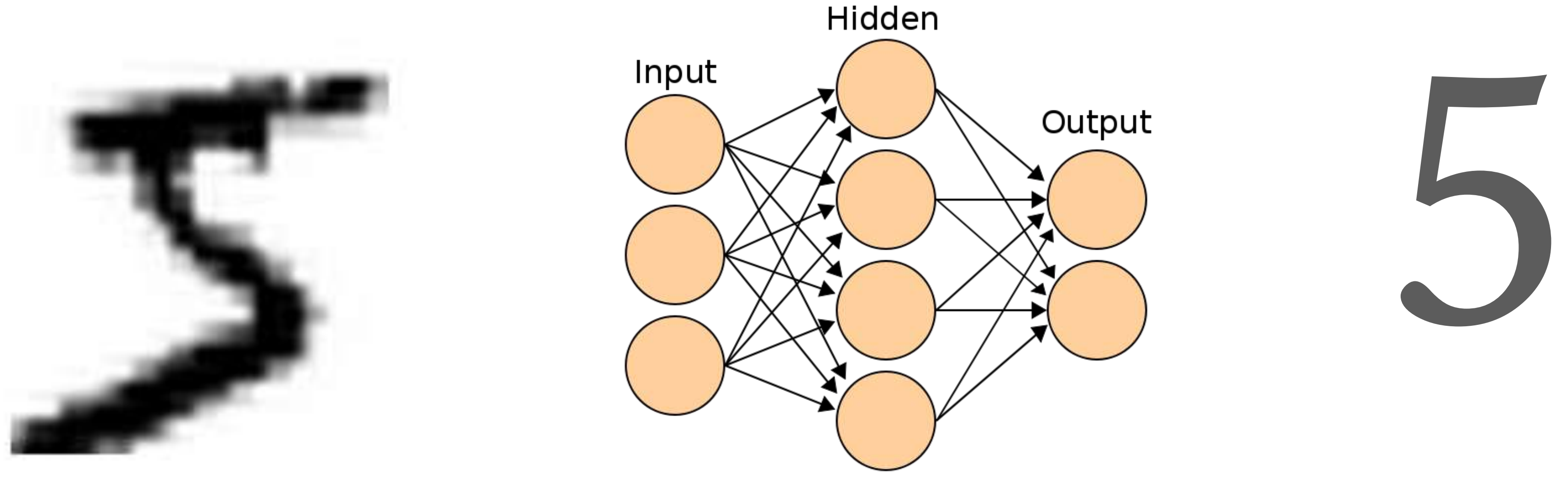
$$M = \frac{1}{N}\sum_i \langle \sigma_i \rangle, \qquad \sigma_i = \pm 1$$

It is a measure of the degree of order in the system

Lars Onsager Phys. Rev. 65, 117 (1944)

Inspiration: fluctuations handwritten digits (mnist)

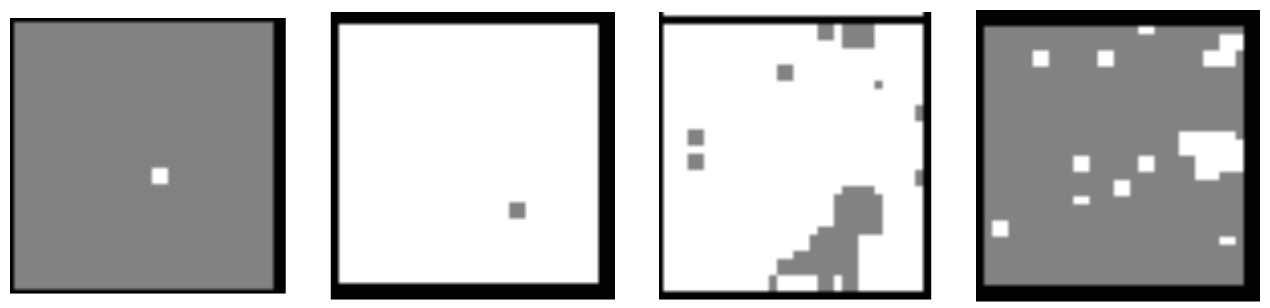


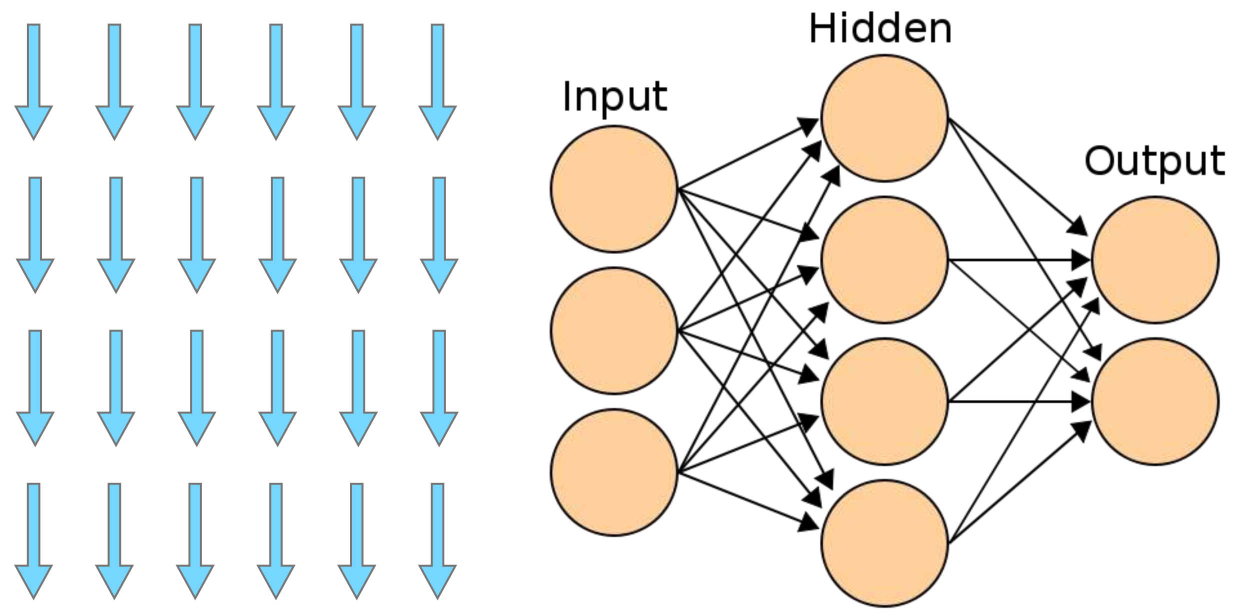


ML community has developed powerful supervised learning algorithms

 $=5+$ fluctuations

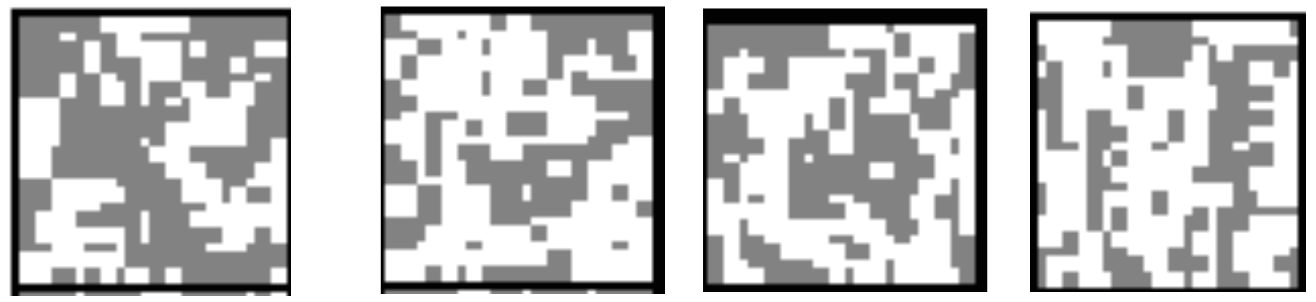
FM phase





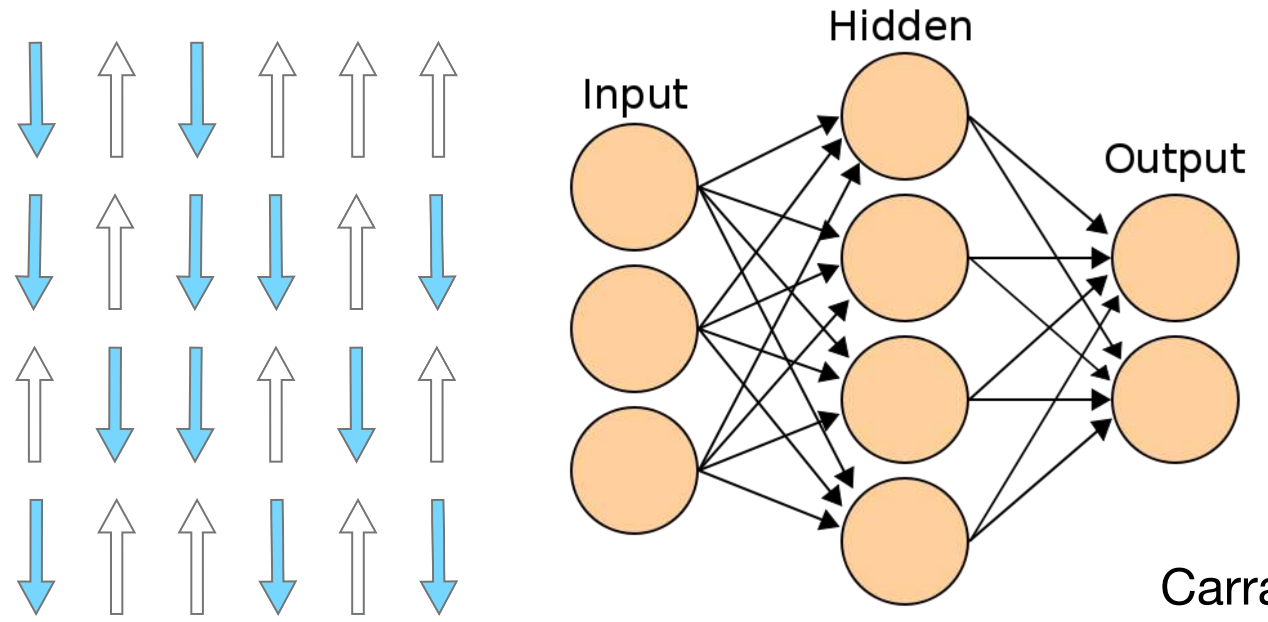
FM (0)

High T phase



grey=spin up

white=spin down



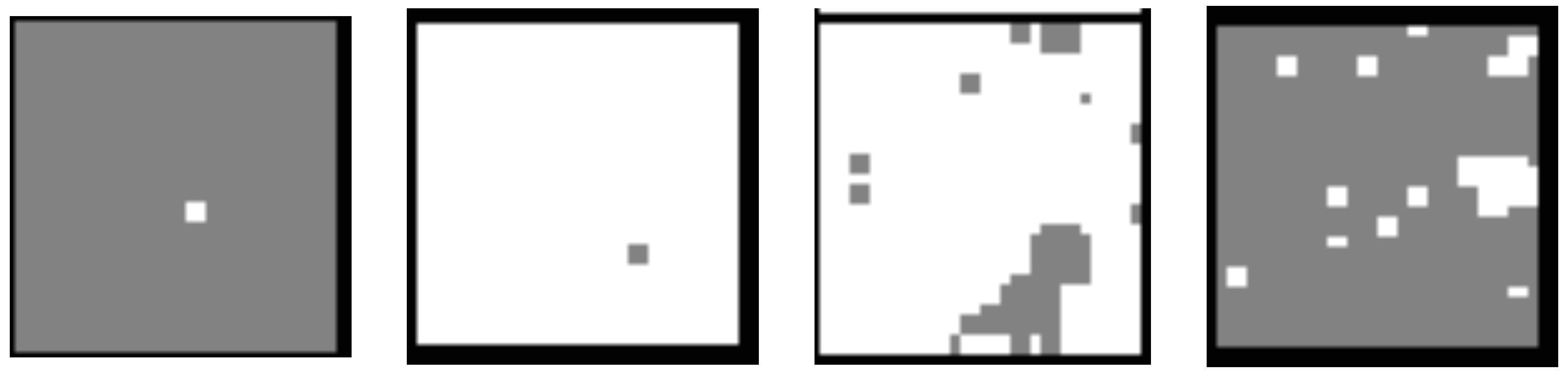
PM (1)

Carrasquilla and Melko. Nature Physics 13, 431–434 (2017)

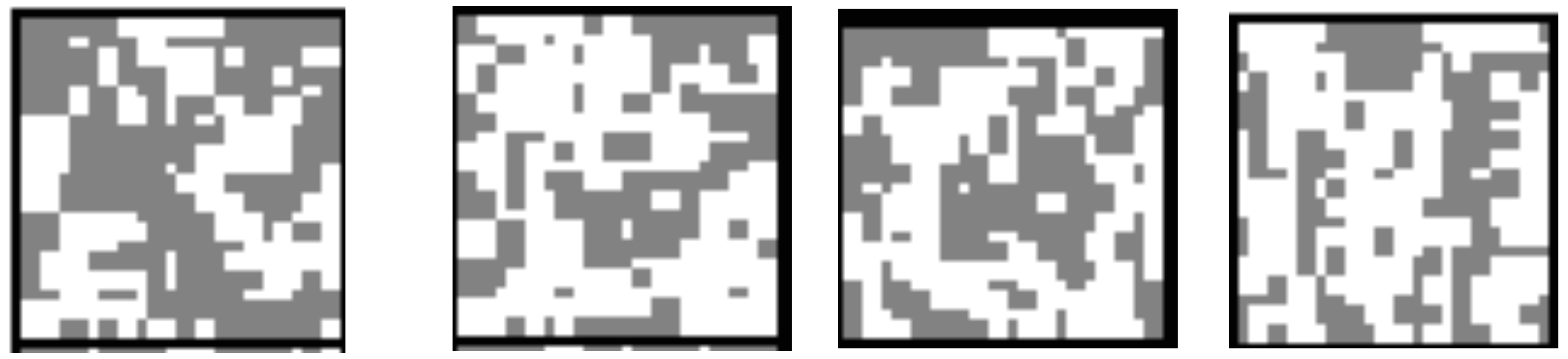
SQUARE LATTICE ISING MODEL

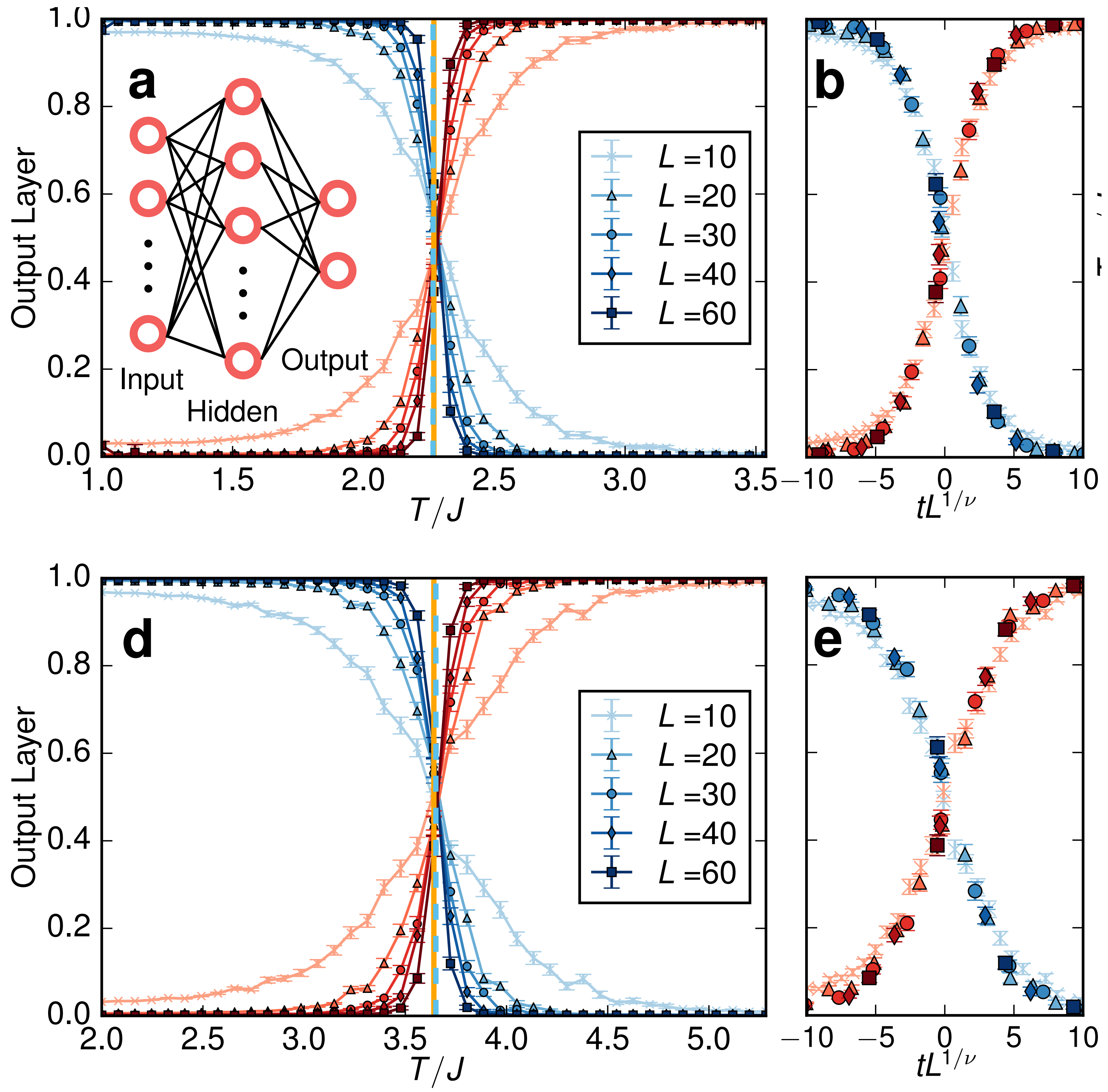
$$T_c/J = 2/\ln\left(1+\sqrt{2}\right)$$

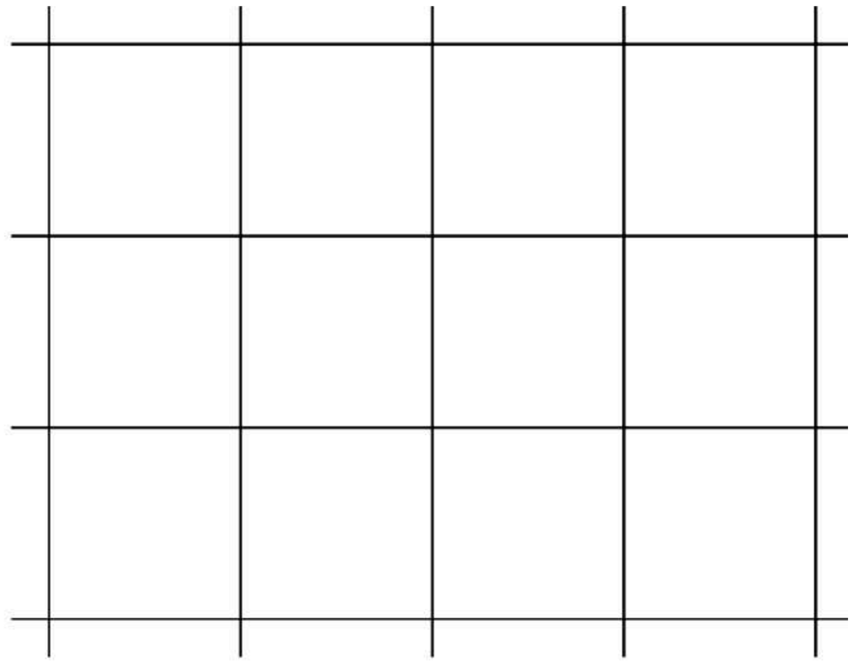
2D Ising model in the ordered phase

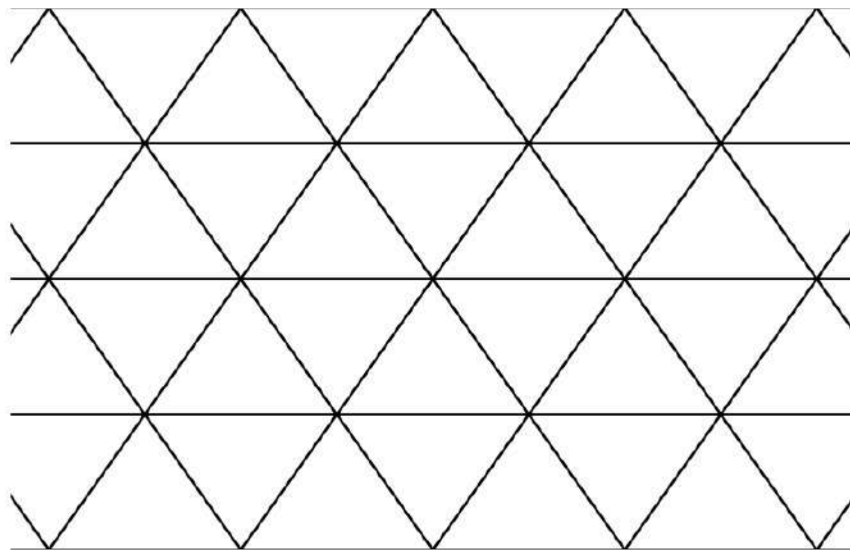


disordered phase









Neural networks get trained to understand physical concepts such as the the order parameter and the energy

Tc of Triangular within <1%

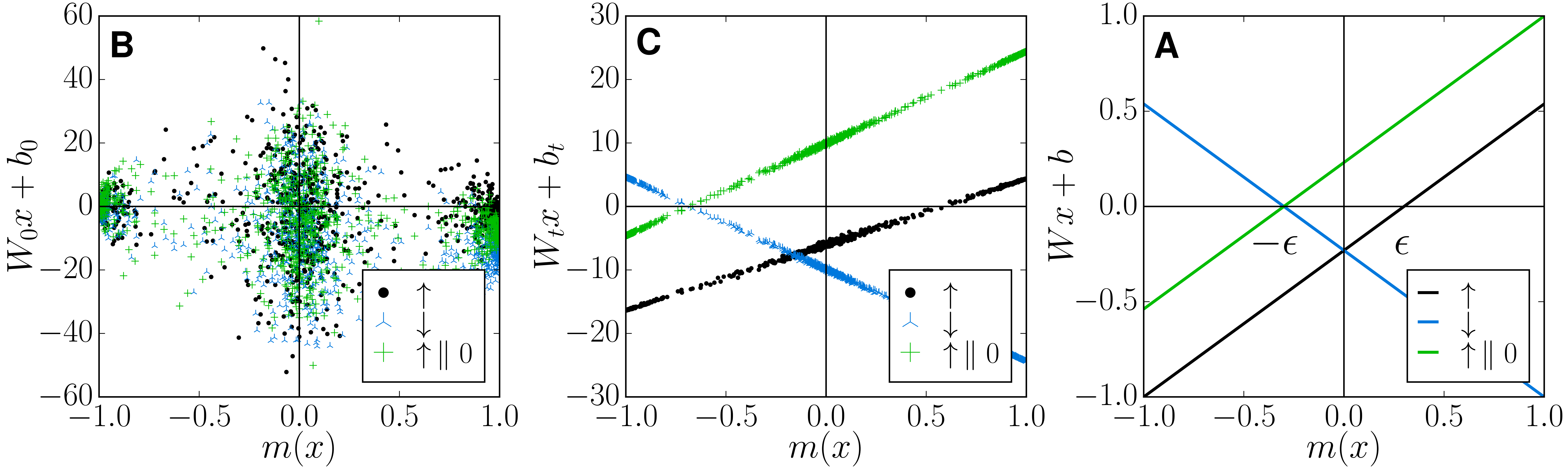
Analytical understanding: toy model with 3 analytically trained neutrons. NN relies on the **magnetization** of the system

$$\nu \approx 1$$

J. Carrasquilla and R. G. Melko. Nature Physics 13, 431–434 (2017)

Toy model

Investigating the argument of the hidden layer during the training



$$W = \frac{1}{N(1+\epsilon)} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ -1 & -1 & \cdots & -1 \\ 1 & 1 & \cdots & 1 \end{pmatrix}, \text{ and } b = \frac{\epsilon}{(1+\epsilon)} \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix}, \qquad Wx + b = \frac{1}{(1+\epsilon)} \begin{pmatrix} m(x) - \epsilon \\ -m(x) - \epsilon \\ m(x) + \epsilon \end{pmatrix},$$

$$x = [\sigma_1 \sigma_2, \ldots, \sigma_N]^{\mathrm{T}} \qquad m(x) = \frac{1}{N} \sum_{i=1}^{N} \sigma_i$$

J. Carrasquilla and R. G. Melko. Nature Physics 13, 431–434 (2017)

Toy model

arXiv > cond-mat > arXiv:2010.00351

Search... All fields Search

Help | Advanced Search

Condensed Matter > Statistical Mechanics

[Submitted on 1 Oct 2020 (v1), last revised 17 Feb 2021 (this version, v2)]

Emergence of a finite-size-scaling function in the supervised learning of the Ising phase transition

Dongkyu Kim, Dong-Hee Kim

We investigate the connection between the supervised learning of the binary phase classification in the ferromagnetic Ising model and the standard finite-size-scaling theory of the second-order phase transition. Proposing a minimal one-free-parameter neural network model, we analytically formulate the supervised learning problem for the canonical ensemble being used as a training data set. We show that just one free parameter is capable enough to describe the data-driven emergence of the universal finite-size-scaling function in the network output that is observed in a large neural network, theoretically validating its critical point prediction for unseen test data from different underlying lattices yet in the same universality class of the Ising criticality. We also numerically demonstrate the interpretation with the proposed one-parameter model by providing an example of finding a critical point with the learning of the Landau mean-field free energy being applied to the real data set from the uncorrelated random scale-free graph with a large degree exponent.

Subjects: **Statistical Mechanics (cond-mat.stat-mech)**; Machine Learning (cs.LG); Machine Learning (stat.ML)

Cite as: arXiv:2010.00351 **[cond-mat.stat-mech]**

(or arXiv:2010.00351v2 **[cond-mat.stat-mech]** for this version)

https://doi.org/10.48550/arXiv.2010.00351

Journal reference: J. Stat. Mech. (2021) 023202

Related DOI: https://doi.org/10.1088/1742-5468/abdc18

Download:

- PDF
- Other formats

(license)

Current browse context:
cond-mat.stat-mech

< prev | next >

new | recent | 2010

Change to browse by:

cond-mat
cs
cs.LG
stat
stat.ML

References & Citations

- NASA ADS
- Google Scholar
- Semantic Scholar

Export Bibtex Citation

Bookmark

Toy model

arXiv > cond-mat > arXiv:2010.00351

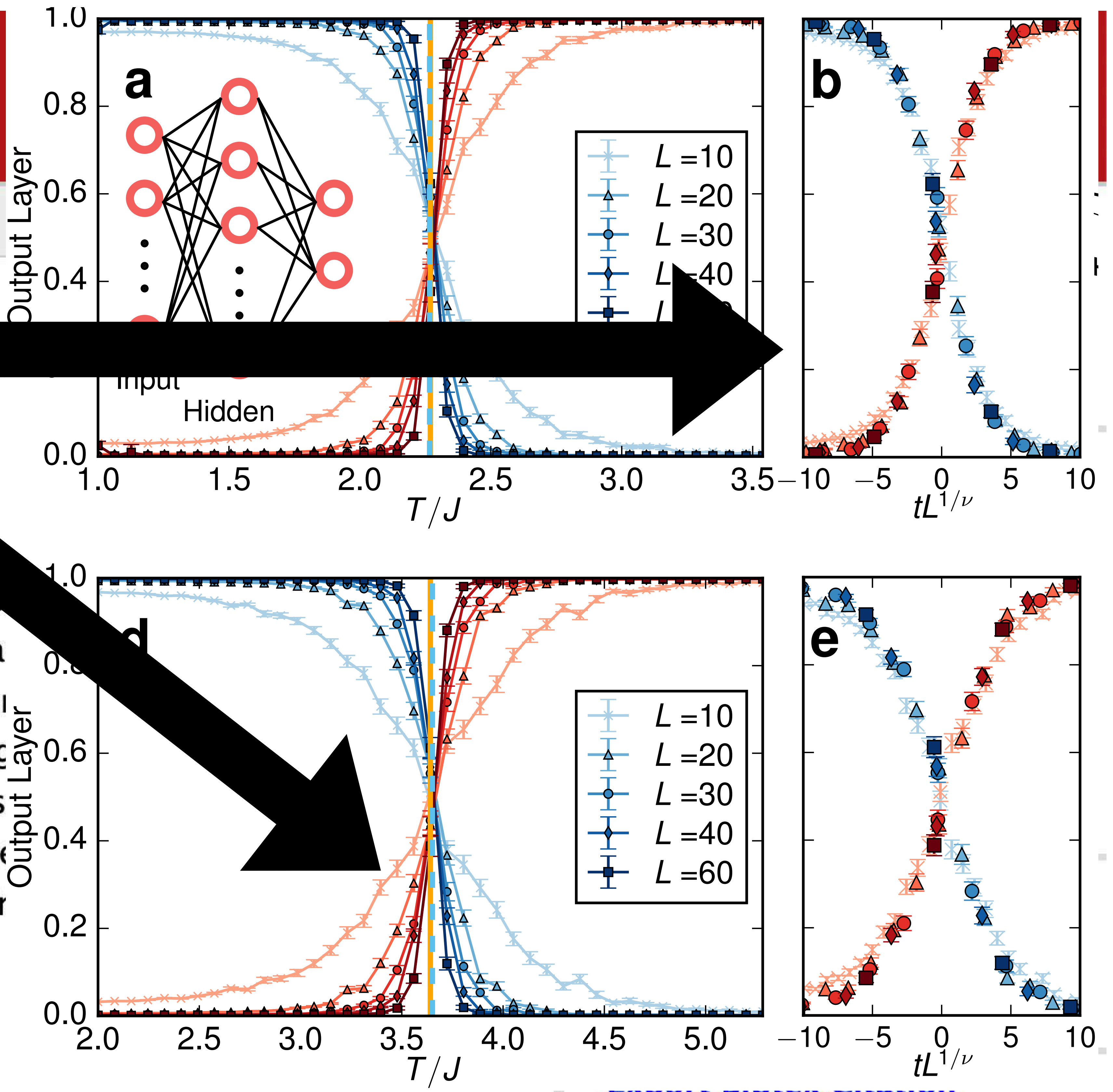
Condensed Matter > Statistical Mechanics

[Submitted on 1 Oct 2020 (v1), last revised 17 Feb 2021 (this version, v2)]

Emergence of a finite-size-scaling function in the supervised
transition

Dongkyu Kim, Dong-Hee Kim

We investigate the connection between the supervised learning of the binary phase classification in the ferromagnetic Ising mo
size-scaling theory of the second-order phase transition. Proposing a minimal one-free-parameter neural network model, we a
supervised learning problem for the canonical ensemble being used as a training data set. We show that just one free parameter
describe the data-driven emergence of the universal finite-size-scaling function in the network output that is observed in a larg
theoretically validating its critical point prediction for unseen test data from different underlying lattices yet in the same univers
criticality. We also numerically demonstrate the interpretation with the proposed one-parameter model by providing an example
with the learning of the Landau mean-field free energy being applied to the real data set from the uncorrelated random scale-fr
degree exponent.



Subjects: **Statistical Mechanics (cond-mat.stat-mech)**; Machine Learning (cs.LG); Machine Learning (stat.ML)
Cite as: arXiv:2010.00351 **[cond-mat.stat-mech]**
(or arXiv:2010.00351v2 **[cond-mat.stat-mech]** for this version)
https://doi.org/10.48550/arXiv.2010.00351
Journal reference: J. Stat. Mech. (2021) 023202
Related DOI: https://doi.org/10.1088/1742-5468/abdc18

Bookmark

Message

- NN analysis of the models enable us to make quantitative predictions about the critical points of the models and can even inform us about universal properties of such critical points without a priori knowledge about the pertinent order parameters or energy.

- Only from raw configurations obtained numerically or experimentally.

- We have made use of the knowledge of the critical point to construct our datasets. This limitation was quickly lifted and there are approaches that do not require any knowledge about the transition. See e.g. “Learning phase transitions by confusion” Evert P. L. van Nieuwenburg, Ye-Hua Liu & Sebastian D. Huber. Nature Physics volume 13, pages 435–439 (2017)

Can we deal with disordered and topological phases not described by order parameters?

Phases of matter without an order parameter at T=0

- Topological phases of matter. Examples: Fractional quantum hall effect, quantum spin liquids, Ising gauge theory. Potential applications in topological quantum computing. These phases defy the Landau symmetry breaking classification.

- Coulomb phases = Highly correlated “spin liquids” described by electrodynamics. Examples: Common water ice and spin ice materials ($Ho_2Ti_2O_7$ and $Dy_2Ti_2O_7$)

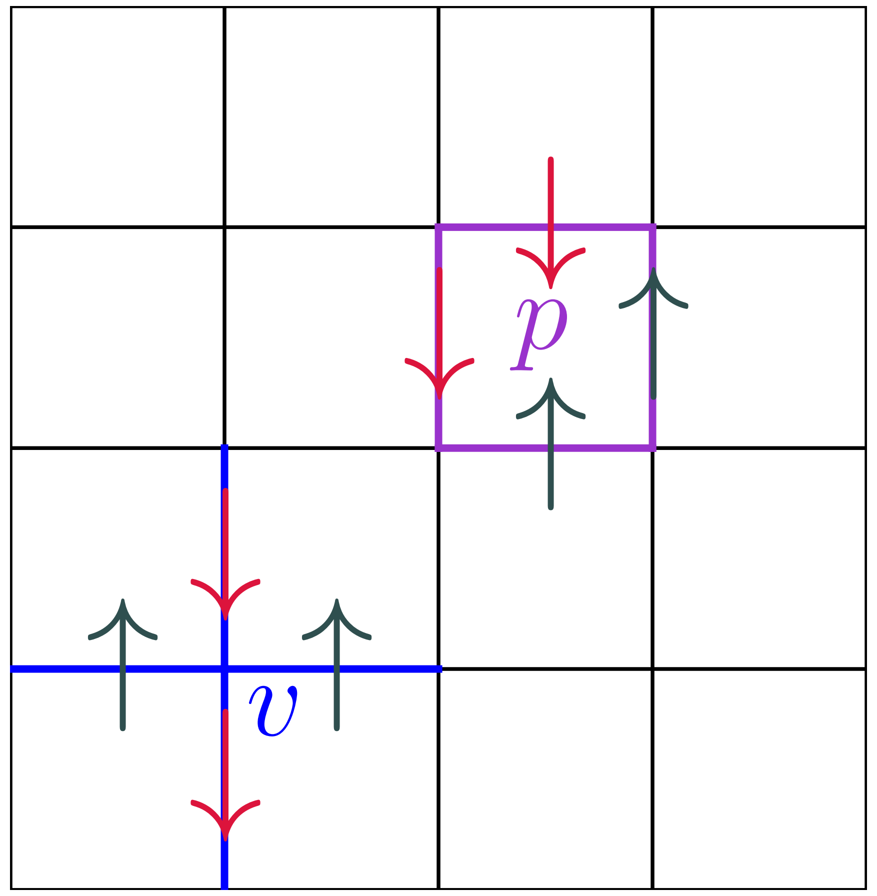
Phases of matter without an order parameter at T=0,∞

Wegner's Ising gauge theory:

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

F.J. Wegner, J. Math. Phys. 12 (1971) 2259

(Kogut Rev. Mod. Phys. 51, 659 (1979))

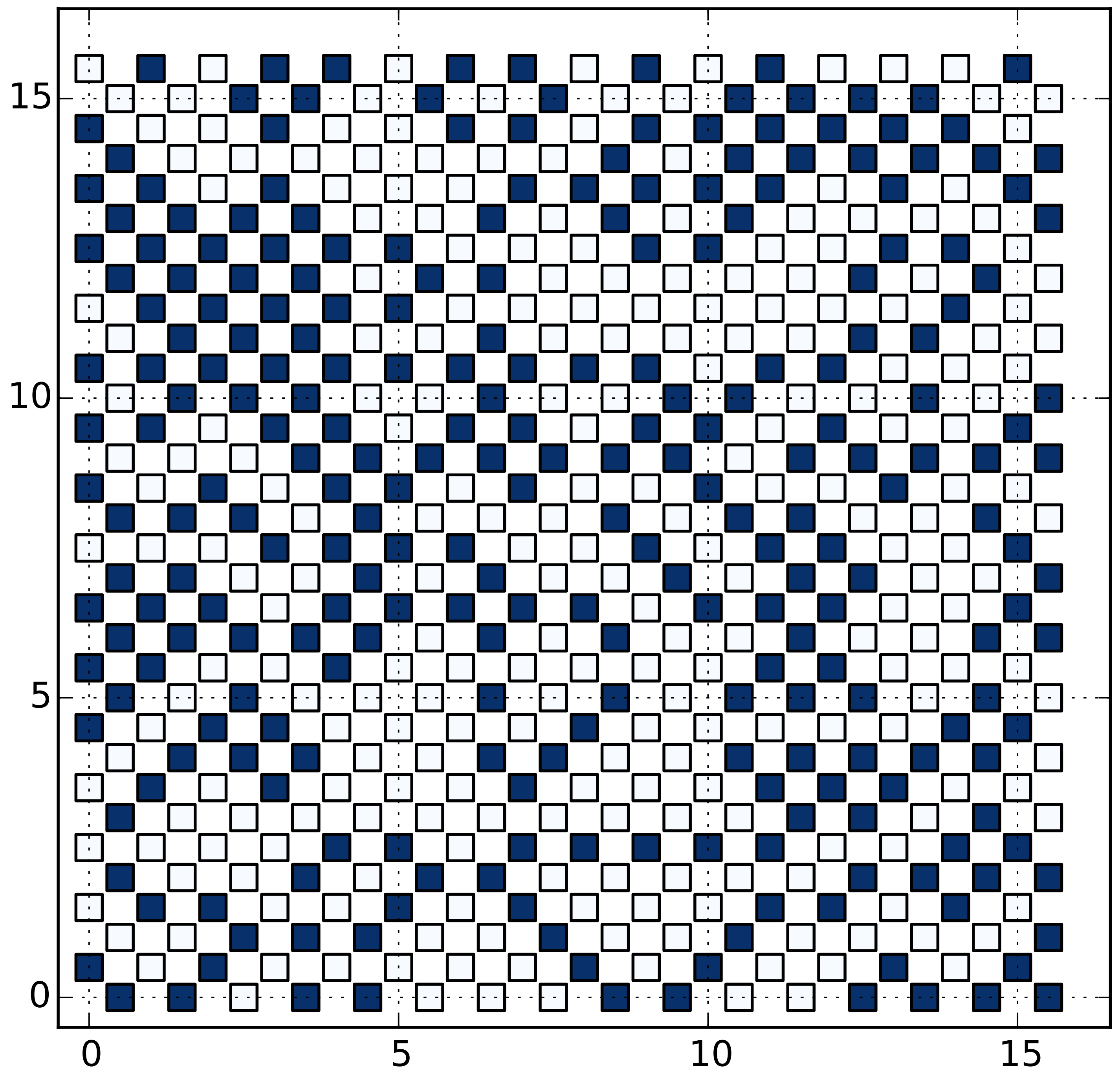


The grandmother of most lattice models for topological quantum computation

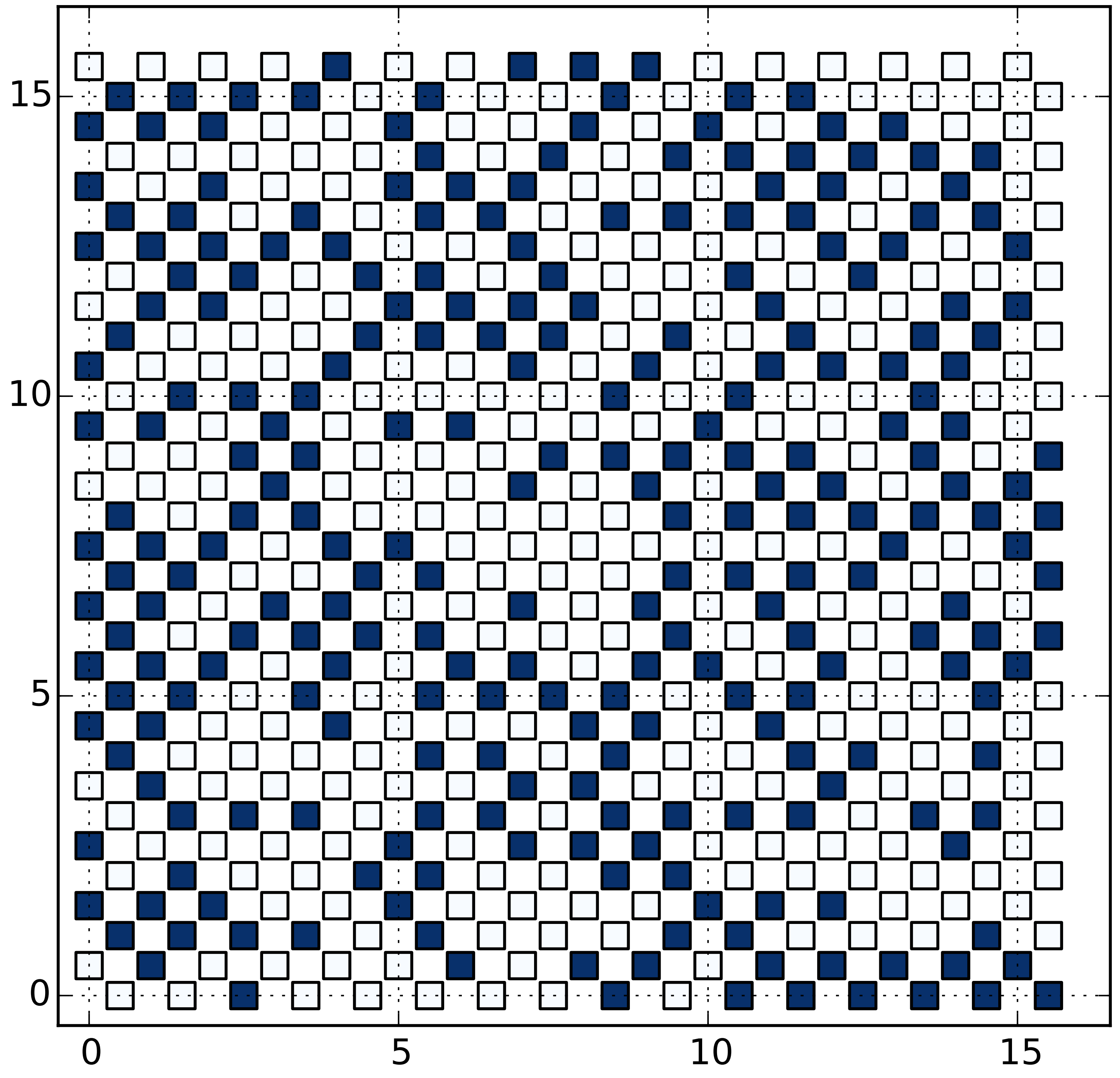
The ground state is a highly degenerate manifold with exponentially decaying spin–spin correlations. Ground state is a disordered topologically ordered phase

Castelnovo and Chamon Phys. Rev. B 76, 174416 (2007)

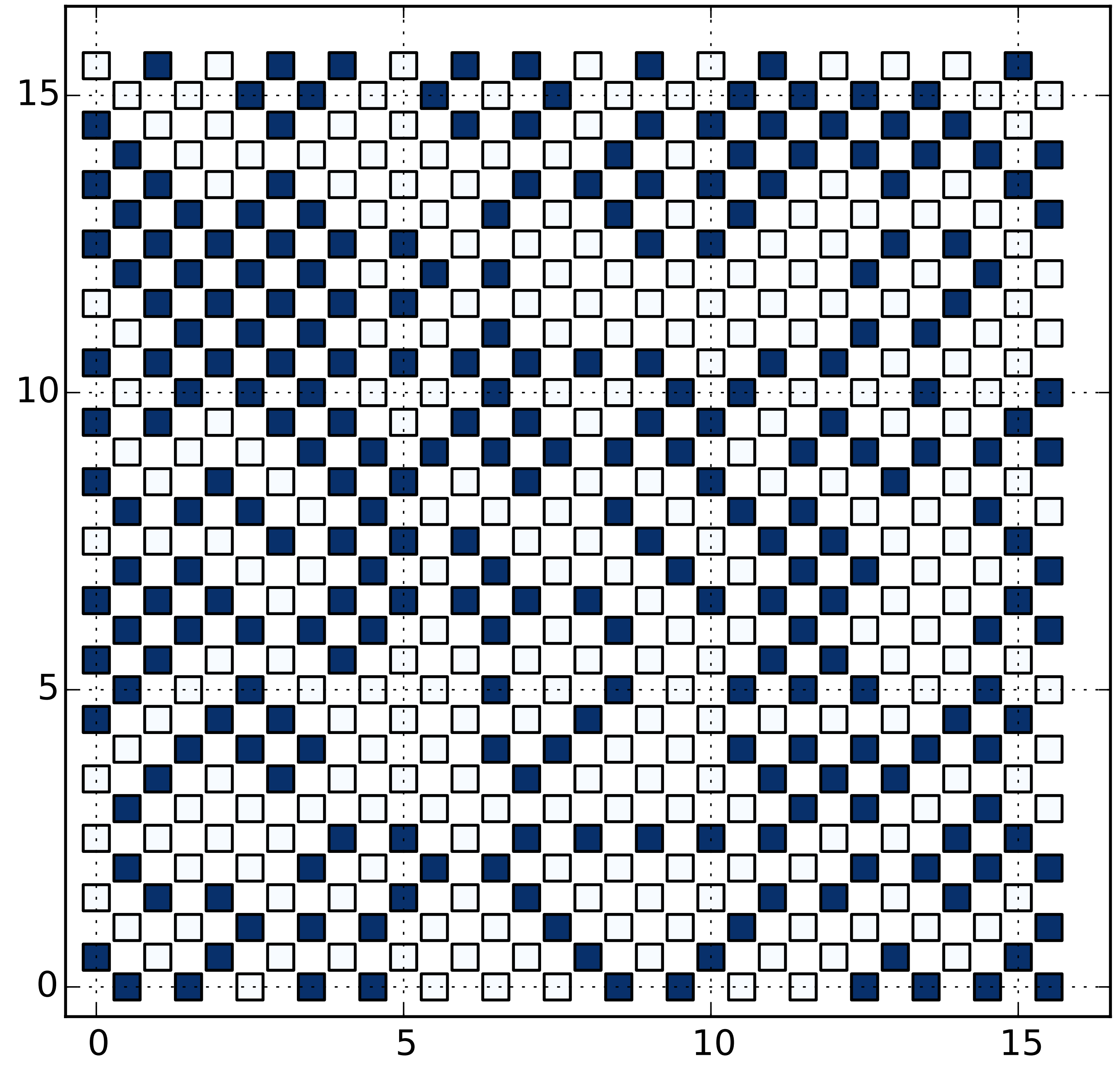
For two configurations



?

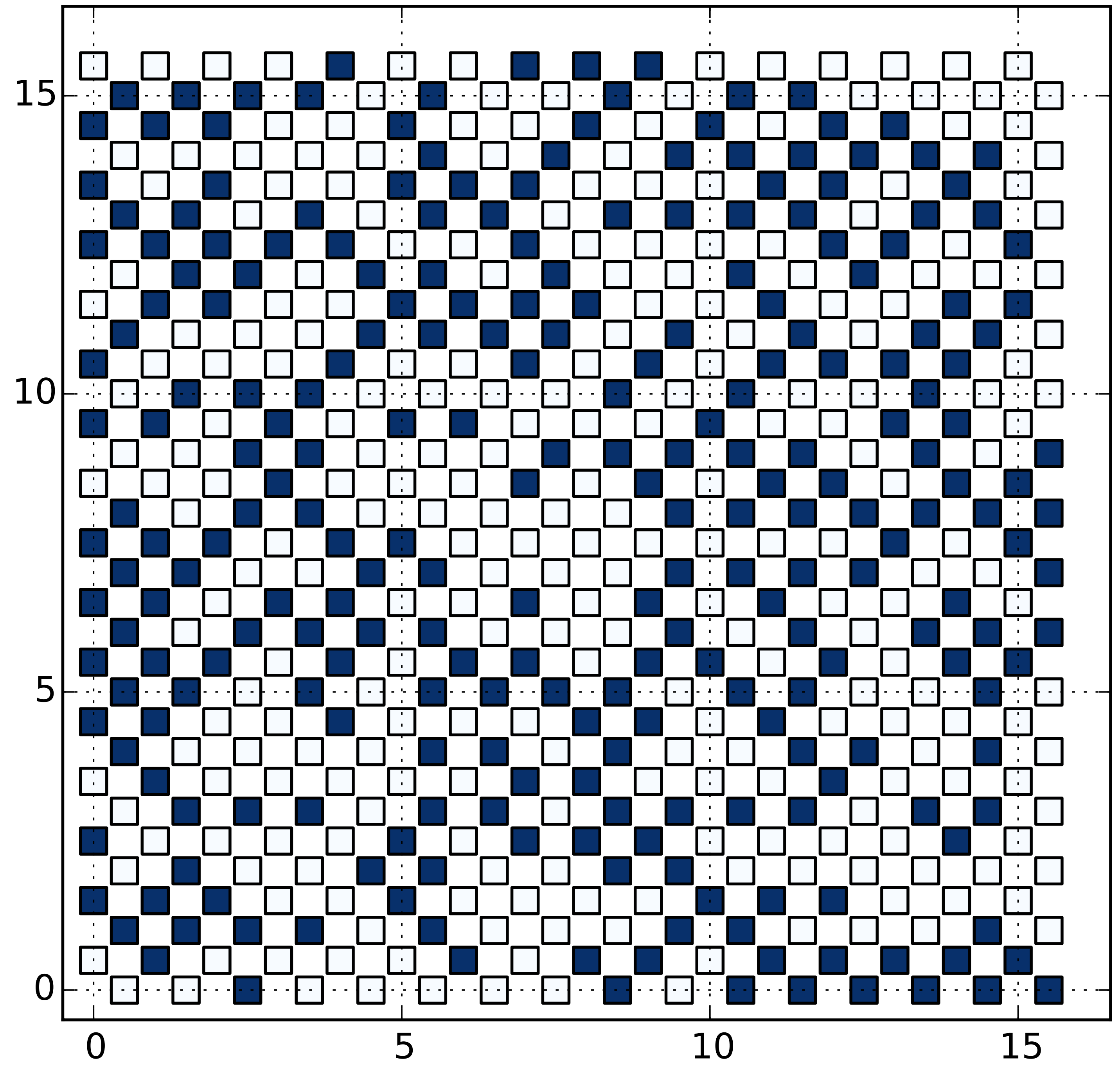


For two configurations



Ground state

?



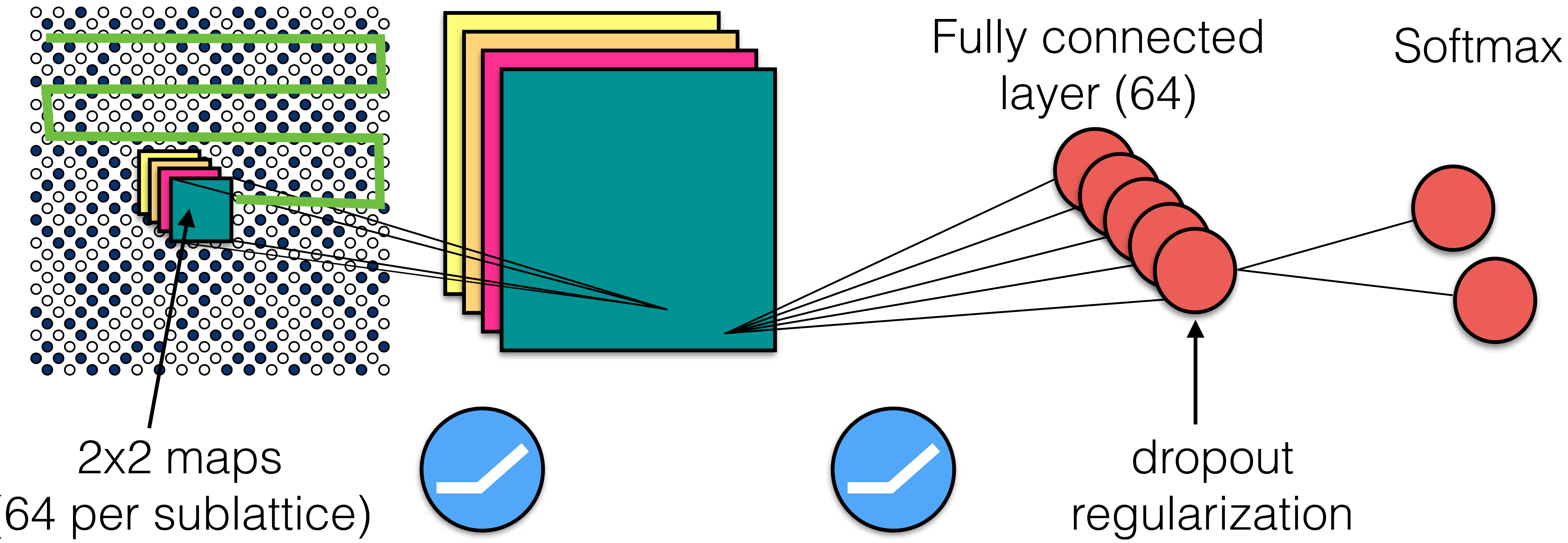
high-temperature state

Feedforward NN are difficult to apply to this problem and lead to 50% accuracy

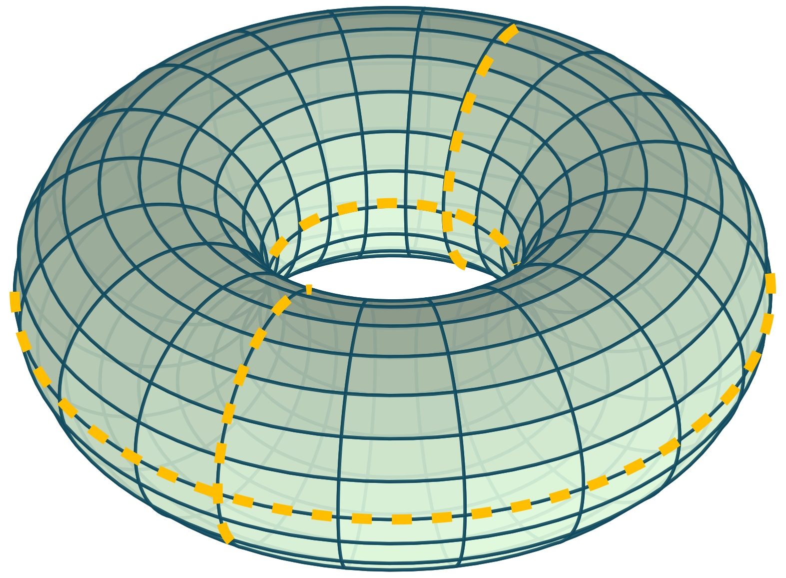
Ising gauge theory

F.J. Wegner, J. Math. Phys. 12 (1971) 2259

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



99% accuracy
easy to train

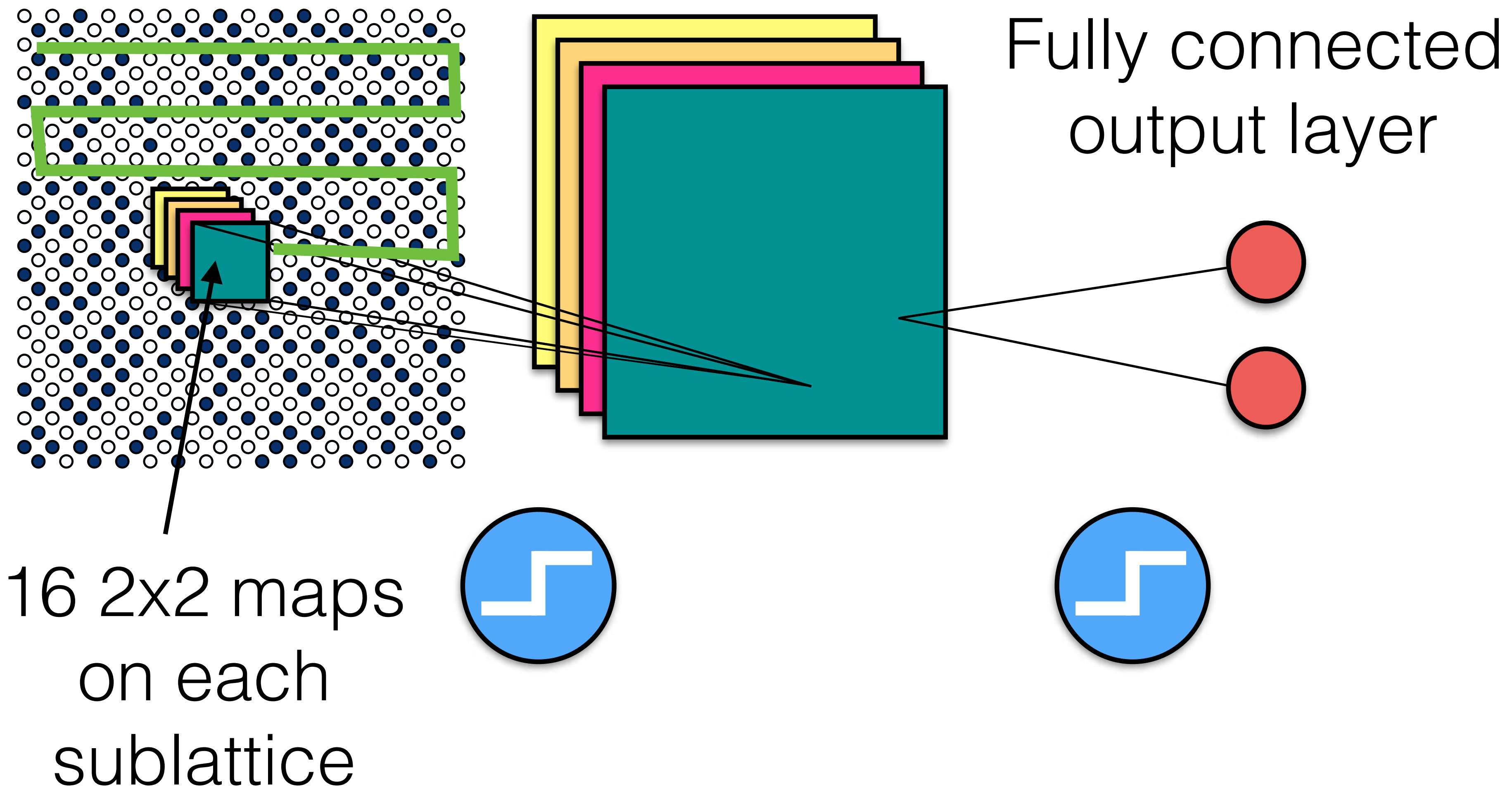


The picture we draw for what the CNN is using to distinguish the phases is that of the detection of satisfied local constraints. In few words, the neural network figures out the energy and uses it to classify states

Analytical understanding: What does the CNN use to make predictions?

- Based on this observation we derived the weights of a streamlined convolutional network analytically designed to work well on our test sets.

$$O_{\text{cold}}(\sigma_1, \ldots, \sigma_N) \propto \lim_{\beta \to \infty} \exp \beta J \sum_p \prod_{i \in p} \sigma_i^z$$



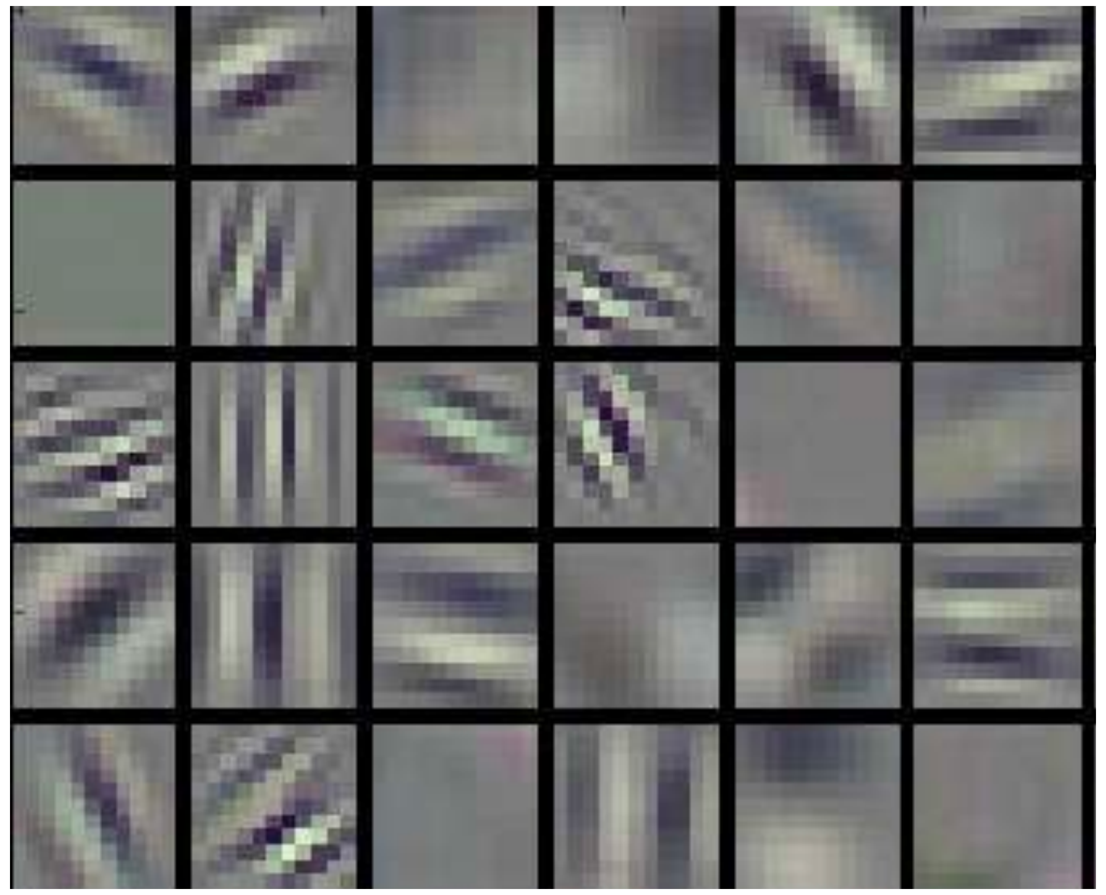
100% accuracy on test sets

Sub-lattices act as color channel in computer vision applications

J. Carrasquilla and R. G. Melko. Nature Physics 13, 431–434 (2017)

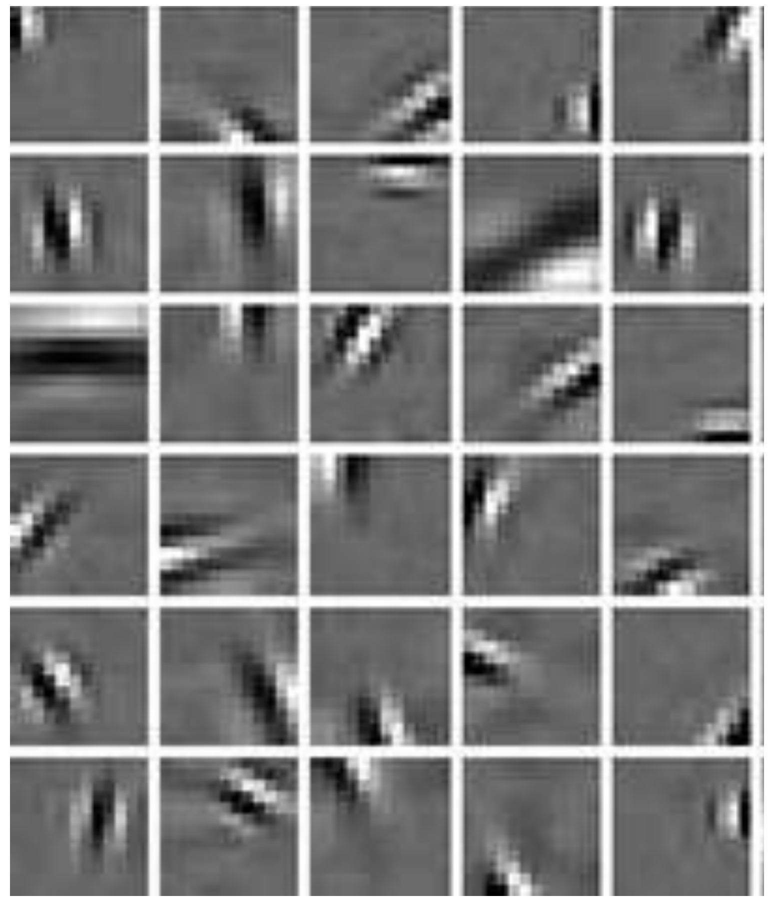
Connection with computer vision

- Convolutional neural networks revolutionized computer vision — beat humans at classifying images since 2015

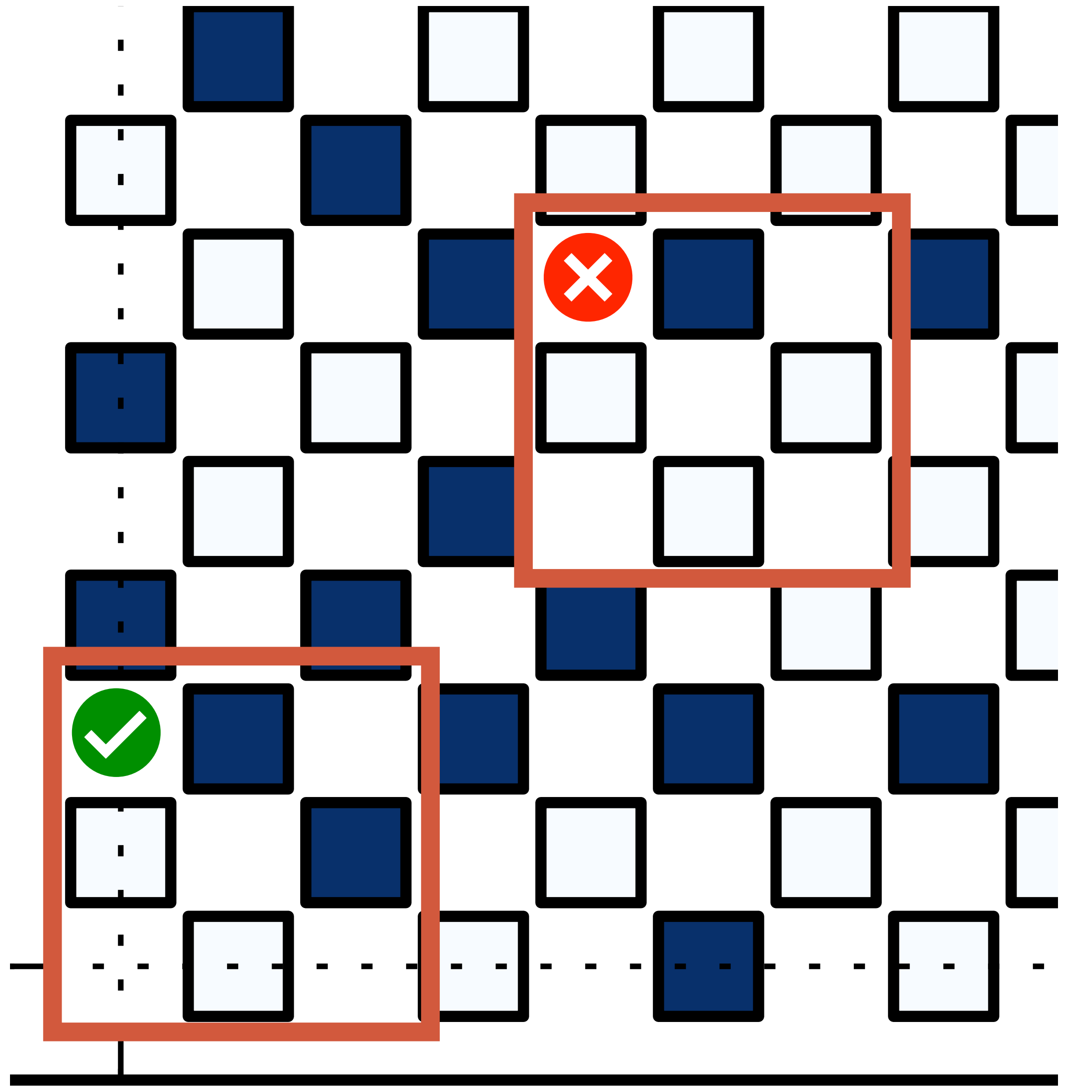


Filters used by the 1st convolutional 1ayer.

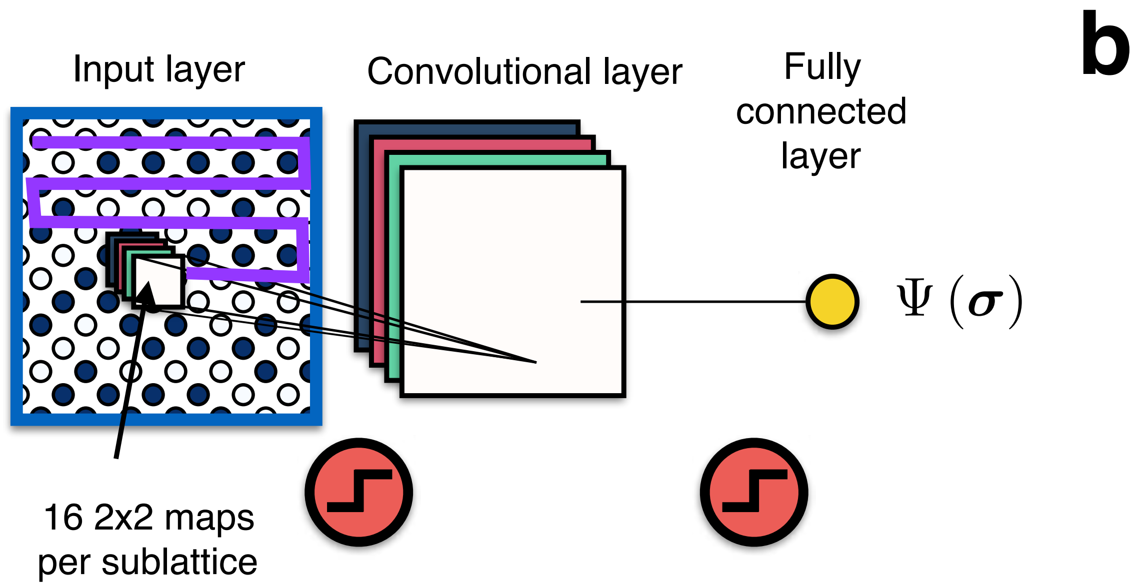
https://cs231n.github.io/understanding-cnn/



Gabor filters, believed to exist in visual processing in the brain (Olshausen & Field, 1997)



In our examples, they see magnetic monopoles or in spin ice, they see the ice rules



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

Analytical model for the Ising gauge theory

Convolutional layer

f	s=A	s=B	f	s=A	s=B
1	$\begin{pmatrix}1 & 0\\ 1 & 0\end{pmatrix}$	$\begin{pmatrix}1 & 1\\ 0 & 0\end{pmatrix}$	9	$\begin{pmatrix}1 & 0\\ 1 & 0\end{pmatrix}$	$\begin{pmatrix}1 & -1\\ 0 & 0\end{pmatrix}$
2	$\begin{pmatrix}-1 & 0\\ -1 & 0\end{pmatrix}$	$\begin{pmatrix}-1 & -1\\ 0 & 0\end{pmatrix}$	10	$\begin{pmatrix}1 & 0\\ 1 & 0\end{pmatrix}$	$\begin{pmatrix}-1 & 1\\ 0 & 0\end{pmatrix}$
3	$\begin{pmatrix}1 & 0\\ 1 & 0\end{pmatrix}$	$\begin{pmatrix}-1 & -1\\ 0 & 0\end{pmatrix}$	11	$\begin{pmatrix}1 & 0\\ -1 & 0\end{pmatrix}$	$\begin{pmatrix}1 & 1\\ 0 & 0\end{pmatrix}$
4	$\begin{pmatrix}1 & 0\\ -1 & 0\end{pmatrix}$	$\begin{pmatrix}-1 & 1\\ 0 & 0\end{pmatrix}$	12	$\begin{pmatrix}-1 & 0\\ 1 & 0\end{pmatrix}$	$\begin{pmatrix}1 & 1\\ 0 & 0\end{pmatrix}$
5	$\begin{pmatrix}-1 & 0\\ -1 & 0\end{pmatrix}$	$\begin{pmatrix}1 & 1\\ 0 & 0\end{pmatrix}$	13	$\begin{pmatrix}-1 & 0\\ -1 & 0\end{pmatrix}$	$\begin{pmatrix}-1 & 1\\ 0 & 0\end{pmatrix}$
6	$\begin{pmatrix}-1 & 0\\ 1 & 0\end{pmatrix}$	$\begin{pmatrix}1 & -1\\ 0 & 0\end{pmatrix}$	14	$\begin{pmatrix}-1 & 0\\ -1 & 0\end{pmatrix}$	$\begin{pmatrix}1 & -1\\ 0 & 0\end{pmatrix}$
7	$\begin{pmatrix}1 & 0\\ -1 & 0\end{pmatrix}$	$\begin{pmatrix}1 & -1\\ 0 & 0\end{pmatrix}$	15	$\begin{pmatrix}-1 & 0\\ 1 & 0\end{pmatrix}$	$\begin{pmatrix}-1 & -1\\ 0 & 0\end{pmatrix}$
8	$\begin{pmatrix}-1 & 0\\ 1 & 0\end{pmatrix}$	$\begin{pmatrix}-1 & 1\\ 0 & 0\end{pmatrix}$	16	$\begin{pmatrix}-1 & 0\\ -1 & 0\end{pmatrix}$	$\begin{pmatrix}-1 & -1\\ 0 & 0\end{pmatrix}$

$$b_c = -(2+\epsilon)\begin{bmatrix}1\\ \vdots\\ 1\end{bmatrix}$$

fully-connected layer

$$W_{\mathrm{o}} = \begin{pmatrix} \overbrace{1 \;\ldots\; 1}^{8L^2 \text{ terms}} & \overbrace{-L^2 \ldots -L^2}^{8L^2 \text{ terms}} \\ -1 \;\ldots\; -1 & L^2 \ldots L^2 \end{pmatrix}, \quad \text{and } b_{\mathrm{o}} = \begin{pmatrix}0\\ 0\end{pmatrix}$$

The purpose of the filters is to individually process each plaquette in the spin configuration and determine whether its energetic constraints are satisfied or not. Basically the Conv. layer encodes the Hamiltonian

https://github.com/carrasqu/IsingGaugeTensorFlow

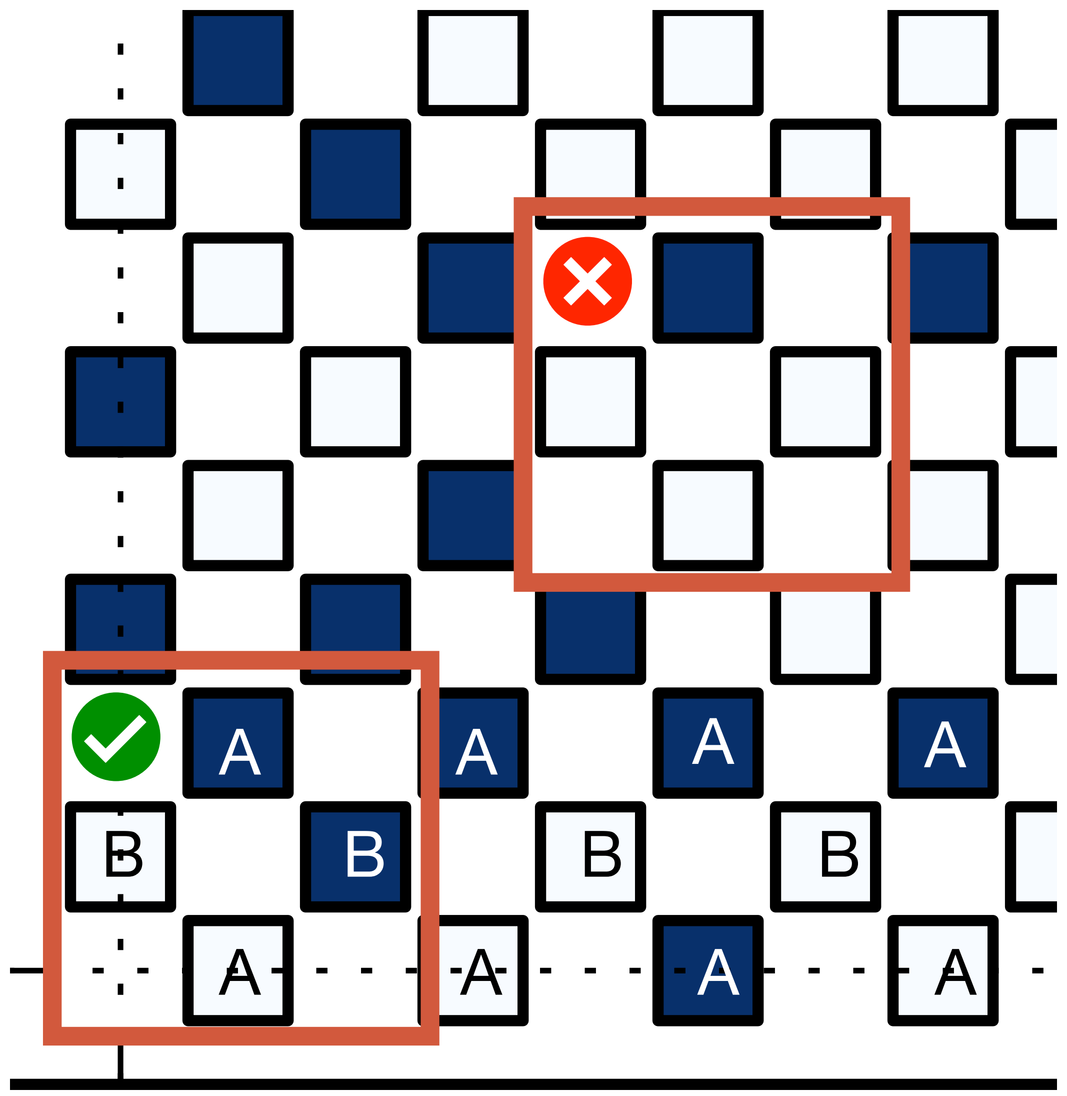
Analytical model for the Ising gauge theory

Convolutional layer

f	s=A	s=B	f	s=A	s=B
1	$\begin{pmatrix}1&0\\1&0\end{pmatrix}$	$\begin{pmatrix}1&1\\0&0\end{pmatrix}$	9	$\begin{pmatrix}1&0\\1&0\end{pmatrix}$	$\begin{pmatrix}1&-1\\0&0\end{pmatrix}$
2	$\begin{pmatrix}-1&0\\-1&0\end{pmatrix}$	$\begin{pmatrix}-1&-1\\0&0\end{pmatrix}$	10	$\begin{pmatrix}1&0\\1&0\end{pmatrix}$	$\begin{pmatrix}-1&1\\0&0\end{pmatrix}$
3	$\begin{pmatrix}1&0\\1&0\end{pmatrix}$	$\begin{pmatrix}-1&-1\\0&0\end{pmatrix}$	11	$\begin{pmatrix}1&0\\-1&0\end{pmatrix}$	$\begin{pmatrix}1&1\\0&0\end{pmatrix}$
4	$\begin{pmatrix}1&0\\-1&0\end{pmatrix}$	$\begin{pmatrix}-1&1\\0&0\end{pmatrix}$	12	$\begin{pmatrix}-1&0\\1&0\end{pmatrix}$	$\begin{pmatrix}1&1\\0&0\end{pmatrix}$
5	$\begin{pmatrix}-1&0\\-1&0\end{pmatrix}$	$\begin{pmatrix}1&1\\0&0\end{pmatrix}$	13	$\begin{pmatrix}-1&0\\-1&0\end{pmatrix}$	$\begin{pmatrix}-1&1\\0&0\end{pmatrix}$
6	$\begin{pmatrix}-1&0\\1&0\end{pmatrix}$	$\begin{pmatrix}1&-1\\0&0\end{pmatrix}$	14	$\begin{pmatrix}-1&0\\-1&0\end{pmatrix}$	$\begin{pmatrix}1&-1\\0&0\end{pmatrix}$
7	$\begin{pmatrix}1&0\\-1&0\end{pmatrix}$	$\begin{pmatrix}1&-1\\0&0\end{pmatrix}$	15	$\begin{pmatrix}-1&0\\1&0\end{pmatrix}$	$\begin{pmatrix}-1&-1\\0&0\end{pmatrix}$
8	$\begin{pmatrix}-1&0\\1&0\end{pmatrix}$	$\begin{pmatrix}-1&1\\0&0\end{pmatrix}$	16	$\begin{pmatrix}-1&0\\-1&0\end{pmatrix}$	$\begin{pmatrix}-1&-1\\0&0\end{pmatrix}$

$$b_c = -(2+\epsilon)\begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix}$$

fully-connected layer



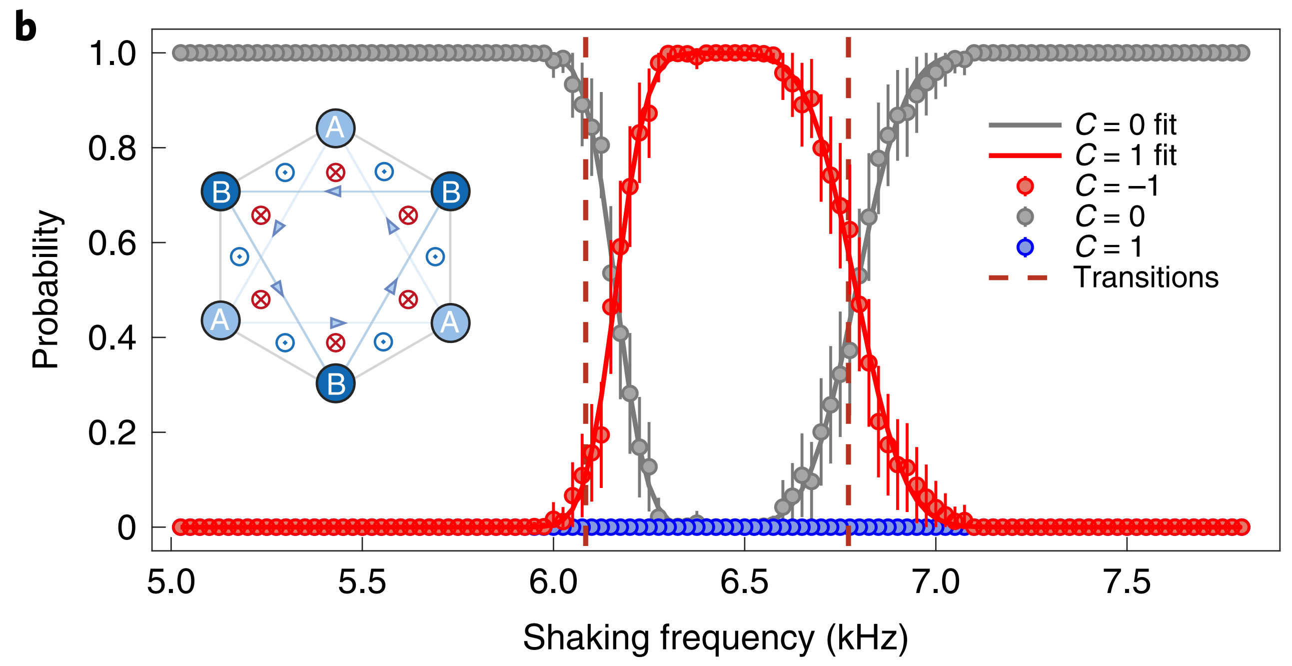
The purpose of the filters is to individually process each plaquette in the spin configuration and determine whether its energetic constraints are satisfied or not. Basically the Conv. layer encodes the Hamiltonian

Analysis of experimental data in quantum systems

nature physics
LETTERS
https://doi.org/10.1038/s41567-019-0554-0

Identifying quantum phase transitions using artificial neural networks on experimental data

Benno S. Rem[1,2], Niklas Käming[1], Matthias Tarnowski[1,2], Luca Asteria[1], Nick Fläschner[1], Christoph Becker[1,3], Klaus Sengstock[1,2,3]* and Christof Weitenberg[1,2]



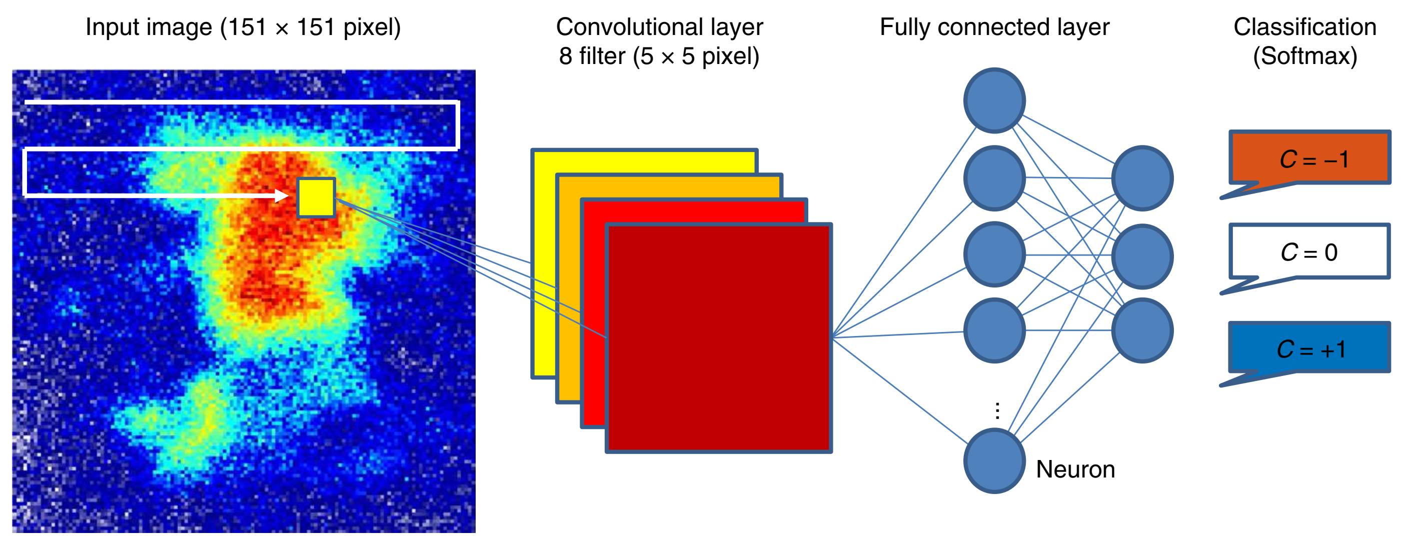


Fig. 1 | Using a neural network to identify physical phases from experimental images. Single images of the density of atoms in momentum space after time-of-flight (false-colour representation of a single-channel image) serve as input for a deep convolutional neural network with a variety of layers including convolutional filters and fully connected layers. The white line represents the sliding of the filters across the input image. The final softmax layer outputs the probability that the image belongs to one of the classes (here, Chern numbers $C = -1$, 0 or 1). The weights of the network are trained on many labelled images and the network can then classify an unknown single image with high confidence. This approach—originally developed for image recognition—works well for identifying physical quantum phases from experimental images.

“Our results point the way to unravel complex phase diagrams of experimental systems, where the Hamiltonian and the order parameters might not be known”

Rem et al. Nature Physics 15, 917 (2019)

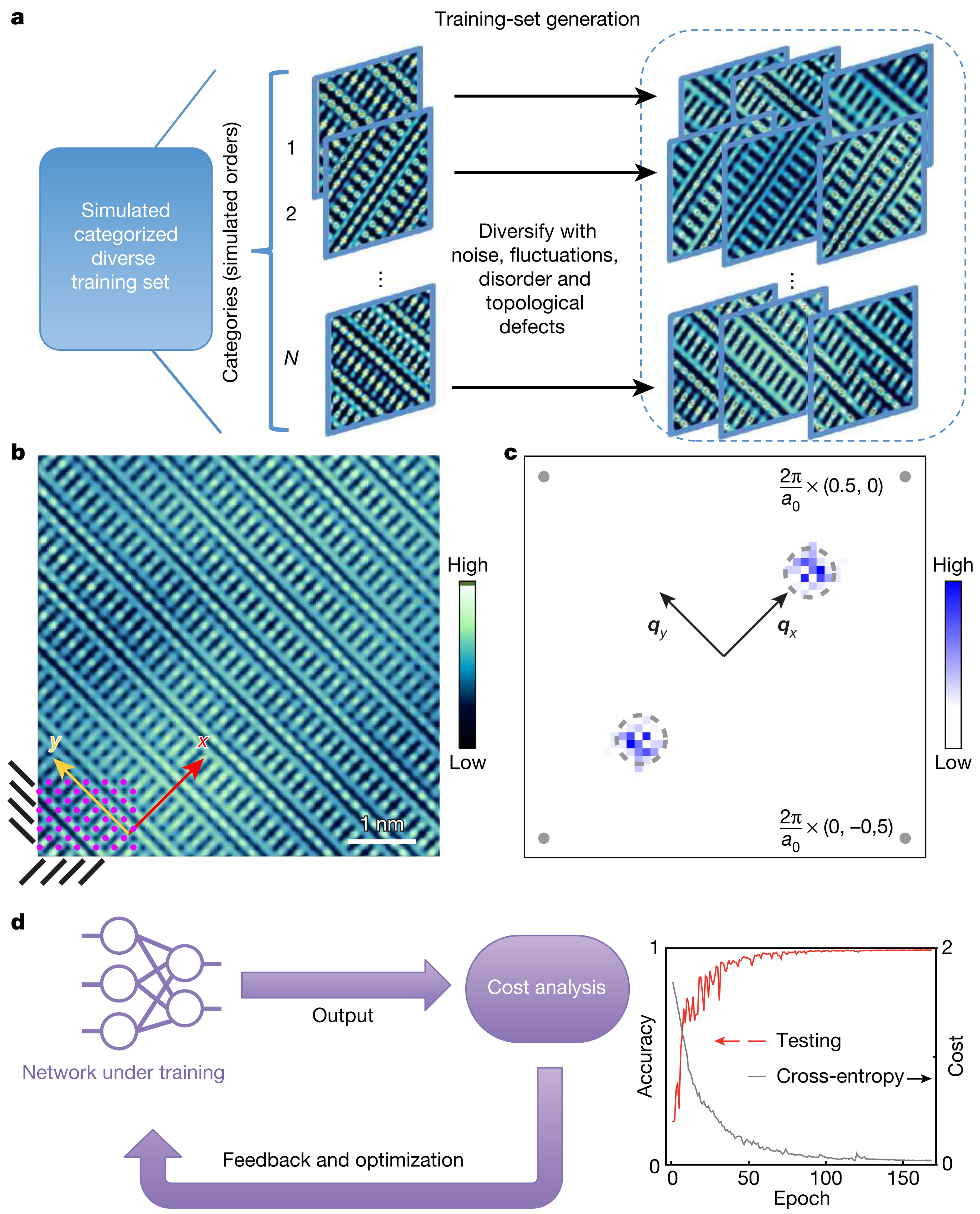
Experimental condensed matter physics

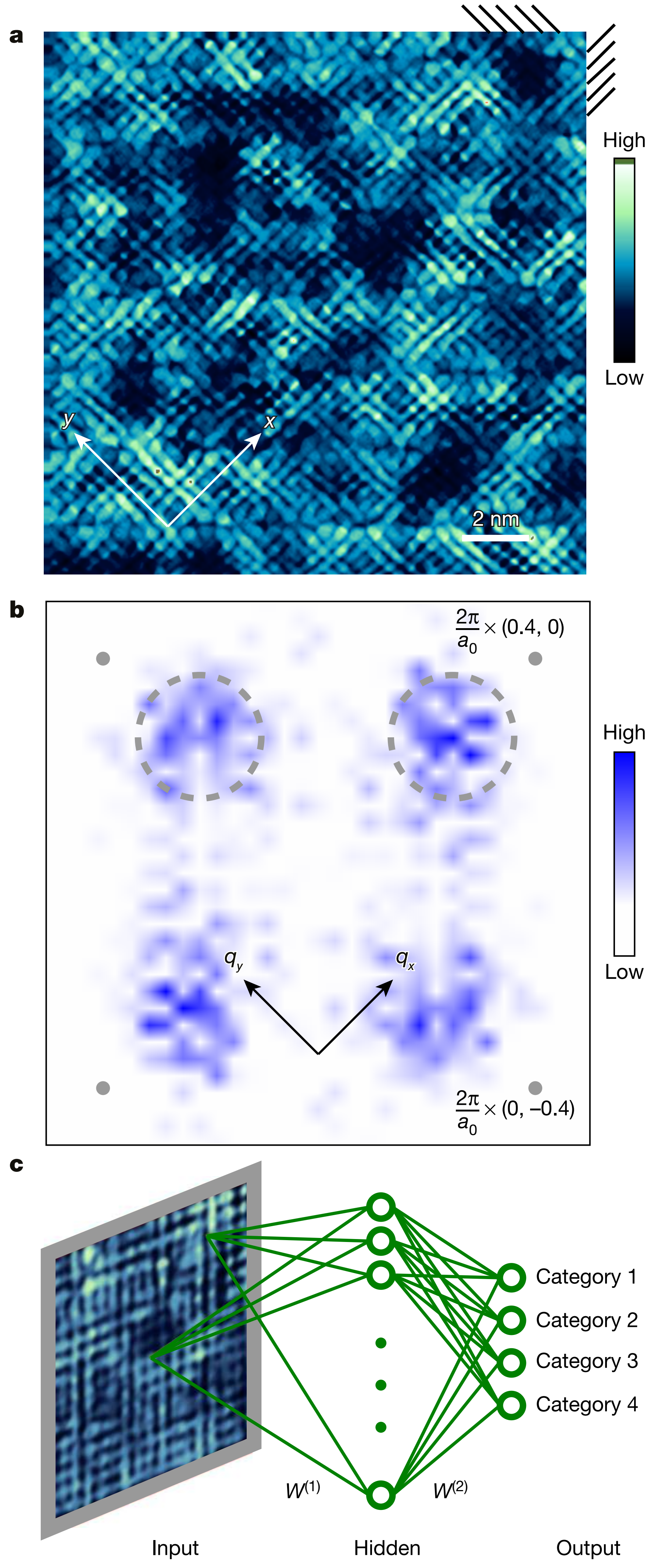
LETTER

https://doi.org/10.1038/s41586-019-1319-8

Machine learning in electronic-quantum-matter imaging experiments

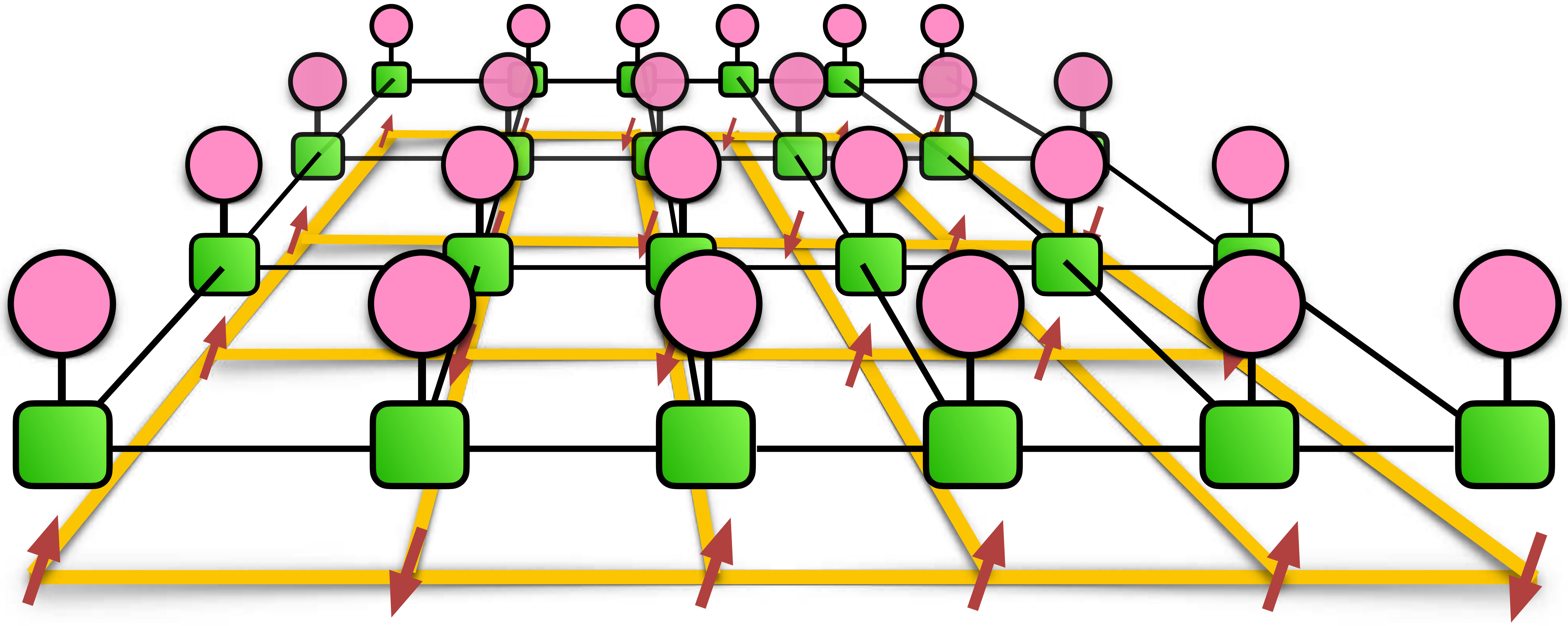
Yi Zhang[1,11], A. Mesaros[1,2,11], K. Fujita[3], S. D. Edkins[1,4], M. H. Hamidian[1,5], K. Ch'ng[6], H. Eisaki[7], S. Uchida[7,8], J. C. Séamus Davis[1,3,9,10], Ehsan Khatami[6] & Eun-Ah Kim[1]*





Zhang et al. Nature 570, 484 (2019)

Neural networks as quantum states



Neural networks as quantum states

$$\Psi_\theta$$

Hamiltonian driven learning —

Finding ground states. This is just one example, but there are many more tasks that are driven by a “physical” principle.

Data driven learning —

quantum state tomography, approximate reconstruction of quantum devices, quantum simulations and Quantum channels

Closest in spirit to ML tasks

Neural networks as quantum states

$$\Psi_\theta$$

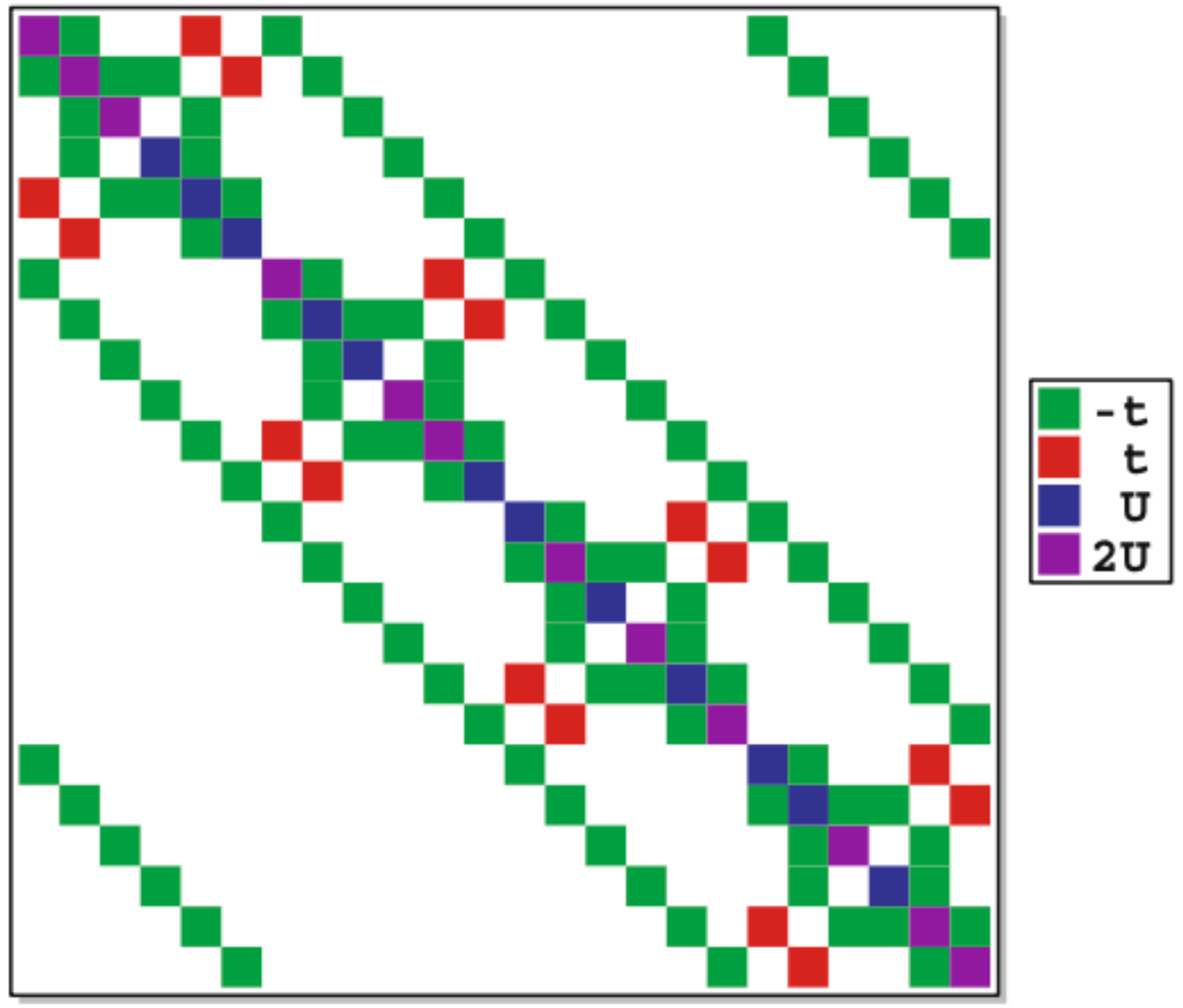
Hamiltonian driven learning —

Finding ground states. This is just one example, but there are many more tasks that are driven by a "physical" principle.

Solving for the low-energy, time-independent solutions of the Schrodinger equation.

Ground states

- Given a problem Hamiltonian H, approximate its ground state.
- H is a very large Hermitian matrix that describes the behaviour of a physical system at the microscopic scale.
- Finding the ground state — the Hamiltonian's lowest energy eigenvector and eigenvalues.
- Typically hard because of exponential complexity.
- Solve the problem **analytically** or propose an Ansatz inspired by machine learning techniques (**neural network**)
- In modern variational quantum algorithms — variational quantum eigensolver (VQE) algorithm.

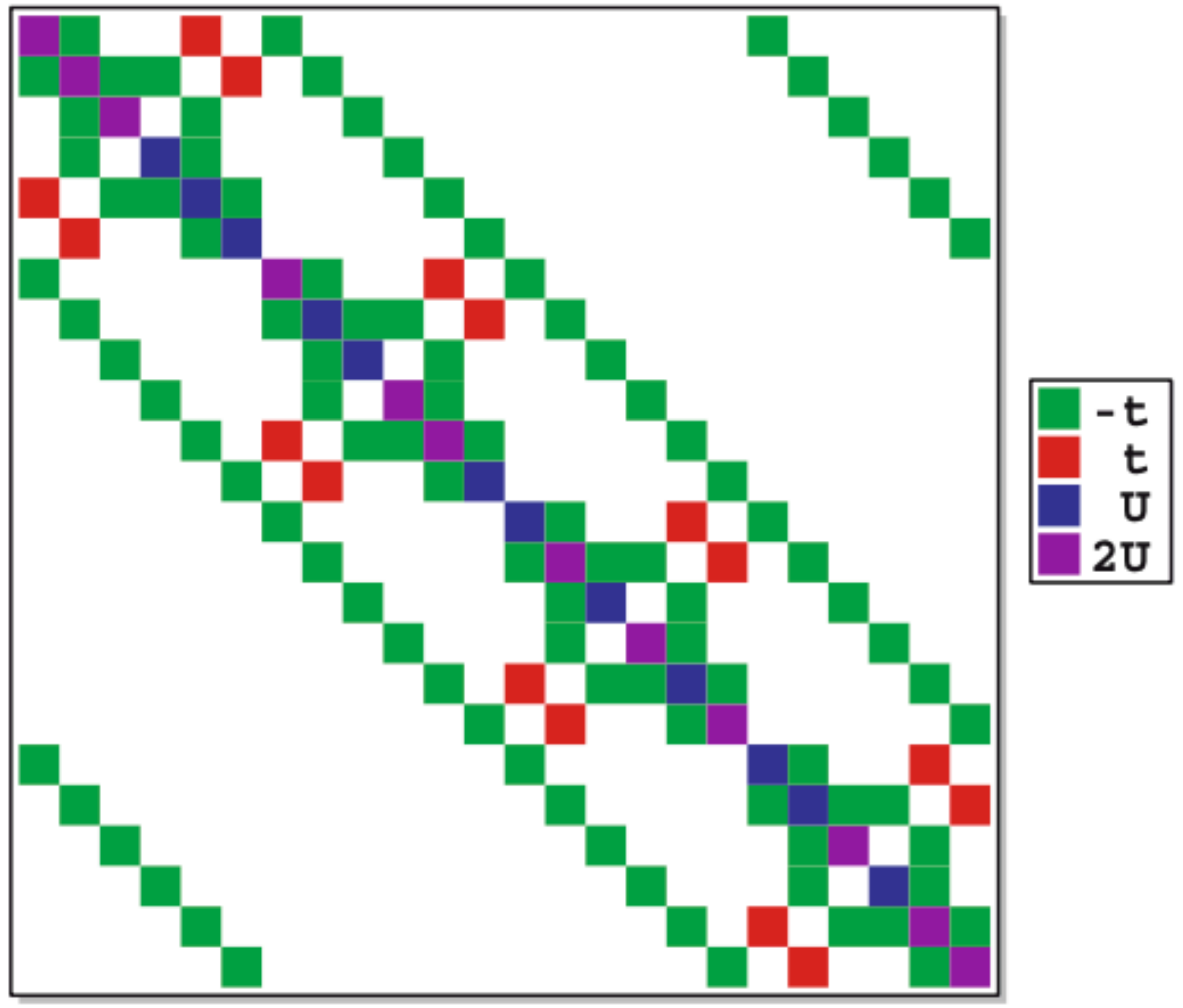


Schematic representation of the Hamiltonian matrix of the Hubbard model with L = 4, $N\uparrow = 3$, $N\downarrow = 2$ (5 particles)

Exact Diagonalization Techniques.
Alexander WeißeHolger Fehske.
Part of the Lecture Notes in Physics book series (LNP, volume 739)

Ground states

- Hamiltonians are very large— problem is computationally difficult.
- Structured and sparse.
- Symmetries (some of which are common to important symmetries and inductive biases in ML).
- The rows and columns indices are usually related to real space configurations of the particles in the system and are usually ordered using bit-strings representations (001010110).



Schematic representation of the Hamiltonian matrix of the Hubbard model with L = 4, N↑ = 3, N↓ = 2 (5 particles)

Approximating ground states with neural networks

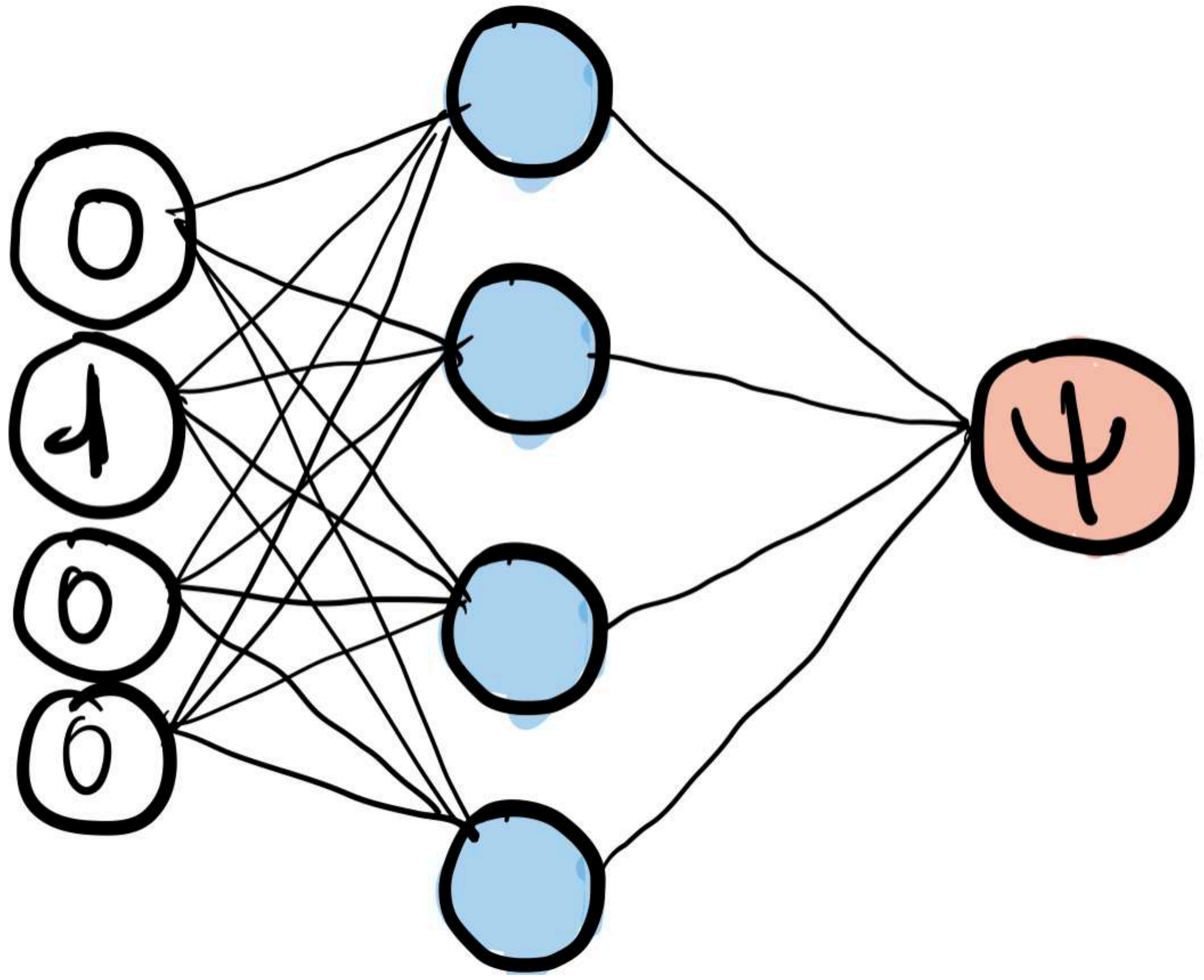
- Recall that we represent a quantum state as a 2^N-dimensional vector of complex entries
- $$|\psi\rangle = \begin{bmatrix} \psi_{0,0,0,\ldots 0,0} \\ \psi_{1,0,0,\ldots 0,0} \\ \psi_{1,1,0,\ldots 0,0} \\ \vdots \\ \psi_{1,1,1,\ldots 1,1} \end{bmatrix}$$ What does it mean that we represent a quantum state as a neural network?

$$\Downarrow$$

$$|\psi_\theta\rangle = \begin{bmatrix} \psi_\theta(0,0,0,\ldots 0,0) \\ \psi_\theta(1,0,0,\ldots 0,0) \\ \psi_\theta(1,1,0,\ldots 0,0) \\ \vdots \\ \psi_\theta(1,1,1,\ldots 1,1) \end{bmatrix}$$

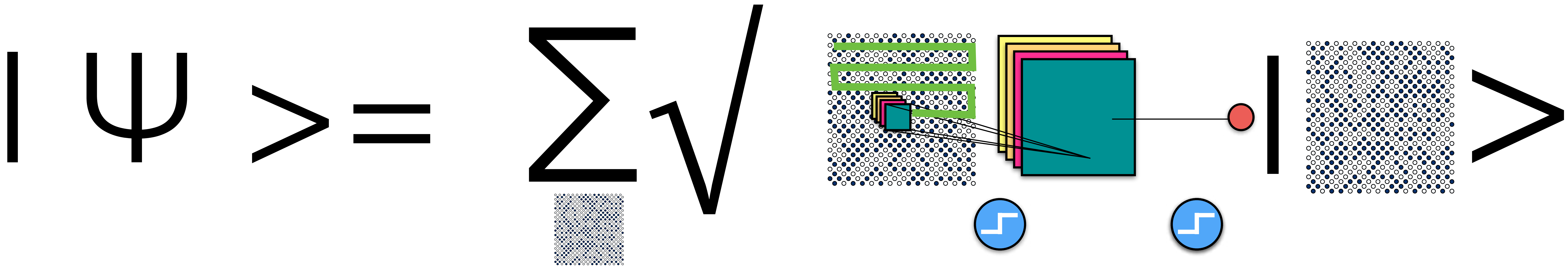
Where the complex-valued boolean function

$\psi_\theta(x_1, x_2, \ldots, x_N) =$ **Neural network**$(x_1, x_2, \ldots, x_N)$

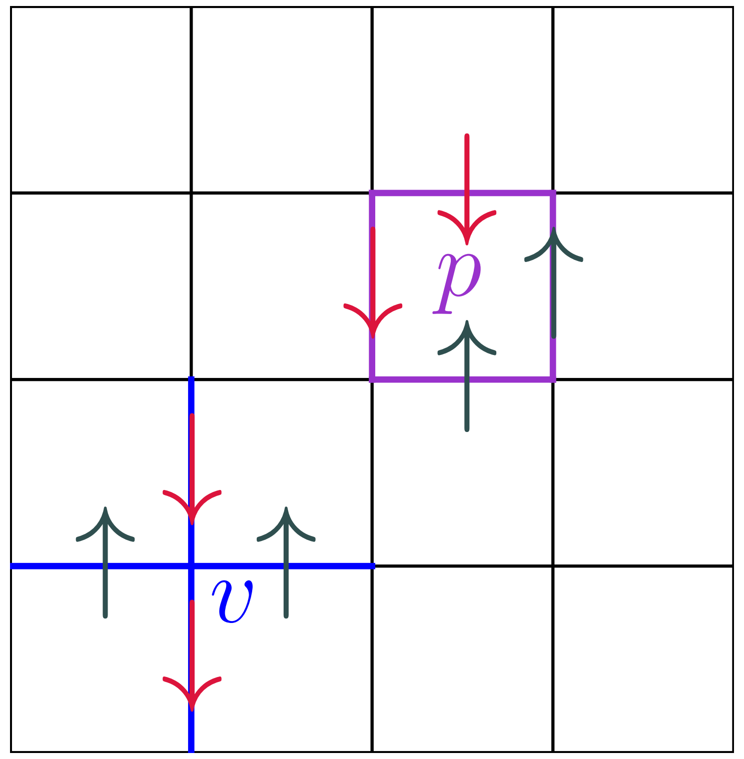


As a consequence, we go from an exponential amount of parameters to a neural network with a few parameters at the cost of constraining the type of functions we can represent.

Cold neuron in our model: Ground state of Kitaev's toric code with convolutional neural networks



Ground state of the toric code

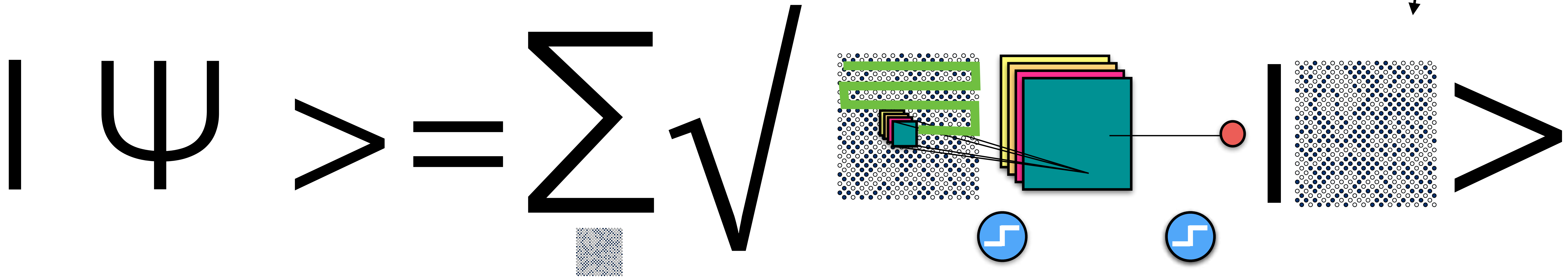


$$H = -J_p \sum_p \prod_{i\in p} \sigma_i^z - J_v \sum_v \prod_{i\in v} \sigma_i^x$$

$$|\Psi_{\mathrm{TC}}\rangle \propto \lim_{\beta\to\infty} \sum_{\sigma_1,\ldots,\sigma_N} e^{\frac{\beta}{2} J \sum_p \prod_{i\in p} \sigma_i^z} |\sigma_1, \ldots, \sigma_N\rangle$$

PEPS : F. Verstraete, M. M. Wolf, D. Perez-Garcia, J. I. Cirac Phys. Rev. Lett. 96, 220601 (2006).

$$O_{\mathrm{cold}}(\sigma_1, \ldots, \sigma_N) \propto \lim_{\beta\to\infty} \exp \beta J \sum_p \prod_{i\in p} \sigma_i^z$$



J. Carrasquilla and R. G. Melko. Nature Physics 13, 431–434 (2017)

Dong-Ling Deng et al Phys. Rev. X 7, 021021 (2017)

Jing Chen, Song Cheng, Haidong Xie, Lei Wang, Tao Xiang arXiv:1701.04831 RBMs

Neural network quantum states

ELSEVIER

Computer Physics Communications 104 (1997) 1–14

Computer Physics Communications

Artificial neural network methods in quantum mechanics

I.E. Lagaris[1], A. Likas, D.I. Fotiadis

Department of Computer Science, University of Ioannina, P.O. Box 1186, GR 45110 Ioannina, Greece

Received 17 March 1997; revised 22 April 1997

3.5. Two-dimensional Schrödinger equation

We consider here the well-studied [2] example of the Henon–Heiles potential. The Hamiltonian is written as

$$H = -\frac{1}{2}\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) + V(x,y),$$

with $V(x,y) = \frac{1}{2}(x^2+y^2) + \frac{1}{4\sqrt{5}}(xy^2 - \frac{1}{3}x^3)$.

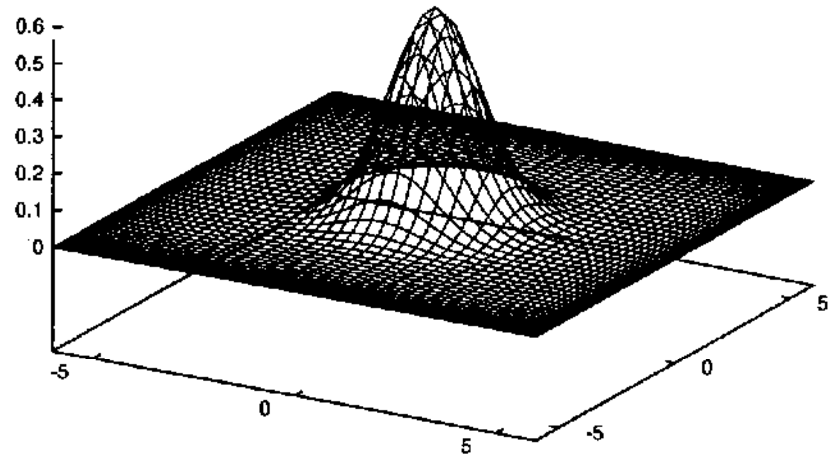


Fig. 4.

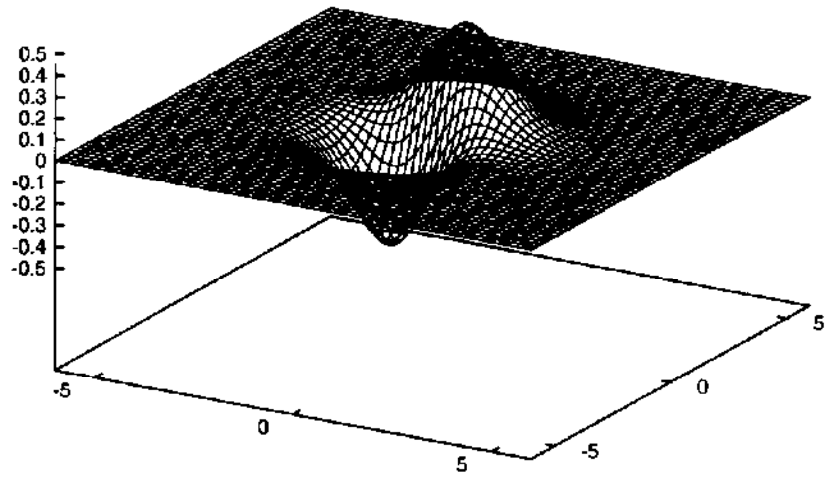
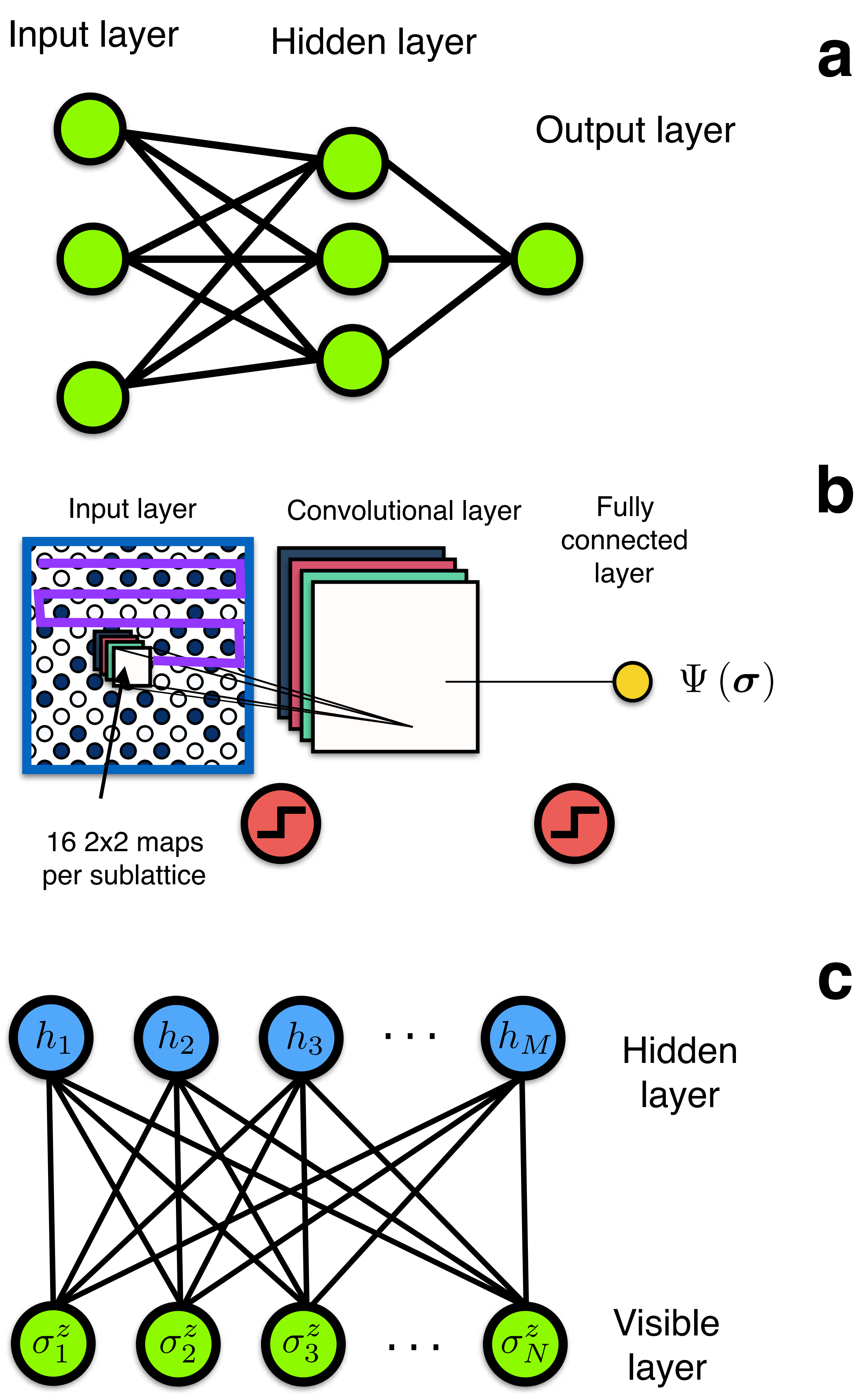


Fig. 5.

Fig. 4. Ground state of the Henon–Heiles problem (ϵ = 0.99866).



Neural network quantum states

nature physics

LETTERS

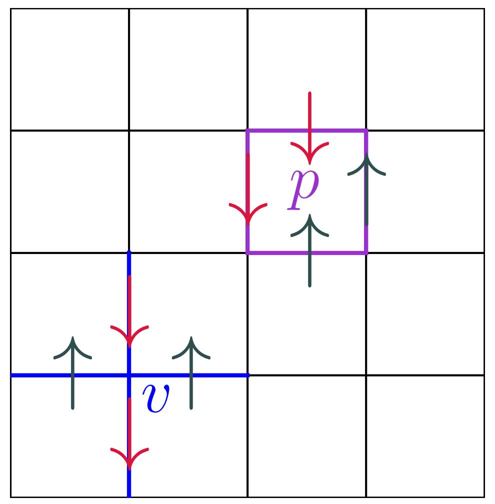
PUBLISHED ONLINE: 13 FEBRUARY 2017 | DOI: 10.1038/NPHYS4035

Machine learning phases of matter

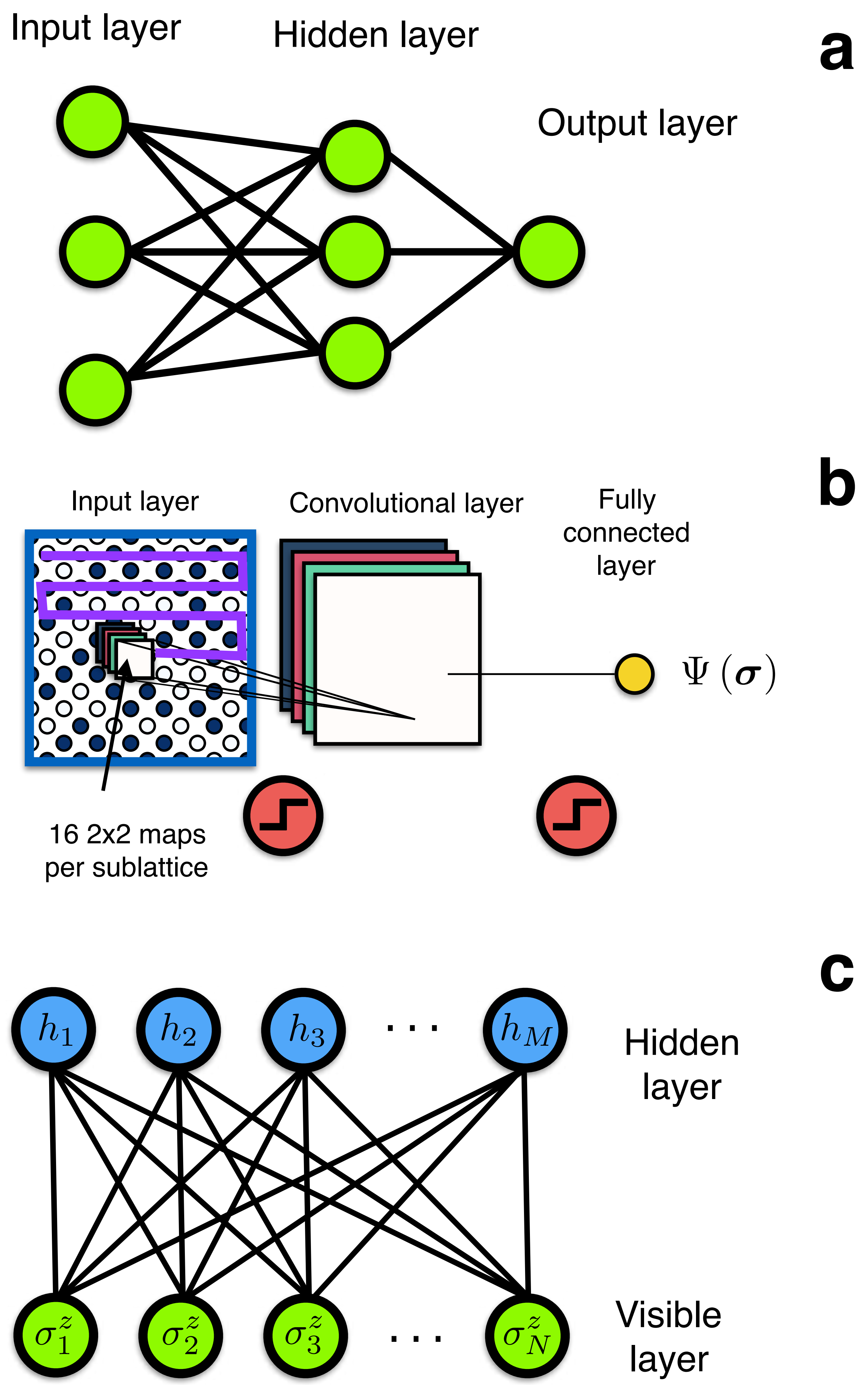
Juan Carrasquilla[1*] and Roger G. Melko[1,2]

KITAEV'S TORIC CODE GROUND STATE

$$H = -J_p \sum_p \prod_{i\in p} \sigma_i^z - J_v \sum_v \prod_{i\in v} \sigma_i^x$$



$$|\Psi_{\mathrm{TC}}\rangle \propto \lim_{\beta\to\infty} \sum_{\sigma_1,\ldots,\sigma_N} e^{\frac{\beta}{2} J \sum_p \prod_{i\in p} \sigma_i^z} |\sigma_1, \ldots, \sigma_N\rangle$$



J. Carrasquilla and R. G. Melko. Nature Physics 13, 431–434 (2017)

Neural network quantum states

RESEARCH

RESEARCH ARTICLE

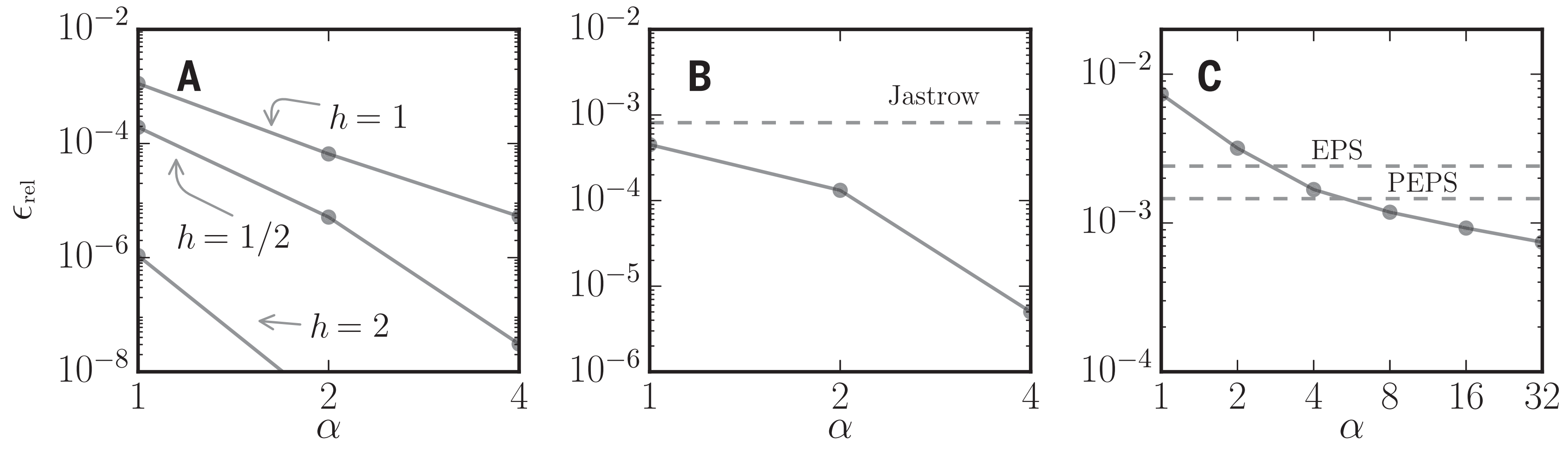
MANY-BODY PHYSICS

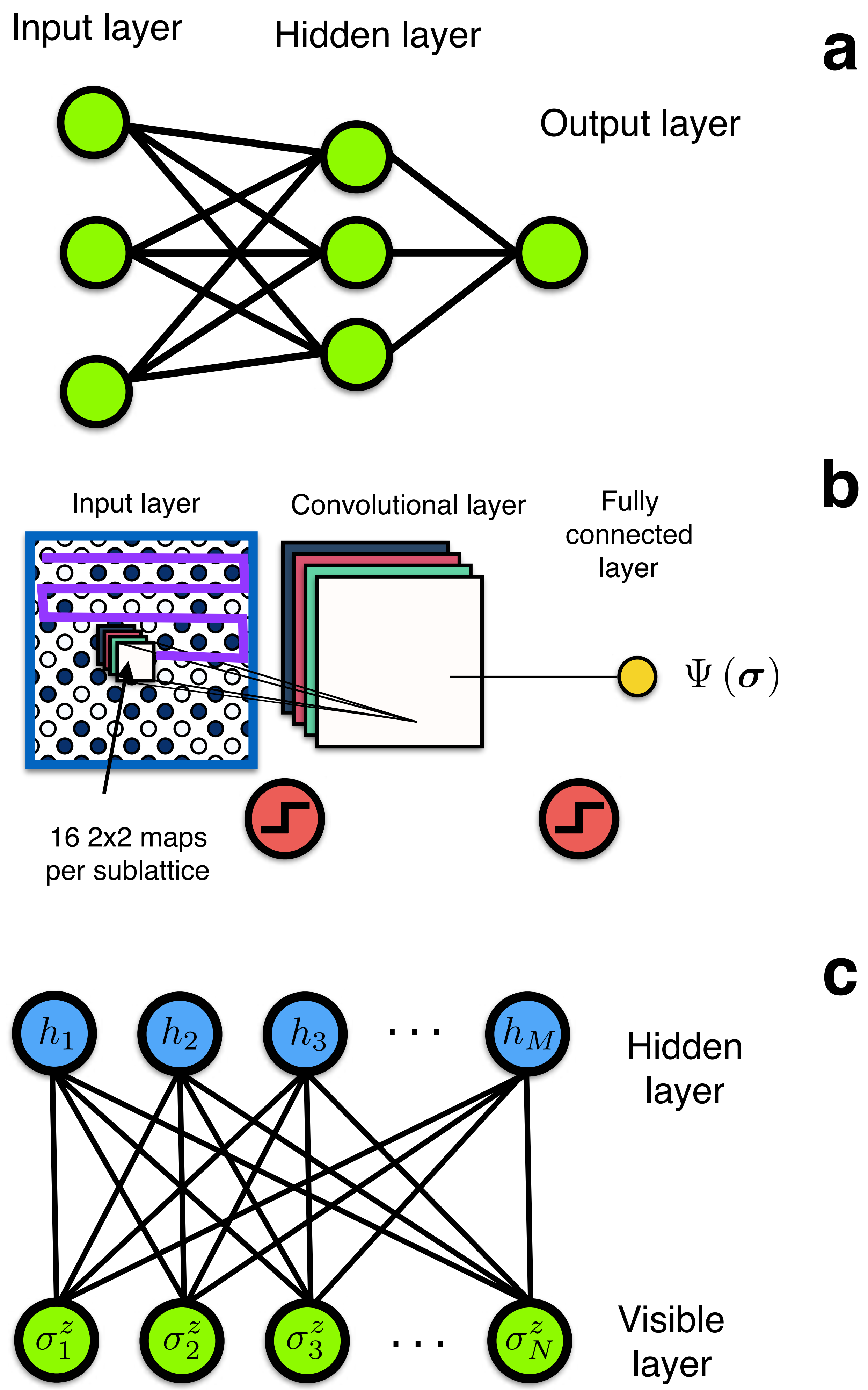
Solving the quantum many-body problem with artificial neural networks

Giuseppe Carleo[1]* and Matthias Troyer[1,2]

The challenge posed by the many-body problem in quantum physics originates from the difficulty of describing the nontrivial correlations encoded in the exponential complexity of the many-body wave function. Here we demonstrate that systematic machine learning of the wave function can reduce this complexity to a tractable computational form for some notable cases of physical interest. We introduce a variational representation of quantum states based on artificial neural networks with a variable number of hidden neurons. A reinforcement-learning scheme we demonstrate is capable of both finding the ground state and describing the unitary time evolution of complex interacting quantum systems. Our approach achieves high accuracy in describing prototypical interacting spins models in one and two dimensions.

$$\mathcal{H}_{\mathrm{TFI}} = -h\sum_i \sigma_i^x - \sum_{ij} \sigma_i^z \sigma_j^z \qquad \mathcal{H}_{\mathrm{AFH}} = \sum_{ij} \sigma_i^x\sigma_j^x + \sigma_i^y\sigma_j^y + \sigma_i^z\sigma_j^z$$





Giuseppe Carleo, Matthias Troyer, Science 355, 602 (2017)

Exploration areas

- Condensed matter physics
- Quantum chemistry
- Materials science
- Atomic physics
- High energy physics and field theory
- Quantum information
- Nuclear physics
- Combinatorial optimization

Juan Carrasquilla (2020) Machine learning for quantum matter, Advances in Physics: X, 5:1, DOI: 10.1080/23746149.2020.1797528

Juan Carrasquilla and Giacomo Torlai. Neural networks in quantum many-body physics: a hands-on tutorial. https://arxiv.org/abs/2101.11099

Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences*. Rev. Mod. Phys. 91, 045002 (2019)

Questions?

Numerical approach based on Variational Monte Carlo

- Ground state search reframed as an optimization problem with an appropriate cost function.
- The variational theorem in quantum physics $E_\theta = \langle \Psi_\theta | H | \Psi_\theta \rangle \geq E_0$ where E_0 is the lowest energy eigenvalue of Hamiltonian matrix H.
- $|\Psi_\theta\rangle$ is a neural network parameterizing the quantum state.
- It is possible to evaluate E_θ and its gradients $\nabla_\theta E_\theta$ via Monte Carlo — use gradient descent techniques
- No data from the exact solution is needed— gradient signal comes from H

F Becca, S Sorella. Quantum Monte Carlo Approaches for Correlated Systems. Cambridge University Press (2017)

Numerical approach based on Variational Monte Carlo

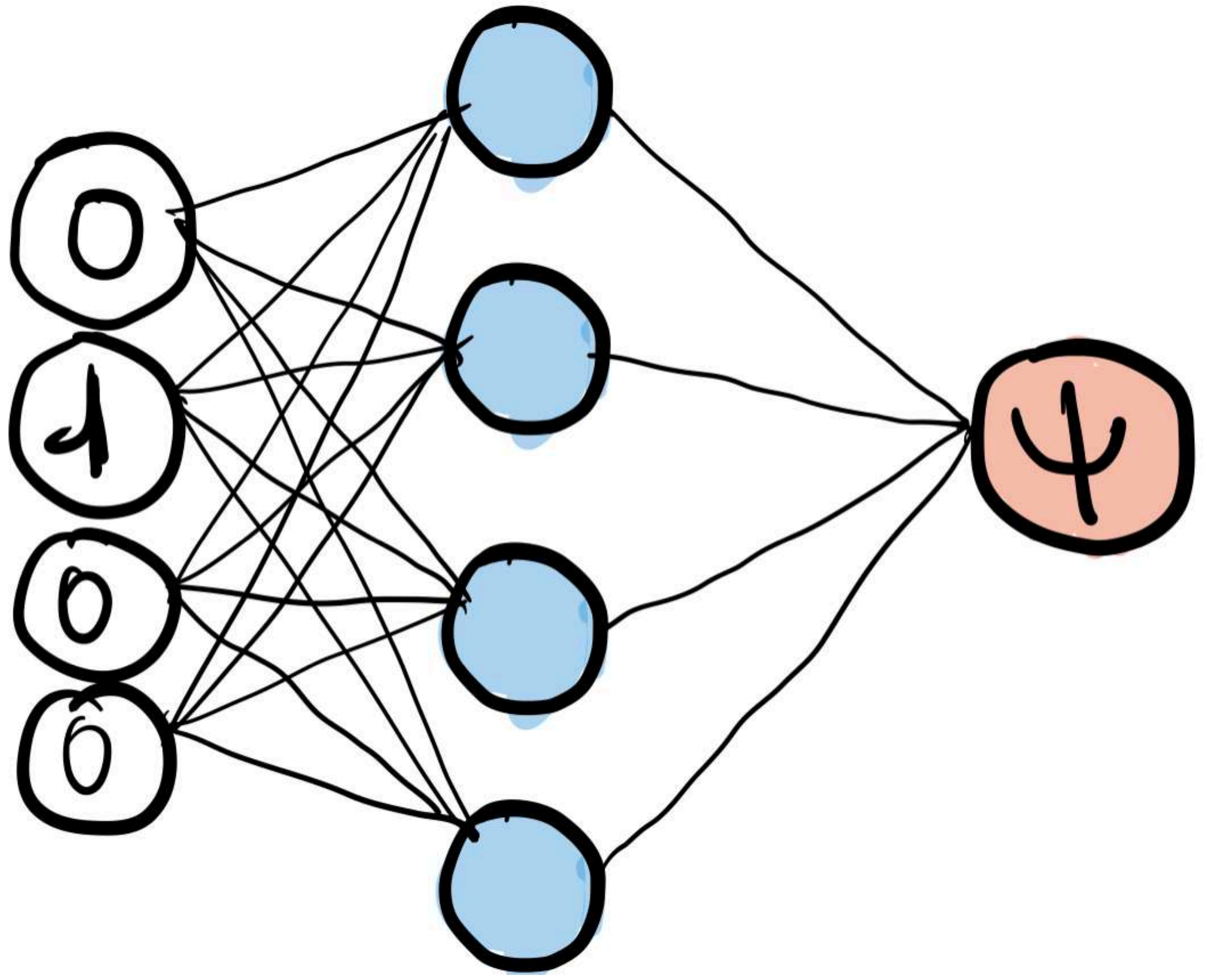
- Ground state search reframed s an optimization problem with an appropriate cost function.

- The variational theorem in quantum physics $E_\theta = \langle \Psi_\theta | H | \Psi_\theta \rangle \geq E_0$ where E_0 is the lowest energy eigenvalue corresponding to the ground matrix H.

- $|\Psi_\theta\rangle$ is a neural network parameterizing the quantum state

- It is possible to evaluate E_θ and its gradients $\nabla_\theta E_\theta$ via Mo descent techniques

- No data from the exact solution is needed— gradient signa

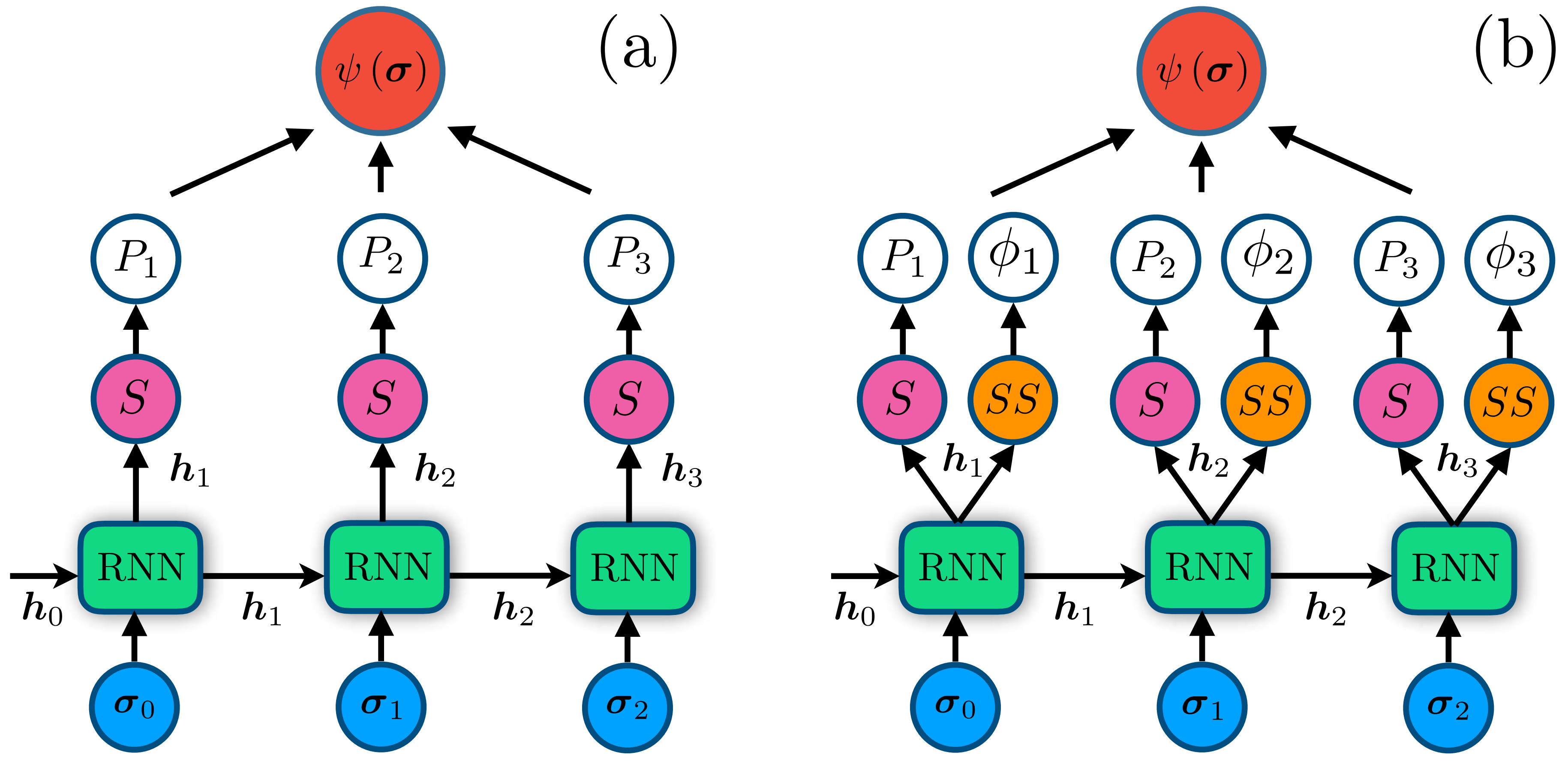


F Becca, S Sorella. Quantum Monte Carlo Approaches for Correlated Systems. Cambridge University Press (2017)

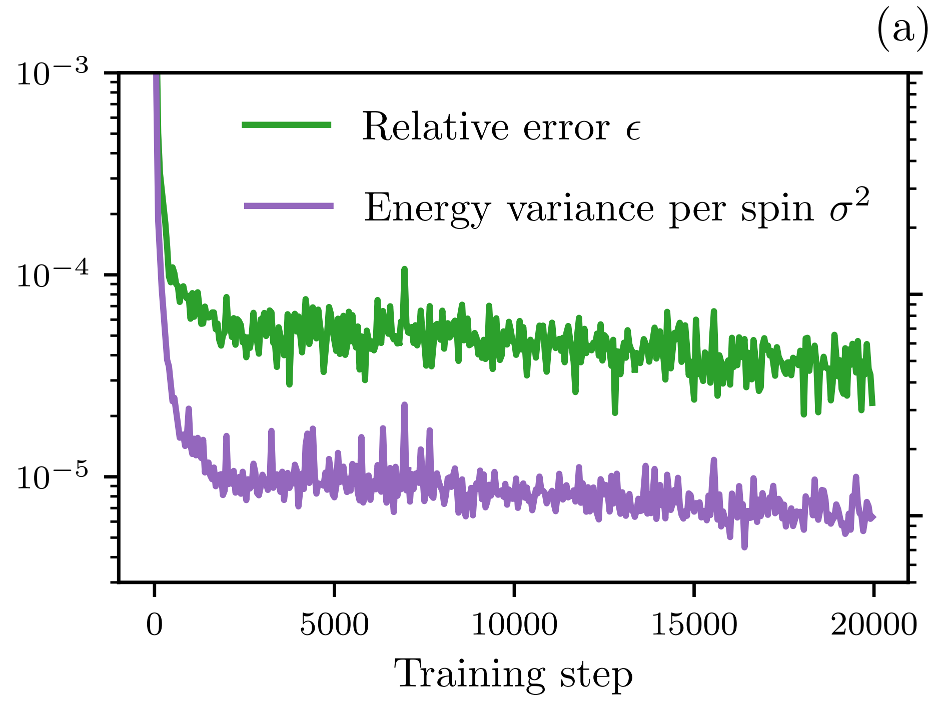
Training the models

- Both $E_{\boldsymbol{\theta}}$ and its gradients available through sampling.
- $E = \langle \Psi_\theta | \hat{H} | \Psi_\theta \rangle = \sum_{\boldsymbol{\sigma}} |\psi_\theta(\boldsymbol{\sigma})|^2 \sum_{\boldsymbol{\sigma}'} H_{\boldsymbol{\sigma}\boldsymbol{\sigma}'} \frac{\psi_\theta(\boldsymbol{\sigma}')}{\psi_\theta(\boldsymbol{\sigma})}$ We interpret $|\psi_\theta(\boldsymbol{\sigma})|^2$ as a probability distribution
- $\equiv \sum_{\boldsymbol{\sigma}} |\psi_\theta(\boldsymbol{\sigma})|^2 E_{loc}(\boldsymbol{\sigma}) \approx \frac{1}{N_S} \sum_{\boldsymbol{\sigma} \sim |\psi_\theta(\boldsymbol{\sigma})|^2} E_{loc}(\boldsymbol{\sigma})$ Evaluate via Monte Carlo average
- Gradients
- $\partial_{\theta_j} E = \sum_{\boldsymbol{\sigma}} |\psi_\theta(\boldsymbol{\sigma})|^2 \frac{\partial_{\theta_j} \psi_\theta^*(\boldsymbol{\sigma})}{\psi_\theta^*(\boldsymbol{\sigma})} E_{loc}(\boldsymbol{\sigma}) + \text{c.c}$
- $\partial_{\theta_j} E \approx \frac{2}{N_S} \mathfrak{Re} \left(\sum_{i=1}^{N_S} \frac{\partial_{\theta_j} \psi_\theta^*(\boldsymbol{\sigma}^{(i)})}{\psi_\theta^*(\boldsymbol{\sigma}^{(i)})} E_{loc}(\boldsymbol{\sigma}^{(i)}) \right)$ Evaluate via Monte Carlo average

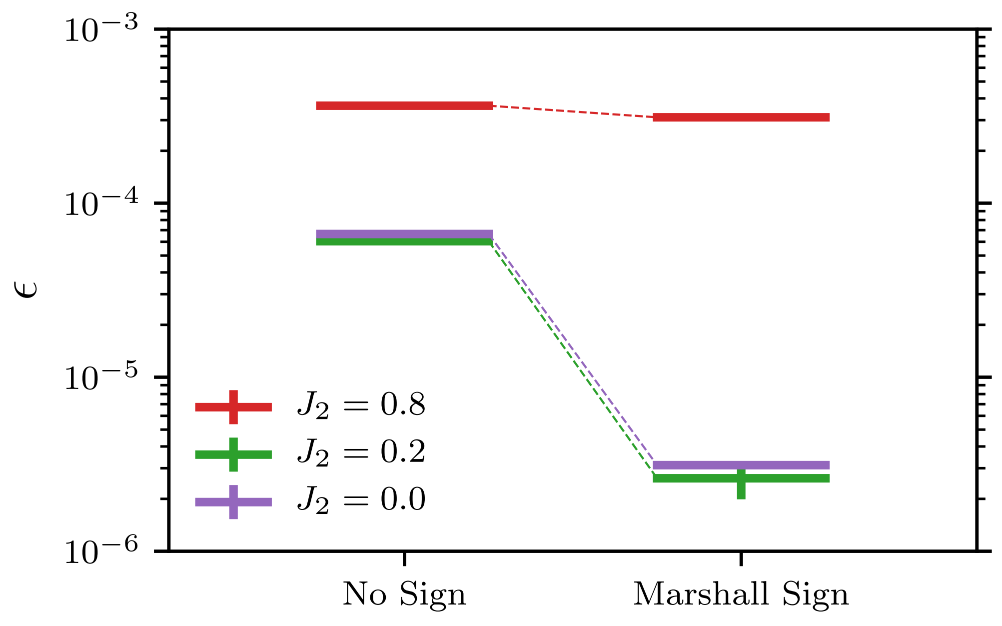
Recurrent neural network wavefunctions



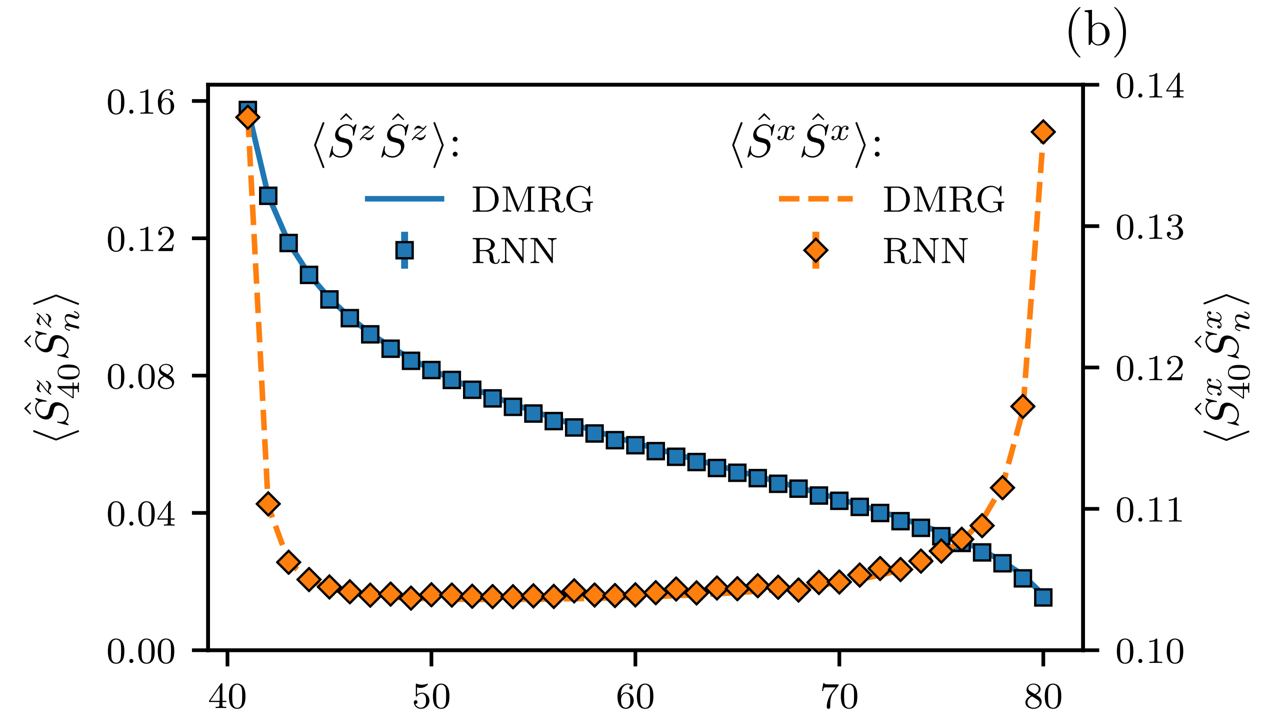
$$|\Psi\rangle = \sum_{\boldsymbol{\sigma}} \psi(\boldsymbol{\sigma})\,|\boldsymbol{\sigma}\rangle = \sum_{\boldsymbol{\sigma}} \sqrt{P(\boldsymbol{\sigma})}\,|\boldsymbol{\sigma}\rangle \qquad |\Psi\rangle = \sum_{\boldsymbol{\sigma}} \exp(\mathrm{i}\phi(\boldsymbol{\sigma}))\sqrt{P(\boldsymbol{\sigma})}\,|\boldsymbol{\sigma}\rangle$$



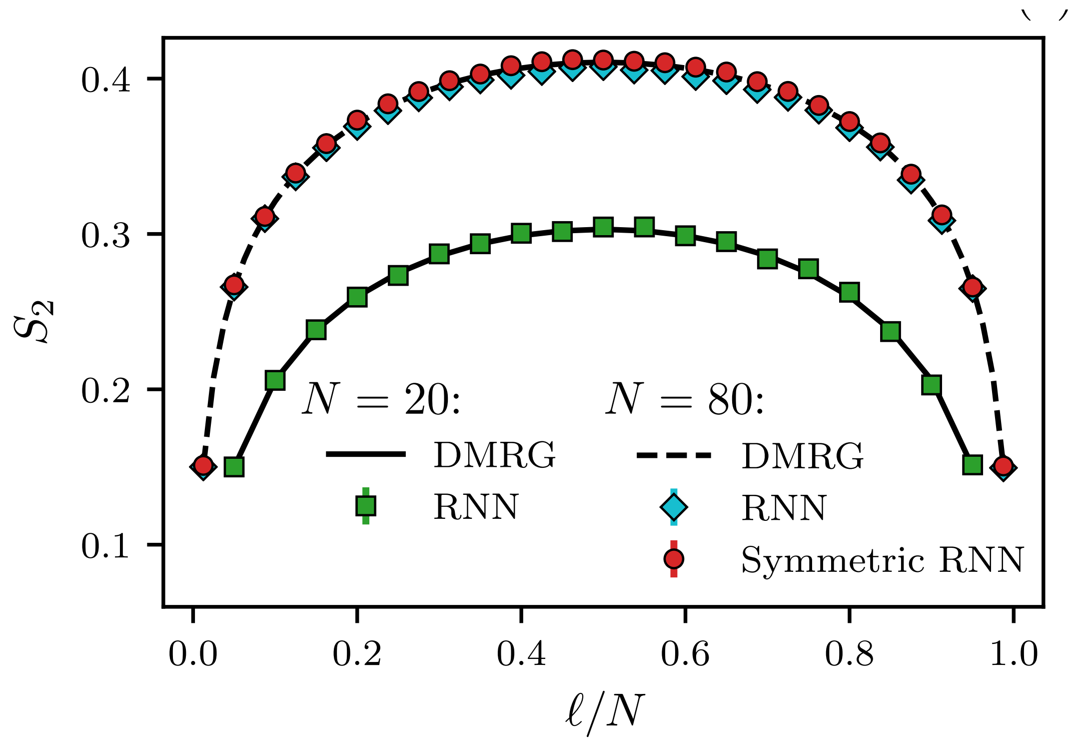
TFIM 1-D N=1000



J1-J2 model in 1-D



TFIM 1-D N=80



TFIM 1-D N=80

Symmetries: Spin inversion, mirror reflection, Sz. Sign: different Marshall signs for the J1-J2 model

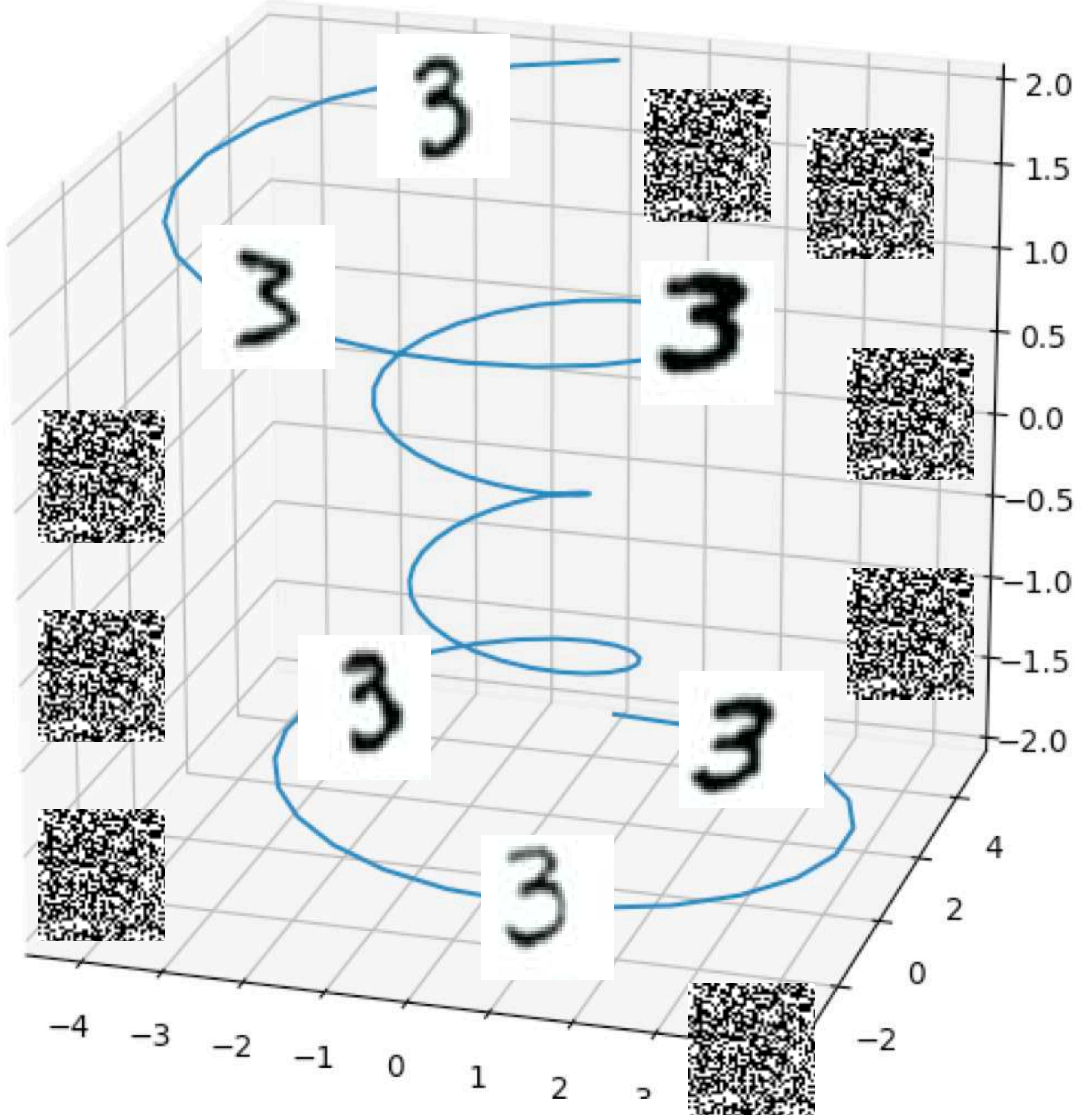
Mohamed Hibat-Allah, Martin Ganahl, Lauren E. Hayward, Roger G. Melko, and Juan Carrasquilla Phys. Rev. Research **2**, 023358 (2020)

Classical topological order

- The peculiar structure of phase space divided into sectors that are connected exclusively by extensive rearrangements of the microscopic degrees of freedom.

Take 28 x 28 binary images

- Size of state space: $2^{28\times 28} = 1.017458 \times 10^{236}$
- Bigger than the number of atoms in the known un noise —> Probability distributions over the image in low-dimensional subspace of these big spaces.



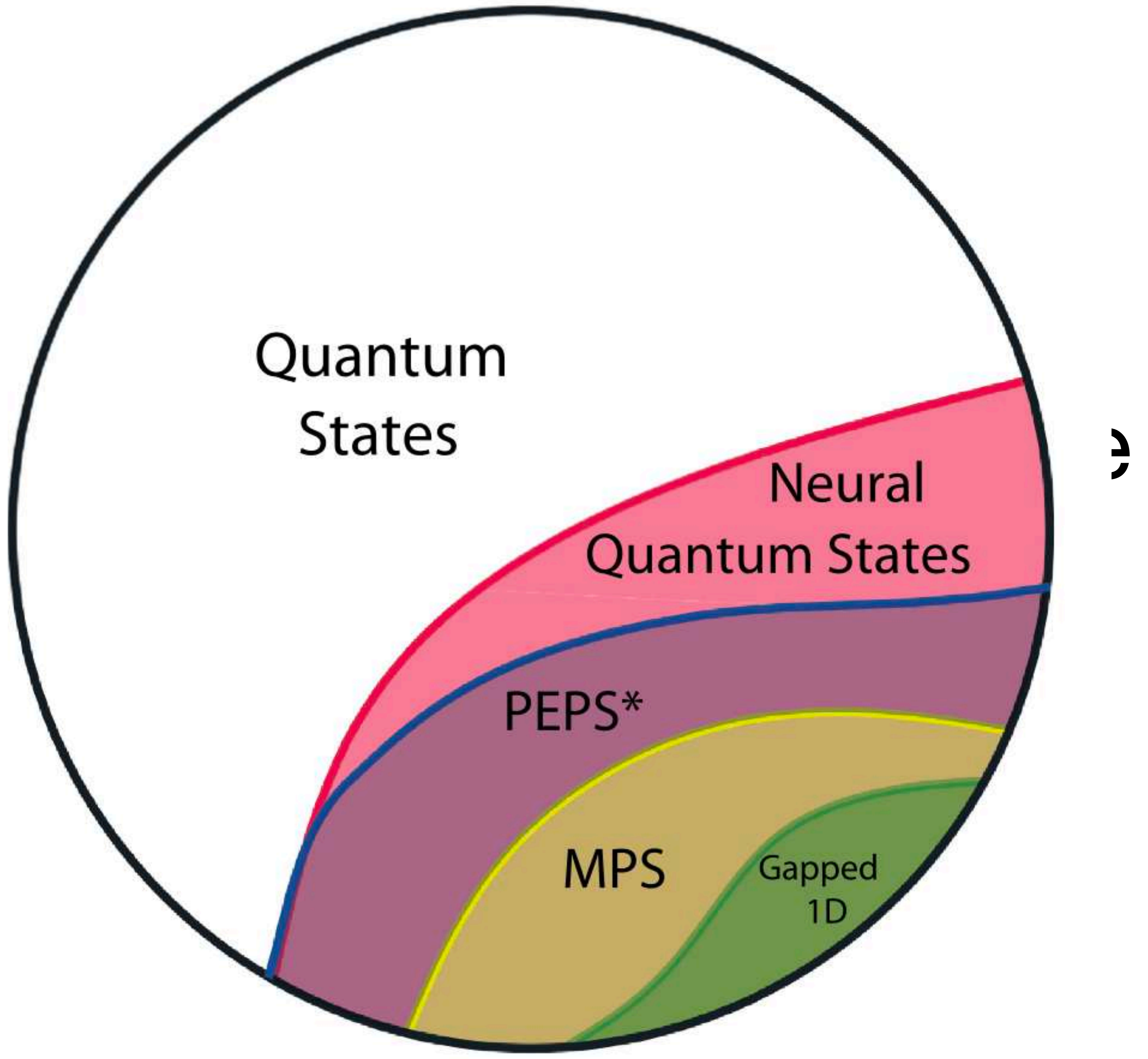


FIG. 3. Expressive power of classically tractable variational quantum states. Different classes of quantum states describing a qudit system with N degrees of freedom and comprising poly(N) variational parameters are compared. MPS can efficiently represent gapped ground states of one-dimensional systems. PEPS* denotes projected entangled pair states of bond dimension χ that are exactly or approximately contracted in poly(N, χ) time on a classical computer. NQS comprise all polynomially tractable TN, thus include MPS, and PEPS∗, while also representing additional states with volume-law entanglement that are not efficiently described by such planar TN.

PHYSICAL REVIEW B 106, 205136 (2022)

ML broad categories

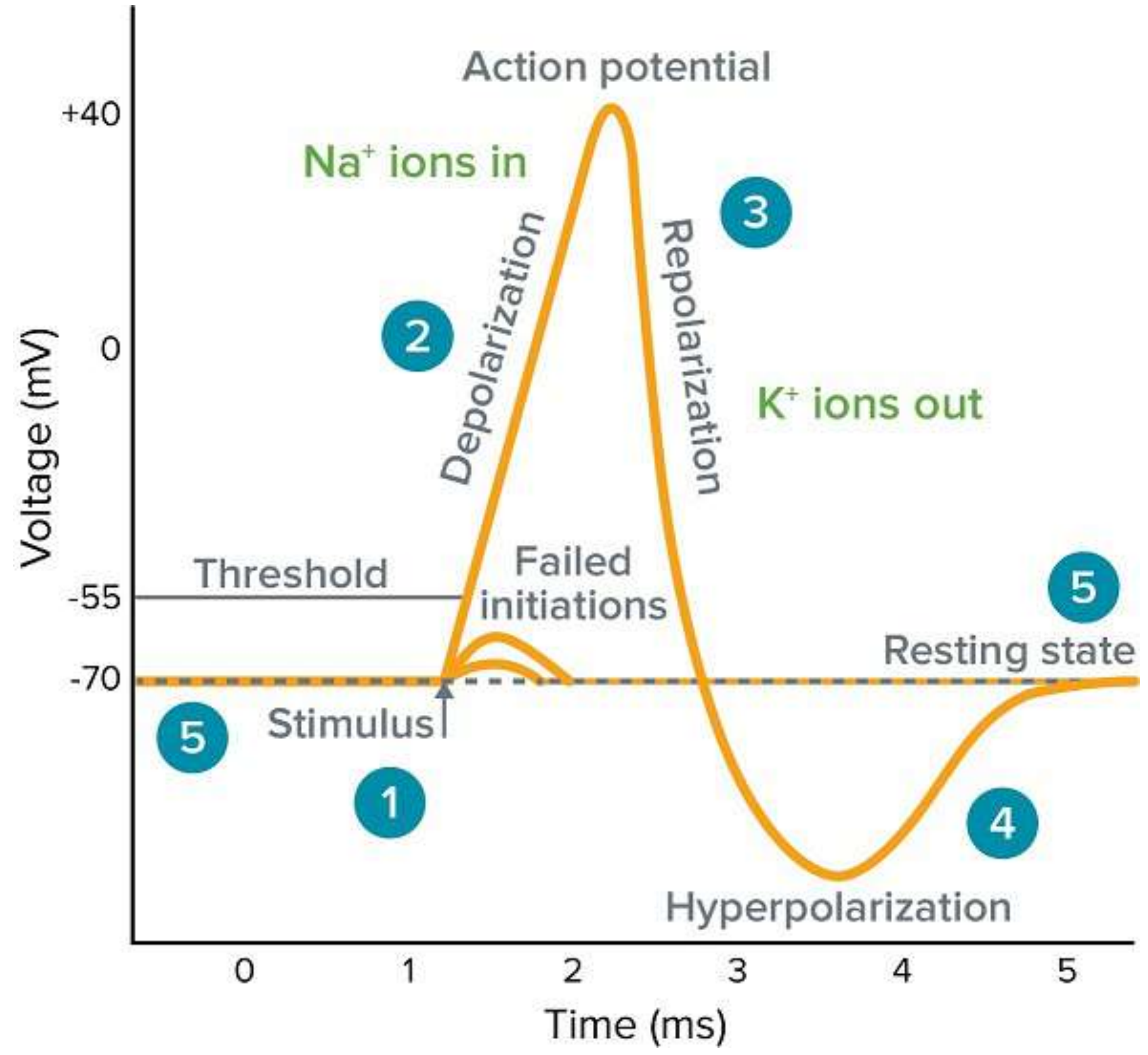
Equation/physical law learning

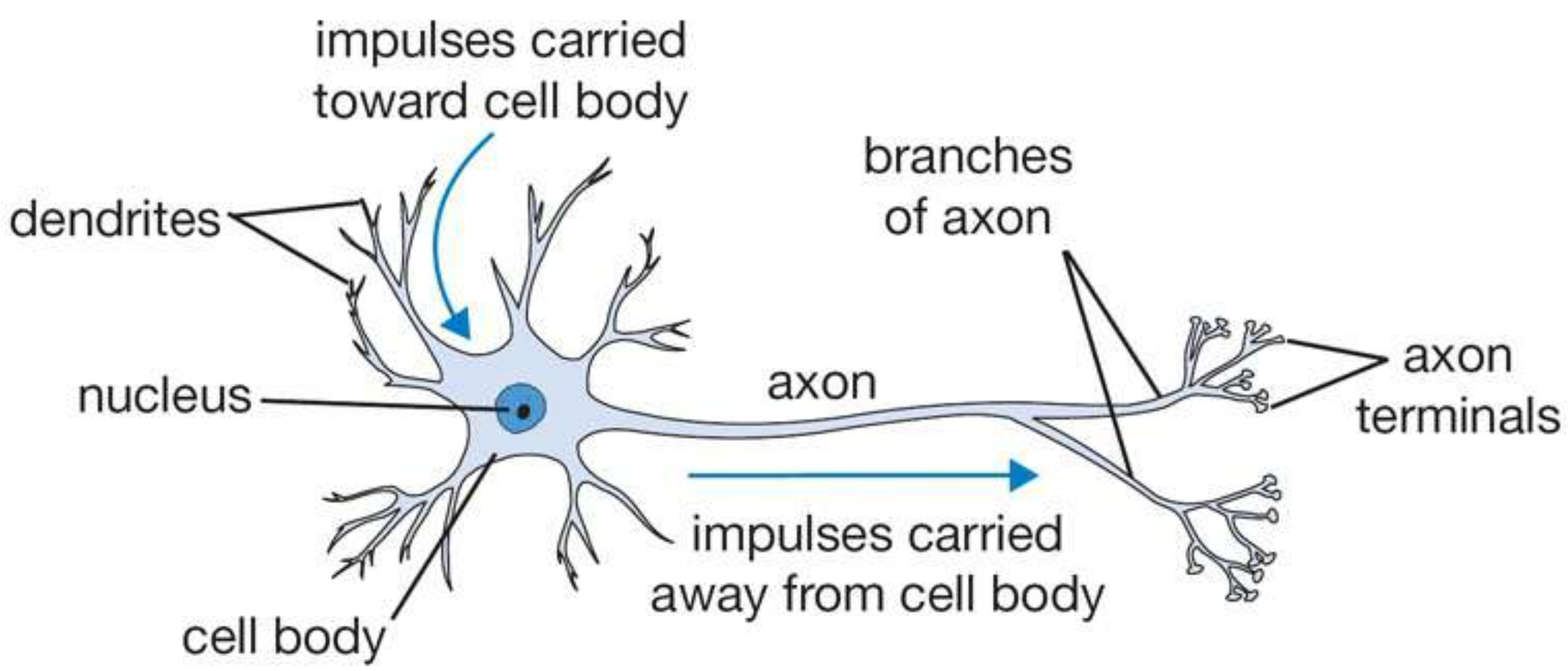
Closed quantum systems	
Time-dependent Schrödinger equation	
Exact	$\mathscr{L}_{\dot{\theta}}$
$i\frac{d}{dt}\lvert\Psi(t)\rangle = H\lvert\Psi(t)\rangle$	$\mathscr{D}\left(\lvert\Psi_{\theta(t)+\delta t\dot{\theta}(t)}\rangle, e^{-iH\delta t}\lvert\Psi_{\theta(t)}\rangle\right)$
Time-Independent Schrödinger equation	
Exact	$\mathscr{L}_{\theta}$
$H\lvert\Psi\rangle = E\lvert\Psi\rangle$	$\langle\Psi_\theta\lvert H\rvert\Psi_\theta\rangle$

Open quantum systems	
Master equation	
Exact	$\mathscr{L}_{\dot{\theta}}$
$\dot{\rho} = L\rho$	$\mathscr{D}\left(\rho_{\theta(t)+\delta t\dot{\theta}(t)}, e^{L\delta t}\rho_{\theta(t)}\right)$
Steady state	
Exact	$\mathscr{L}_{\theta}$
$\dot{\rho} = L\rho = 0$	$\lVert\dot{\rho}_\theta\rVert$
Gibbs state	
Exact	$\mathscr{L}_{\theta}$
$\rho(T) = \frac{e^{-H/T}}{Z(T)}$	$\mathrm{Tr}\left[\rho_\theta H\right] - TS\left(\rho_\theta\right)$

Inspiration: The brain

- Our brain has $\sim 10^{11}$ neurons, each of which communicates to other $\sim 10^4$ neurons





- Neurons receive input signals and accumulate voltage. After some threshold they will fire spiking responses.
- Pic credit: www.moleculardevices.com, http://cs231n.github.io/neural-networks-1/