A Video from Google DeepMind

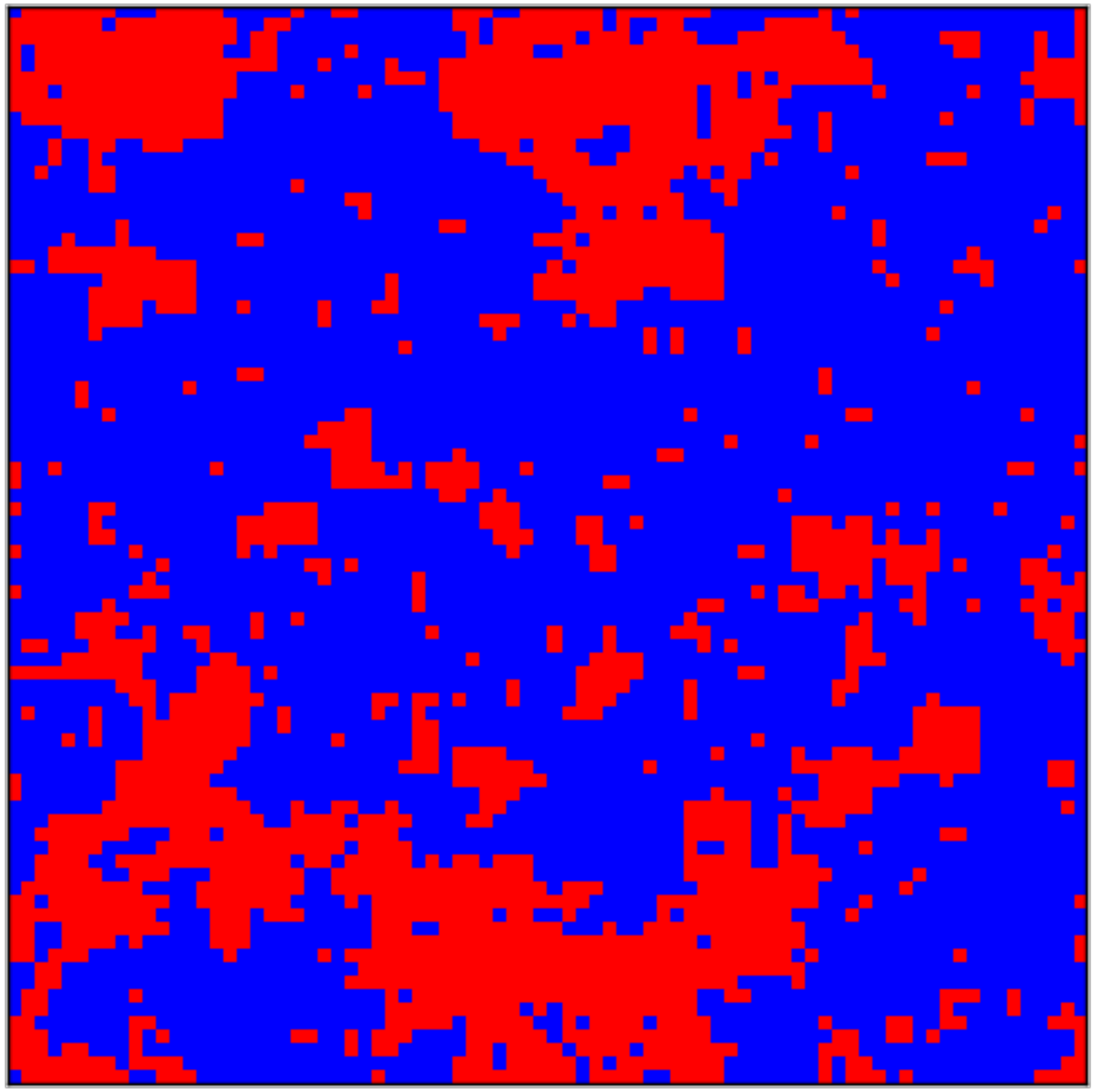
http://www.nature.com/nature/journal/v518/n7540/fig_tab/nature14236_SV2.html

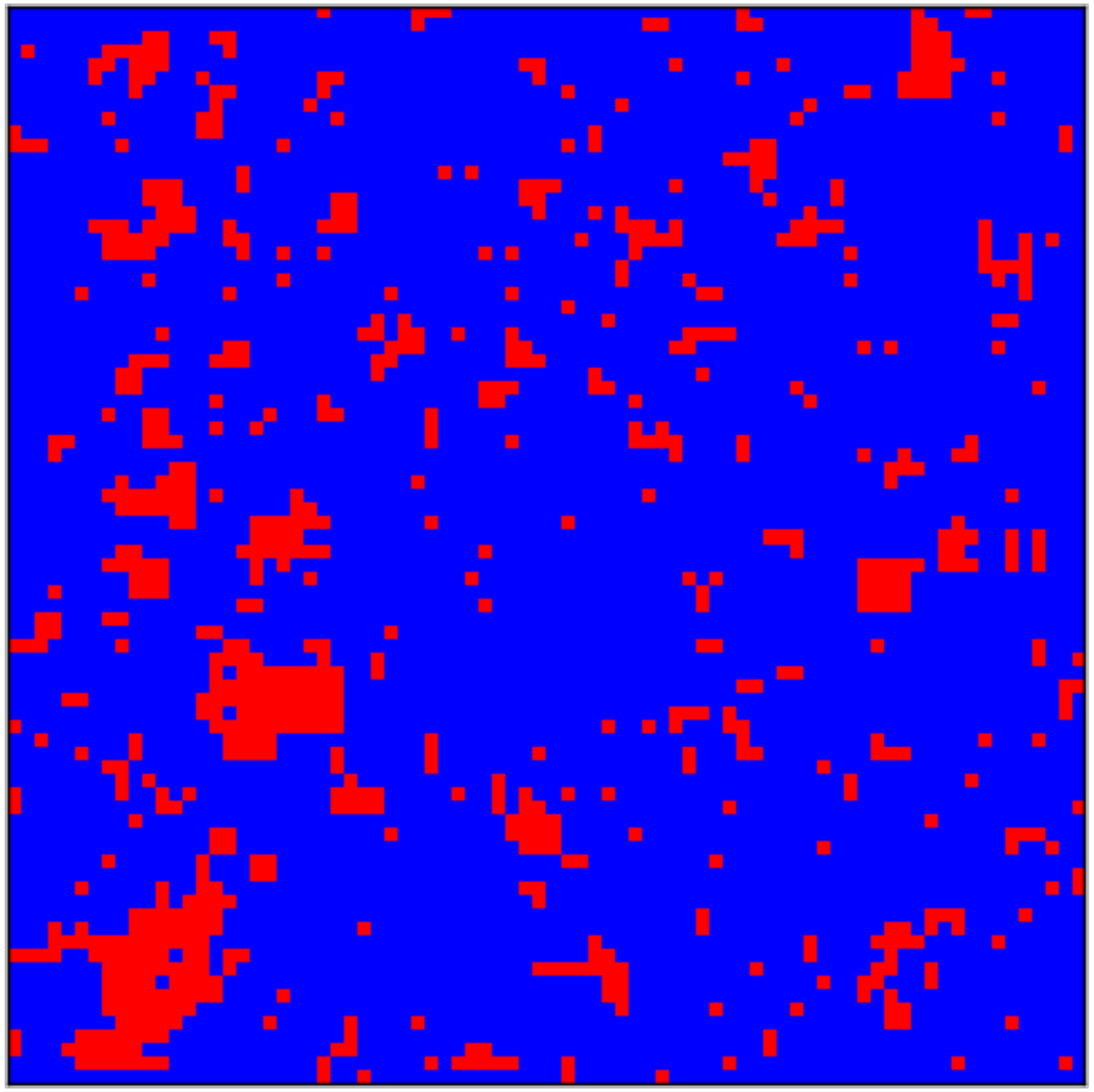
Can machine learning teach us cluster updates ?

Lei Wang
Institute of Physics, CAS
https://wangleiphy.github.io

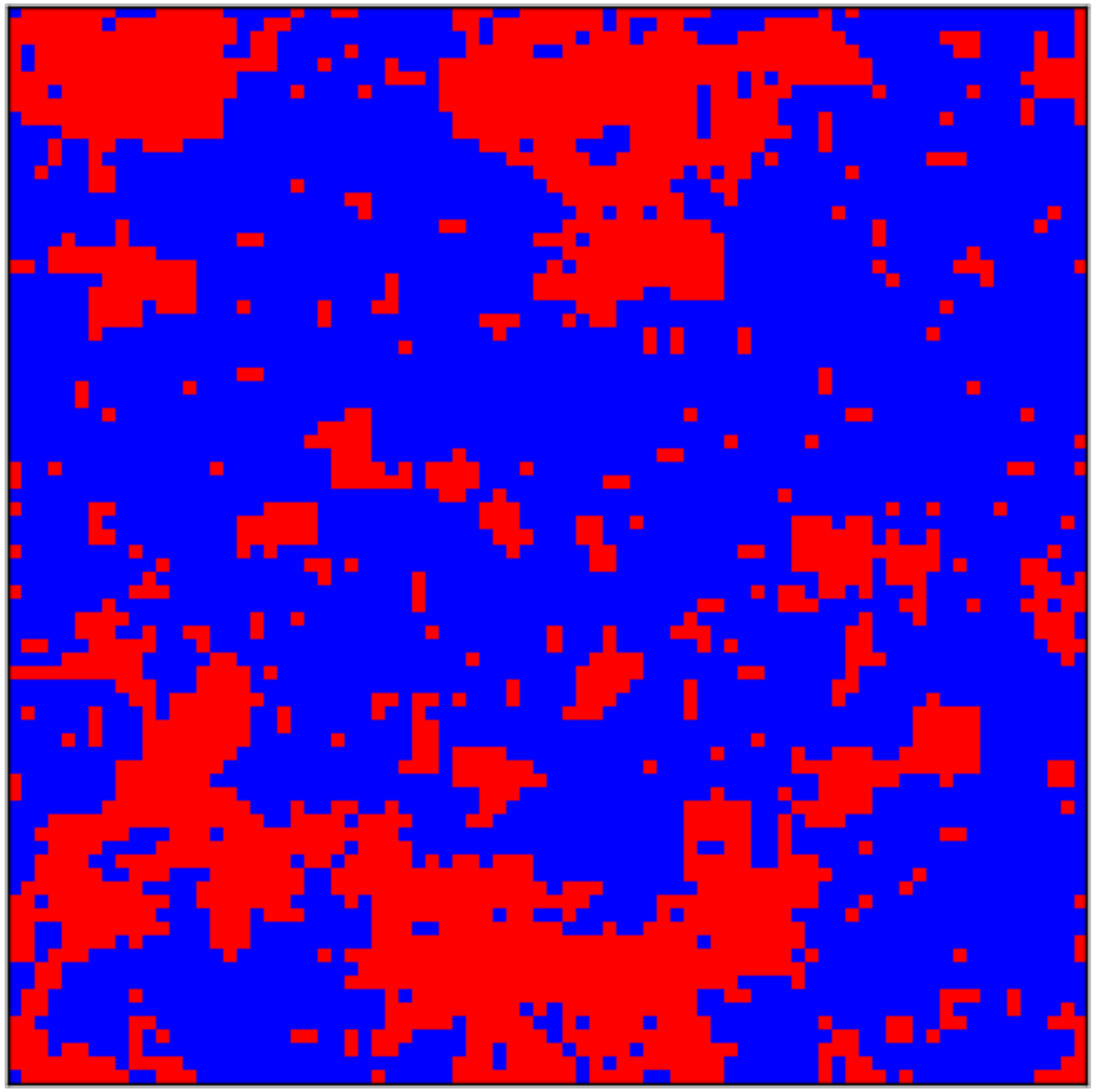
Li Huang and LW, 1610.02746
LW, 1702.08586

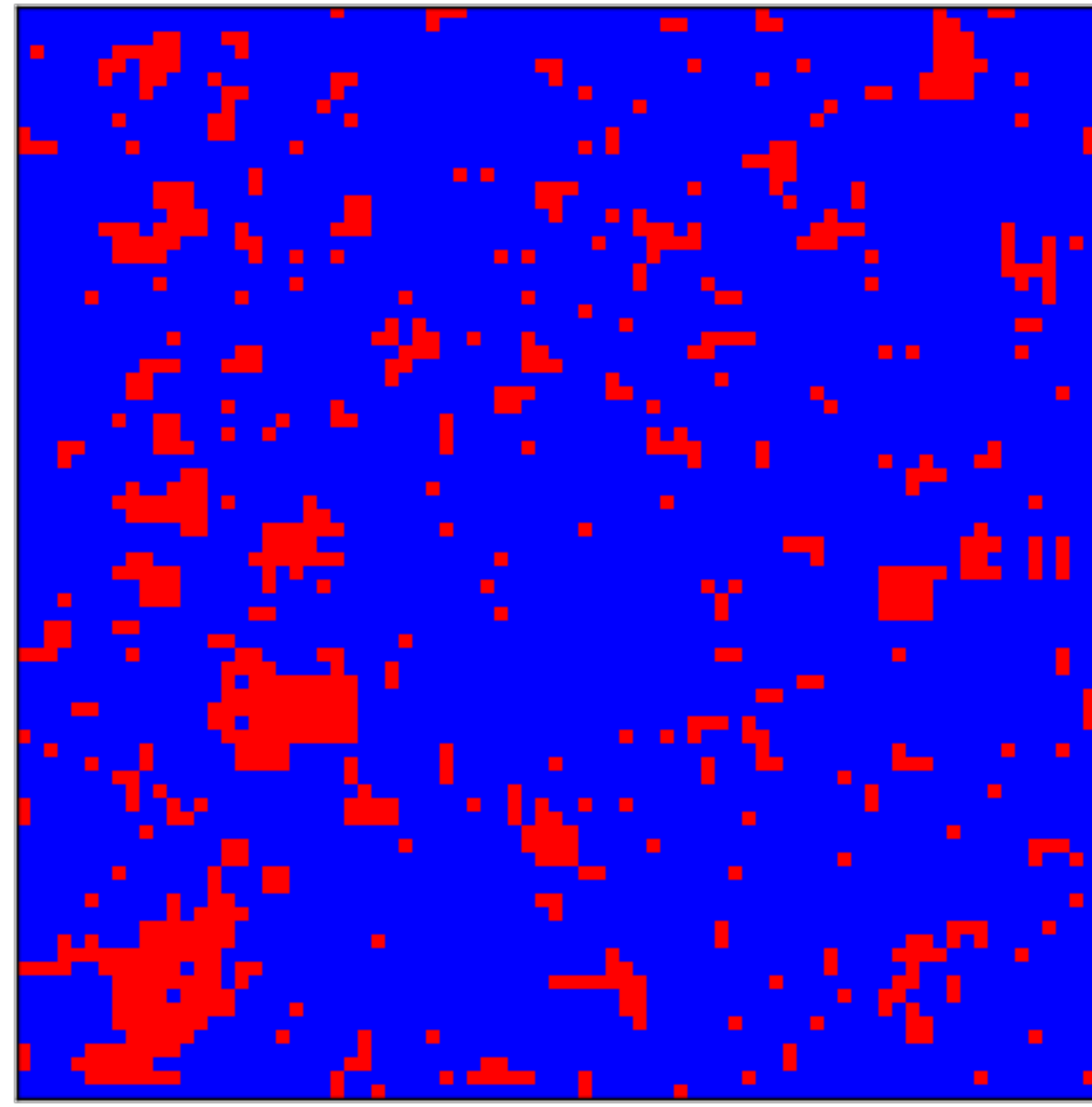
Local vs Cluster algorithms





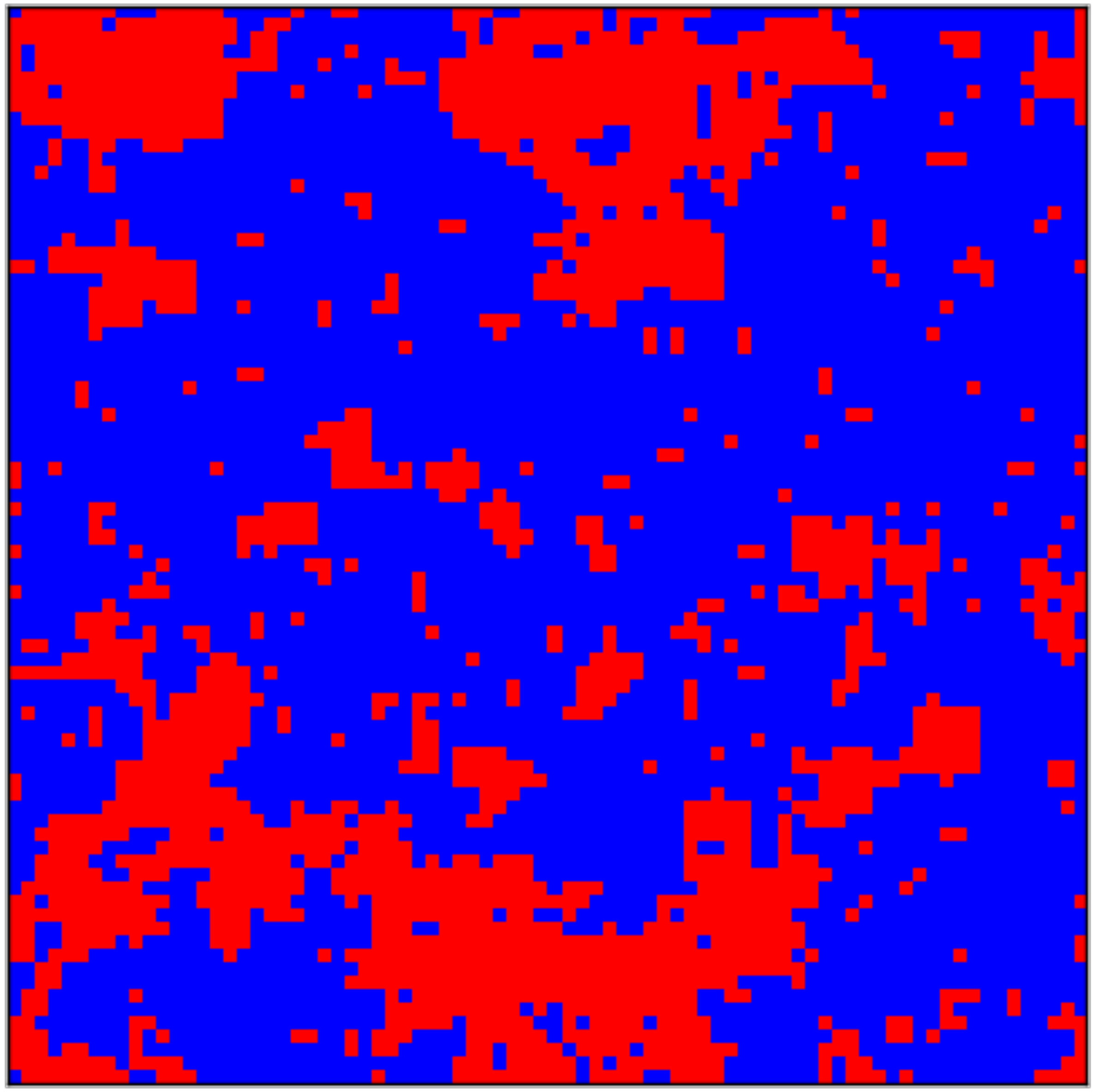
Local vs Cluster algorithms

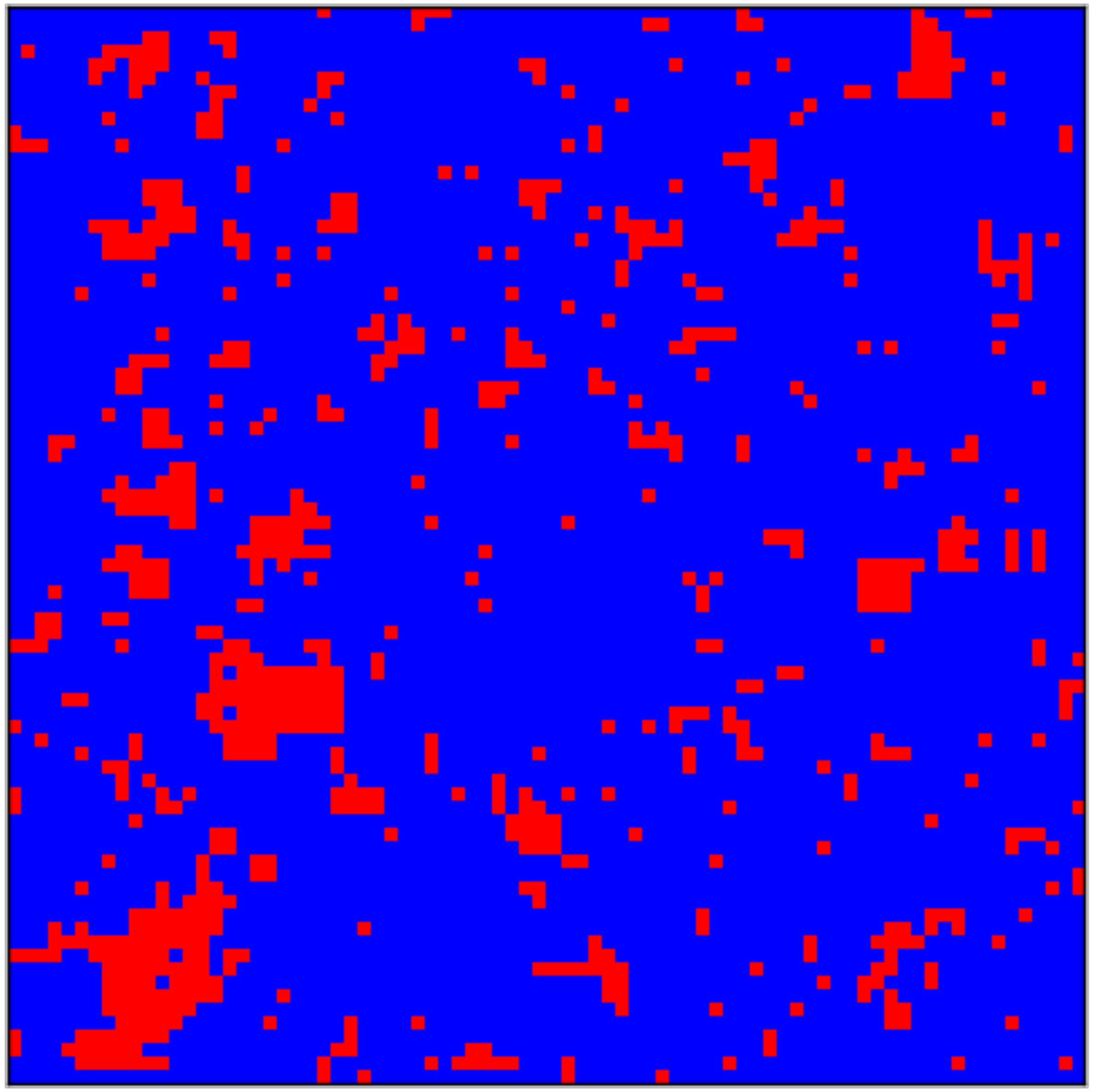




is slower than

Local vs Cluster algorithms





Algorithmic innovation outperforms Moore's law!

Recommender Systems

NETFLIX

amazon

Learn preferences

Recommendations

Recommender Systems

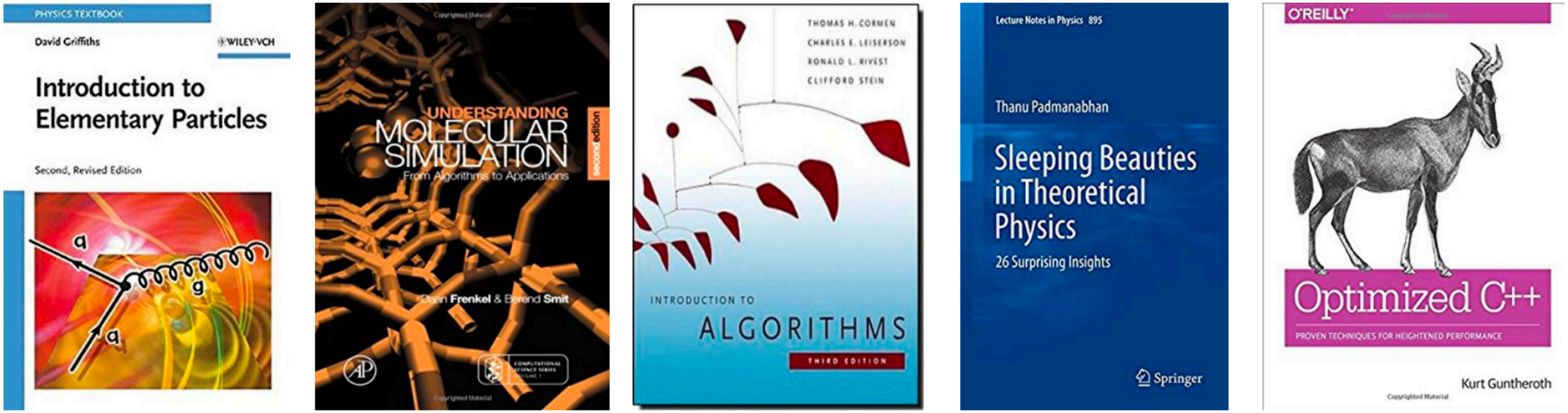
NETFLIX

amazon

Learn preferences

Recommendations

Recommendations for you in Books



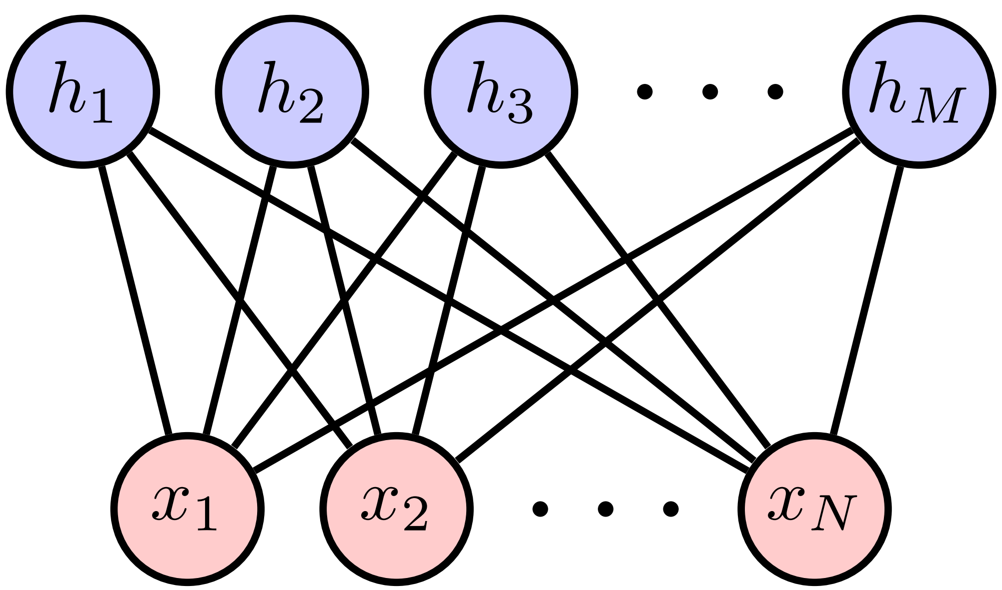
Restricted Boltzmann Machine

Smolensky 1986 Hinton and Sejnowski 1986

$$E(\mathbf{x}, \mathbf{h}) = -\sum_{i=1}^{N} a_i x_i - \sum_{j=1}^{M} b_j h_j - \sum_{i=1}^{N} \sum_{j=1}^{M} x_i W_{ij} h_j$$

Energy based model

$\mathbf{h} \in \{0,1\}^M$



$\mathbf{x} \in \{0,1\}^N$

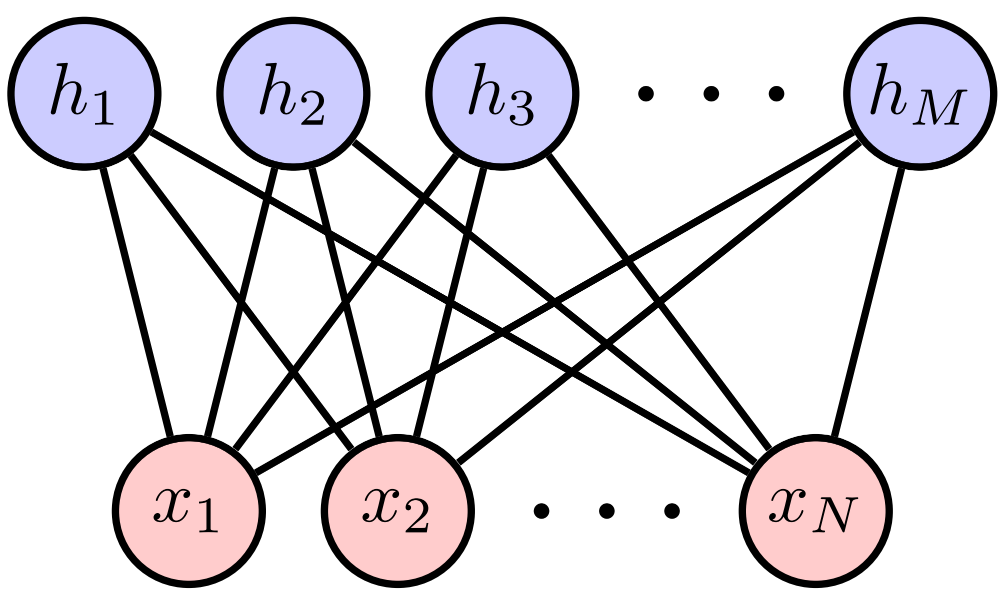
Restricted Boltzmann Machine

Smolensky 1986 Hinton and Sejnowski 1986

$$E(\mathbf{x}, \mathbf{h}) = -\sum_{i=1}^{N} a_i x_i - \sum_{j=1}^{M} b_j h_j - \sum_{i=1}^{N} \sum_{j=1}^{M} x_i W_{ij} h_j$$

Energy based model

$\mathbf{h} \in \{0,1\}^M$



$\mathbf{x} \in \{0,1\}^N$

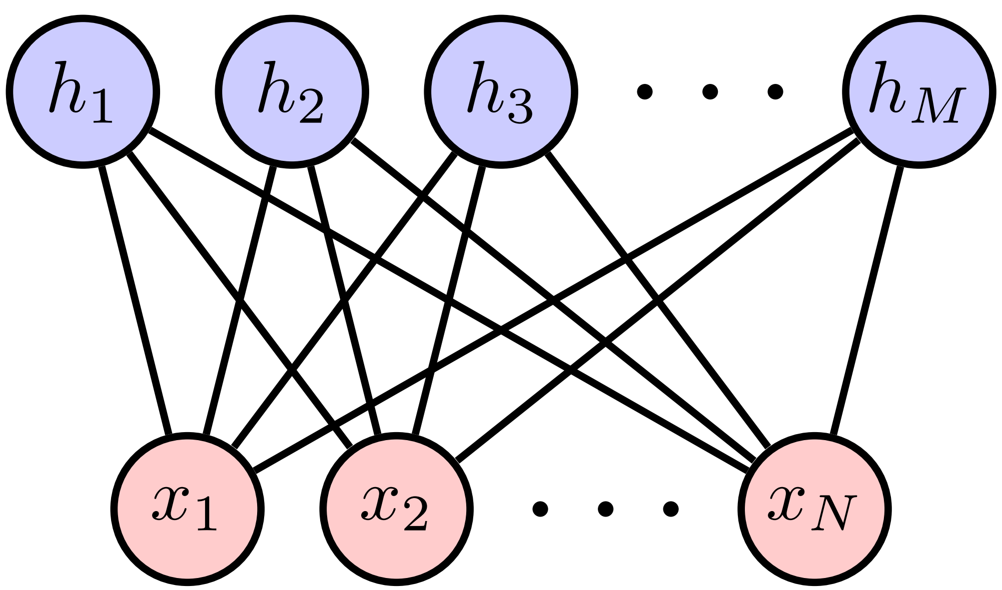
Restricted Boltzmann Machine

Smolensky 1986 Hinton and Sejnowski 1986

$$E(\mathbf{x}, \mathbf{h}) = -\sum_{i=1}^{N} \boxed{a_i} x_i - \sum_{j=1}^{M} \boxed{b_j} h_j - \sum_{i=1}^{N} \sum_{j=1}^{M} x_i \boxed{W_{ij}} h_j$$

Energy based model

$\mathbf{h} \in \{0, 1\}^M$



$\mathbf{x} \in \{0, 1\}^N$

Probabilities

$$p(\mathbf{x}, \mathbf{h}) \sim e^{-E(\mathbf{x}, \mathbf{h})}$$

$$p(\mathbf{x}) = \sum_{\mathbf{h}} p(\mathbf{x}, \mathbf{h})$$

$$p(\mathbf{h}|\mathbf{x}) = p(\mathbf{x}, \mathbf{h}) / p(\mathbf{x})$$

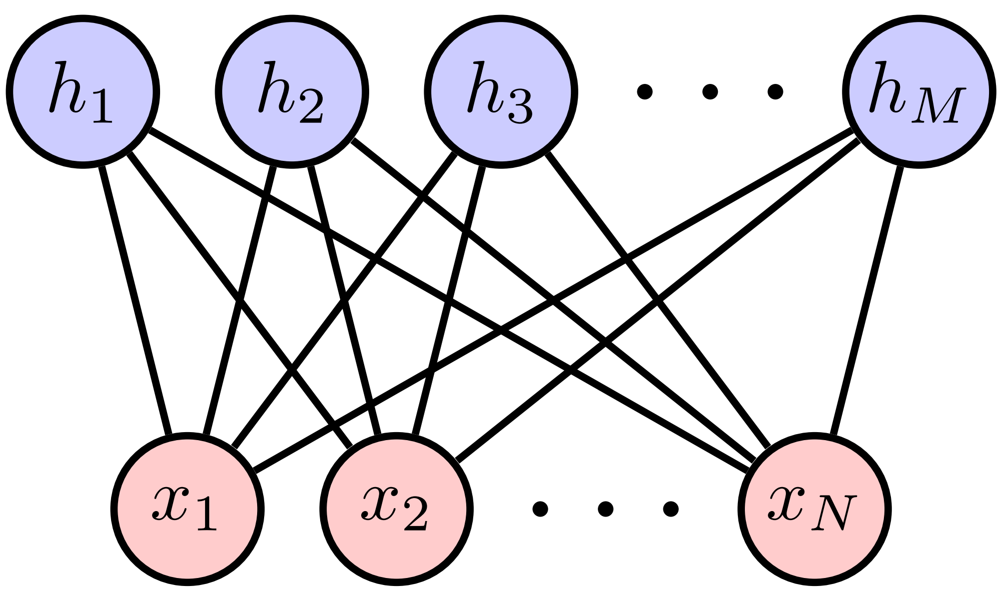
Restricted Boltzmann Machine

Smolensky 1986 Hinton and Sejnowski 1986

$$E(\mathbf{x},\mathbf{h}) = -\sum_{i=1}^{N} \boxed{a_i} x_i - \sum_{j=1}^{M} \boxed{b_j} h_j - \sum_{i=1}^{N}\sum_{j=1}^{M} x_i \boxed{W_{ij}} h_j$$

Energy based model

$\mathbf{h} \in \{0,1\}^M$



$\mathbf{x} \in \{0,1\}^N$

Probabilities

$$p(\mathbf{x},\mathbf{h}) \sim e^{-E(\mathbf{x},\mathbf{h})}$$

$$p(\mathbf{x}) = \sum_{\mathbf{h}} p(\mathbf{x},\mathbf{h})$$

$$p(\mathbf{h}|\mathbf{x}) = p(\mathbf{x},\mathbf{h})/p(\mathbf{x})$$

Universal approximator of probability distributions

. Freund and Haussler, 1989 Le Roux and Bengio, 2008

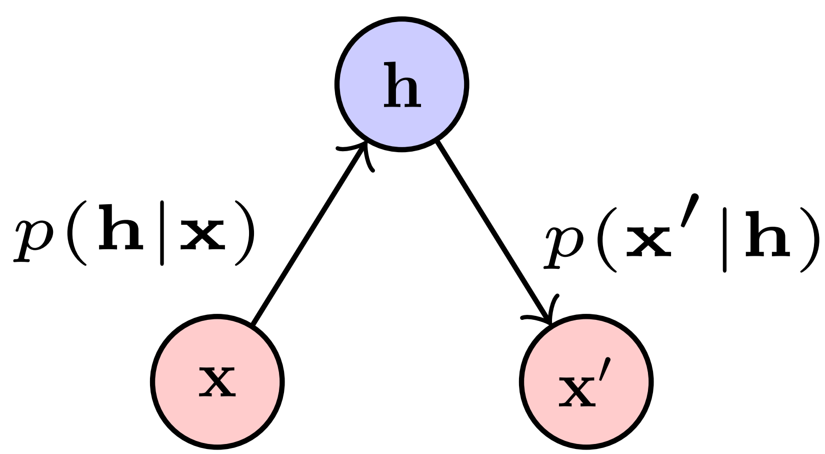
Generative Learning

Learning: Fit to the target distribution

$$p(\mathbf{x}) \sim \pi(\mathbf{x})$$ target probability

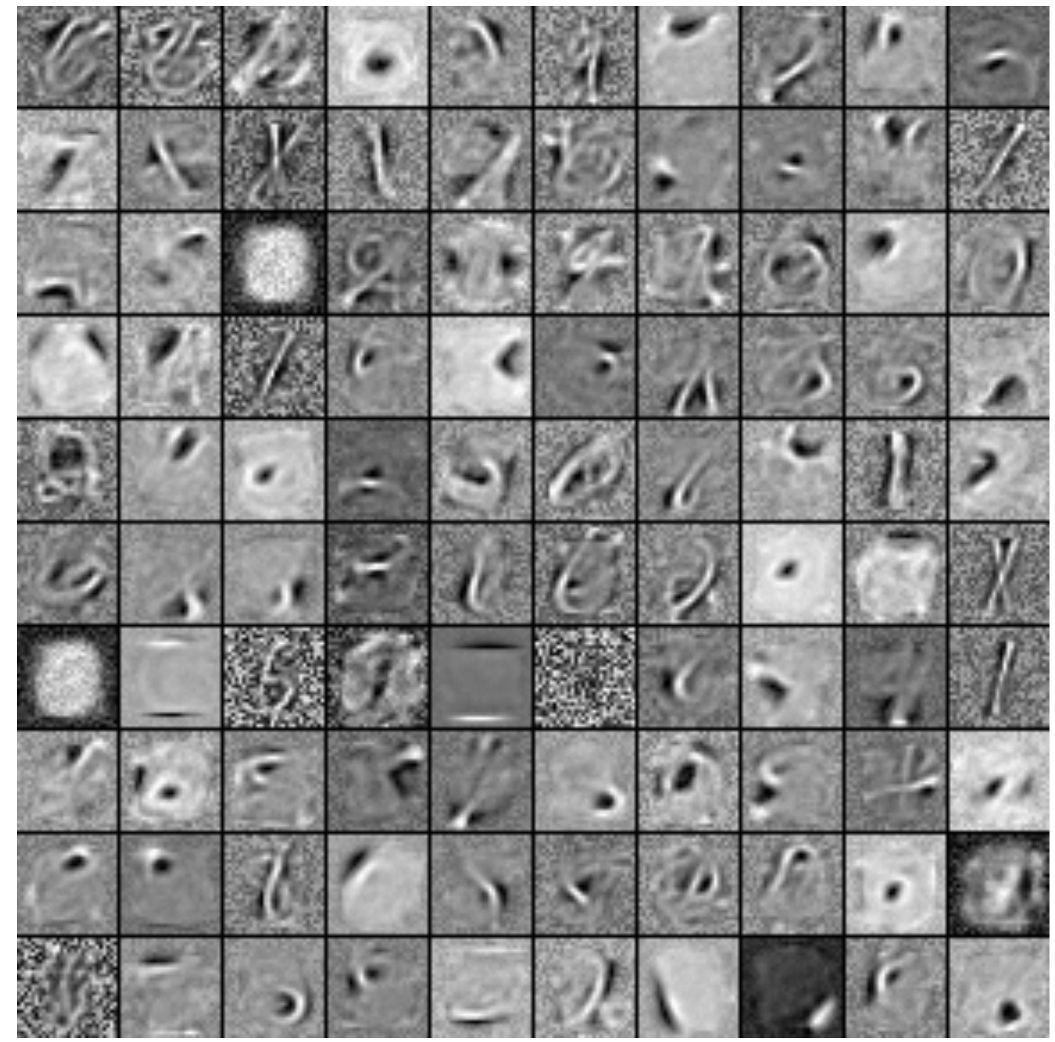
Learn the model from data Hinton 2002

Generating: Blocked Gibbs sampling



Generate more data from the learned model

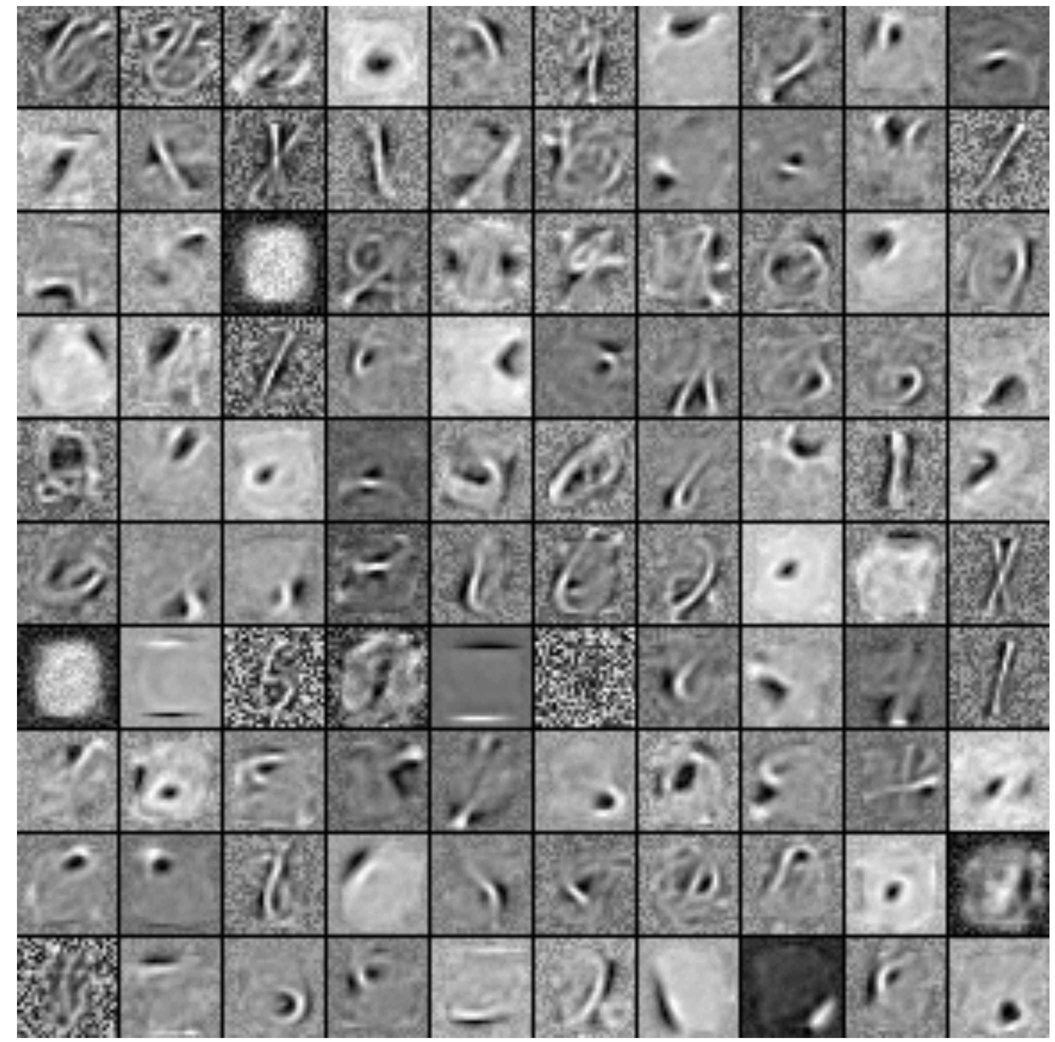
MNIST database
of handwritten digits



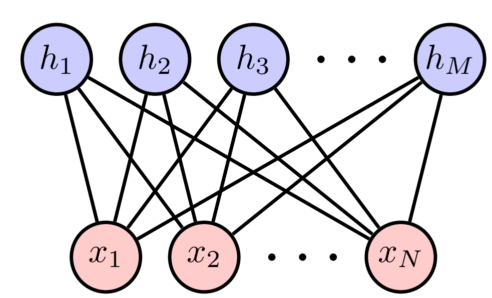
learned weight W_{ij}

Theano deep learning tutorial http://www.deeplearning.net/tutorial/rbm.html#rbm

MNIST database
of handwritten digits



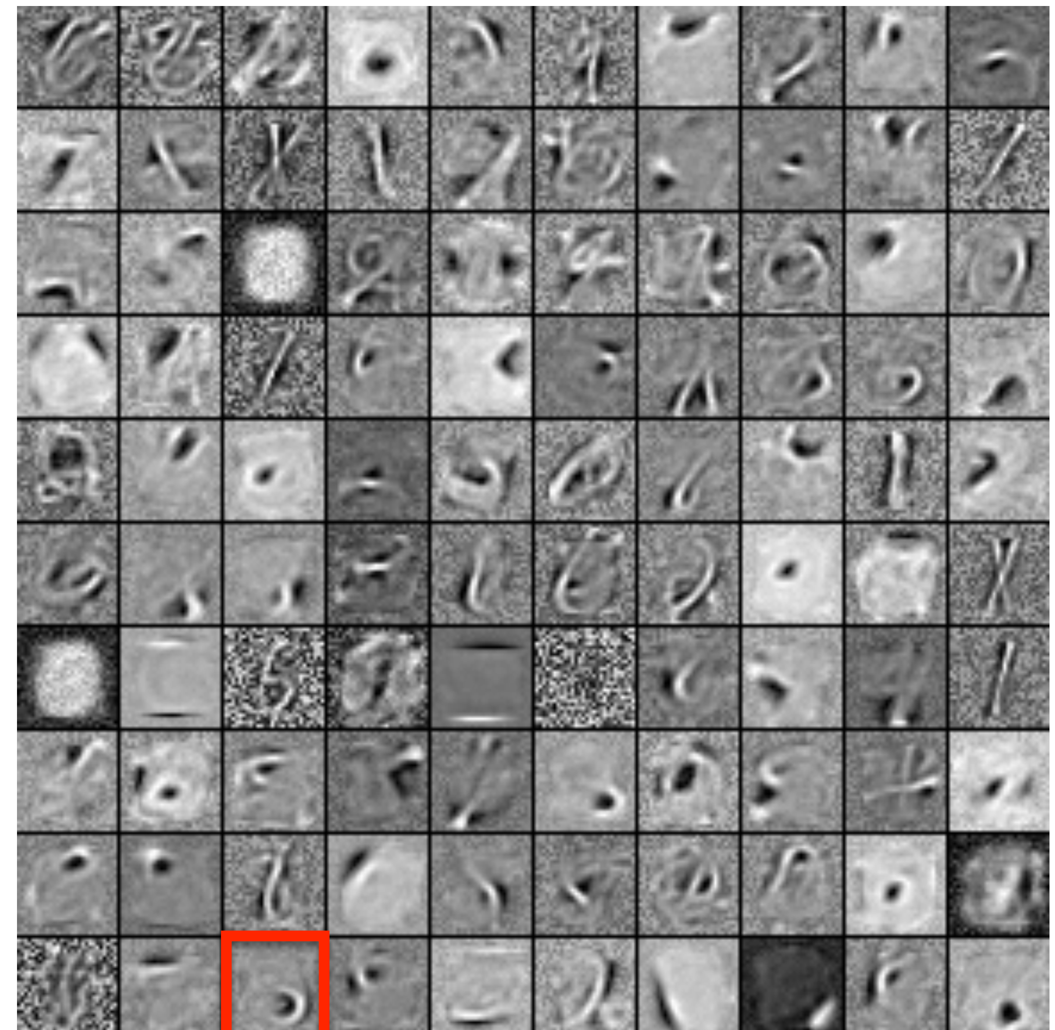
learned weight W_{ij}



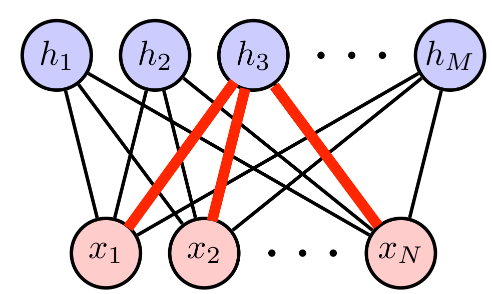
784 100

Theano deep learning tutorial http://www.deeplearning.net/tutorial/rbm.html#rbm

MNIST database
of handwritten digits



learned weight W_{ij}



784 100

Theano deep learning tutorial http://www.deeplearning.net/tutorial/rbm.html#rbm

Challenge

Which one is written by human ?

Salakhutdinov, 2015 Deep learning summer school

Idea

Model the QMC data with an RBM, then sample from the RBM

Li Huang and LW, 1610.02746

cf. Liu, Qi, Meng, Fu,1610.03137

Liu, Shen, Qi, Meng, Fu,1611.09364

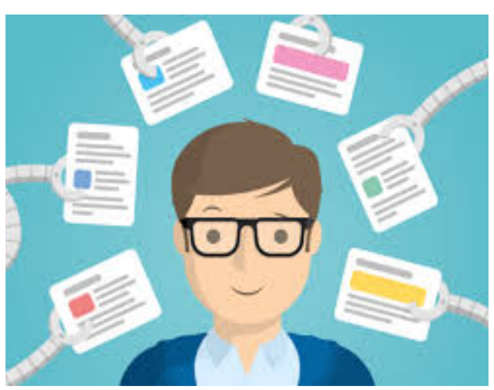
Xu, Qi, Liu, Fu, Meng,1612.03804

Why is it useful ?

- When the fitting is perfect, we can completely bypass the “quantum” part of the QMC

cf. Torlai, Melko,1606.02718

- Even with an imperfect fitting, the RBM can still guide the QMC sampling



RBM is a recommender system for the QMC simulations

Why is it useful ?

- When the fitting is perfect, we can completely bypass the “quantum” part of the QMC

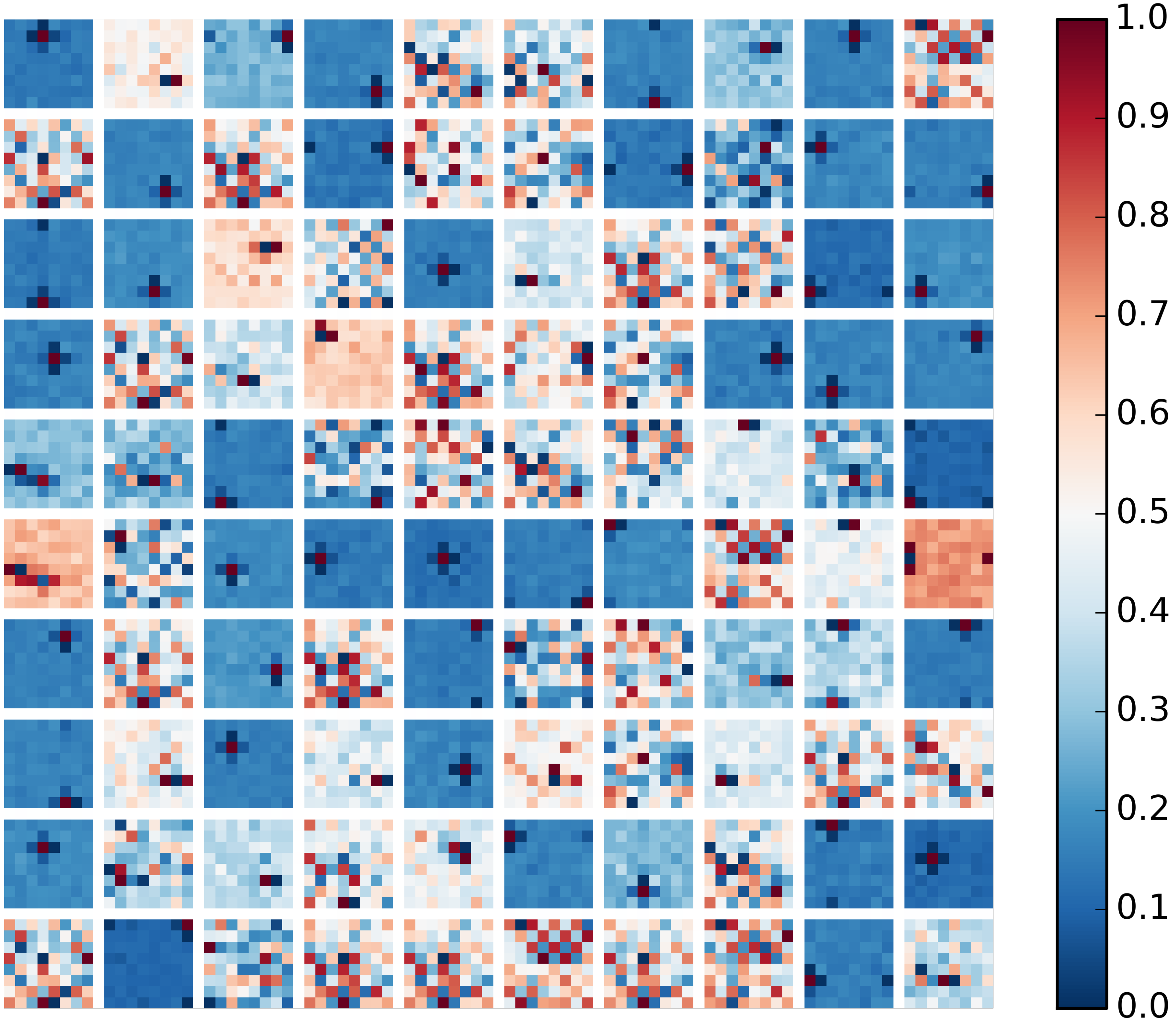
cf. Torlai, Melko,1606.02718

- Even with an imperfect fitting, the RBM can still guide the QMC sampling

RBM is a recommender system for the QMC simulations

Bonus: it is also fun to see what it discovers for the physical model

Learned Weight



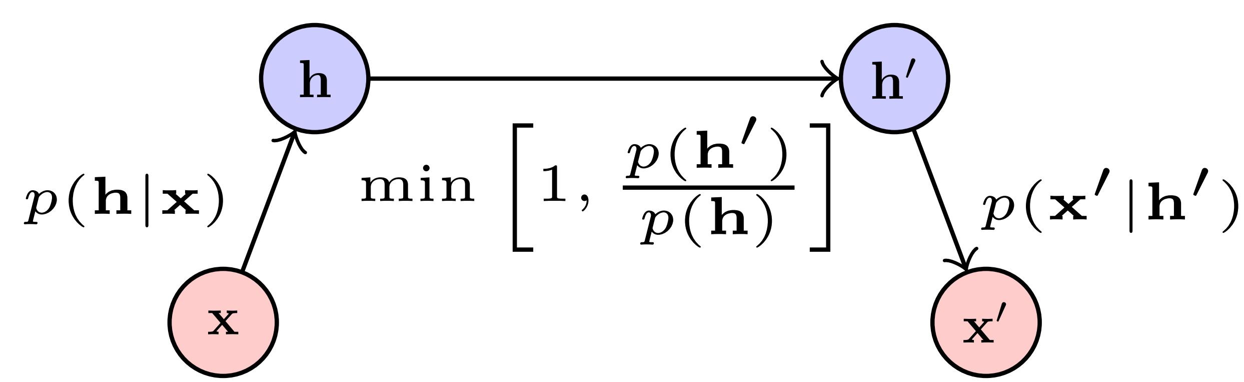
of a 8*8 Falicov-Kimball model

Accept or not?

$$A(\mathbf{x} \to \mathbf{x}') = \min\left[1, \frac{T(\mathbf{x}' \to \mathbf{x})}{T(\mathbf{x} \to \mathbf{x}')} \frac{\pi(\mathbf{x}')}{\pi(\mathbf{x})}\right]$$

The art of Monte Carlo methods

Sample the RBM, and propose the move to QMC



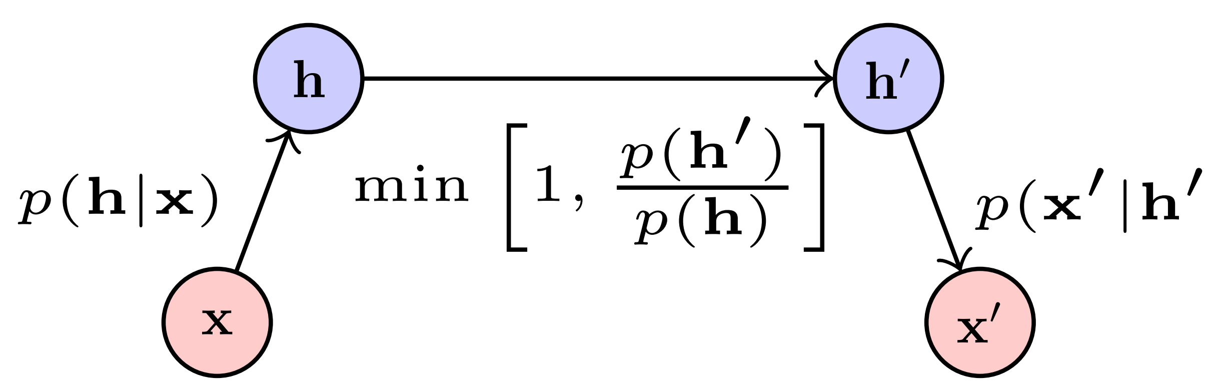
Accept or not?

$$A(\mathbf{x} \to \mathbf{x}') = \min\left[1, \frac{T(\mathbf{x}' \to \mathbf{x})}{T(\mathbf{x} \to \mathbf{x}')} \frac{\pi(\mathbf{x}')}{\pi(\mathbf{x})}\right]$$

Detailed balance condition for the RBM

$$\frac{T(\mathbf{x}' \to \mathbf{x})}{T(\mathbf{x} \to \mathbf{x}')} = \frac{p(\mathbf{x})}{p(\mathbf{x}')}$$

Sample the RBM, and propose the move to QMC



Accept or not?

Acceptance rate of recommended update

Li Huang and LW, 1610.02746

$$A(\mathbf{x} \to \mathbf{x}') = \min\left[1, \underbrace{\frac{p(\mathbf{x})}{p(\mathbf{x}')}}_{\text{RBM}} \cdot \underbrace{\frac{\pi(\mathbf{x}')}{\pi(\mathbf{x})}}_{\text{Physical model}}\right]$$

"surrogate function" R. M. Neal, Bayesian learning for neural networks, 1996

. J. S. Liu, Monte Carlo strategies in scientific computing, 2008

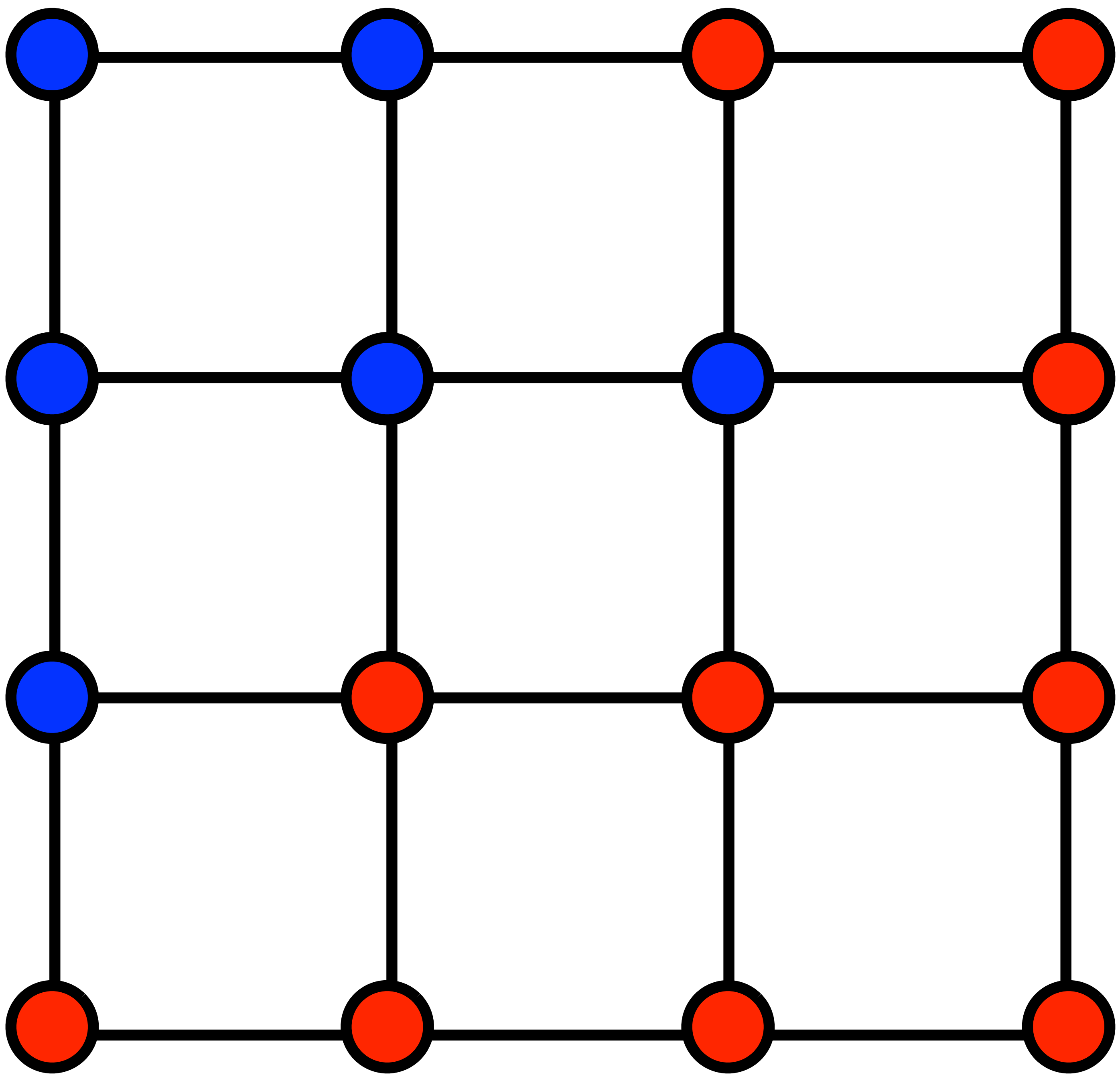
"force bias" S. Zhang, Auxiliary-field QMC for correlated electron systems, 2013

Question

Can Boltzmann Machines Discover Cluster Updates ?

LW, 1702.08586

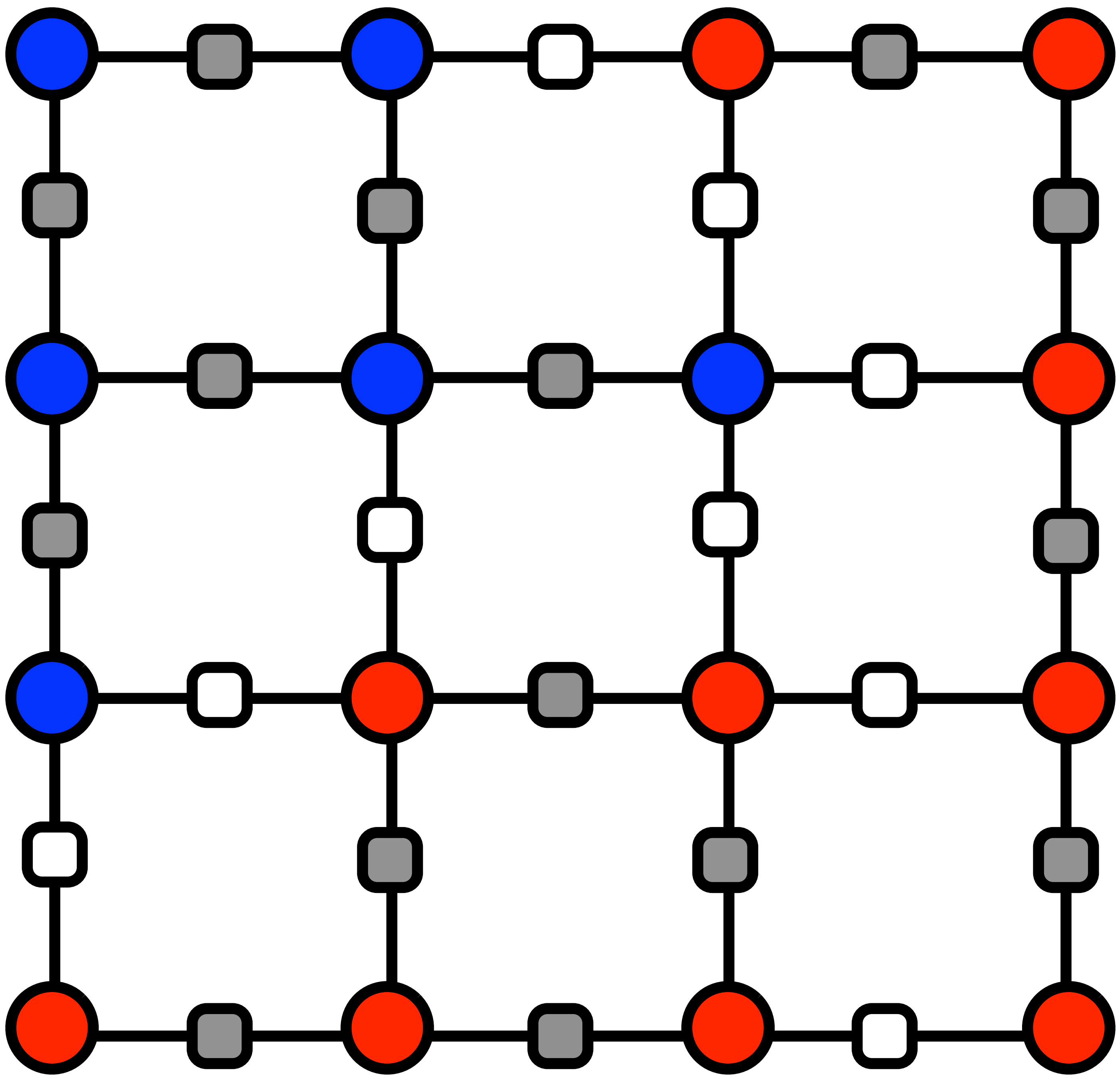
Cluster Update in a Nutshell



Swendsen and Wang,1987

Cluster Update in a Nutshell

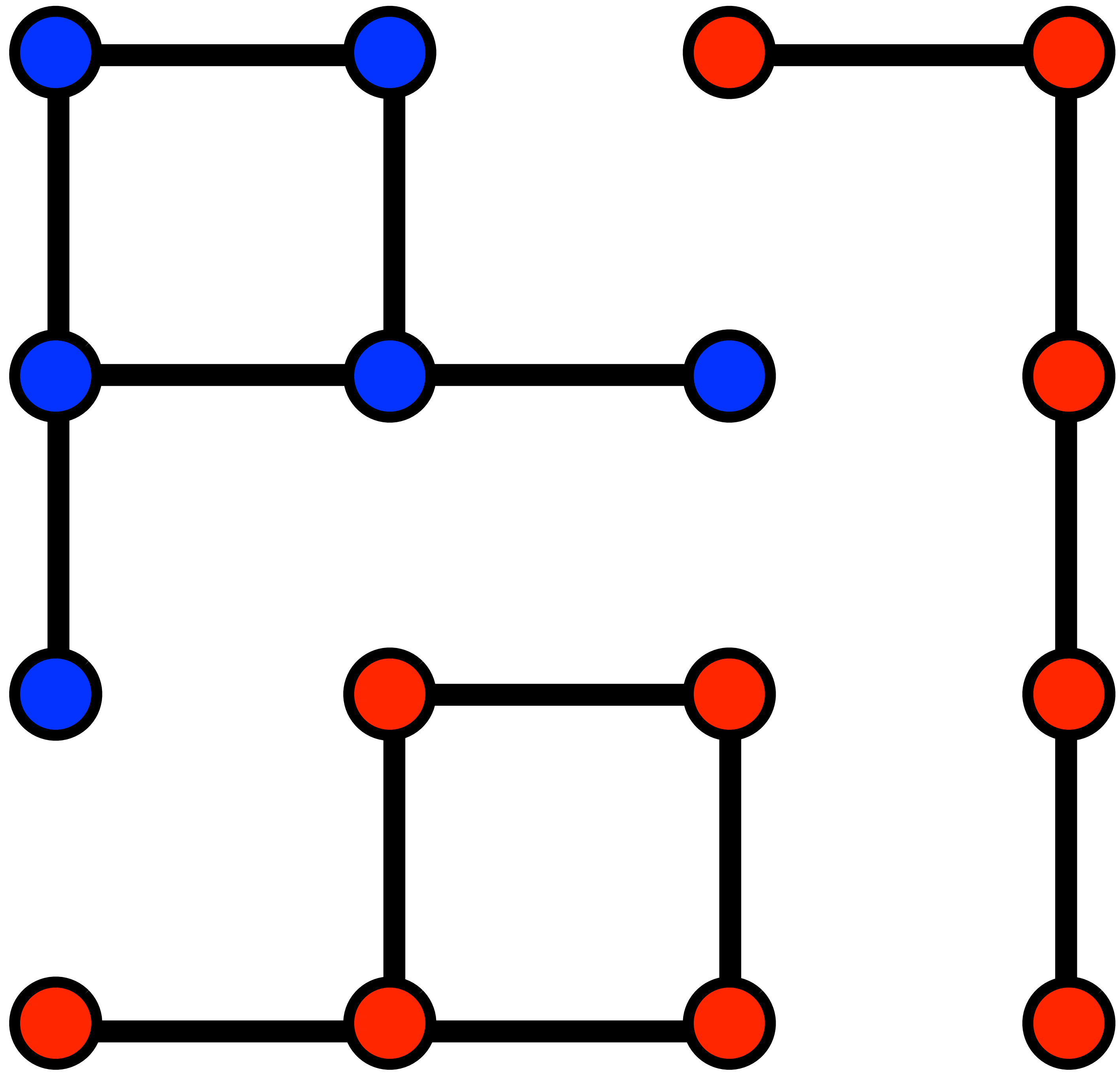
Sample auxiliary bond variables



Swendsen and Wang, 1987

Cluster Update in a Nutshell

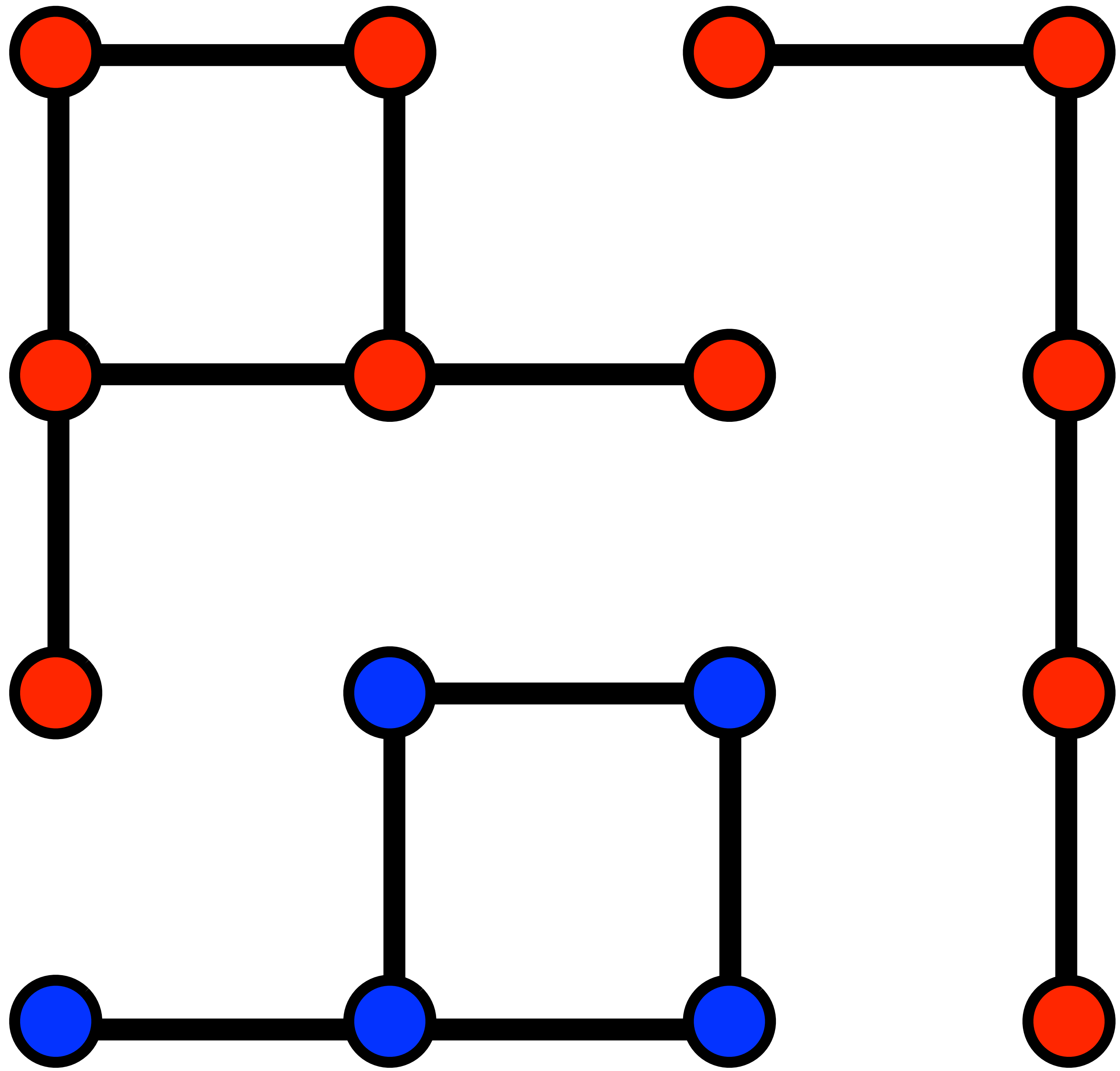
Build clusters



Swendsen and Wang,1987

Cluster Update in a Nutshell

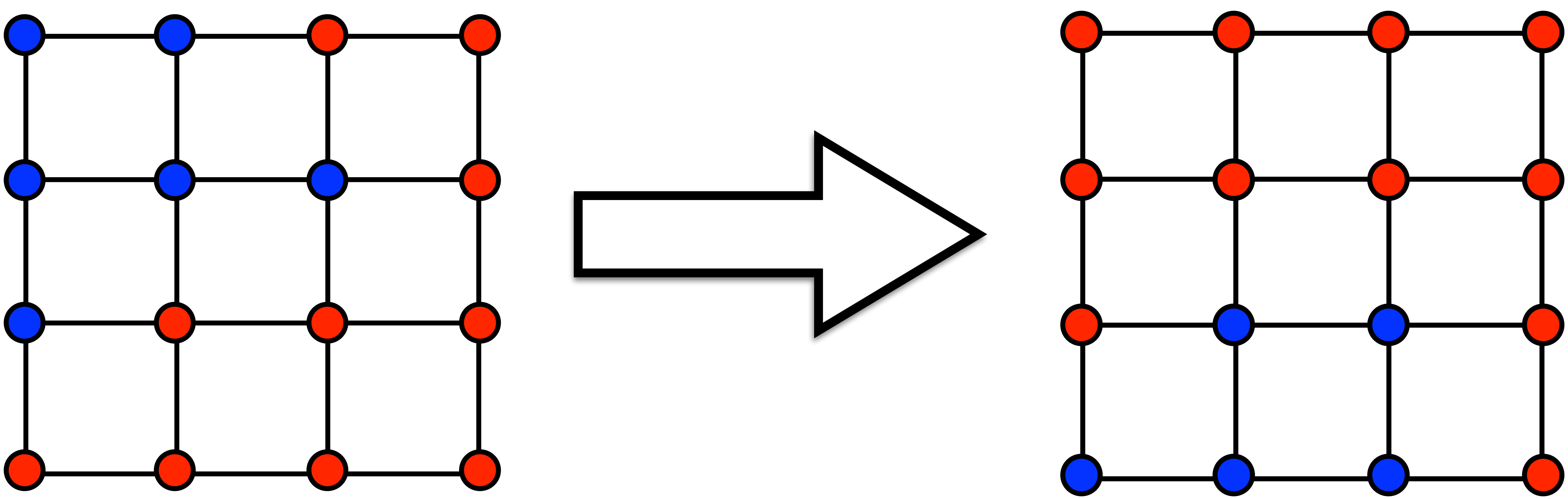
Flip clusters randomly



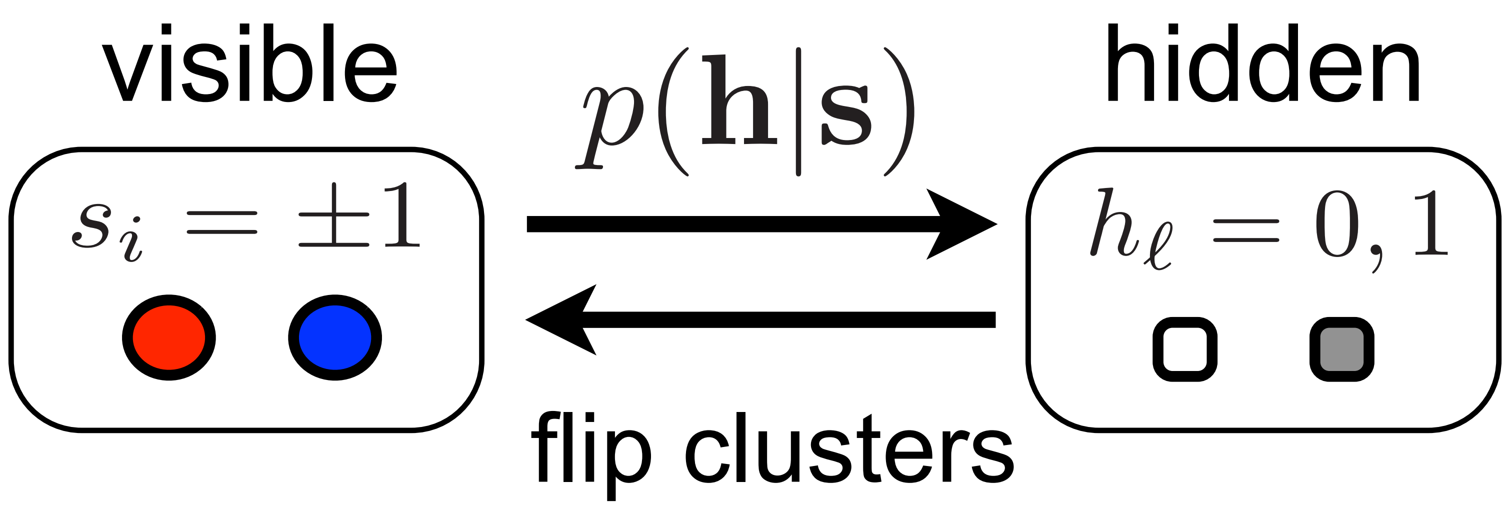
Swendsen and Wang, 1987

Cluster Update in a Nutshell

Rejection free cluster update!



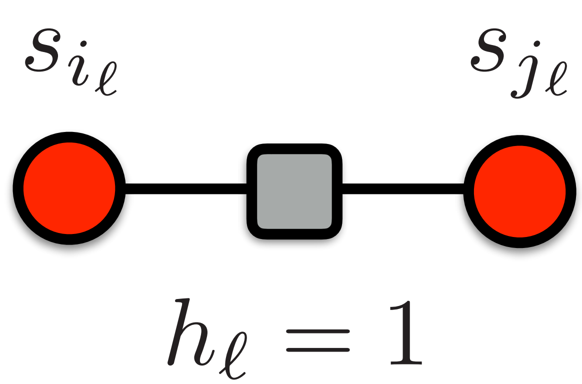
Swendsen and Wang, 1987



Swendsen-Wang	Boltzmann Machine
Ising spins	Visible units
Auxiliary bond variables	Hidden units
Build clusters	Sample h given s
Flip clusters	Sample s given h

Boltzmann Machine for cluster update

$$E(\mathbf{s},\mathbf{h}) = -\sum_{\ell} \left(W s_{i_\ell} s_{j_\ell} + b\right) h_\ell$$

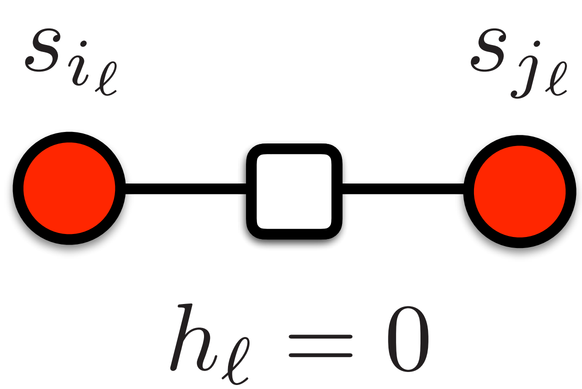


$$\mathbf{s} \in \{-1,+1\}^{|\text{Vertex}|}$$

$$\mathbf{h} \in \{0,1\}^{|\text{Edge}|}$$

Boltzmann Machine for cluster update

$$E(\mathbf{s}, \mathbf{h}) = -\sum_{\ell} \left(W s_{i_\ell} s_{j_\ell} + b\right) h_\ell$$

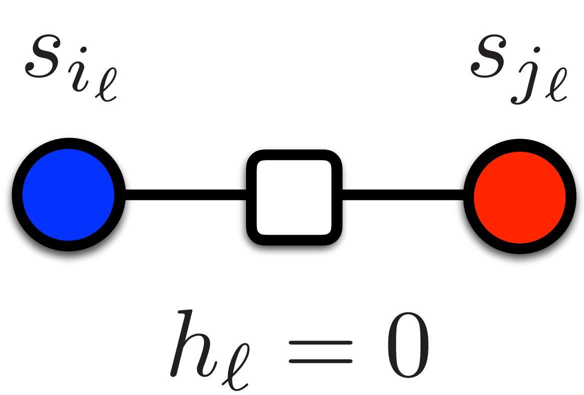


$\mathbf{s} \in \{-1, +1\}^{|\mathrm{Vertex}|}$

$\mathbf{h} \in \{0, 1\}^{|\mathrm{Edge}|}$

Boltzmann Machine for cluster update

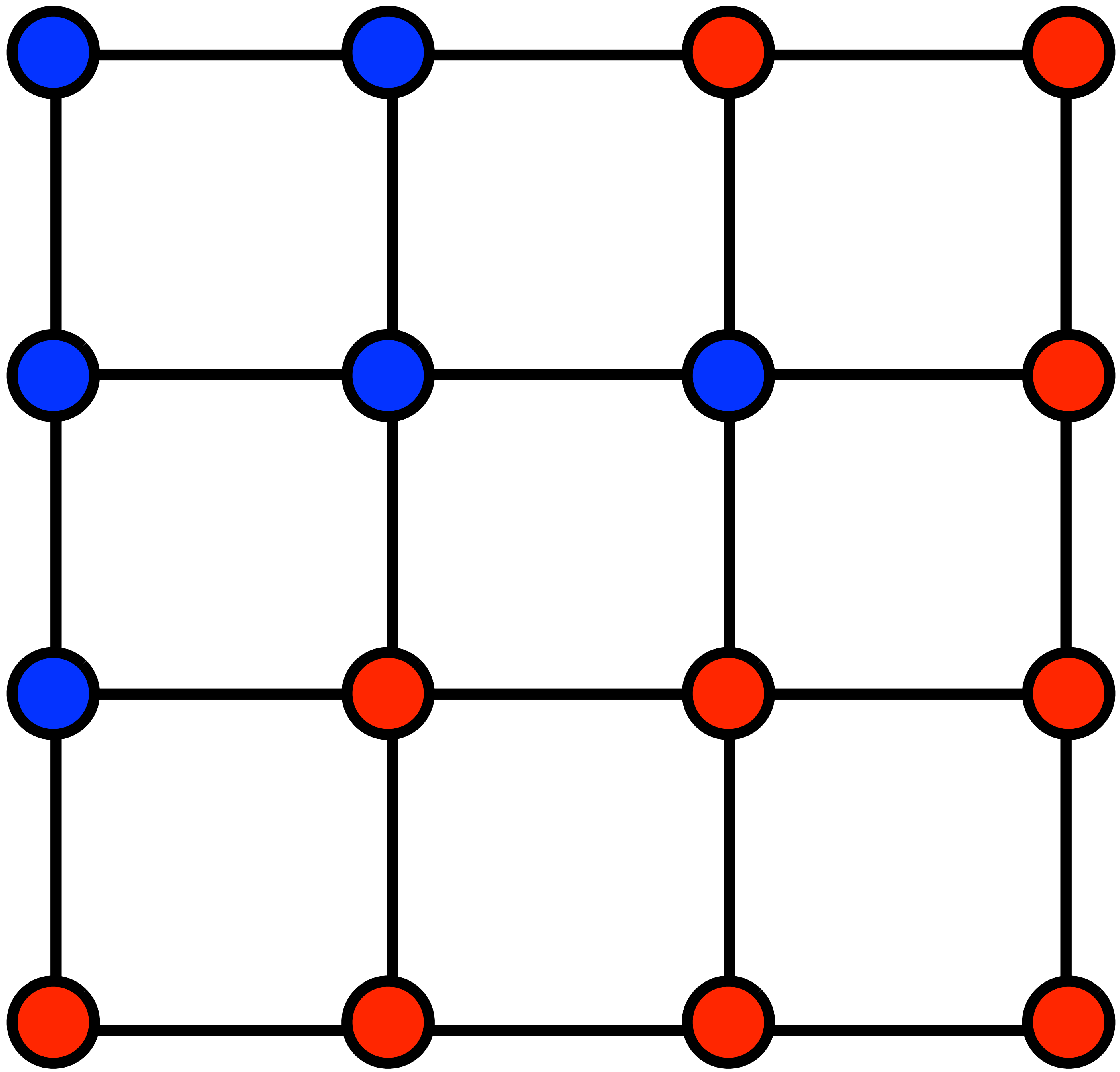
$$E(\mathbf{s},\mathbf{h}) = -\sum_{\ell} \left(W s_{i_\ell} s_{j_\ell} + b\right) h_\ell$$



$$\mathbf{s} \in \{-1, +1\}^{|\mathrm{Vertex}|}$$

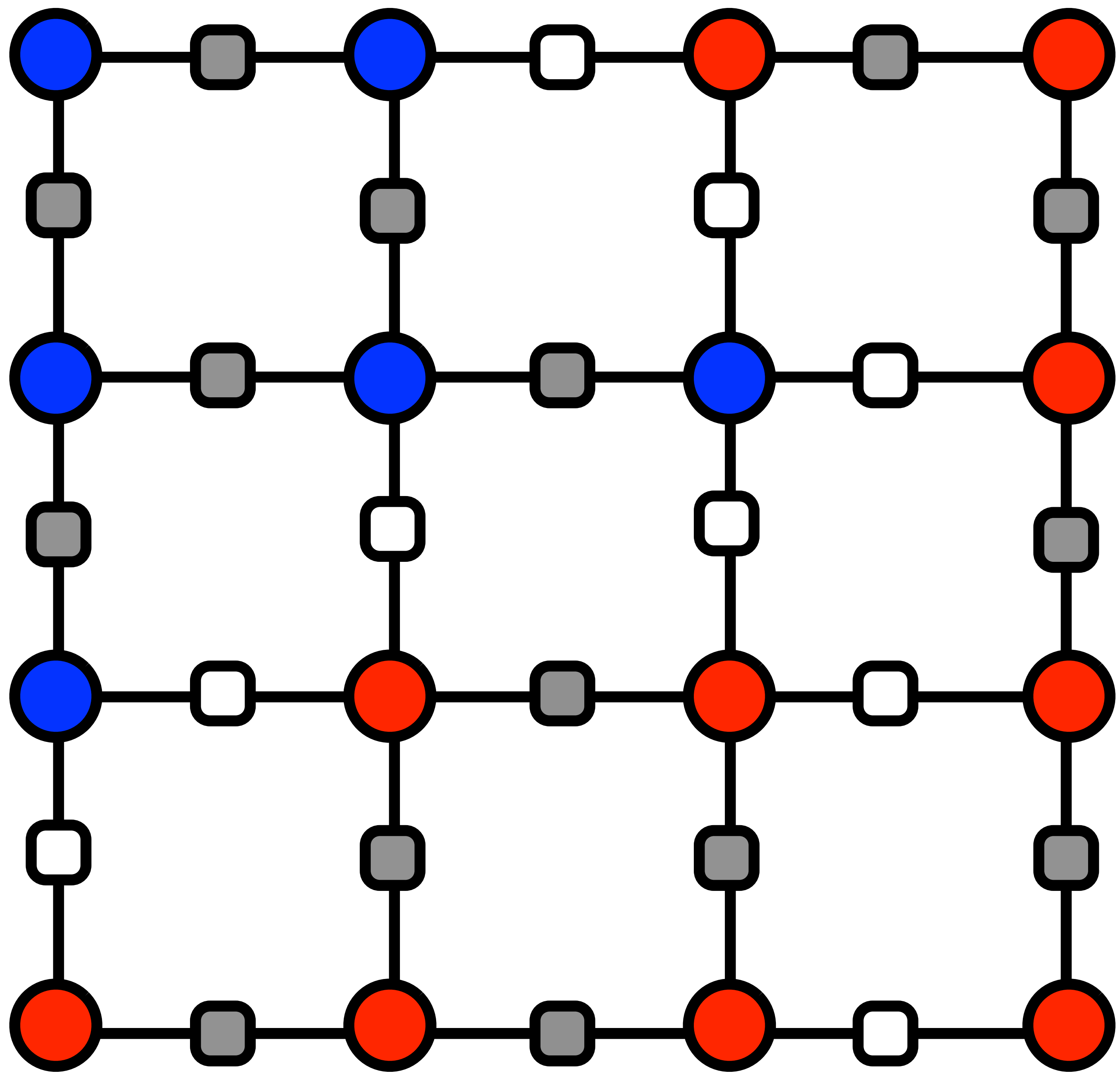
$$\mathbf{h} \in \{0, 1\}^{|\mathrm{Edge}|}$$

Cluster Update in the BM language



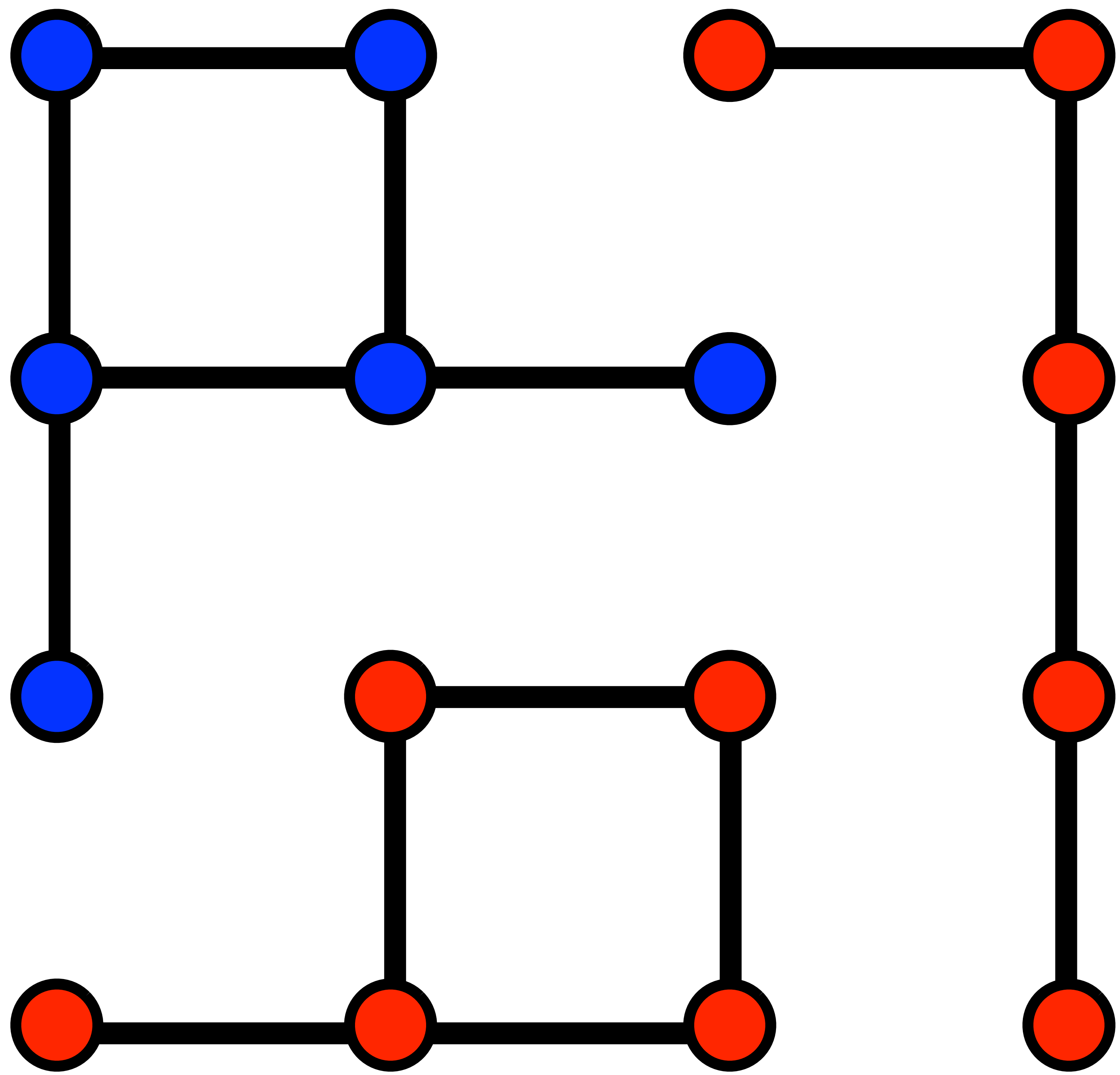
Cluster Update in the BM language

Sample hidden variables given visible units



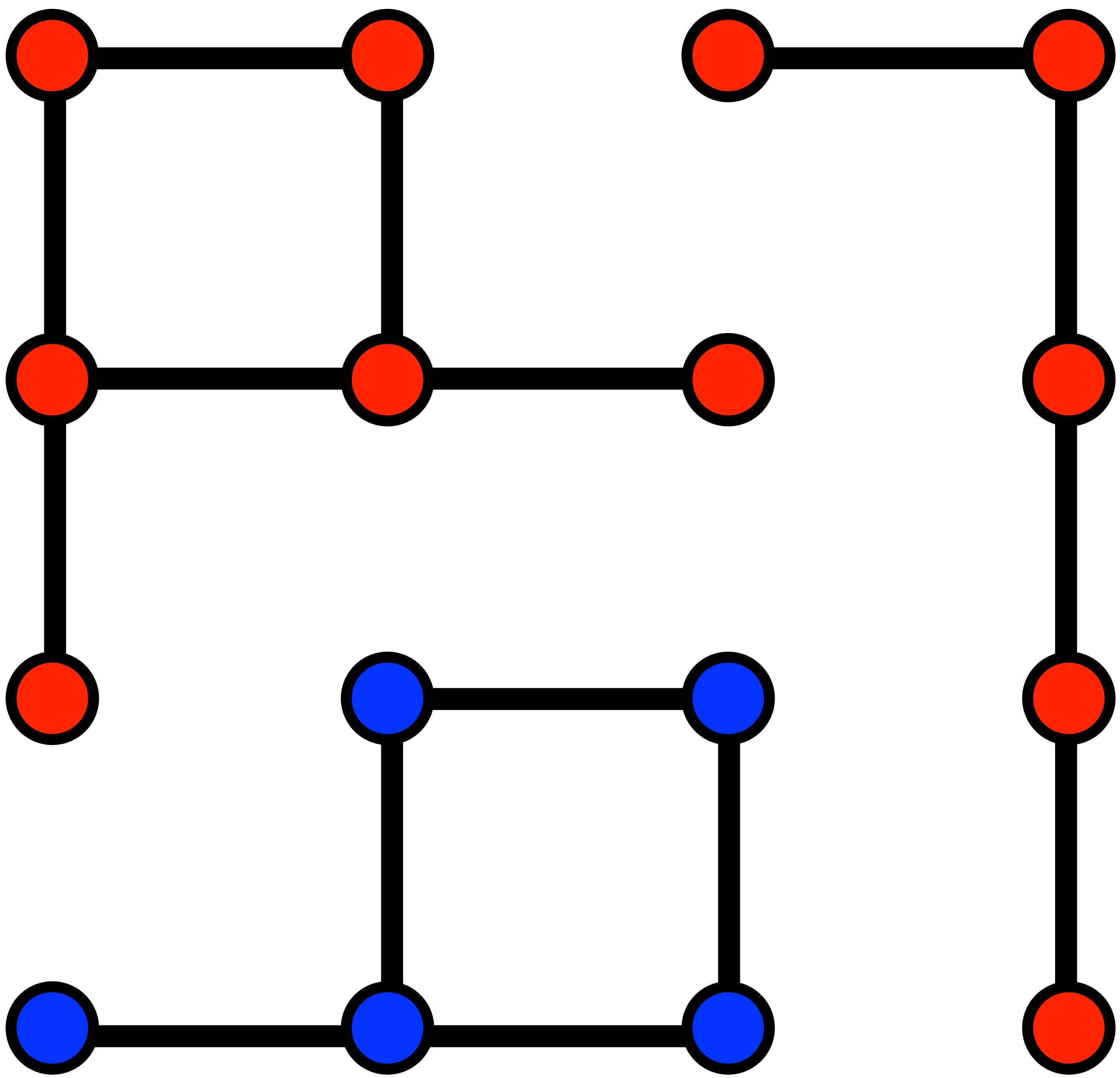
Cluster Update in the BM language

Inactive hidden units break the links



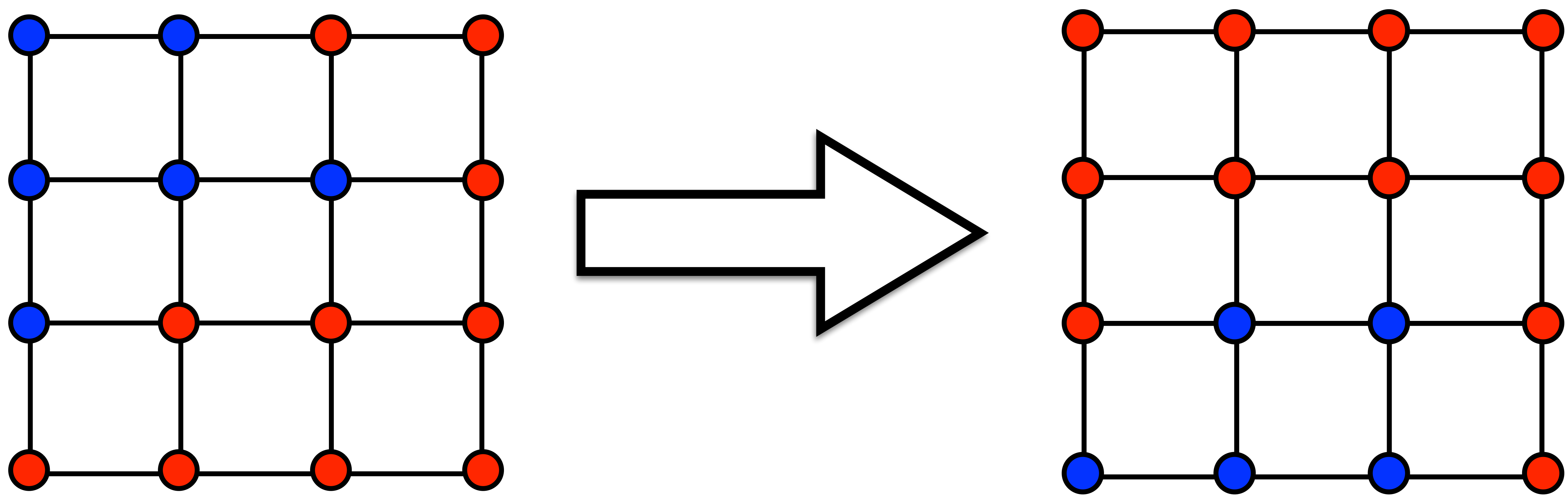
Cluster Update in the BM language

Randomly flip clusters of visible units



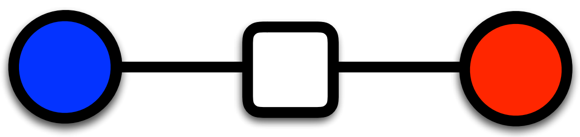
Cluster Update in the BM language

Voila!



Boltzmann Machine for cluster update

$$E(\mathbf{s},\mathbf{h}) = -\sum_{\ell} \left(W s_{i_\ell} s_{j_\ell} + b\right) h_\ell$$



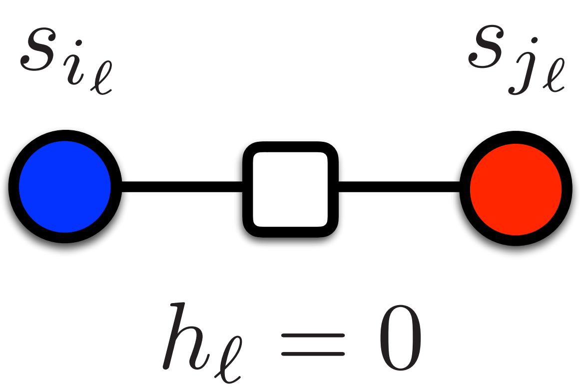
$$\mathbf{s} \in \{-1,+1\}^{|\mathrm{Vertex}|}$$

$$\mathbf{h} \in \{0,1\}^{|\mathrm{Edge}|}$$

- Rejection free Monte Carlo updates when $\dfrac{1+e^{b+W}}{1+e^{b-W}} = e^{2\beta J}$

Boltzmann Machine for cluster update

$$E(\mathbf{s}, \mathbf{h}) = -\sum_{\ell} (W s_{i_\ell} s_{j_\ell} + b)\, h_\ell$$



$$\mathbf{s} \in \{-1, +1\}^{|\mathrm{Vertex}|}$$

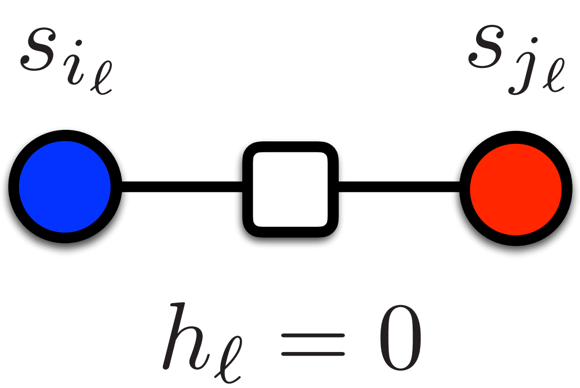
$$\mathbf{h} \in \{0, 1\}^{|\mathrm{Edge}|}$$

- Rejection free Monte Carlo updates when $\dfrac{1+e^{b+W}}{1+e^{b-W}} = e^{2\beta J}$
- Encompass general cluster algorithm frameworks

Niedermeyer, 1988 Kandel and Domany, 1991 Kawashima and Gubernatis, 1995

Boltzmann Machine for cluster update

$$E(\mathbf{s}, \mathbf{h}) = -\sum_{\ell} \left(W s_{i_\ell} s_{j_\ell} + b\right) h_\ell$$



$\mathbf{s} \in \{-1, +1\}^{|\mathrm{Vertex}|}$

$\mathbf{h} \in \{0, 1\}^{|\mathrm{Edge}|}$

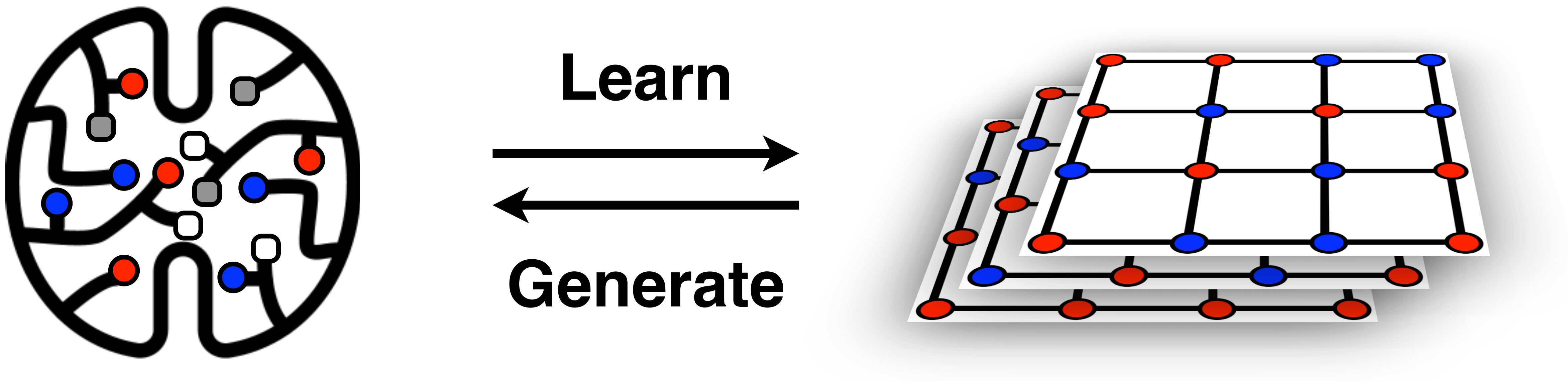
- Rejection free Monte Carlo updates when $\dfrac{1 + e^{b+W}}{1 + e^{b-W}} = e^{2\beta J}$

- Encompass general cluster algorithm frameworks

Niedermeyer, 1988 Kandel and Domany, 1991 Kawashima and Gubernatis, 1995

- In general $E(\mathbf{s}, \mathbf{h}) = E(\mathbf{s}) - \sum_{\alpha} \left[W_\alpha \mathcal{F}_\alpha(\mathbf{s}) + b_\alpha\right] h_\alpha$

"feature"

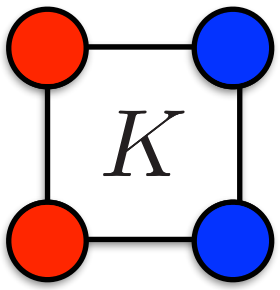


- Boltzmann Machines are learnable so they can adapt to various physical problems

- Boltzmann Machines parametrize Monte Carlo policies which can be optimized for efficiency

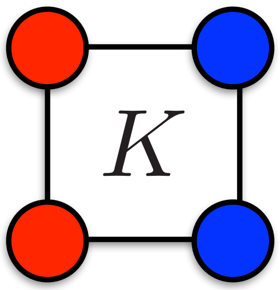
- The hidden units learn to play smart roles:
  Fortuin-Kasteleyn and Hubbard-Stratonovich transformations

Plaquette Ising model



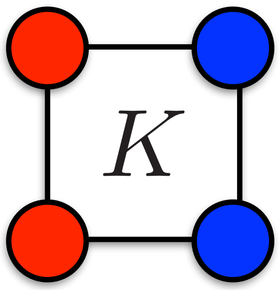
"However, for K≠0, **NO** simple and efficient global update method is known." —1610.03137

Plaquette Ising model



$$e^{\beta K s_1 s_2 s_3 s_4} \to \sum_h e^{W(s_1 s_2 + s_3 s_4) h}$$

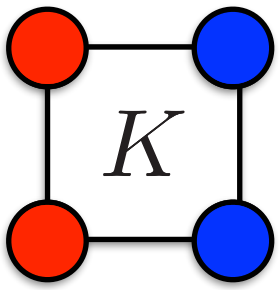
Plaquette Ising model



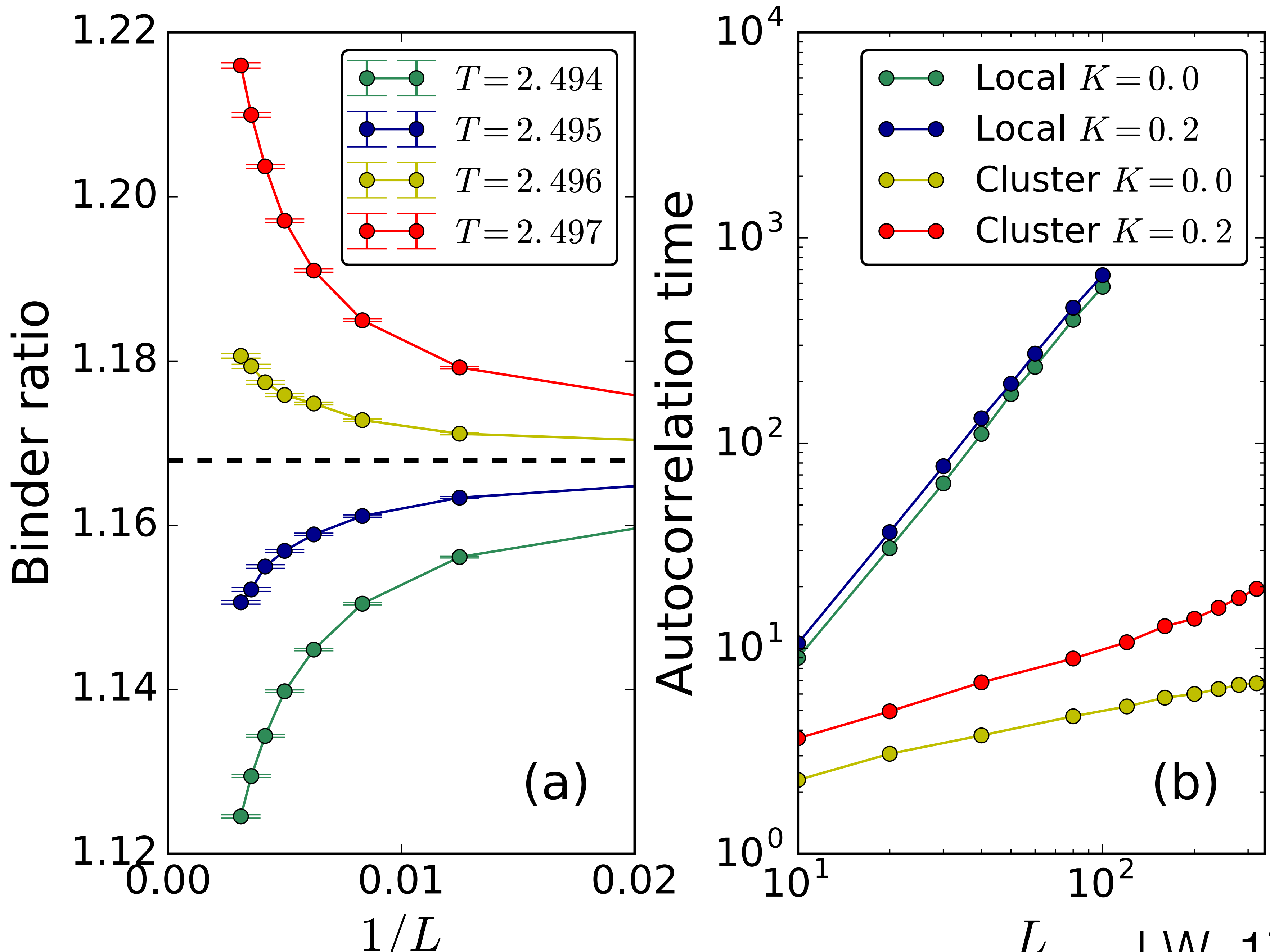
$$e^{\beta K s_1 s_2 s_3 s_4} \to \sum_h e^{W(s_1 s_2 + s_3 s_4)h}$$

- Given s, sample h on each plaquette independently
- Given h, the Boltzmann Machine is an ordinary Ising model with modulated interactions, which can be sampled efficiently
- Rejection free cluster update!

Plaquette Ising model



$$e^{\beta K s_1 s_2 s_3 s_4} \to \sum_h e^{W(s_1 s_2 + s_3 s_4)h}$$



LW, 1702.08586

Machine learning for many-body physics

- New tools for (quantum) many-body problems
- Moreover, it offers a new way of thinking
- Can we make new scientific discovery with it ?
- Can one design better algorithms with it ?

LW, 1606.00318

Li Huang and LW, 1610.02746

Li Huang, Yi-feng Yang and LW, 1612.01871

LW, 1702.08586

Machine learning for many-body physics

- New tools for (quantum) many-body problems
- Moreover, it offers a new way of thinking
- Can we make new scientific discovery with it ?
- Can one design better algorithms with it ?

The fun just starts!

LW, 1606.00318

Li Huang and LW, 1610.02746

Li Huang, Yi-feng Yang and LW, 1612.01871

LW, 1702.08586

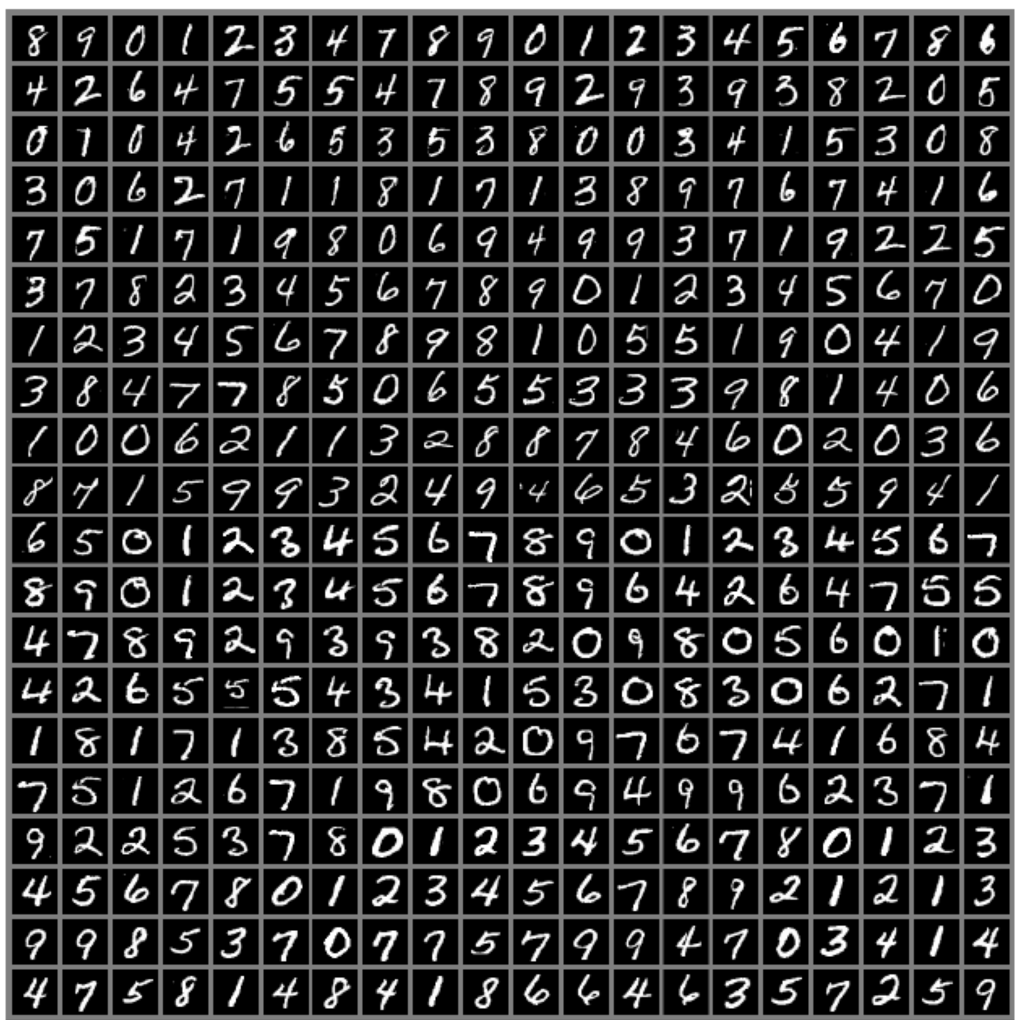
Quantum many-body physics for ML

Quantum entanglement perspective on deep learning

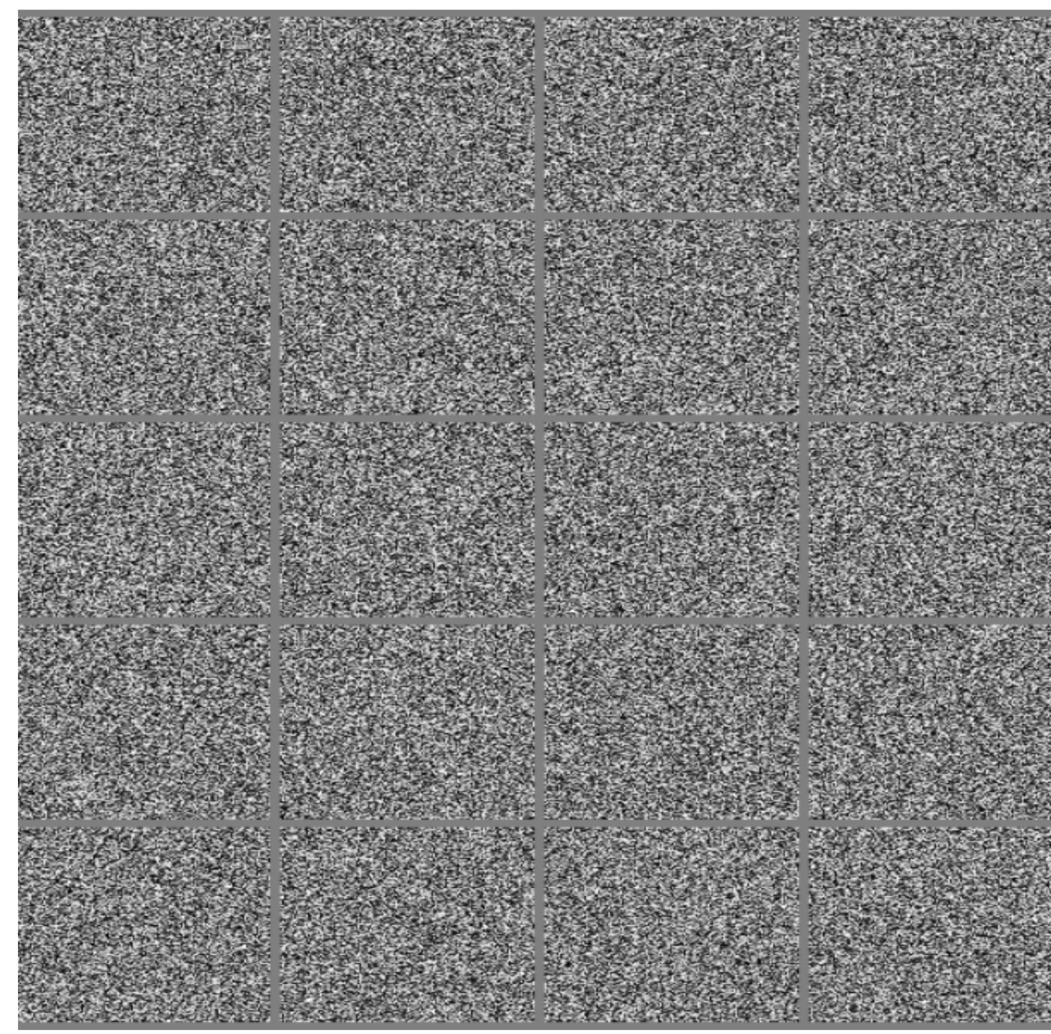
Jing Chen, Song Cheng, Haidong Xie, LW, and Tao Xiang, 1701.04831

Dong-Ling Deng, Xiaopeng Li and S. Das Sarma, 1701.04844

Xun Gao, L.-M. Duan, 1701.05039 Y. Huang and J. E. Moore, 1701.06246



MNIST database



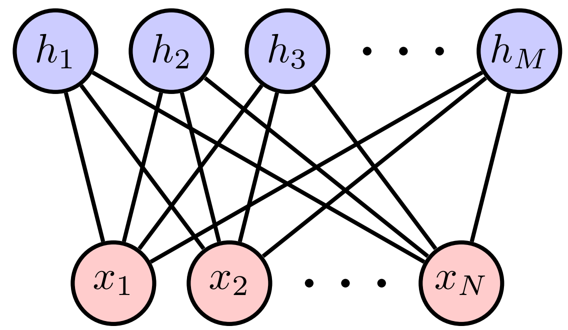
random images

from the "Deep Learning" book by Goodfellow, Bengio, Courville https://www.deeplearningbook.org/

$$A(\mathbf{x} \to \mathbf{x}') = \min \left[1, \; \underbrace{\frac{p(\mathbf{x})}{p(\mathbf{x}')}}_{\text{RBM}} \cdot \underbrace{\frac{\pi(\mathbf{x}')}{\pi(\mathbf{x})}}_{\text{Physical model}} \right]$$

$\mathbf{h} \in \{0,1\}^M$

$\mathbf{x} \in \{0,1\}^N$



RBM

Physical model

e.g. Falikov-Kimball (classical field + fermions) model

$$\ln \boxed{\pi(\mathbf{x})} = \frac{\beta U}{2} \sum_{i=1}^{N} x_i + \ln \det \left[1 + e^{-\beta \mathcal{H}(\mathbf{x})} \right]$$

$$\mathcal{H}_{ij} = \mathcal{K}_{ij} + \delta_{ij} U \left(x_i - 1/2 \right)$$

while for the RBM

$$\ln \boxed{p(\mathbf{x})} = \sum_{i=1}^{N} \boxed{a_i} x_i + \sum_{j=1}^{M} \ln \left(1 + e^{\boxed{b_j} + \sum_{i=1}^{N} x_i \boxed{W_{ij}}} \right)$$

Train the RBM

$$\boxed{p(\mathbf{x}) \sim \pi(\mathbf{x})}$$

Supervised learning of the RBM

$$\sum_{i=1}^{N} a_i x_i + \sum_{j=1}^{M} \ln\left(1 + e^{b_j + \sum_{i=1}^{N} x_i W_{ij}}\right)$$

$$= \frac{\beta U}{2} \sum_{i=1}^{N} x_i + \ln\det\left(1 + e^{-\beta \mathcal{H}}\right)$$