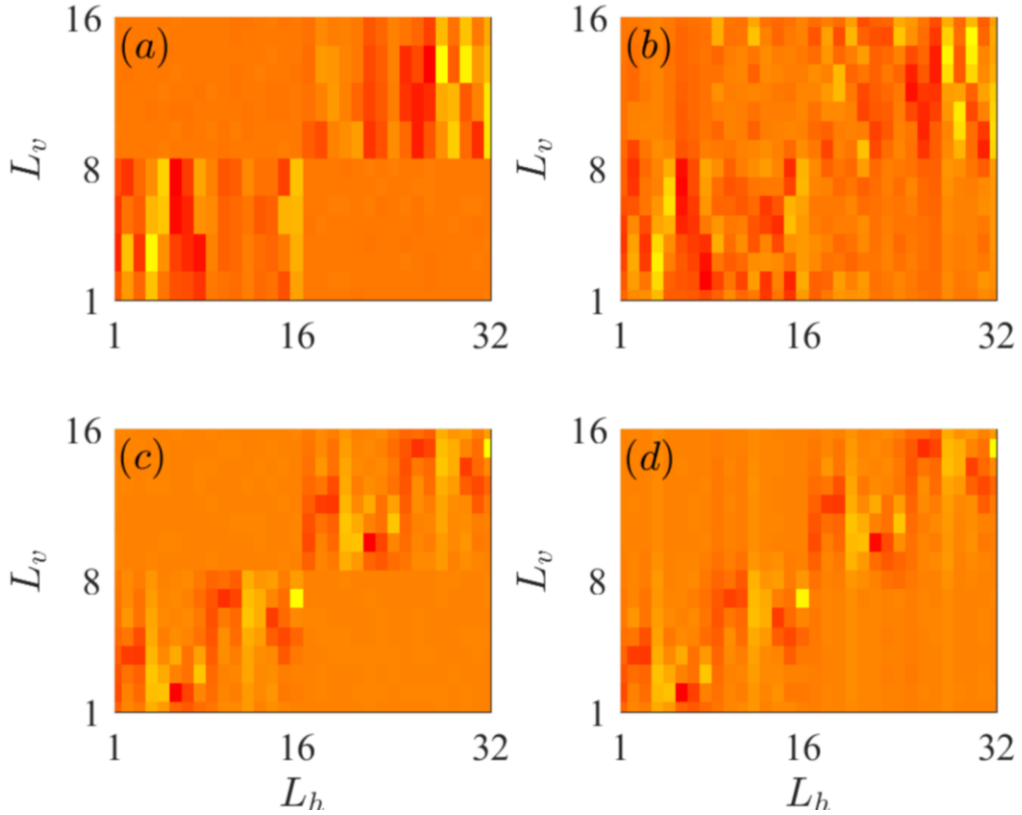
Neural Network Quantum States and Transfer Learning



Dario Poletti

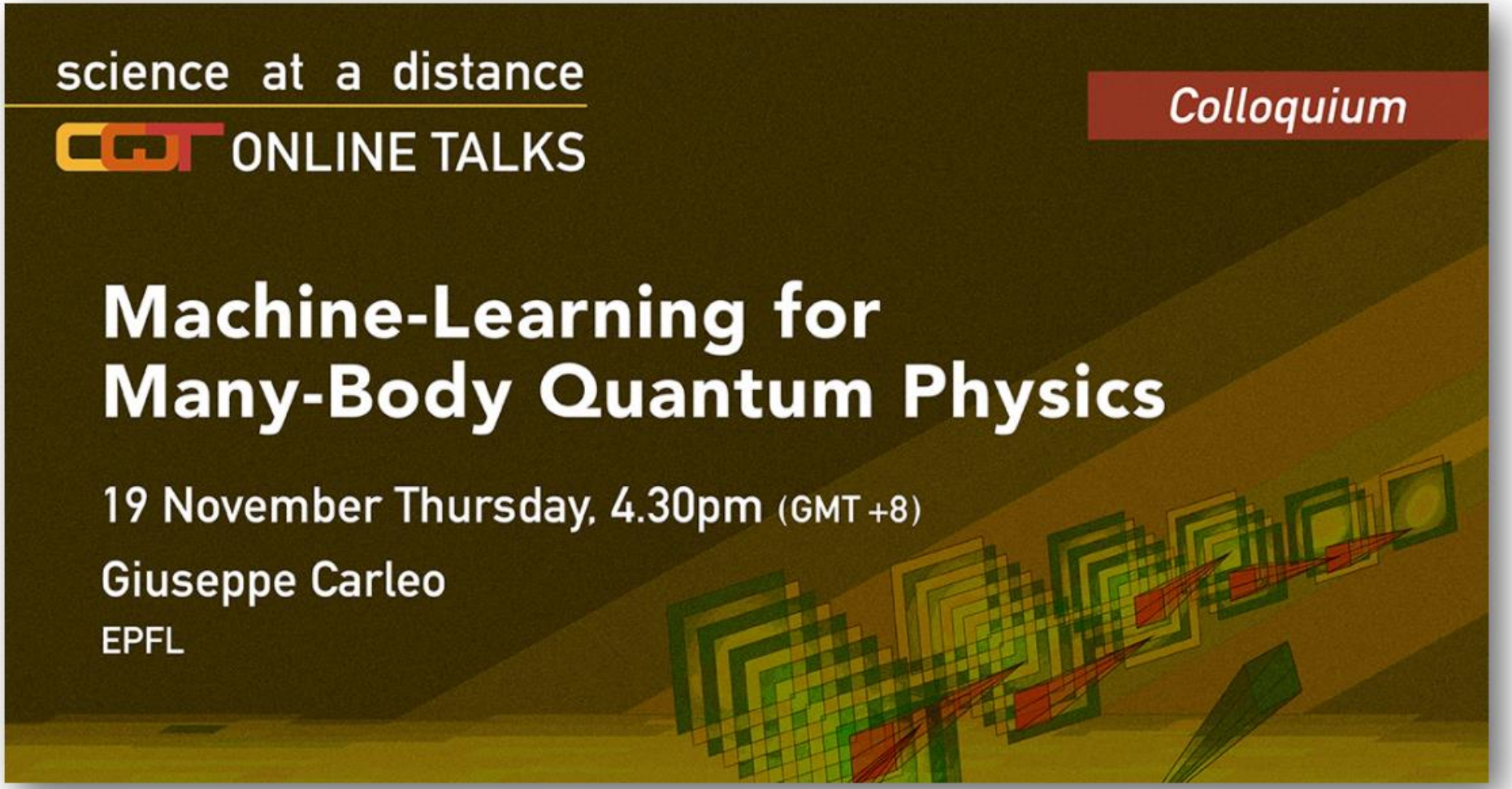
Singapore University of Technology and Design
Majulab

Ministry of Education SINGAPORE

SG NSCC SINGAPORE National Supercomputing Centre

MajuLab

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN



https://www.quantumlah.org/page/key/upcomingevents

Plan of the presentation

1) **Neural Network Quantum States**

2) **Restricted Boltzmann machines**

3) **Restricted Boltzmann machines and transfer learning**

4) **Conclusions and Outlook**

Plan of the presentation

1) **Neural Network Quantum States**

2) **Restricted Boltzmann machines**

3) **Restricted Boltzmann machines and transfer learning**

4) **Conclusions and Outlook**

Neural network quantum states

Neural network (NN) can represent probability distributions

In reality they can represent functions … so why not using neural networks to represent wavefunctions?

Carleo and Troyer (Science 2017) showed that neural networks can be used as a variational ansatz for many-body quantum systems.
This worked open the field of neural-network quantum states NNQS.

- Why now?
- How are NN used in physics?
- How effective NNQS are?
- What is the main working principle for NNQS?
- What architectures for NNQS?

Neural network quantum states: why now?

The always more ubiquitous use of NN could not be missed by physicist.
Their effectiveness in daily applications is resounding.

Natural language processing, image recognition, games …

Great progress comes from

- Use of deep networks
- Advances in hardware (GPUs)

Neural network quantum states: how are NN used in physics?

Guide experiments

Particle physics and cosmology

Recognize phases of matters

Quantum states and processes tomography

Quantum error correction

Finding the ground state of a Hamiltonian

Finding the steady state of an open quantum system

Perform the time-evolution of a quantum system

Accelerate routines

...

Zdeborová, L., and F. Krzakala, Adv. Phys. (2016)
G Carleo, et al., Reviews of Modern Physics (2019)

Neural network quantum states: how effective are NN?

Universal Approximation Theorem -> NN can represent any well-behaved function.
But does it work for the quantum many-body problems we aim to solve?

States described my matrix tensor networks can also be represented by restricted Boltzmann machines (RBM). Chen et al. PRB 2018
This implies that states that can be well represented with tensor networks (see Guo's talk), can also be represented with RBMs.

Restricted Boltzmann machines can represent volume law states. Deng et al. PRX 2017
This implies that there are states which cannot be represented efficiently with tensor networks, but which can be represented efficiently with RBMs.

Neural network quantum states: how effective are NN?

Effectiveness varies significantly on the application and how mature it is.

In particle physics they have been applied for a very long time and they are very effective.
As for neural network quantum states (NNQS), the field is still young and growing

Neural network quantum states: how effective are NN?

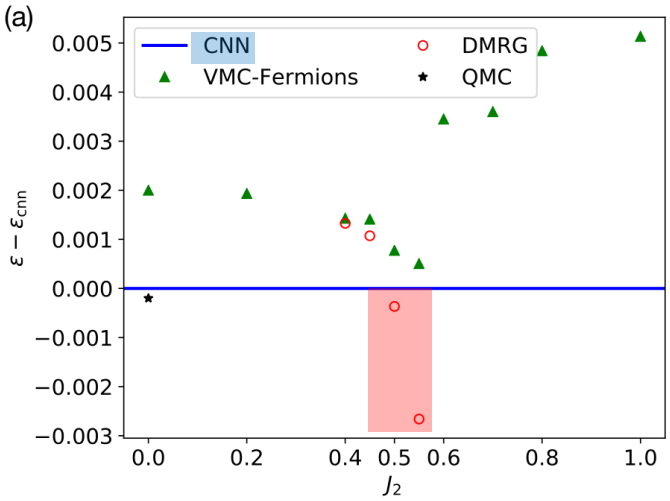
Effectiveness varies significantly on the application and how mature it is.

In particle physics they have been applied for a very long time and they are very effective.
As for neural network quantum states (NNQS), the field is still young and growing

For groundstate, the best comparison is the study of a J1-J2 frustrated system.

Choo et al., Phys. Rev. B 100, 125124 (2019)
10x10 sites

In the frustrated region DMRG still performs better than any other method.



Neural network quantum states: how effective are NN?

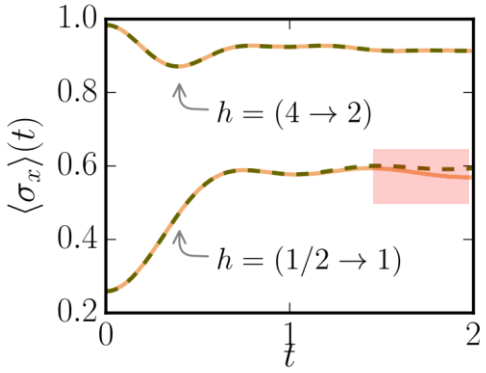
Effectiveness varies significantly on the application and how mature it is.

In particle physics they have been applied for a very long time and they are very effective.
As for neural network quantum states (NNQS), the field is still young and growing

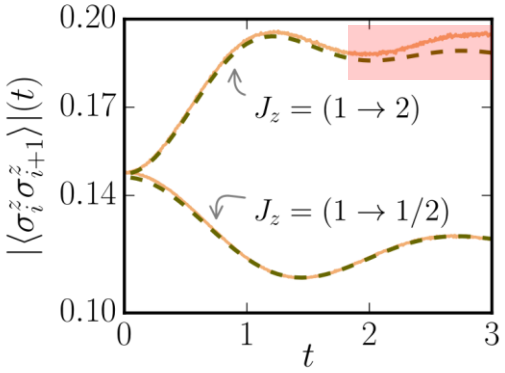
Time evolution in many-body quantum systems.

Carleo and Troyer, Science (2017)

Transverse field Ising model
comparison to exact results



Heisenberg model
comparison to DMRG



Solid lines results from NNQS

Neural network quantum states: what is the main working principle for NNQS?

Variational principle!

$$E[\psi] = \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle} \geq E_0$$

Neural network quantum states: what is the main working principle for NNQS?

Variational principle!

$$E[\psi] = \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} \geq E_0$$

We group all the parameters of the network in the vector **θ**

$$E(\boldsymbol{\theta}) = \frac{\sum_{\boldsymbol{x},\boldsymbol{x'}} \psi^*(\boldsymbol{x};\boldsymbol{\theta}) \langle \boldsymbol{x} \mid H \mid \boldsymbol{x'} \rangle \psi(\boldsymbol{x'};\boldsymbol{\theta})}{\sum_{\boldsymbol{x''}} |\psi(\boldsymbol{x''};\boldsymbol{\theta})|^2}$$

Neural network quantum states: what is the main working principle for NNQS?

Variational principle!

$$E[\psi] = \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} \geq E_0$$

We group all the parameters of the network in the vector **θ**

$$E(\boldsymbol{\theta}) = \frac{\sum_{\boldsymbol{x},\boldsymbol{x}'} \psi^*(\boldsymbol{x};\boldsymbol{\theta}) \langle \boldsymbol{x} \mid H \mid \boldsymbol{x}' \rangle \psi(\boldsymbol{x}';\boldsymbol{\theta})}{\sum_{\boldsymbol{x}''} |\psi(\boldsymbol{x}'';\boldsymbol{\theta})|^2}$$

$$= \sum_{\boldsymbol{x}} \left(\frac{|\psi(\boldsymbol{x};\boldsymbol{\theta})|^2}{\sum_{\boldsymbol{x}''} |\psi(\boldsymbol{x}'';\boldsymbol{\theta})|^2} \sum_{\boldsymbol{x}'} \langle \boldsymbol{x} \mid H \mid \boldsymbol{x}' \rangle \frac{\psi(\boldsymbol{x}';\boldsymbol{\theta})}{\psi(\boldsymbol{x};\boldsymbol{\theta})} \right)$$

Neural network quantum states: what is the main working principle for NNQS?

Variational principle!

$$E[\psi] = \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} \geq E_0$$

We group all the parameters of the network in the vector **θ**

$$E(\boldsymbol{\theta}) = \frac{\sum_{\boldsymbol{x},\boldsymbol{x}'} \psi^*(\boldsymbol{x};\boldsymbol{\theta}) \langle \boldsymbol{x} \mid H \mid \boldsymbol{x}' \rangle \psi(\boldsymbol{x}';\boldsymbol{\theta})}{\sum_{\boldsymbol{x}''} |\psi(\boldsymbol{x}'';\boldsymbol{\theta})|^2}$$

$$= \sum_{\boldsymbol{x}} \left(\frac{|\psi(\boldsymbol{x};\boldsymbol{\theta})|^2}{\sum_{\boldsymbol{x}''} |\psi(\boldsymbol{x}'';\boldsymbol{\theta})|^2} \sum_{\boldsymbol{x}'} \langle \boldsymbol{x} \mid H \mid \boldsymbol{x}' \rangle \frac{\psi(\boldsymbol{x}';\boldsymbol{\theta})}{\psi(\boldsymbol{x};\boldsymbol{\theta})} \right)$$

$$= \sum_{\boldsymbol{x}} p_\psi(\boldsymbol{x};\boldsymbol{\theta}) E_{\mathrm{loc}}(\boldsymbol{x};\boldsymbol{\theta})$$

Neural network quantum states: what is the main working principle for NNQS?

Variational principle!

$$E[\psi] = \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} \geq E_0$$

We group all the parameters of the network in the vector **θ**

$$E(\boldsymbol{\theta}) = \frac{\sum_{\boldsymbol{x},\boldsymbol{x}'} \psi^*(\boldsymbol{x};\boldsymbol{\theta}) \langle \boldsymbol{x} \mid H \mid \boldsymbol{x}' \rangle \psi(\boldsymbol{x}';\boldsymbol{\theta})}{\sum_{\boldsymbol{x}''} |\psi(\boldsymbol{x}'';\boldsymbol{\theta})|^2}$$

$$= \sum_{\boldsymbol{x}} \left(\frac{|\psi(\boldsymbol{x};\boldsymbol{\theta})|^2}{\sum_{\boldsymbol{x}''} |\psi(\boldsymbol{x}'';\boldsymbol{\theta})|^2} \sum_{\boldsymbol{x}'} \langle \boldsymbol{x} \mid H \mid \boldsymbol{x}' \rangle \frac{\psi(\boldsymbol{x}';\boldsymbol{\theta})}{\psi(\boldsymbol{x};\boldsymbol{\theta})} \right)$$

$$= \sum_{\boldsymbol{x}} p_\psi(\boldsymbol{x};\boldsymbol{\theta}) E_{\mathrm{loc}}(\boldsymbol{x};\boldsymbol{\theta})$$

$$E_{\mathrm{loc}}(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{\boldsymbol{x}'} \langle \boldsymbol{x} \mid H \mid \boldsymbol{x}' \rangle \frac{\psi(\boldsymbol{x}';\boldsymbol{\theta})}{\psi(\boldsymbol{x};\boldsymbol{\theta})}$$

Neural network quantum states: what is the main working principle for NNQS?

Variational principle!

$$E[\psi] = \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} \geq E_0$$

We group all the parameters of the network in the vector **θ**

$$E(\boldsymbol{\theta}) = \frac{\sum_{\boldsymbol{x},\boldsymbol{x}'} \psi^*(\boldsymbol{x};\boldsymbol{\theta}) \langle \boldsymbol{x} \mid H \mid \boldsymbol{x}' \rangle \psi(\boldsymbol{x}';\boldsymbol{\theta})}{\sum_{\boldsymbol{x}''} |\psi(\boldsymbol{x}'';\boldsymbol{\theta})|^2}$$

$$= \sum_{\boldsymbol{x}} \left(\frac{|\psi(\boldsymbol{x};\boldsymbol{\theta})|^2}{\sum_{\boldsymbol{x}''} |\psi(\boldsymbol{x}'';\boldsymbol{\theta})|^2} \sum_{\boldsymbol{x}'} \langle \boldsymbol{x} \mid H \mid \boldsymbol{x}' \rangle \frac{\psi(\boldsymbol{x}';\boldsymbol{\theta})}{\psi(\boldsymbol{x};\boldsymbol{\theta})} \right)$$

$$= \sum_{\boldsymbol{x}} p_\psi(\boldsymbol{x};\boldsymbol{\theta}) E_{\mathrm{loc}}(\boldsymbol{x};\boldsymbol{\theta})$$

$$E_{\mathrm{loc}}(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{\boldsymbol{x}'} \langle \boldsymbol{x} \mid H \mid \boldsymbol{x}' \rangle \frac{\psi(\boldsymbol{x}';\boldsymbol{\theta})}{\psi(\boldsymbol{x};\boldsymbol{\theta})}$$

From the neural network.

$\psi(\boldsymbol{x};\boldsymbol{\theta})$

Neural network quantum states: what is the main working principle for NNQS?

Variational principle!

$$E[\psi] = \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} \geq E_0$$

We group all the parameters of the network in the vector **θ**

$$E(\boldsymbol{\theta}) = \frac{\sum_{\boldsymbol{x},\boldsymbol{x}'} \psi^*(\boldsymbol{x};\boldsymbol{\theta}) \langle \boldsymbol{x} \mid H \mid \boldsymbol{x}' \rangle \psi(\boldsymbol{x}';\boldsymbol{\theta})}{\sum_{\boldsymbol{x}''} |\psi(\boldsymbol{x}'';\boldsymbol{\theta})|^2}$$

$$= \sum_{\boldsymbol{x}} \left(\frac{|\psi(\boldsymbol{x};\boldsymbol{\theta})|^2}{\sum_{\boldsymbol{x}''} |\psi(\boldsymbol{x}'';\boldsymbol{\theta})|^2} \sum_{\boldsymbol{x}'} \langle \boldsymbol{x} \mid H \mid \boldsymbol{x}' \rangle \frac{\psi(\boldsymbol{x}';\boldsymbol{\theta})}{\psi(\boldsymbol{x};\boldsymbol{\theta})} \right)$$

$$= \sum_{\boldsymbol{x}} p_\psi(\boldsymbol{x};\boldsymbol{\theta}) E_{\mathrm{loc}}(\boldsymbol{x};\boldsymbol{\theta})$$

$$E_{\mathrm{loc}}(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{\boldsymbol{x}'} \langle \boldsymbol{x} \mid H \mid \boldsymbol{x}' \rangle \frac{\psi(\boldsymbol{x}';\boldsymbol{\theta})}{\psi(\boldsymbol{x};\boldsymbol{\theta})}$$

From the neural network.

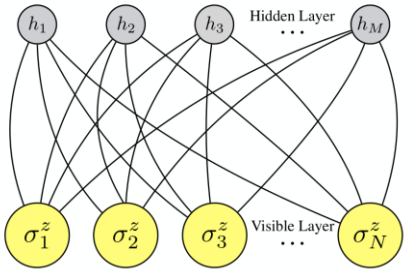
$\psi(\boldsymbol{x};\boldsymbol{\theta})$

Sample E_{loc} from the neural network using $p_\psi = |\psi|^2$

In essence, a variational Monte Carlo method

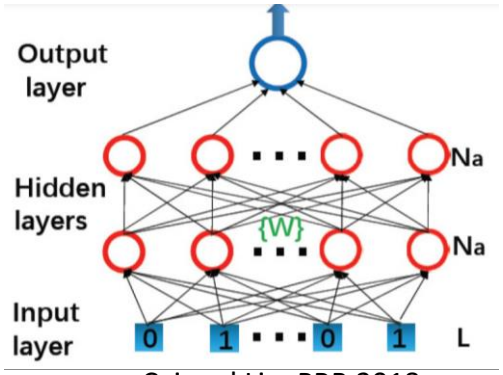
Neural network quantum states: what architectures for NNQS?

Restricted Boltzmann machines and variations …



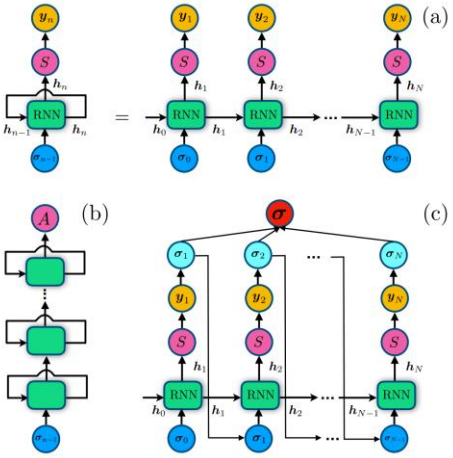
Carlo and Troyer, Science 2017

Multilayer perceptron



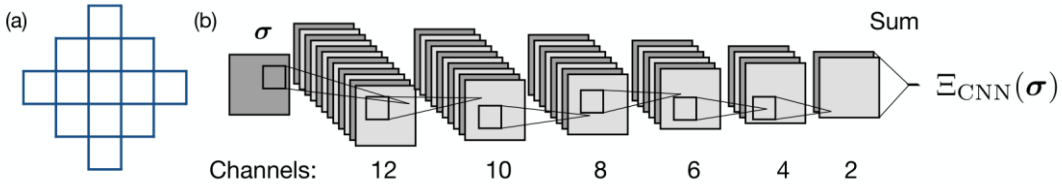
Cai and Liu, PRB 2018

Recurrent neural networks



Hibat-Allah et al., PRR 2020

Convolutional neural networks

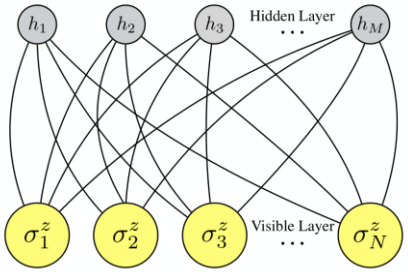


Choo et al., PRB 2019

… and more …

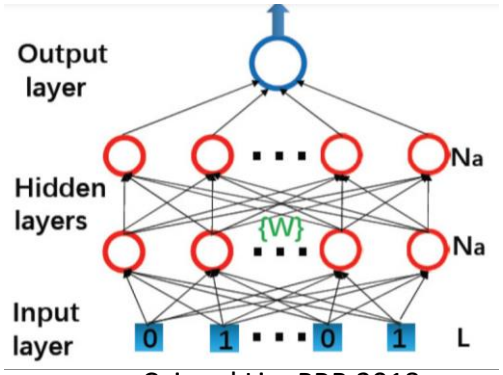
Neural network quantum states: what architectures for NNQS?

Restricted Boltzmann machines and variations …



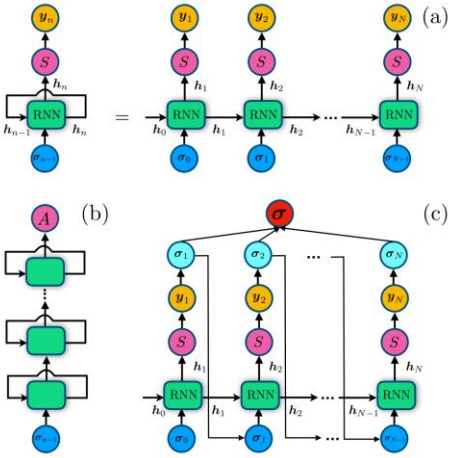
Carlo and Troyer, Science 2017

Multilayer perceptron



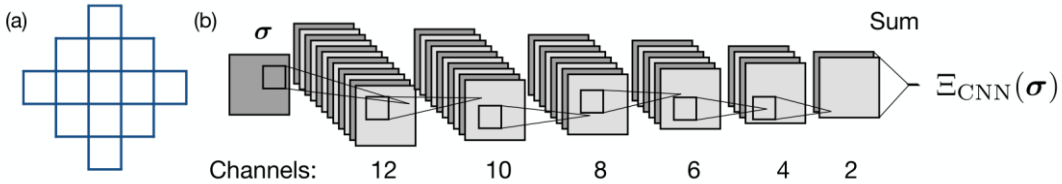
Cai and Liu, PRB 2018

Recurrent neural networks



Hibat-Allah et al., PRR 2020

Convolutional neural networks



Choo et al., PRB 2019

… and more …

+ **optimizers** to find the minimum
e.g. stochastic gradient descent

+ **samplers** to sample effectively and efficiently
e.g. Metropolis-Hastings, Gibbs

Plan of the presentation

1) **Neural Network Quantum States**

2) **Restricted Boltzmann machines**

3) **Restricted Boltzmann machines and transfer learning**

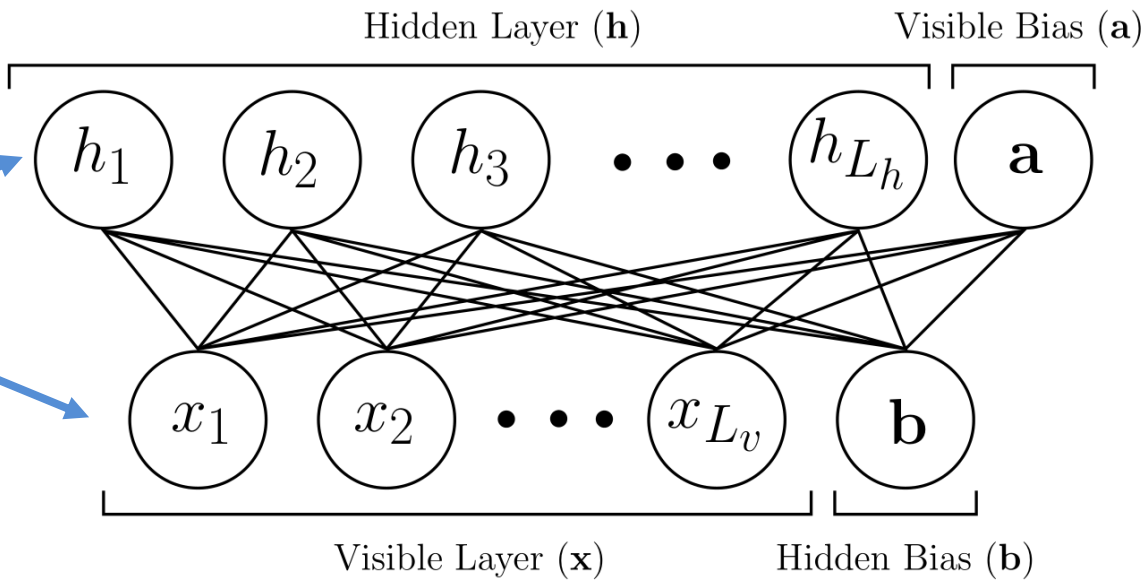
4) **Conclusions and Outlook**

Restricted Boltzmann machines

Neural networks can be used as a variational ansatz for many-body quantum systems (Carleo and Troyer Science 2017).

The first example of NNQS has been Boltzmann machines.

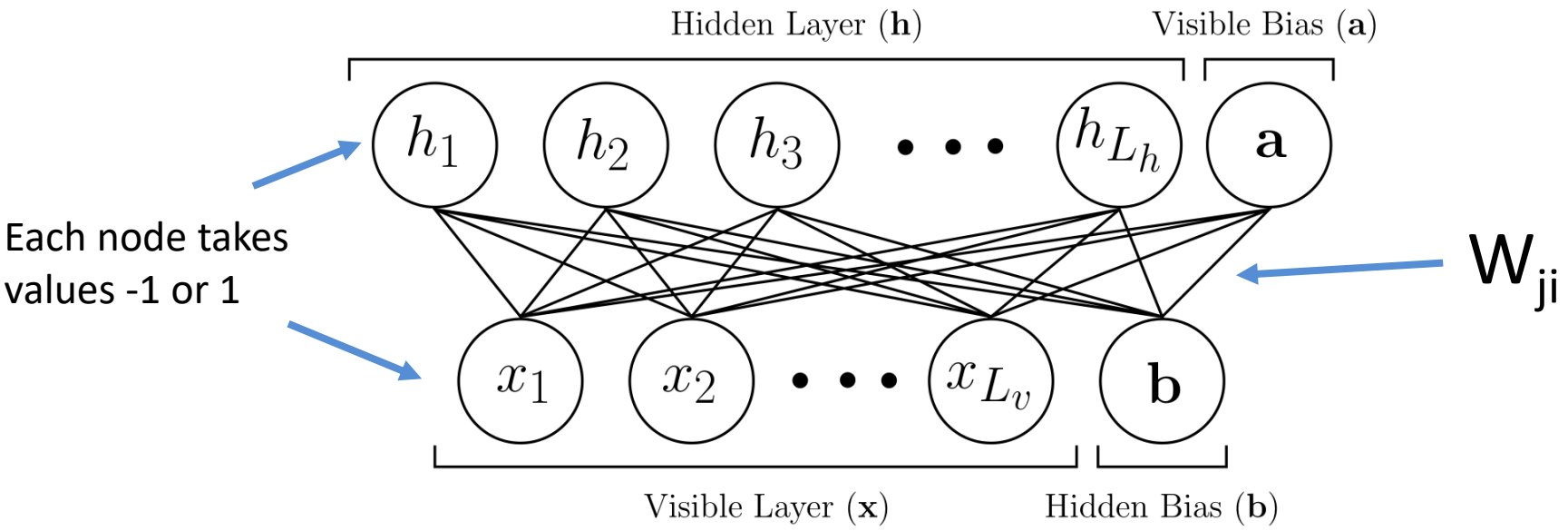
Each node takes values -1 or 1



Restricted Boltzmann machines

Neural networks can be used as a variational ansatz for many-body quantum systems (Carleo and Troyer Science 2017).

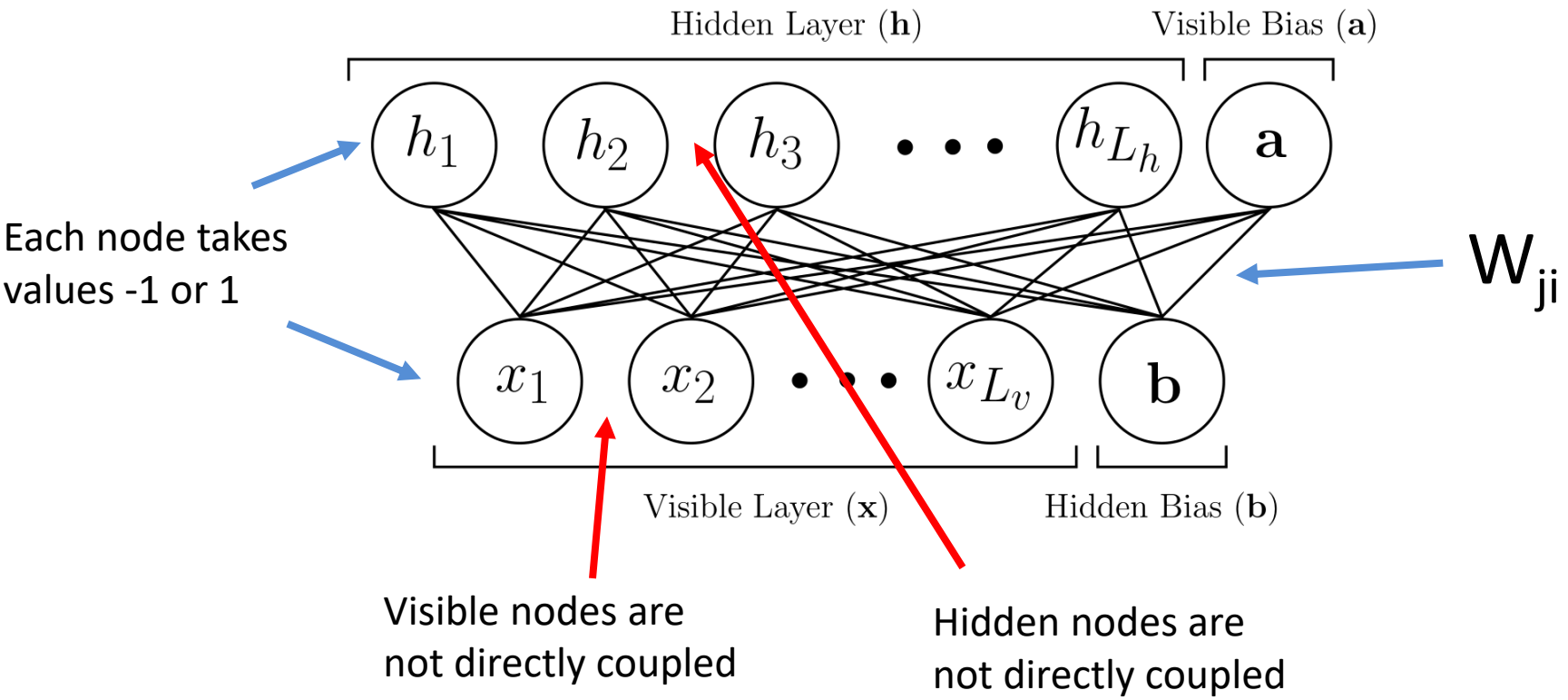
The first example of NNQS has been Boltzmann machines.



Restricted Boltzmann machines

Neural networks can be used as a variational ansatz for many-body quantum systems (Carleo and Troyer Science 2017).

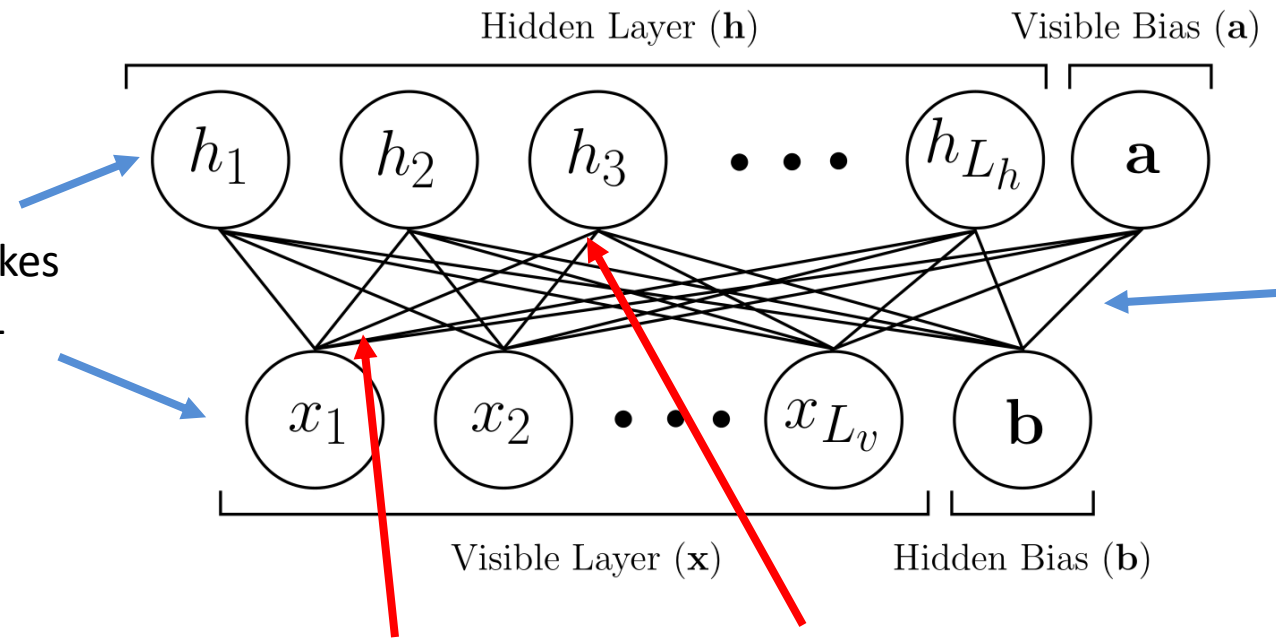
The first example of NNQS has been Boltzmann machines.



Restricted Boltzmann machines

Neural networks can be used as a variational ansatz for many-body quantum systems (Carleo and Troyer Science 2017).

The first example of NNQS has been Boltzmann machines.



Each node takes values -1 or 1

W_{ji}

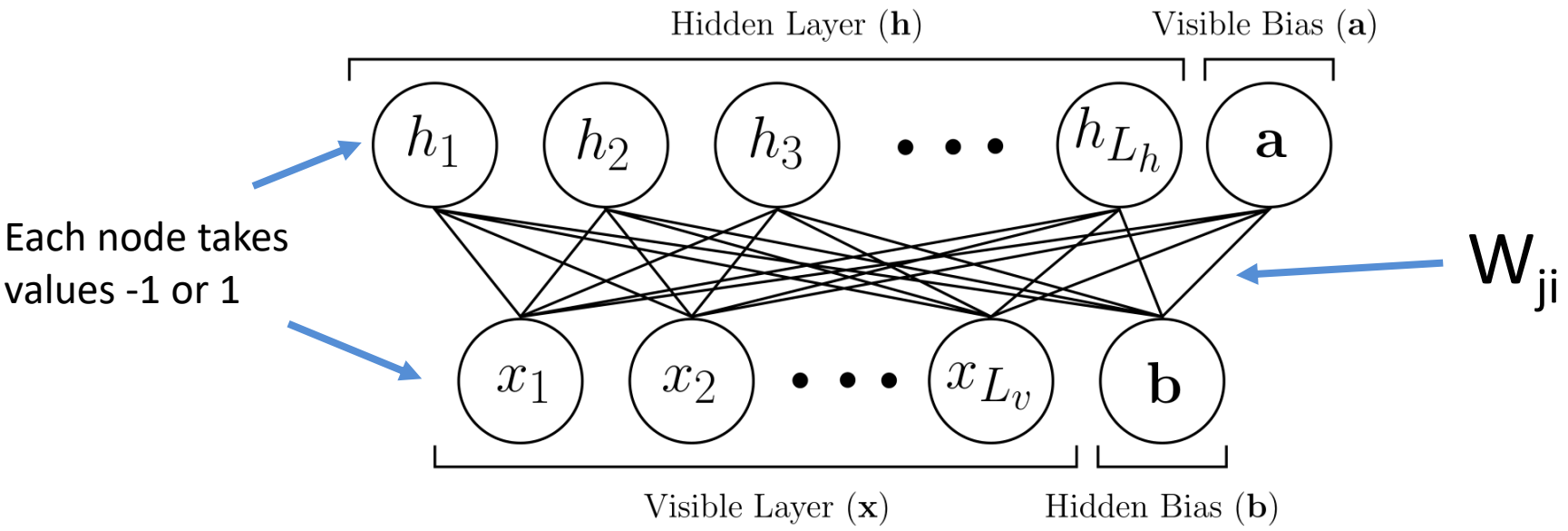
Each visible node is directly coupled to all hidden nodes

Each hidden node is directly coupled to all visible nodes

Restricted Boltzmann machines

Neural networks can be used as a variational ansatz for many-body quantum systems (Carleo and Troyer Science 2017).

The first example of NNQS has been Boltzmann machines.

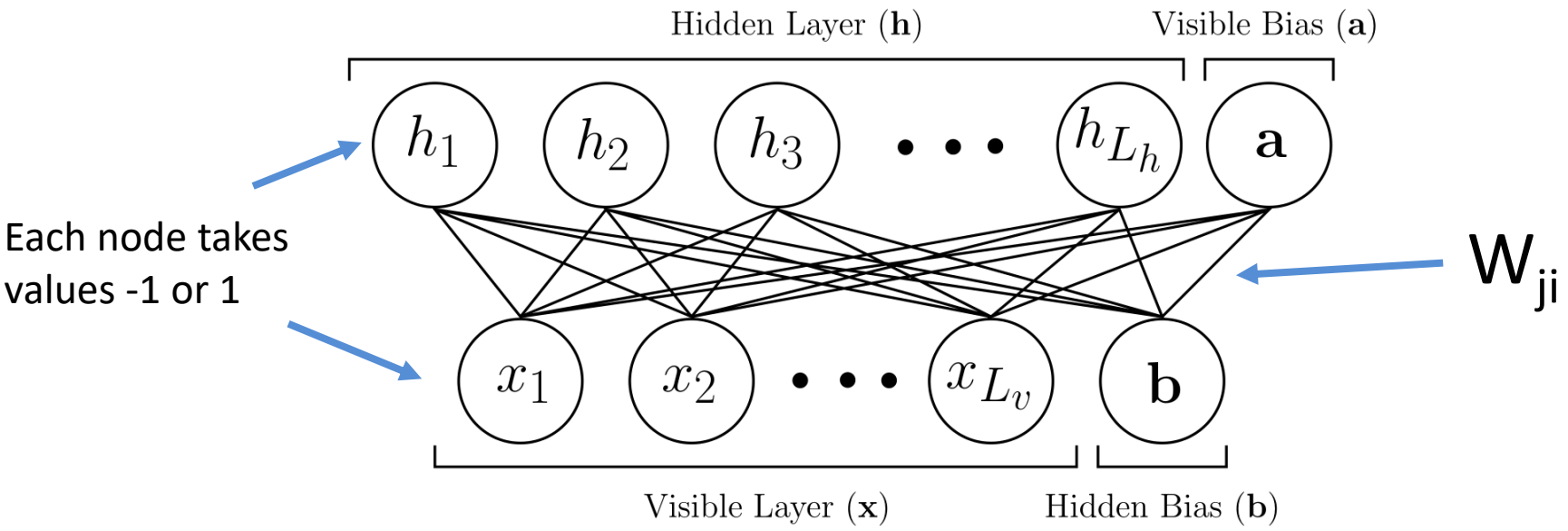


$$RBM(\boldsymbol{x}, \boldsymbol{h}; \boldsymbol{W}, \boldsymbol{a}, \boldsymbol{b}) = \frac{1}{Z} e^{\sum_j a_j x_j + \sum_i b_i h_i + \sum_{i,j} x_j W_{ji} h_i}$$

Restricted Boltzmann machines

Neural networks can be used as a variational ansatz for many-body quantum systems (Carleo and Troyer Science 2017).

The first example of NNQS has been Boltzmann machines.

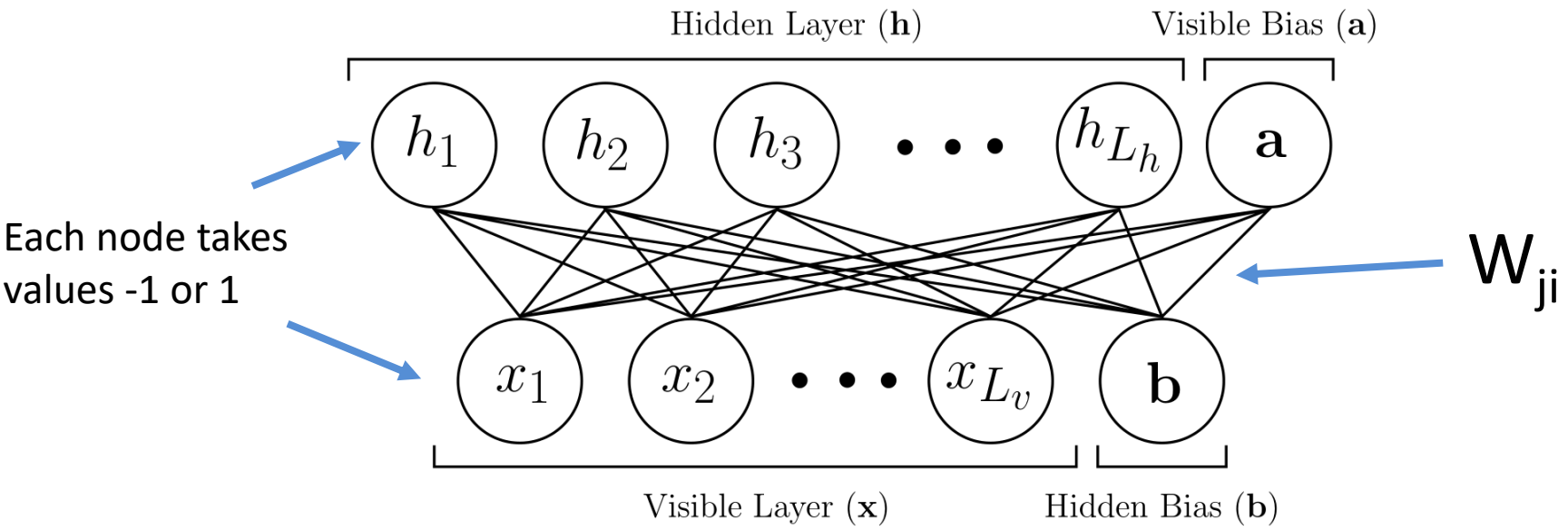


$$RBM(\boldsymbol{x}, \boldsymbol{h}; \boldsymbol{W}, \boldsymbol{a}, \boldsymbol{b}) = \frac{1}{Z} e^{\sum_j a_j x_j + \sum_i b_i h_i + \sum_{i,j} x_j W_{ji} h_i}$$

Restricted Boltzmann machines

Neural networks can be used as a variational ansatz for many-body quantum systems (Carleo and Troyer Science 2017).

The first example of NNQS has been Boltzmann machines.

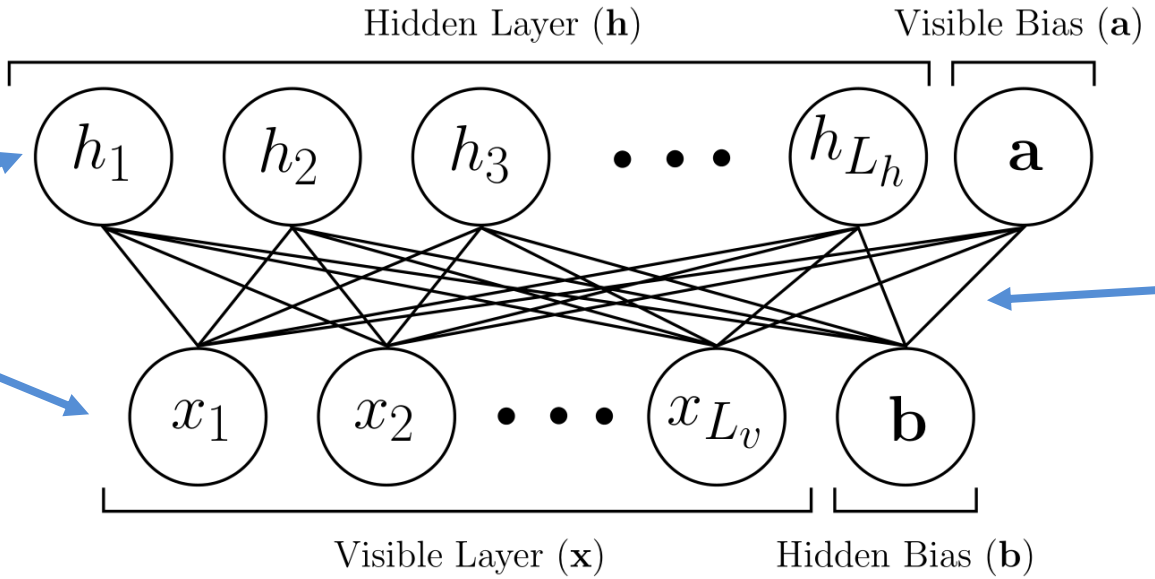


$$RBM\,(\boldsymbol{x};\boldsymbol{W},\boldsymbol{a},\boldsymbol{b}) = \frac{1}{Z}\exp\left(\sum_j a_j x_j\right)\prod_i 2\cosh\left(\sum_j x_j W_{ji} + b_i\right)$$

Restricted Boltzmann machines

Neural networks can be used as a variational ansatz for many-body quantum systems (Carleo and Troyer Science 2017).

The first example of NNQS has been Boltzmann machines.



Each node takes values -1 or 1

W_{ji}

This is your ansatz for

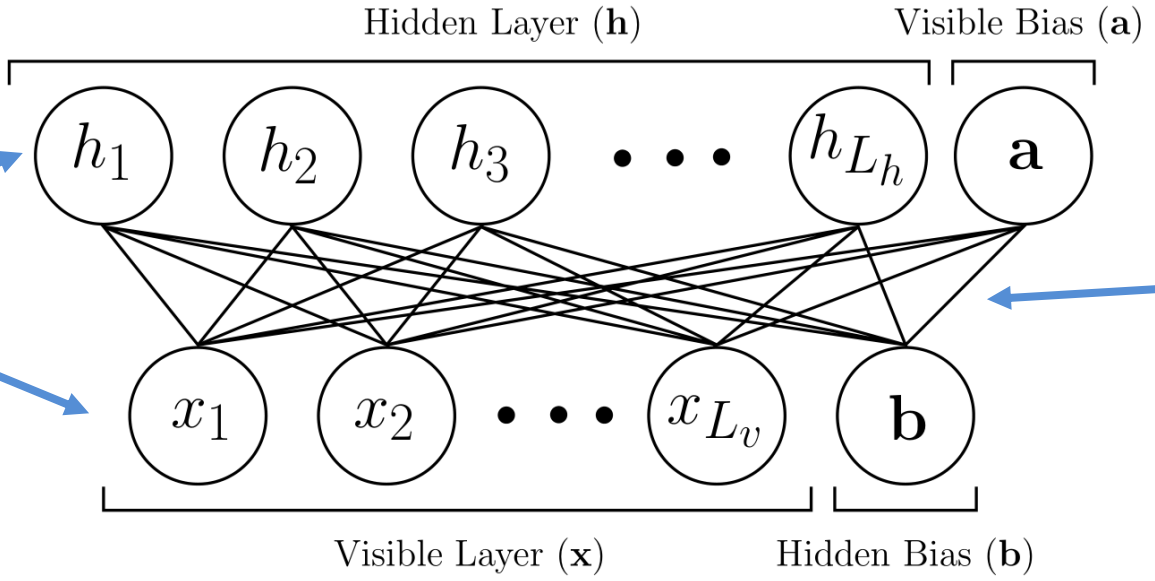
ψ

$$RBM\,(\boldsymbol{x};\boldsymbol{W},\boldsymbol{a},\boldsymbol{b}) = \frac{1}{Z}\exp\left(\sum_j a_j x_j\right)\prod_i 2\cosh\left(\sum_j x_j W_{ji} + b_i\right)$$

Restricted Boltzmann machines

Neural networks can be used as a variational ansatz for many-body quantum systems (Carleo and Troyer Science 2017).

The first example of NNQS has been Boltzmann machines.



Each node takes values -1 or 1

W_{ji}

This is your ansatz for

ψ

$$RBM\,(\boldsymbol{x};\boldsymbol{W},\boldsymbol{a},\boldsymbol{b}) = \frac{1}{Z}\exp\left(\sum_j a_j x_j\right)\prod_i 2\cosh\left(\sum_j x_j W_{ji} + b_i\right)$$

$p_\psi = |\psi|^2$

When you know that is real and positive

Restricted Boltzmann machines

$$RBM(\boldsymbol{x}, \boldsymbol{h}; \boldsymbol{W}, \boldsymbol{a}, \boldsymbol{b}) = \frac{1}{Z} e^{\sum_j a_j x_j + \sum_i b_i h_i + \sum_{i,j} x_j W_{ji} h_i}$$

ψ

$p_\psi = |\psi|^2$

$$RBM(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{a}, \boldsymbol{b}) = \frac{1}{Z} \exp\left(\sum_j a_j x_j\right) \prod_i 2\cosh\left(\sum_j x_j W_{ji} + b_i\right)$$

When you know that is real and positive

$$E(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} p_\psi(\boldsymbol{x}; \boldsymbol{\theta}) E_{\text{loc}}(\boldsymbol{x}; \boldsymbol{\theta})$$

$$E_{\text{loc}}(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{\boldsymbol{x'}} \langle \boldsymbol{x} \mid H \mid \boldsymbol{x'} \rangle \frac{\psi(\boldsymbol{x'}; \boldsymbol{\theta})}{\psi(\boldsymbol{x}; \boldsymbol{\theta})}$$

The content of these two boxes allows you do find an approximation for the ground state

Plan of the presentation

1) **Neural Network Quantum States**

2) **Restricted Boltzmann machines**

3) **Restricted Boltzmann machines and transfer learning**

4) **Conclusions and Outlook**

Restricted Boltzmann machines and transfer learning

Singaporean-French collaboration involving SUTD – NUS – Majulab – UCA
with 60% physicists and 40% computer scientists.

Remmy Zen, Long My, Ryan Tan, Frederic Hebert, Mario Gattobigio, Christian Miniatura, DP, Stephane Bressan

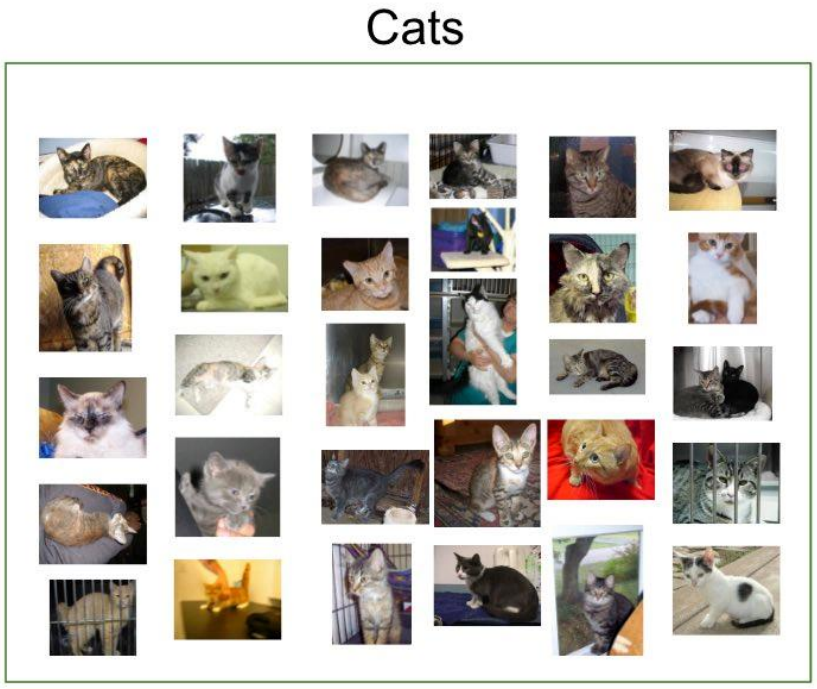
Singapore

France

Zen et al., Physical Review E 2020

Restricted Boltzmann machines and transfer learning

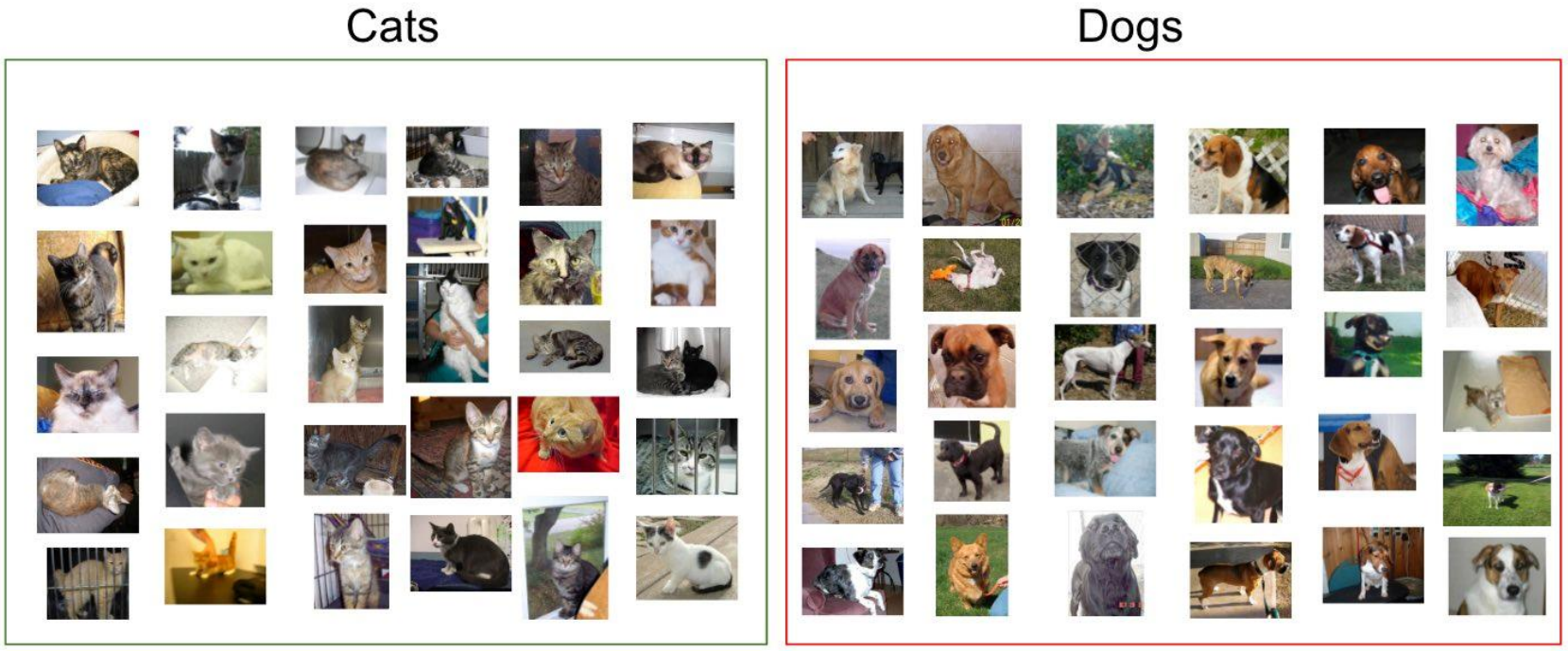
Typically transfer learning is used in scenarios like this:
you have a large number of data to train a neural network (for example many photos of cats), and a small amount of data for a different but similar problem (e.g. photos of dogs). **Then you can first train the network first with the data and refine it with the cats.**



Sample of cats & dogs i

Restricted Boltzmann machines and transfer learning

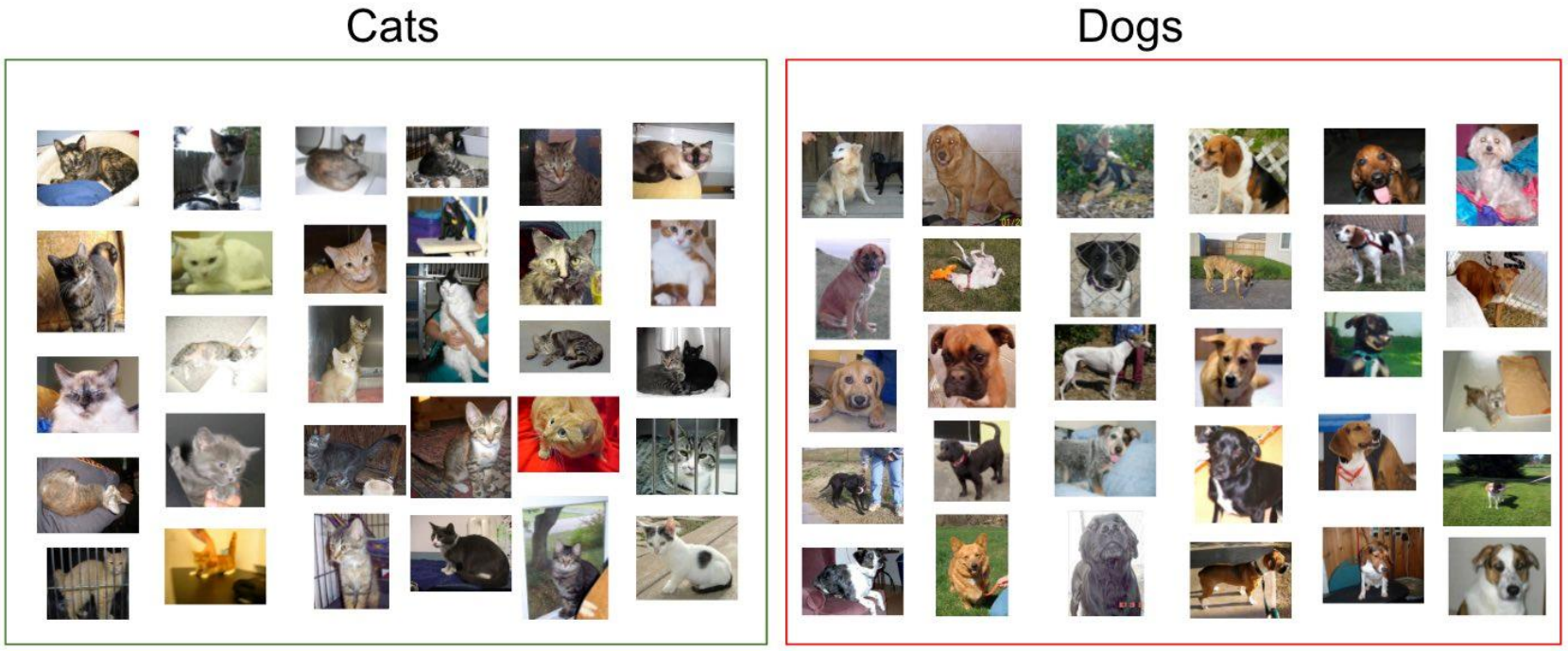
Typically transfer learning is used in scenarios like this:
you have a large number of data to train a neural network (for example many photos of cats), and a small amount of data for a different but similar problem (e.g. photos of dogs). **Then you can first train the network first with the data and refine it with the cats.**



Sample of cats & dogs images from Kaggle Dataset

Restricted Boltzmann machines and transfer learning

Typically transfer learning is used in scenarios like this:
you have a large number of data to train a neural network (for example many photos of cats), and a small amount of data for a different but similar problem (e.g. photos of dogs). **Then you can first train the network first with the data and refine it with the cats.**



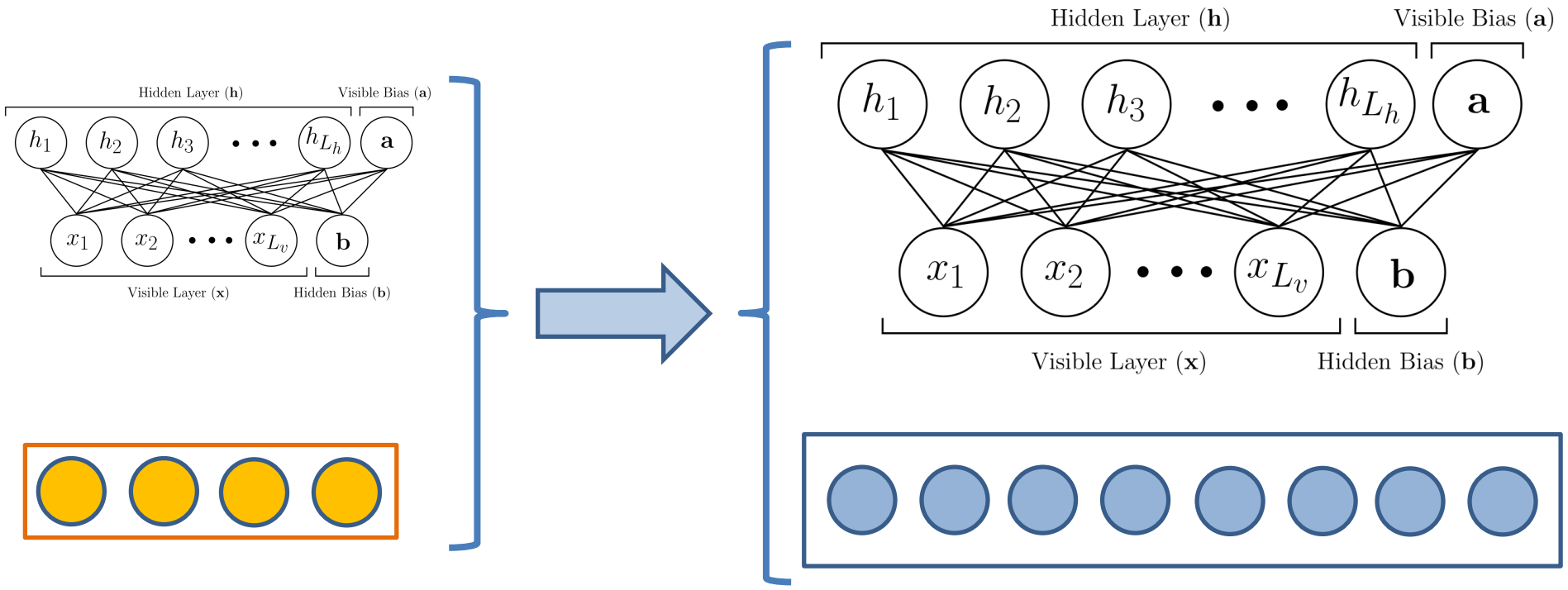
Sample of cats & dogs images from Kaggle Dataset

We can extend this to problem in quantum physics and also **beyond supervised learning**.

Restricted Boltzmann machines and transfer learning

Here we aim to work similarly:

- Train a network for a system of a certain size
- Transfer the weights of the network to another network for the description of a larger system.



Restricted Boltzmann machines and transfer learning

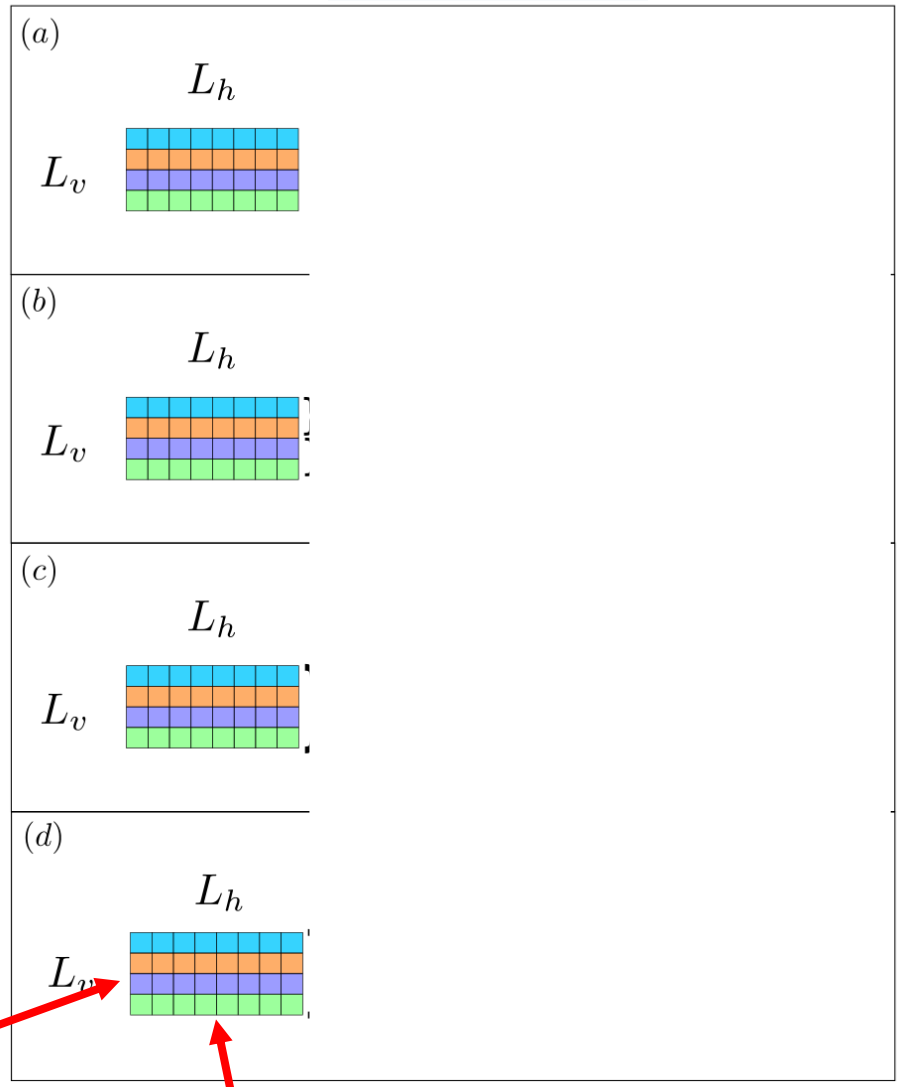
$W_{ji} \rightarrow W'_{ji}$

(1,2)-tiling

(2,2)-tiling

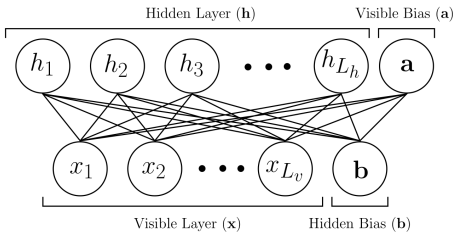
(L,2)-tiling

(L,4)-tiling



Visible nodes

Hidden nodes



W_{ji}

Restricted Boltzmann machines and transfer learning

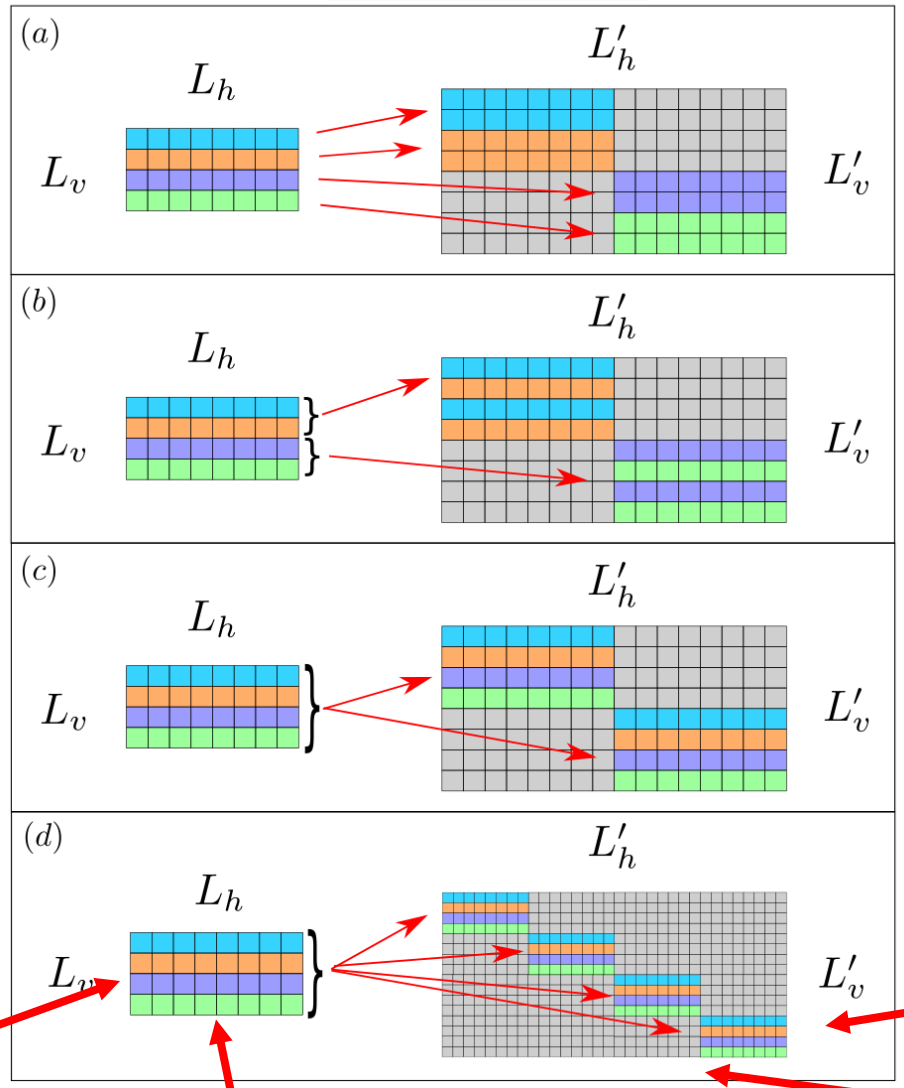
$W_{ji} \rightarrow W'_{ji}$

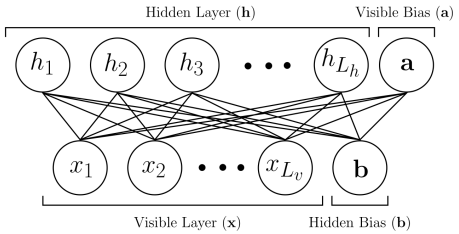
(1,2)-tiling

(2,2)-tiling

(L,2)-tiling

(L,4)-tiling





W_{ji}

Visible nodes

Hidden nodes

Visible nodes

Hidden nodes

Restricted Boltzmann machines and transfer learning

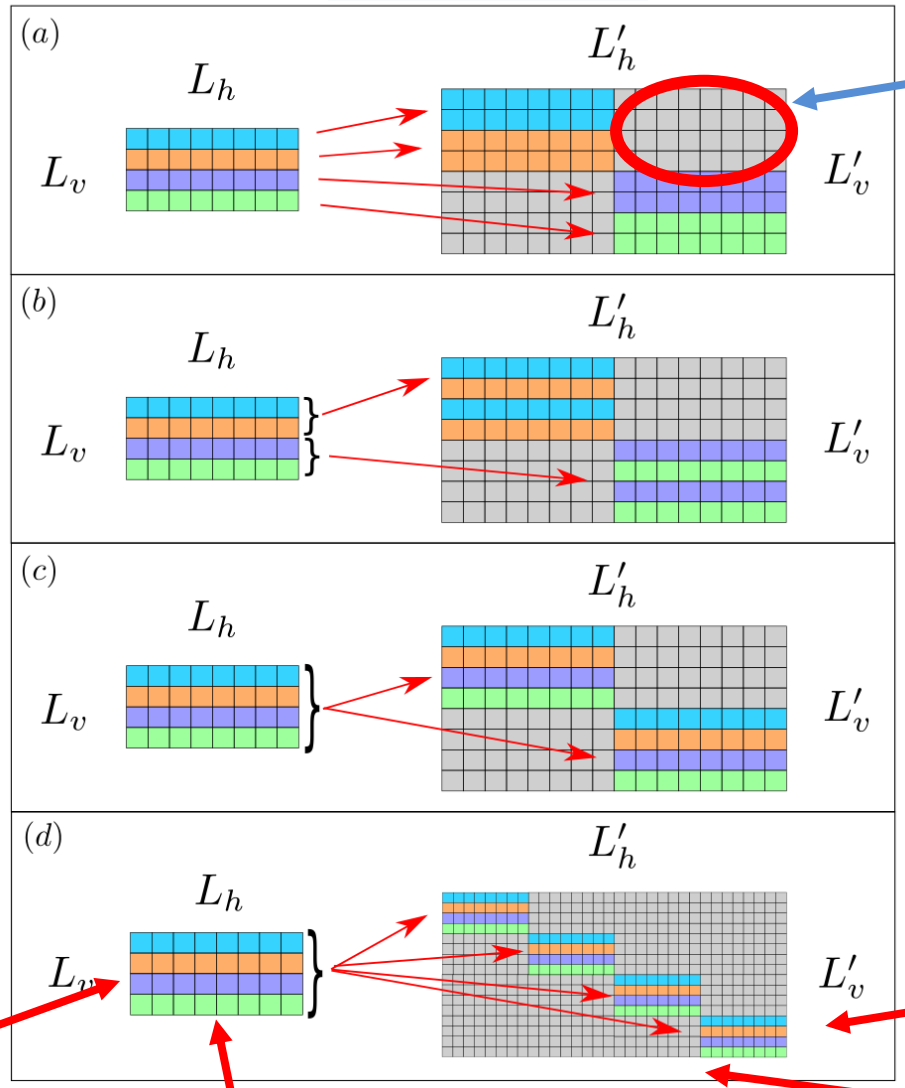
$W_{ji} \rightarrow W'_{ji}$

(1,2)-tiling

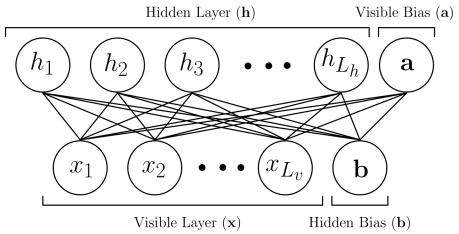
(2,2)-tiling

(L,2)-tiling

(L,4)-tiling



Set up randomly from Gaussian distribution



W_{ji}

Visible nodes

Hidden nodes

Visible nodes

Hidden nodes

Restricted Boltzmann machines and transfer learning

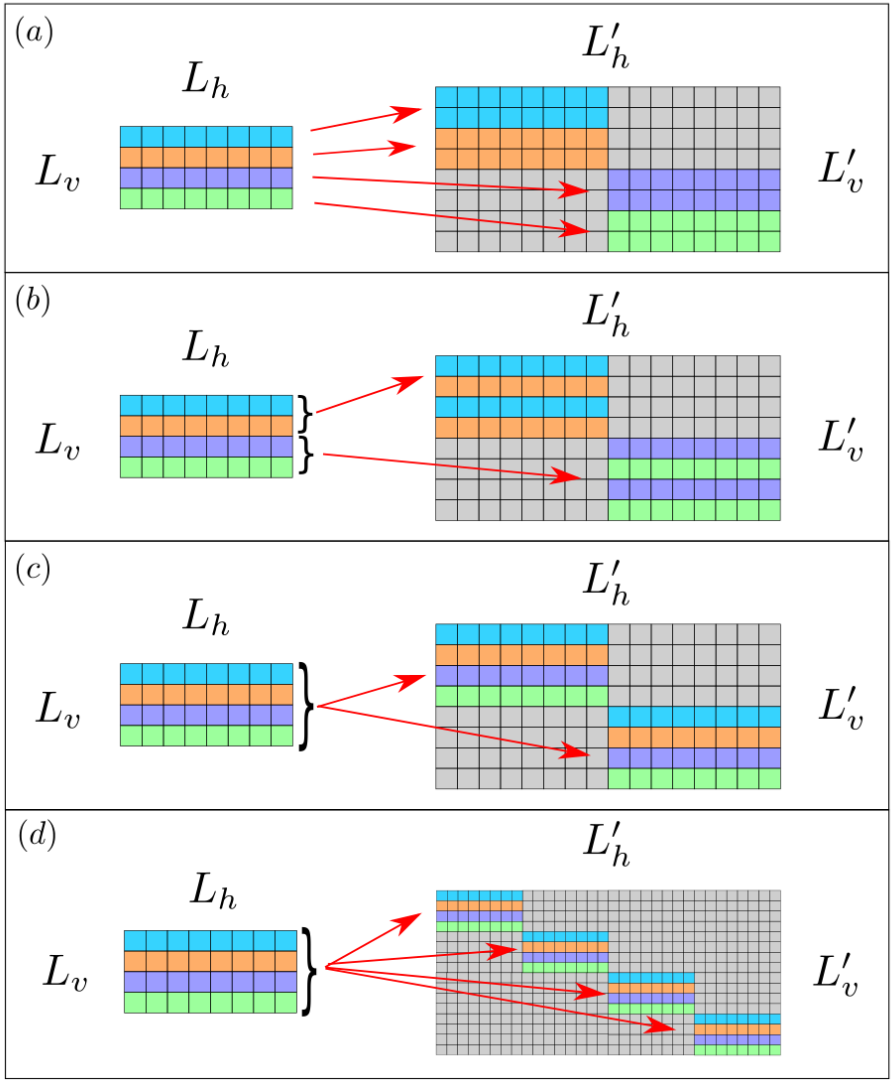
$W_{ji} \rightarrow W'_{ji}$

(1,2)-tiling

(2,2)-tiling

(L,2)-tiling

(L,4)-tiling



Roughly a bias of this type

$|\uparrow\uparrow\downarrow\downarrow\rangle \rightarrow |\uparrow\uparrow\uparrow\uparrow\downarrow\downarrow\downarrow\downarrow\rangle$

$|\uparrow\downarrow\uparrow\downarrow\rangle \rightarrow |\uparrow\uparrow\downarrow\downarrow\uparrow\uparrow\downarrow\downarrow\rangle$

$|\uparrow\uparrow\downarrow\downarrow\rangle \rightarrow |\uparrow\uparrow\ \uparrow\uparrow\ \downarrow\downarrow\ \downarrow\downarrow\rangle$

$|\uparrow\downarrow\uparrow\downarrow\rangle \rightarrow |\uparrow\downarrow\ \uparrow\downarrow\ \uparrow\downarrow\ \uparrow\downarrow\rangle$

Restricted Boltzmann machines and transfer learning

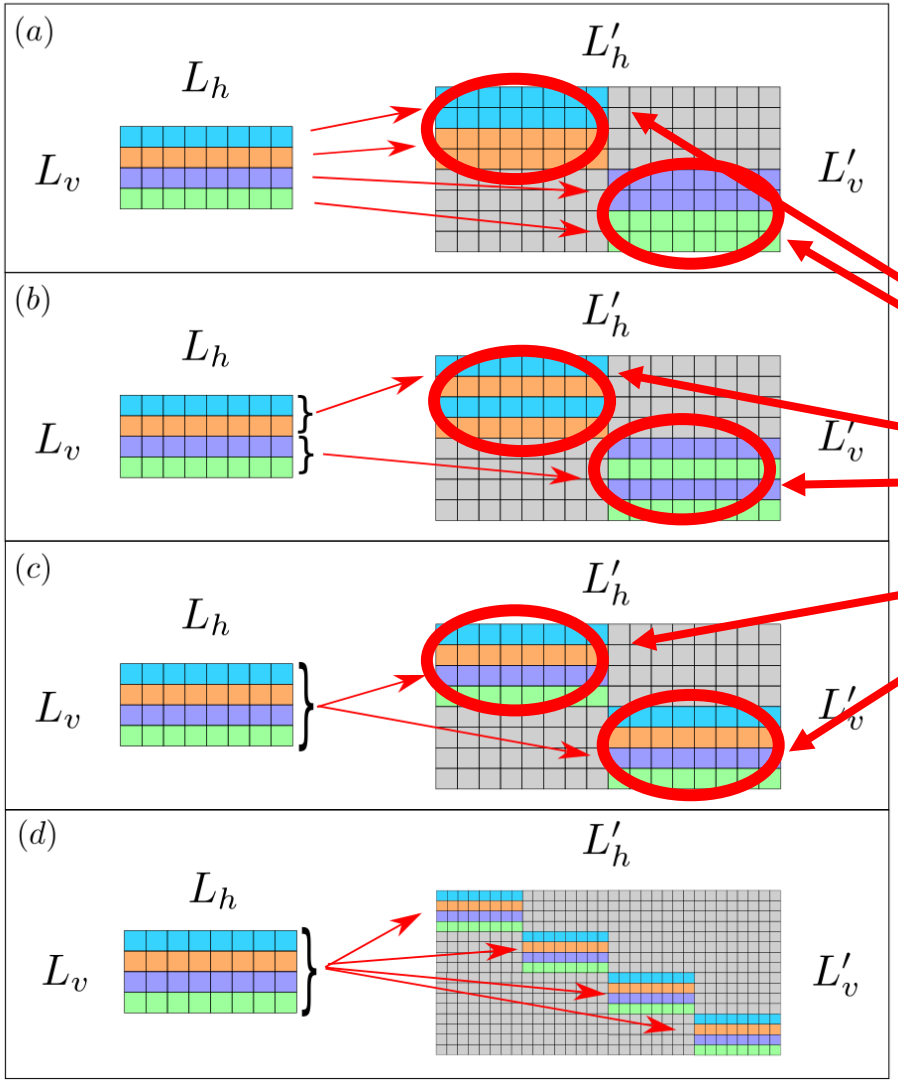
$W_{ji} \rightarrow W'_{ji}$

(1,2)-tiling

(2,2)-tiling

(L,2)-tiling

(L,4)-tiling



Decoupling the halves of the system via the hidden layers enhances significantly the performance.

Restricted Boltzmann machines and transfer learning

How do we implement the algorithm?

- To evaluate the energy and the gradients we take 10^4 samples
- To optimize we use a gradient descent algorithm (adaptive learning rate strategies RMSProp and the learning rate of the RMSProp to 0.001
- For the Ising model, the samples are obtained by a single iteration Gibbs sampling.
- For the Heisenberg XXZ model, we use a Metropolis-Hastings sampling with exchange. Strategy to conserve magnetization.
- We use NVIDIA Titan V GPUs

Restricted Boltzmann machines and transfer learning

How do we implement the algorithm?

- To evaluate the energy and the gradients we take 10^4 samples
- To optimize we use a gradient descent algorithm (adaptive learning rate strategies RMSProp and the learning rate of the RMSProp to 0.001
- For the Ising model, the samples are obtained by a single iteration Gibbs sampling.
- For the Heisenberg XXZ model, we use a Metropolis-Hastings sampling with exchange. Strategy to conserve magnetization.
- We use NVIDIA Titan V GPUs

- We use twice as many hidden layers as visible ones.

Restricted Boltzmann machines and transfer learning

How do we implement the algorithm?

- To evaluate the energy and the gradients we take 10^4 samples
- To optimize we use a gradient descent algorithm (adaptive learning rate strategies RMSProp and the learning rate of the RMSProp to 0.001
- For the Ising model, the samples are obtained by a single iteration Gibbs sampling.
- For the Heisenberg XXZ model, we use a Metropolis-Hastings sampling with exchange. Strategy to conserve magnetization.
- We use NVIDIA Titan V GPUs

- We use twice as many hidden layers as visible ones.

- Stopping criterion. We compute the variance of "local energy" $E_{\mathrm{loc}}(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{\boldsymbol{x}'} \langle \boldsymbol{x} \mid H \mid \boldsymbol{x}' \rangle \frac{\psi(\boldsymbol{x}';\boldsymbol{\theta})}{\psi(\boldsymbol{x};\boldsymbol{\theta})}$

$$\sigma_{E_{loc}} = \sqrt{\sum_{\boldsymbol{x}} (E_{loc}(\boldsymbol{x},\boldsymbol{\theta}) - E(\boldsymbol{\theta}))^2}/\sqrt{N}$$

and we stop when $\sigma_{E_{loc}}/E(\boldsymbol{\theta})$ is less than $\varepsilon_\sigma = 0.005$ or when the number of epochs reaches $\varepsilon_{epoch} = 30,000.$

Restricted Boltzmann machines and transfer learning

We consider two models

$$H_I = -J_I \sum_{\langle l,m \rangle} \sigma_l^z \sigma_m^z - h \sum_l \sigma_l^x$$ Ising

$$H_{XXZ} = -J_{XXZ} \sum_{\langle l,m \rangle} \left(\sigma_l^x \sigma_m^x + \sigma_l^y \sigma_m^y + \Delta \sigma_l^z \sigma_m^z \right)$$ Heisenberg (number conserving)

Ising	For large negative J_I the Ising model is antiferromagnetic (AF)	$\|\uparrow\downarrow\uparrow\downarrow\rangle$
	For large positive J_I the Ising model is ferromagnetic (FM)	$\|\uparrow\uparrow\uparrow\uparrow\rangle$
Heisenberg	For large negative Δ the Heisenberg model is antiferromagnetic (AF)	$\|\uparrow\downarrow\uparrow\downarrow\rangle$
	For large positive Δ the Heisenberg model is ferromagnetic (FM)	$\|\uparrow\uparrow\downarrow\downarrow\rangle$

Restricted Boltzmann machines and transfer learning

We consider two models

$$H_I = -J_I \sum_{\langle l,m \rangle} \sigma_l^z \sigma_m^z - h \sum_l \sigma_l^x$$ Ising

$$H_{XXZ} = -J_{XXZ} \sum_{\langle l,m \rangle} \left(\sigma_l^x \sigma_m^x + \sigma_l^y \sigma_m^y + \Delta \sigma_l^z \sigma_m^z \right)$$ Heisenberg (number conserving)

System sizes $\{4, 8, 16, 32, 64, 128\}$ 1D

$\{2 \times 2, 4 \times 4, 8 \times 8\}$ 2D

Restricted Boltzmann machines and transfer learning

We consider two models

$$H_I = -J_I \sum_{\langle l,m \rangle} \sigma_l^z \sigma_m^z - h \sum_l \sigma_l^x$$

Ising

$$H_{XXZ} = -J_{XXZ} \sum_{\langle l,m \rangle} \left(\sigma_l^x \sigma_m^x + \sigma_l^y \sigma_m^y + \Delta \sigma_l^z \sigma_m^z \right)$$

Heisenberg
(number conserving)

We evaluate

- Efficiency: how quickly the ground state is reached

- Effectiveness: how well the ground state is represented

Restricted Boltzmann machines and transfer learning

We consider two models

$$H_I = -J_I \sum_{\langle l,m \rangle} \sigma_l^z \sigma_m^z - h \sum_l \sigma_l^x$$

Ising

$$H_{XXZ} = -J_{XXZ} \sum_{\langle l,m \rangle} \left(\sigma_l^x \sigma_m^x + \sigma_l^y \sigma_m^y + \Delta \sigma_l^z \sigma_m^z \right)$$

Heisenberg
(number conserving)

For efficiency we measure time.
For effectiveness we look at a number of properties and we compare to accurate results from Matrix Product States simulations.

$$C_{l,m}^z = \sum_{\boldsymbol{x}} p(\boldsymbol{x}) \langle \boldsymbol{x} | \sigma_l^z \sigma_m^z | \boldsymbol{x} \rangle.$$

Local magnetization

Antiferromagnetic order

$$C_d^A = \frac{1}{d-1} \sum_{l=2}^{d} (-1)^{l-1} C_{1,l}^z,$$

$$C_d^F = \frac{1}{d-1} \sum_{l=2}^{d} C_{1,l}^z$$

Ferromagnetic order

Restricted Boltzmann machines and transfer learning

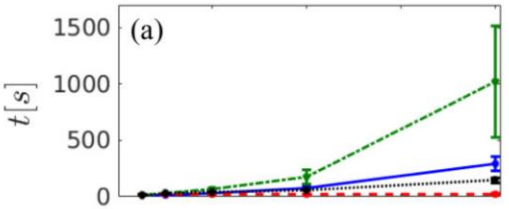
We compute the **time** required to reach the stopping criterion, adding the time required at previous steps.

Restricted Boltzmann machines and transfer learning

We compute the **time** required to reach the stopping criterion, adding the time required at previous steps.

Ising

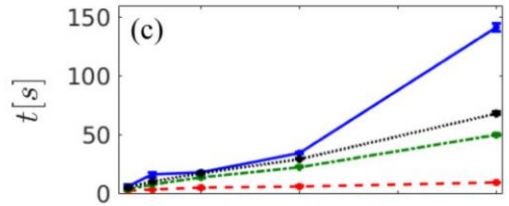
Antiferro



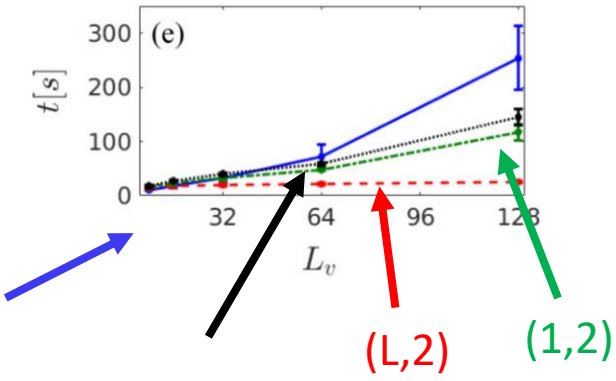
$|\uparrow\downarrow\uparrow\downarrow\rangle$

$|\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\rangle$

Gapless



Ferro



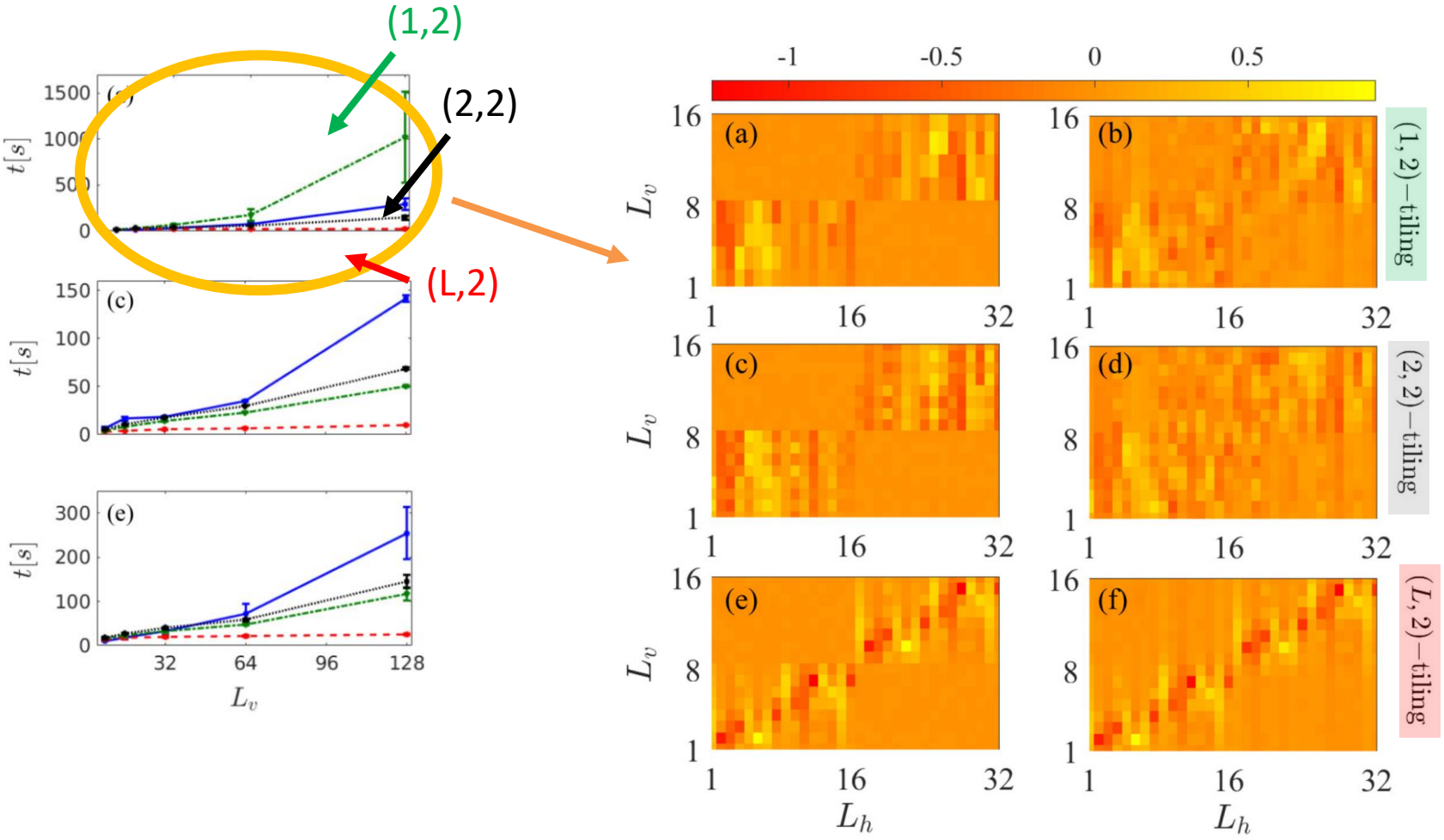
$|\uparrow\uparrow\uparrow\uparrow\rangle$

$|\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\rangle$

Restricted Boltzmann machines and transfer learning

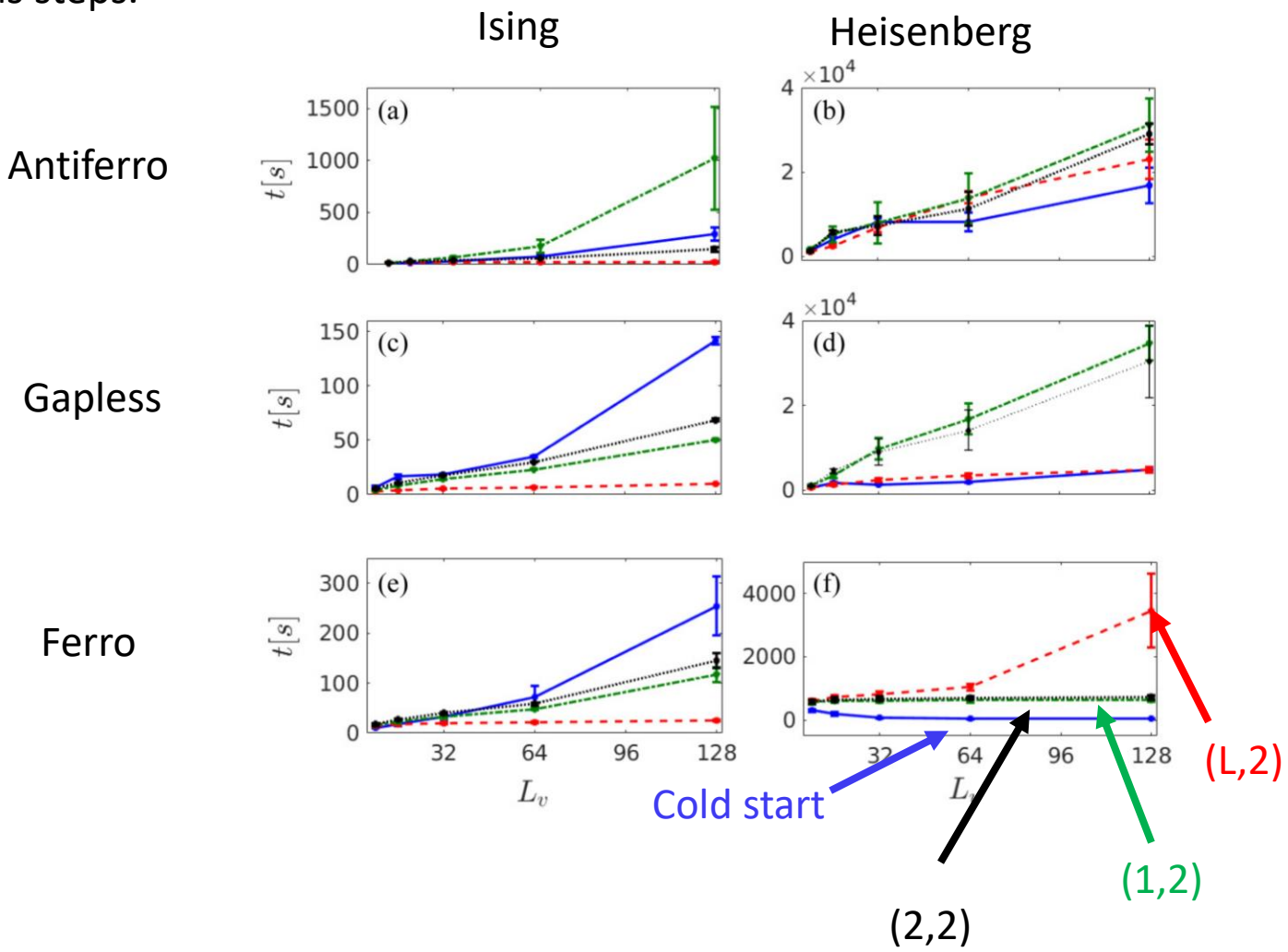
Let us look at the details of why one transfer learning technique is faster than the other.

We consider only Ising and we compare (1,2) vs (L,2) tiling in the ferromagmetic phase.



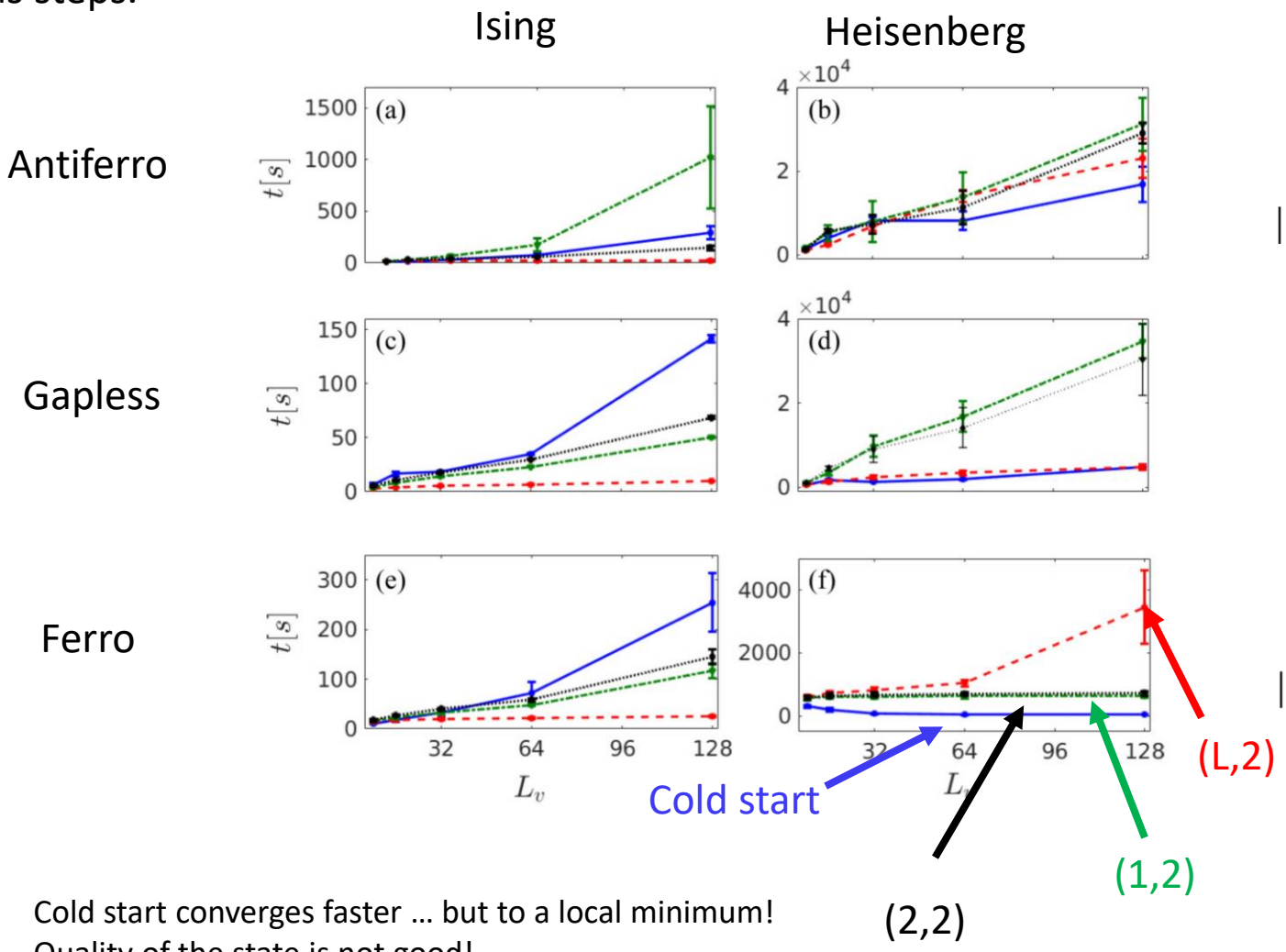
Restricted Boltzmann machines and transfer learning

We compute the **time** required to reach the stopping criterion, adding the time required at previous steps.



Restricted Boltzmann machines and transfer learning

We compute the **time** required to reach the stopping criterion, adding the time required at previous steps.



Cold start converges faster … but to a local minimum!
Quality of the state is not good!

Restricted Boltzmann machines and transfer learning

How good are the states generated?

How close is the energy from the matrix product states value?

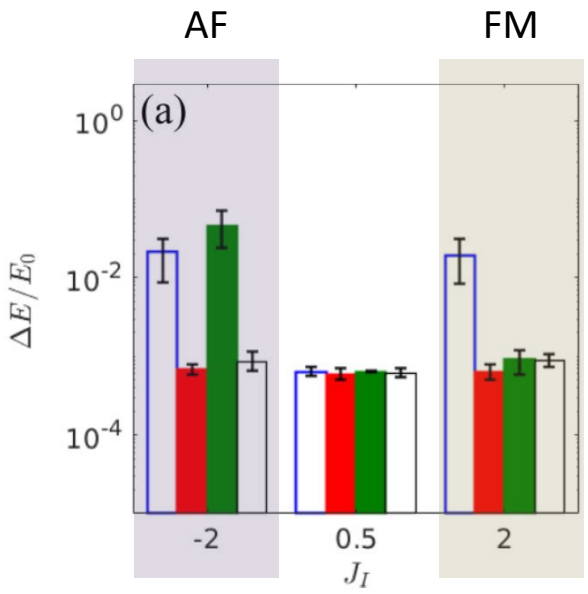
We consider a state for 64 spins and we double its size with different transfer learning techniques and then we do **comparisons to Matrix Product States** results.
For a fair comparison, **we consider the same number of epochs for all transfer techniques**.

cold start

(L,2)

(2,2)

(1,2)



Ising

Restricted Boltzmann machines and transfer learning

How good are the states generated?

How close is the energy from the matrix product states value?

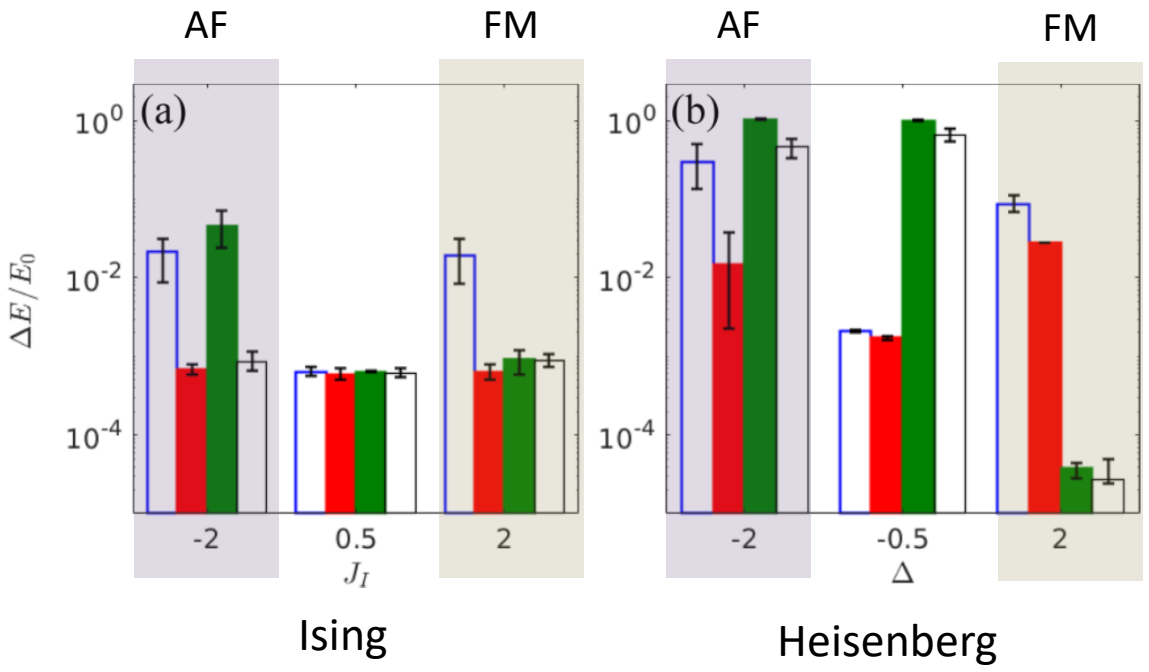
We consider a state for 64 spins and we double its size with different transfer learning techniques and then we do **comparisons to Matrix Product States** results.
For a fair comparison, **we consider the same number of epochs for all transfer techniques**.

cold start

(L,2)

(2,2)

(1,2)



Restricted Boltzmann machines and transfer learning

How good are the states generated?

How close is the energy from the matrix product states value?

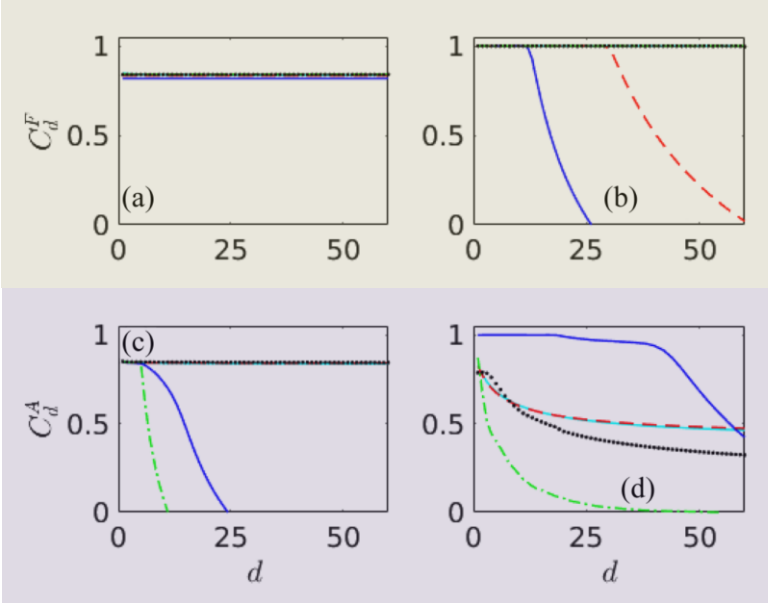
We consider a state for 64 spins and we double its size with different transfer learning techniques and then we do **comparisons to Matrix Product States** results.
For a fair comparison, **we consider the same number of epochs for all transfer techniques**.

cold start

(L,2)

(2,2)

(1,2)



Ising Heisenberg

Not only the energy, but also the correlations can be predicted well when using the appropriate transfer learning

Restricted Boltzmann machines and transfer learning

How good are the states generated?

How close is the energy from the matrix product states value?

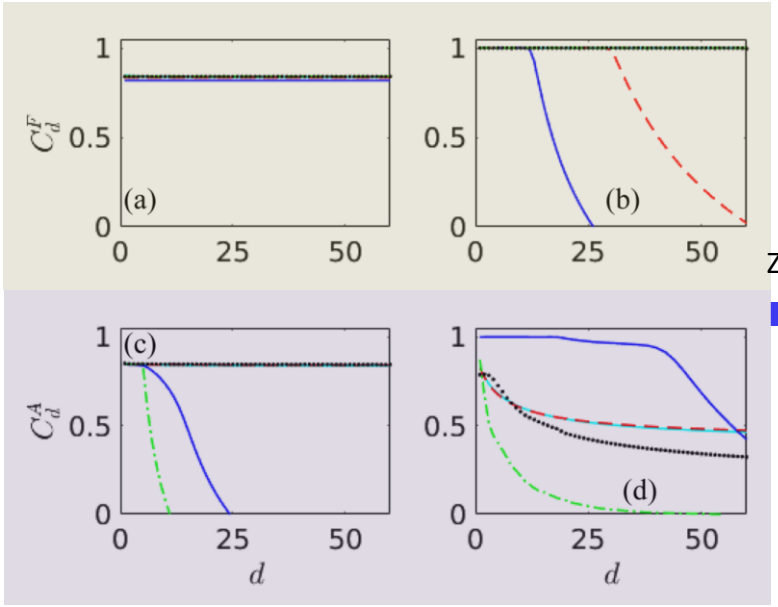
We consider a state for 64 spins and we double its size with different transfer learning techniques and then we do **comparisons to Matrix Product States** results.
For a fair comparison, **we consider the same number of epochs for all transfer techniques**.

cold start

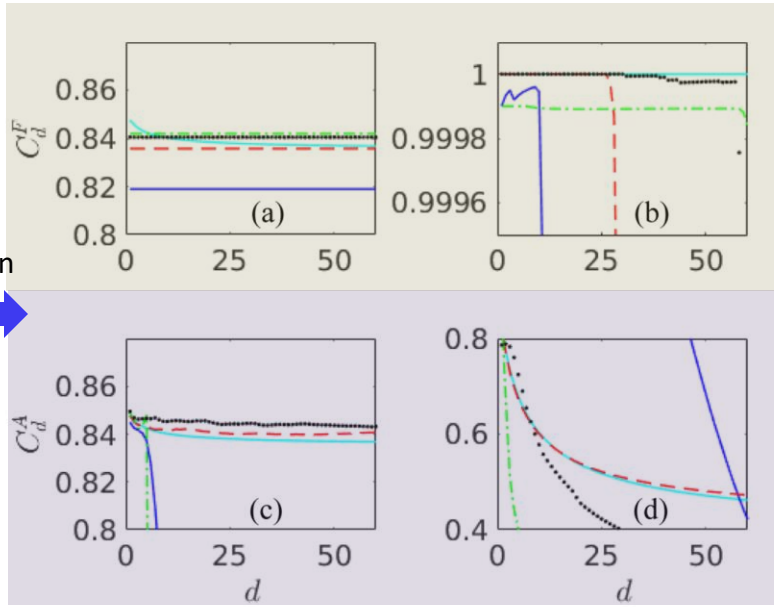
(L,2)

(2,2)

(1,2)



Zoom in



Ising

Heisenberg

Plan of the presentation

1) **Neural Network Quantum States**

2) **Restricted Boltzmann machines**

3) **Restricted Boltzmann machines and transfer learning**

4) **Conclusions and Outlook**

Conclusions and outlook

Transfer learning seems to provide an effective and efficient way to study larger quantum many-body systems. What about frustrated Hamiltonians?

Zen et al., Physical Review E 2020

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

MajuLab

Thank you!

Support

Ministry of Education SINGAPORE

NSCC SINGAPORE National Supercomputing Centre

Conclusions and outlook

Transfer learning seems to provide an effective and efficient way to study larger quantum many-body systems. What about frustrated Hamiltonians?

Zen et al., Physical Review E 2020

Many lines of research at the interface between machine learning and many-body physics. A lot is yet to be done

- What are the limits of efficiency and effectiveness in finding ground states?
- Are NNQS the way to go for 2D and 3D systems?
- What are the limits of time evolution?
- Can we use more physically-inspired architectures and optimizers so as to improve the performance?
- Can physics teach us how to do better neural networks for any sort of problems?
- Can physics help us to interpret neural networks better?

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

MajuLab

Thank you!

Support

Ministry of Education SINGAPORE

SG NSCC SINGAPORE National Supercomputing Centre