# A quantum annealing approach for learning Boltzmann machines as function approximators and/or samplers

Siddhartha Srivastava, Veera Sundararaghavan

Multi-Scale Structural Simulations Laboratory

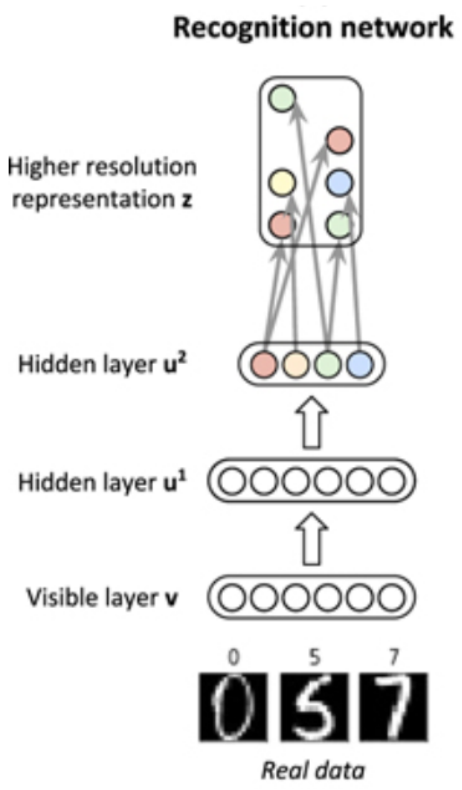
University of Michigan, Ann Arbor

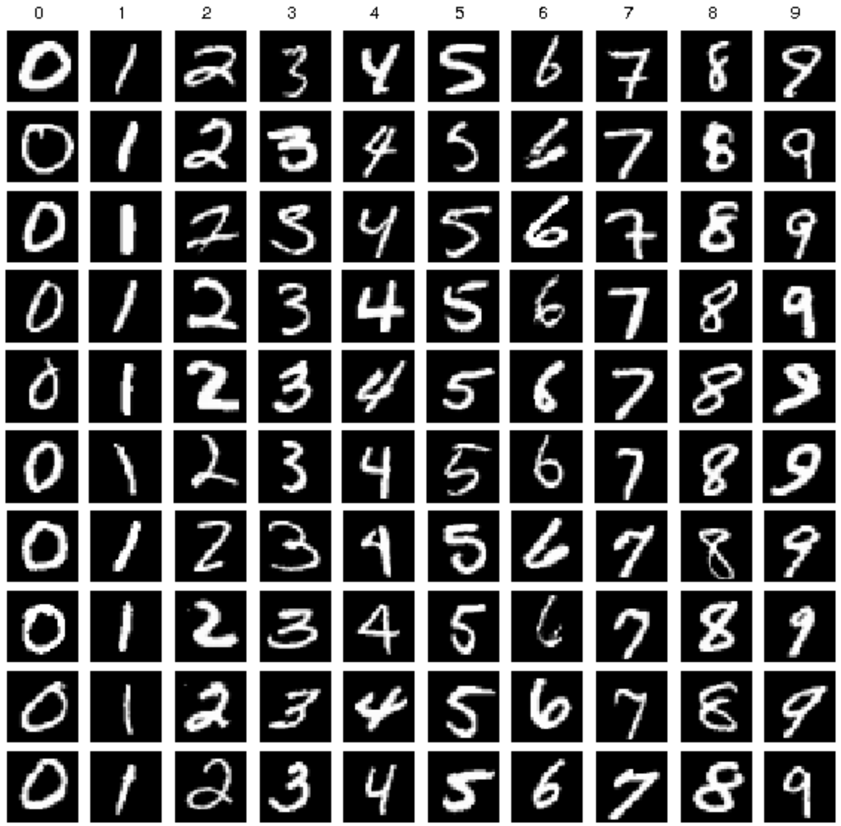
UNIVERSITY OF MICHIGAN

# Outline

- Applications in machine learning
- Definition and properties
- Review of some classical training strategies
- Proposed training method using Quantum annealing
- New challenges and their resolution

# Application: Labeled data generation



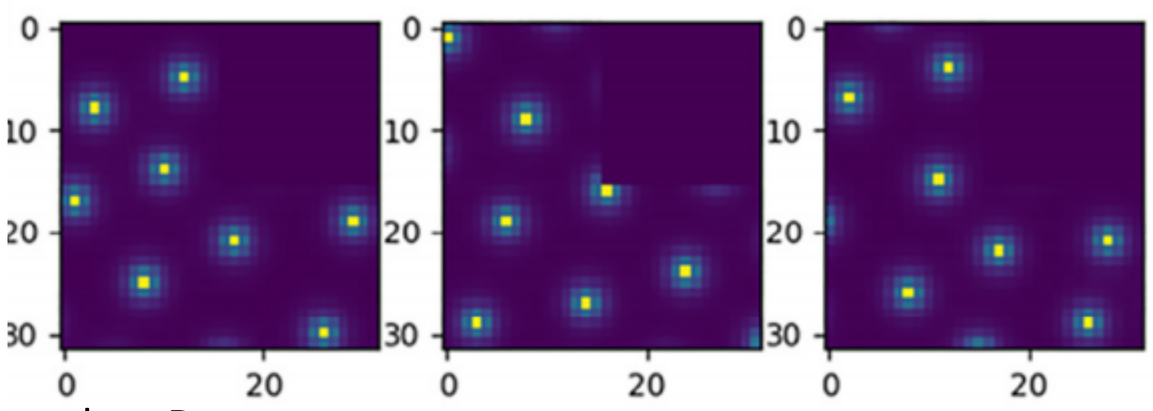




Labeled training Data (mnist)

Handwritten numbers

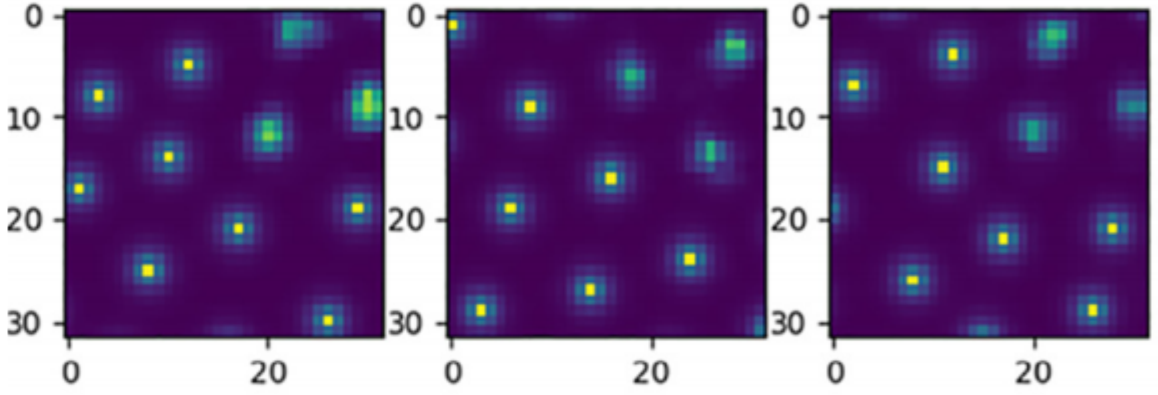
Benedetti, Marcello, John Realpe-Gómez, and Alejandro Perdomo-Ortiz. "Quantum-assisted helmholtz machines: a quantum–classical deep learning framework for industrial datasets in near-term devices." *Quantum Science and Technology* 3.3 (2018): 034007.

# Application: Recovering missing data



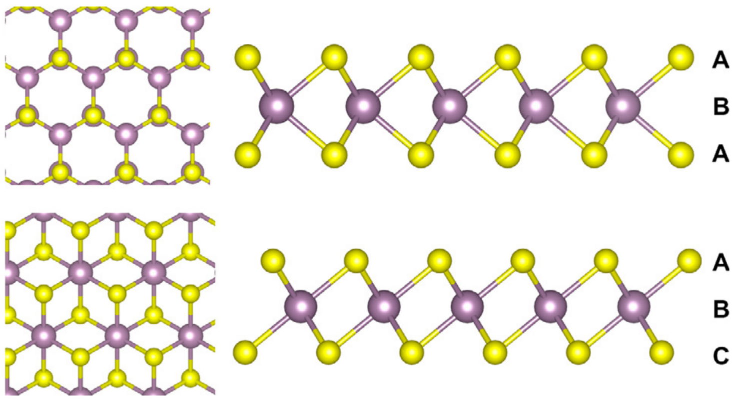

Incomplete Data

Reconstructed Data

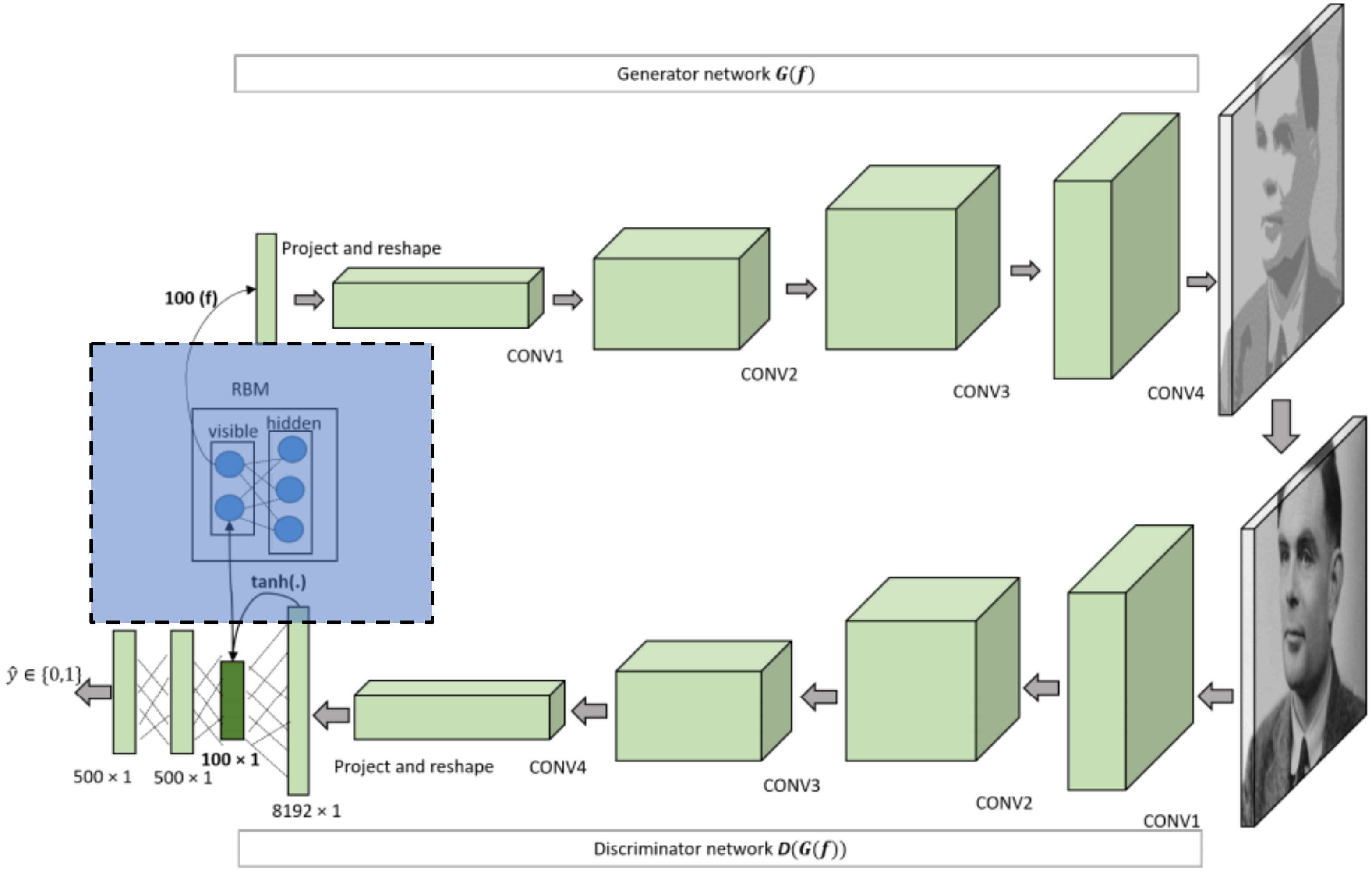



Chemical vapor deposition (CVD) growth for a MoS2 monolayer

Liu, Jeremy, et al. "Boltzmann machine modeling of layered MoS2 synthesis on a quantum annealer." *Computational Materials Science* 173 (2020): 109429.



# Application: Machine Learning architectures

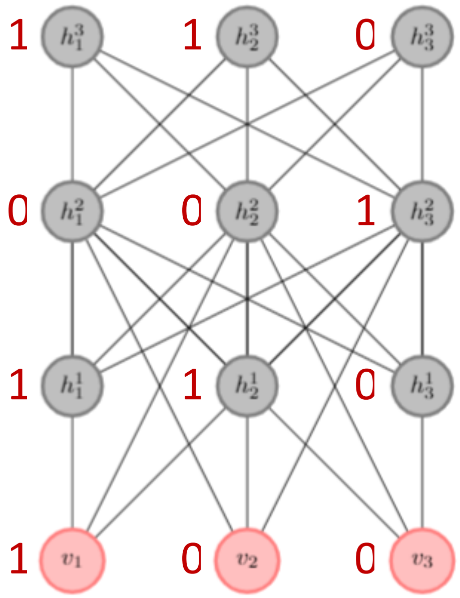



Associative adversarial networks

Arici, Tarik, and Asli Celikyilmaz. "Associative adversarial networks." *arXiv preprint arXiv:1611.06953* (2016).

- Intermediate layer of the discriminator reads the visible layer of the RBM network (the associative memory).
- RBM Samples generate inputs for the generator network (as opposed to noise sampling).
- This layer that is visible to the associative memory represents a feature space that can capture latent factors of variations in the data

# Boltzmann machine are probabilistic energy-based graph models




- **Graph models** – Nodes connected via edges (undirected)

- **Energy based** – Each node takes 0/1 value
- Energy determined by an Ising-type energy

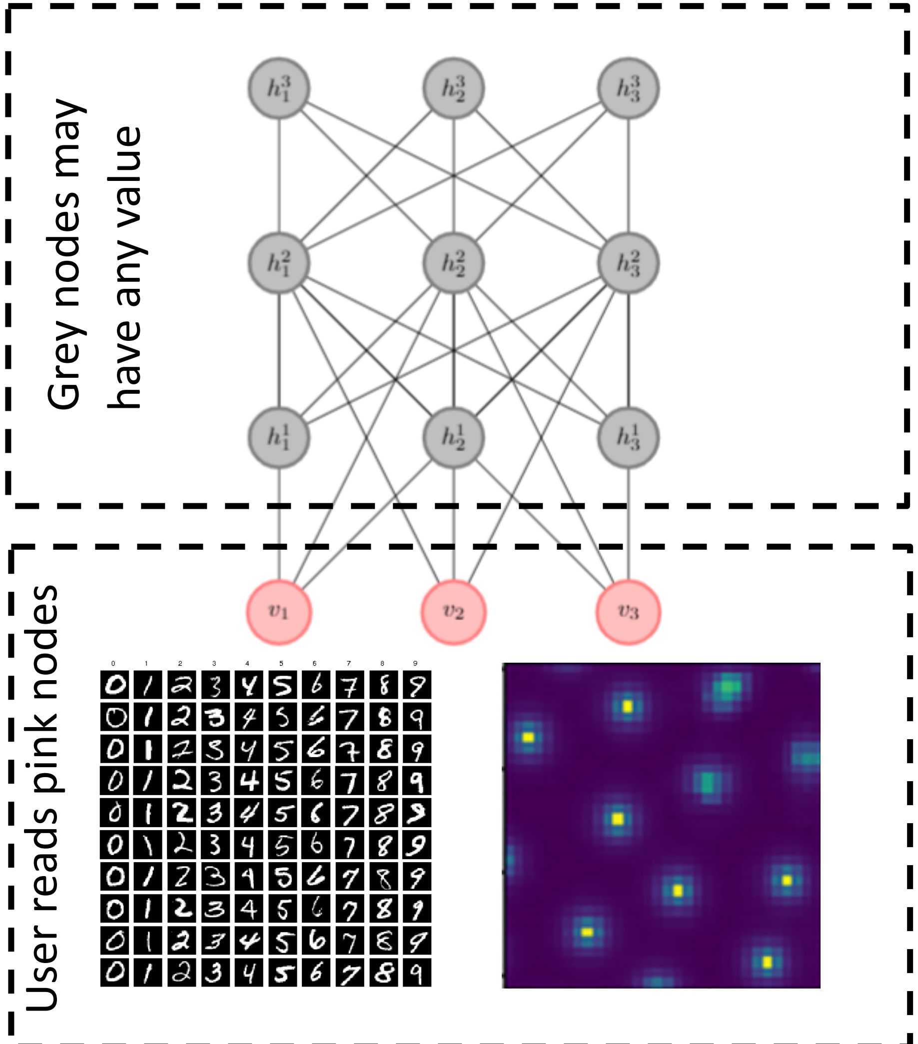
$$E(S) = \sum_{i \in Nodes} H_i S_i + \sum_{(i,j) \in Edges} J_{ij} S_i S_j$$

- **Probabilistic** – Each state is determined via Boltzmann distribution

$$p(S) = \frac{e^{-\beta E(S)}}{Z}, \qquad Z = \sum e^{-\beta E(S)}$$

$\beta$ is the inverse temperature

# User can only read part of the nodes




- Nodes segregated into Visible and Hidden nodes

$$S = [v, h]$$

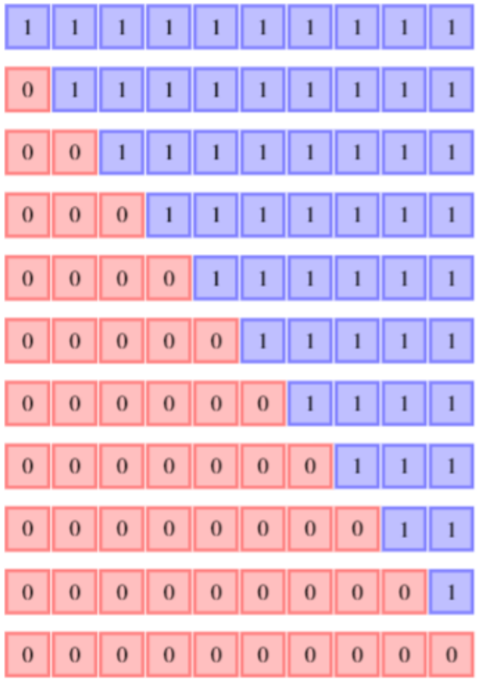
- Only data on the visible nodes can be read.

- Probability of visible nodes determined by marginalizing over hidden nodes

$$p(v;\theta) = \sum p([v, h])$$

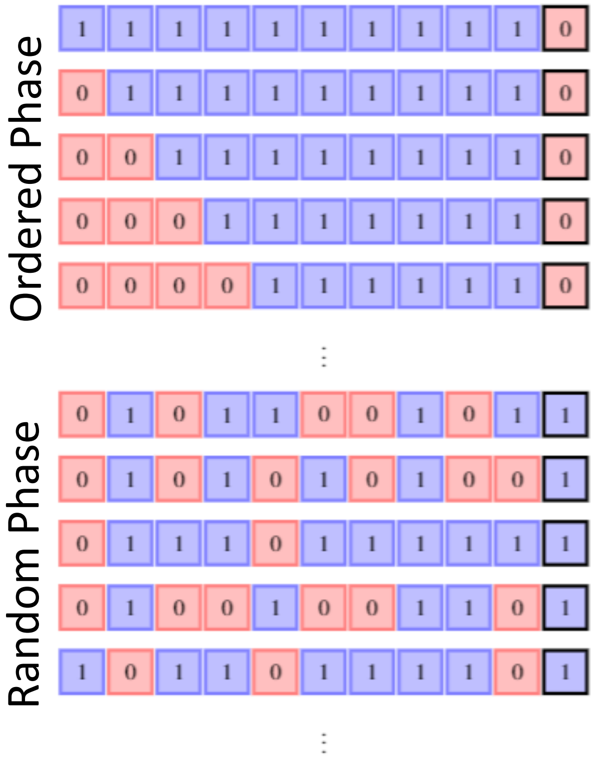
- This step allows to model complicated probability mass functions

# Representing data-sets for visible nodes




Set of states with '0' on left and '1' on right

$\text{size}\, x = 10$




Distinguish between Random and ordered phase

$\text{size}\, x = 10$

$\text{size}\, f(x) = 1$

- Each row is a data, and each column is a node
- Left Sample set: Generative Learning

  Samples state $(x)$ from this data set
- Right Sample set : Adding classification

  Samples state $(x,\, f(x))$ from this data set

Note that we may be interested in complete sampling or reconstruction

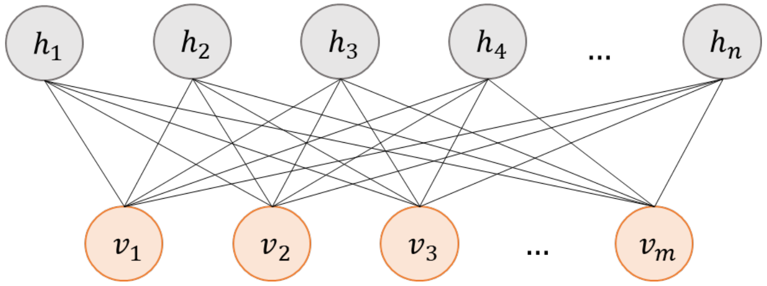
# Estimation of gradients is challenging

- Optimize for Log-likelihood based cost (KL Divergence, Negative Log-likelihood)

$$\frac{\partial\left(-\log p(v^*)\right)}{\partial\theta} = \mathbb{E}_h\left(\frac{\partial E(v,h)}{\partial\theta}\middle| v^*\right) - \mathbb{E}_{v,h}\left(\frac{\partial E(v,h)}{\partial\theta}\right)$$

- Exact estimation prohibited due to exponentially large number of states
- Estimating expectation using Monte Carlo-based techniques takes time to equilibrate
- Another idea: Use “simpler” graph-structures

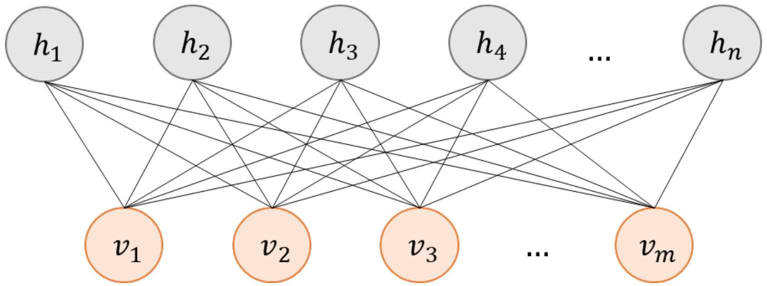
Restricted Boltzmann machine - Bipartite graph of hidden and visible layer




Contrastive Divergence / Negative Sampling

Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets." *Neural computation* 18.7 (2006): 1527-1554

# Computational Complexity is determined by the topology of the graph




$$\frac{\partial\left(-\log p(v^*)\right)}{\partial\theta} = \mathbb{E}_h\left(\frac{\partial E(v,h)}{\partial\theta}\middle| v^*\right) - \mathbb{E}_{v,h}\left(\frac{\partial E(v,h)}{\partial\theta}\right)$$

Maximizing likelihood of a data state

Contrastive Divergence / Negative Sampling

Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets." *Neural computation* 18.7 (2006): 1527-1554

**<u>Idea</u>:** Start with a data (desired) state and check if you are moving away from it.

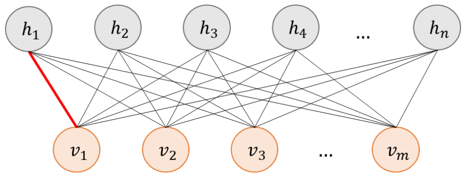

Positive Phase:

$$\mathbb{E}_h\left(\frac{\partial E(v,h)}{\partial\theta}\middle| v^*\right) \equiv \frac{\partial E\left(v^*,h\right)}{\partial\theta}$$

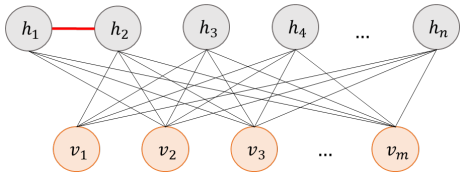
Negative Phase:

# Computational Complexity is determined by the topology of the graph

- Ease of computation doesn't depend on just sparsity but the overall topology of graph, e.g., presence of cycles, multipartite graph etc.

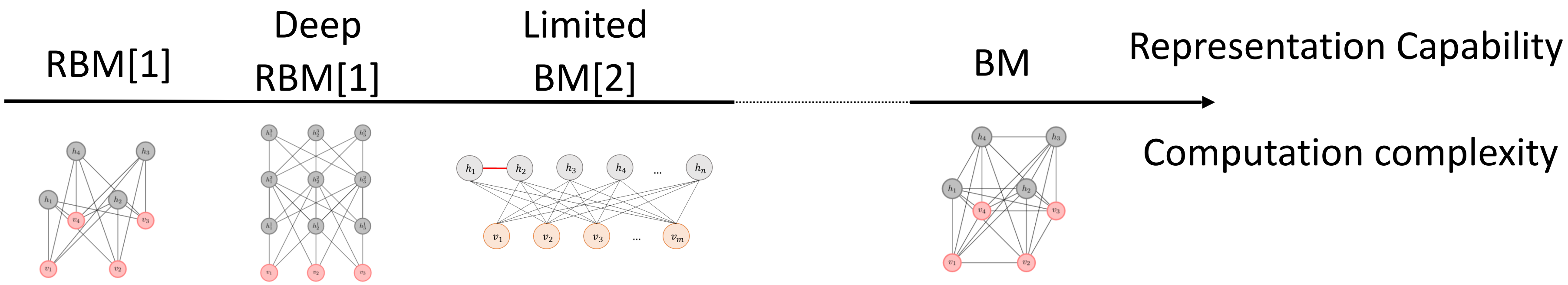


Less complex



Moderately complex

- In general, adding edges to a network increases representation capability but also the cost of computation



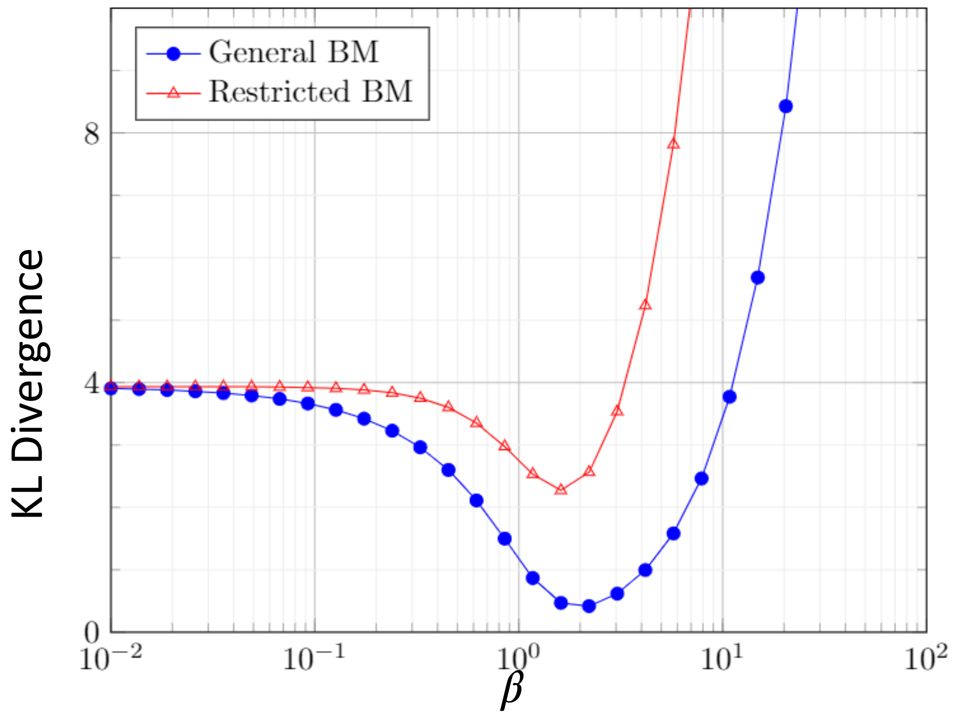


[1] Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In Artificial intelligence and statistics, (2009)
[2] Liu, Jeremy, et al. "Boltzmann machine modeling of layered MoS2 synthesis on a quantum annealer." Computational Materials Science (2020)

# Tradeoffs between representability and computational cost

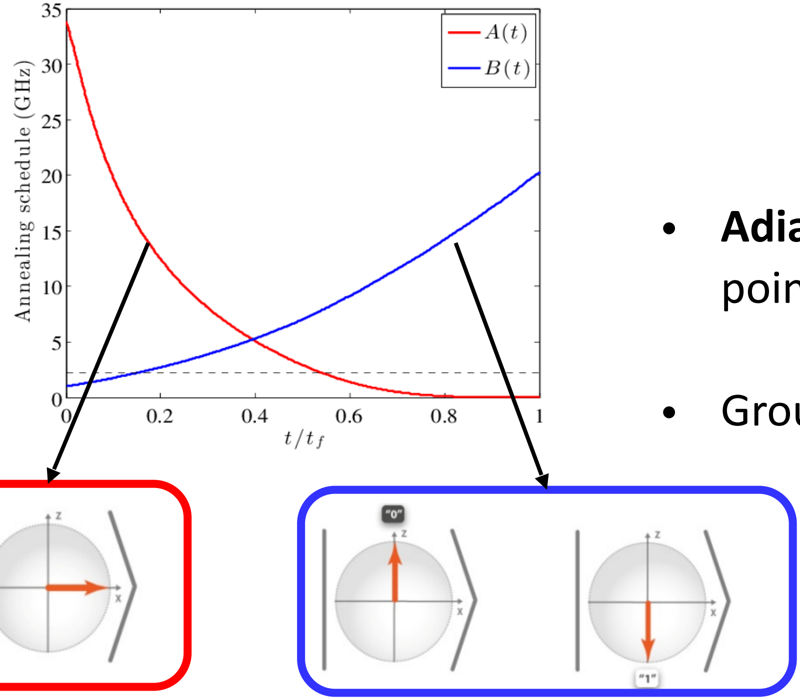
- Gradient based approximations for general BM is difficult due to calculation of expectations
- Use Contrastive Divergence techniques for simpler graphs – RBM
- But General BM is more representable than RBM
- The solution to this problem is an effective low-cost sampler for Boltzmann machine

  **Quantum Annealer**




Generative training with 4 hidden nodes

# Quantum Annealing




- The annealing procedure evolves energy on super-conducting qubits

$$E(t) = A(t)\sum_i S_i^x + B(t)\Big(\sum_i H_i S_i^z + \sum_{<i,j>} J_{ij} S_i^z S_j^z\Big)$$

- **Adiabatic theorem:** If this process is done slowly and band gap is positive at every point then state equilibrates to the ground state of blue Hamiltonian

- Ground state of Blue Hamiltonian same as that of classical spin energy

$$E(S) = \sum_i H_i S_i + \sum_{\langle ij \rangle} J_{ij} S_i S_j$$

**Benefits:**

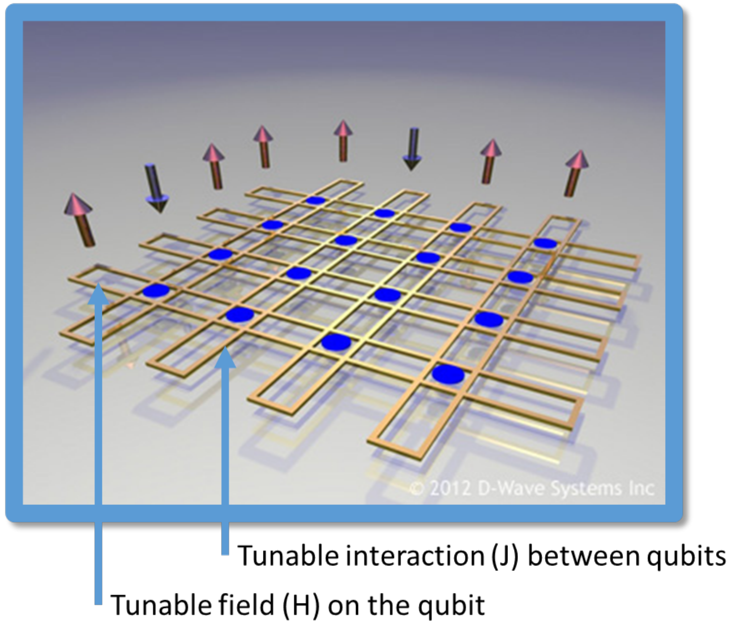
1. Finds the minimum in a single computation
2. Savings in energy consumption by reduced computation time

# Quantum Annealing

- Currently available hardware like D-Wave where parameters are tunable using analog controls

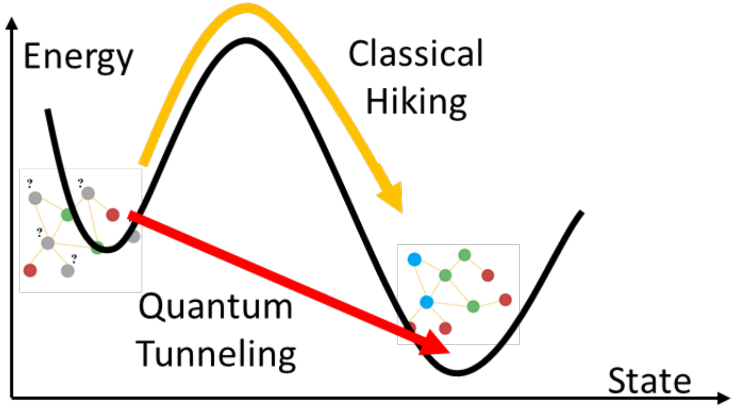
- Employs Quantum Annealing with short simulation time ($\sim 20 \mu s$) and finite temperature ($\sim 15 mK$)

- Adiabatic theorem no longer valid.




What does Quantum annealing give?

- Independent samples based on **Boltzmann distribution**



# Generative learning

**Estimate statistics from QA Samples**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

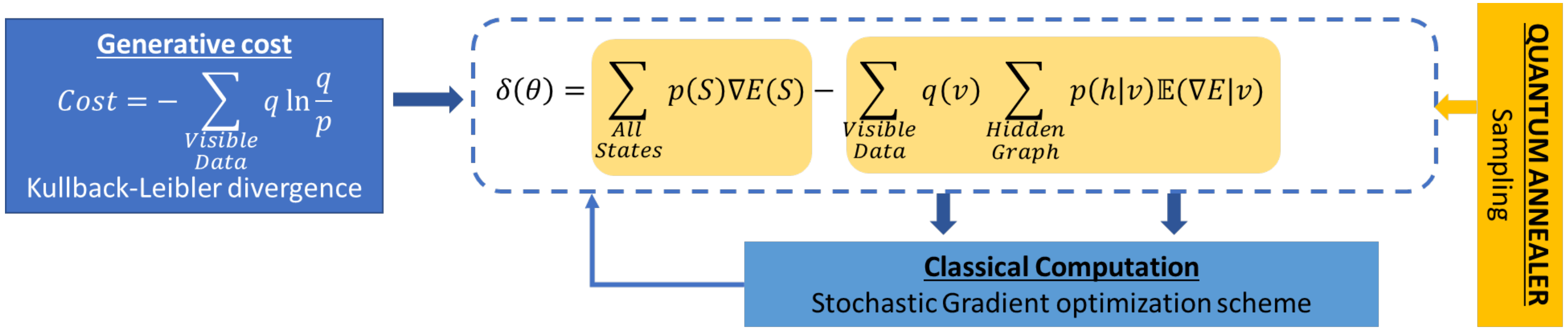
Data Set

- Cost function is chosen to be KL Divergence:

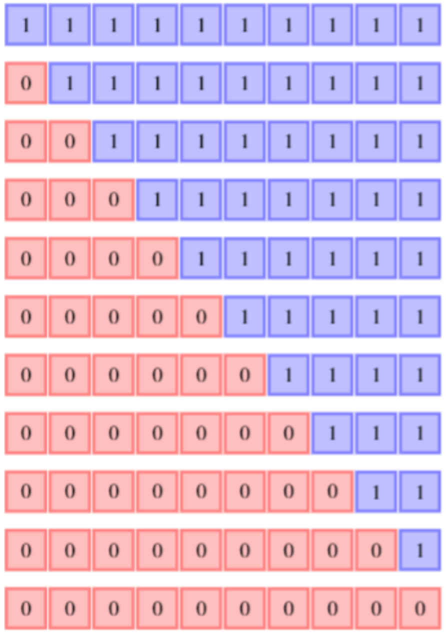
$$D_{KL} = \sum_{Visible\ data} q \log \frac{q}{p}$$

$$q = (\#\text{Data})^{-1},\ \text{p} = \text{Model probability}$$

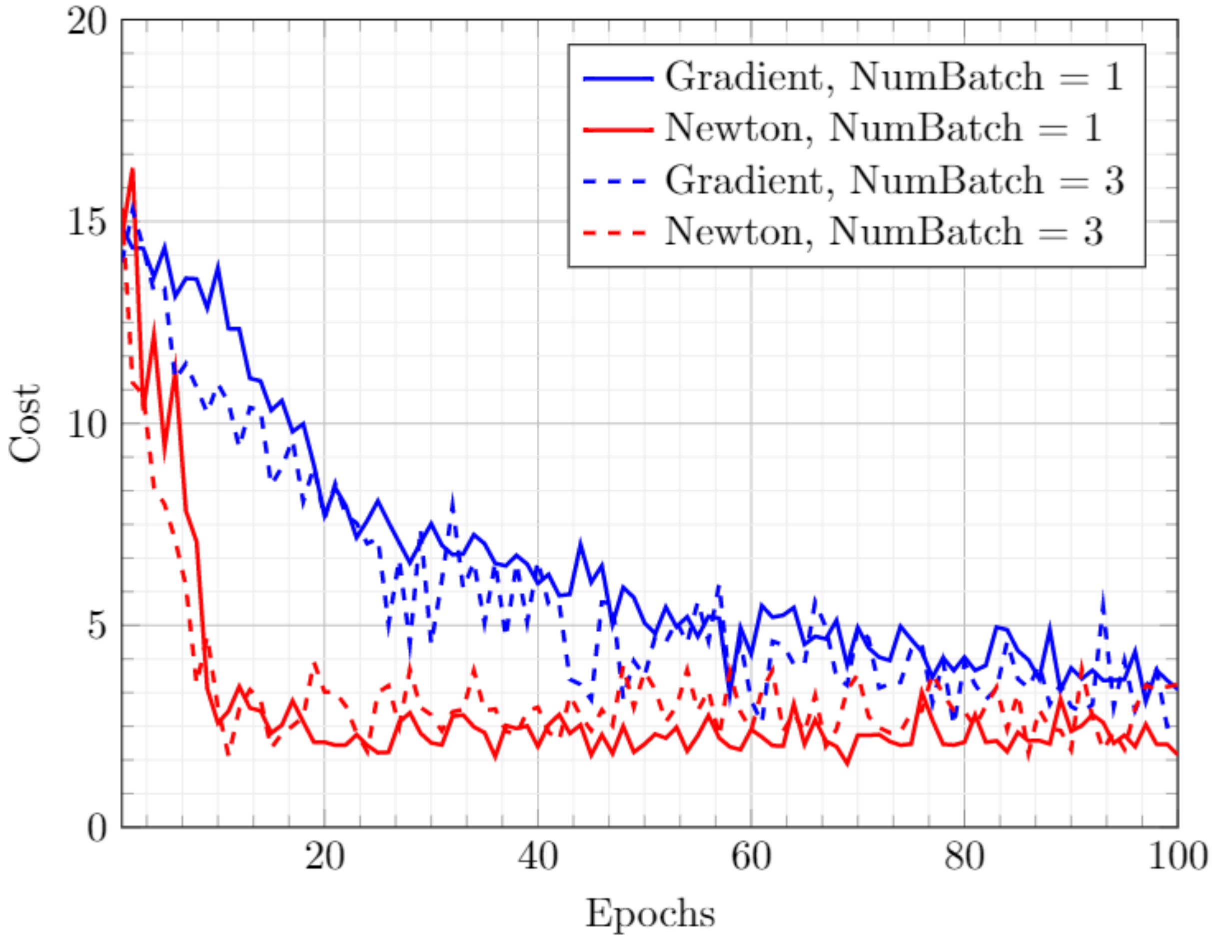
- Approximate gradients and even Hessian (in terms of Covariances) for a little premium on cost
- Use Stochastic Gradient/Newton method for optimization



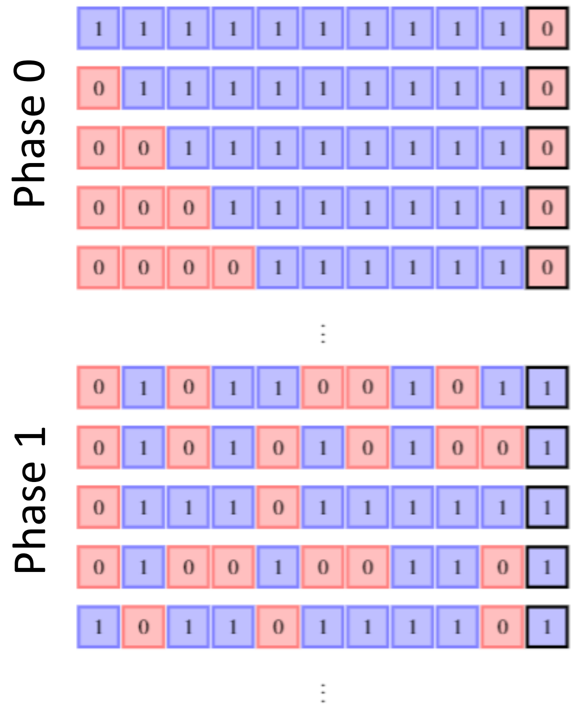
# Generative learning




Data Set



# Classification of state (Discriminative learning)



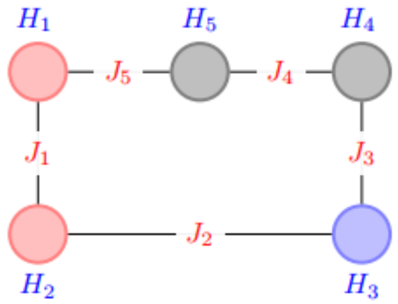

State $(x, f(x))$

Graph decomposed as:

(a) Visible Input (Pink)

(b) Visible output (Blue)

(c) Hidden (grey)



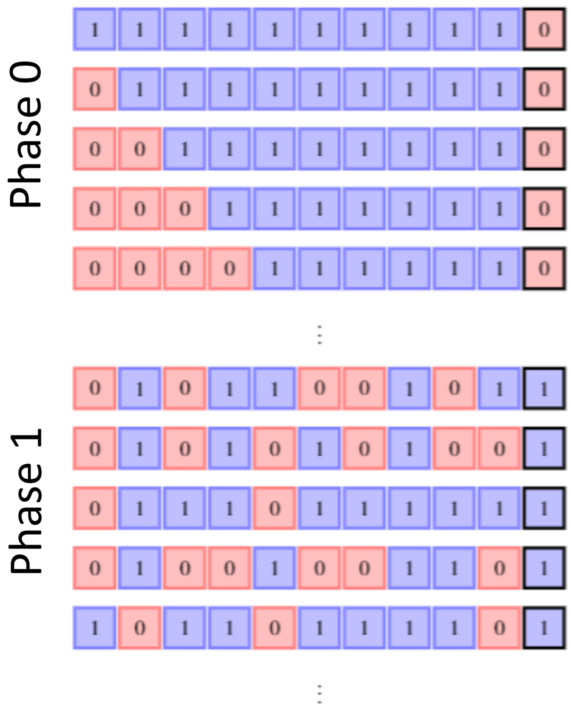
# Including classification cost

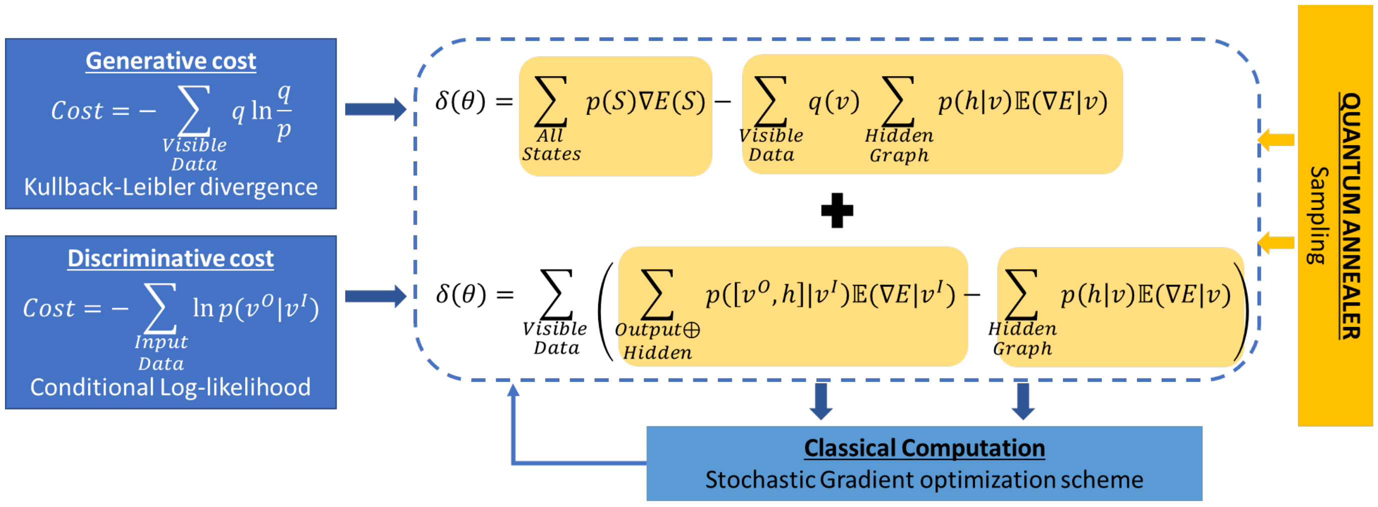
- Optimize for $p(f(x)\,|\,x)$

Negative Conditional Log-Likelihood

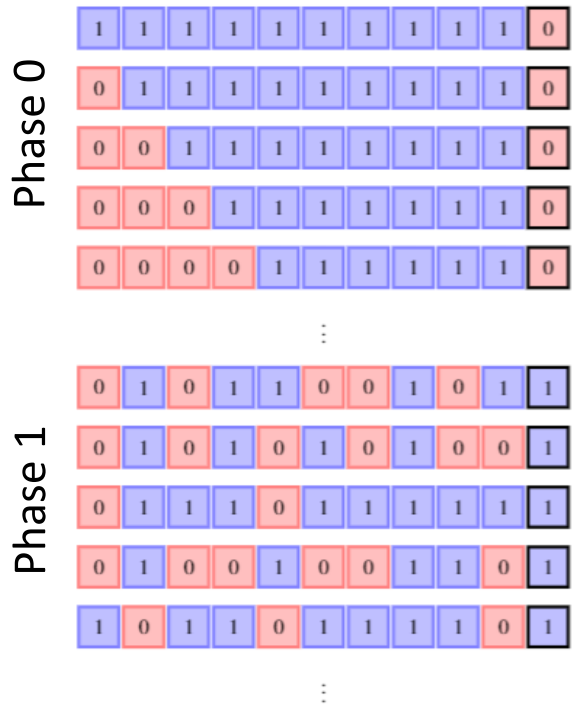
$$\mathcal{N} = \sum_{[x, f(x)] \in \{v^1, \dots, v^D\}} \log p(f(x)\,|\,x; \theta, \beta)$$

$$\rightarrow \quad Cost = \alpha D_{KL} + \frac{(1-\alpha)}{D}\mathcal{N}$$

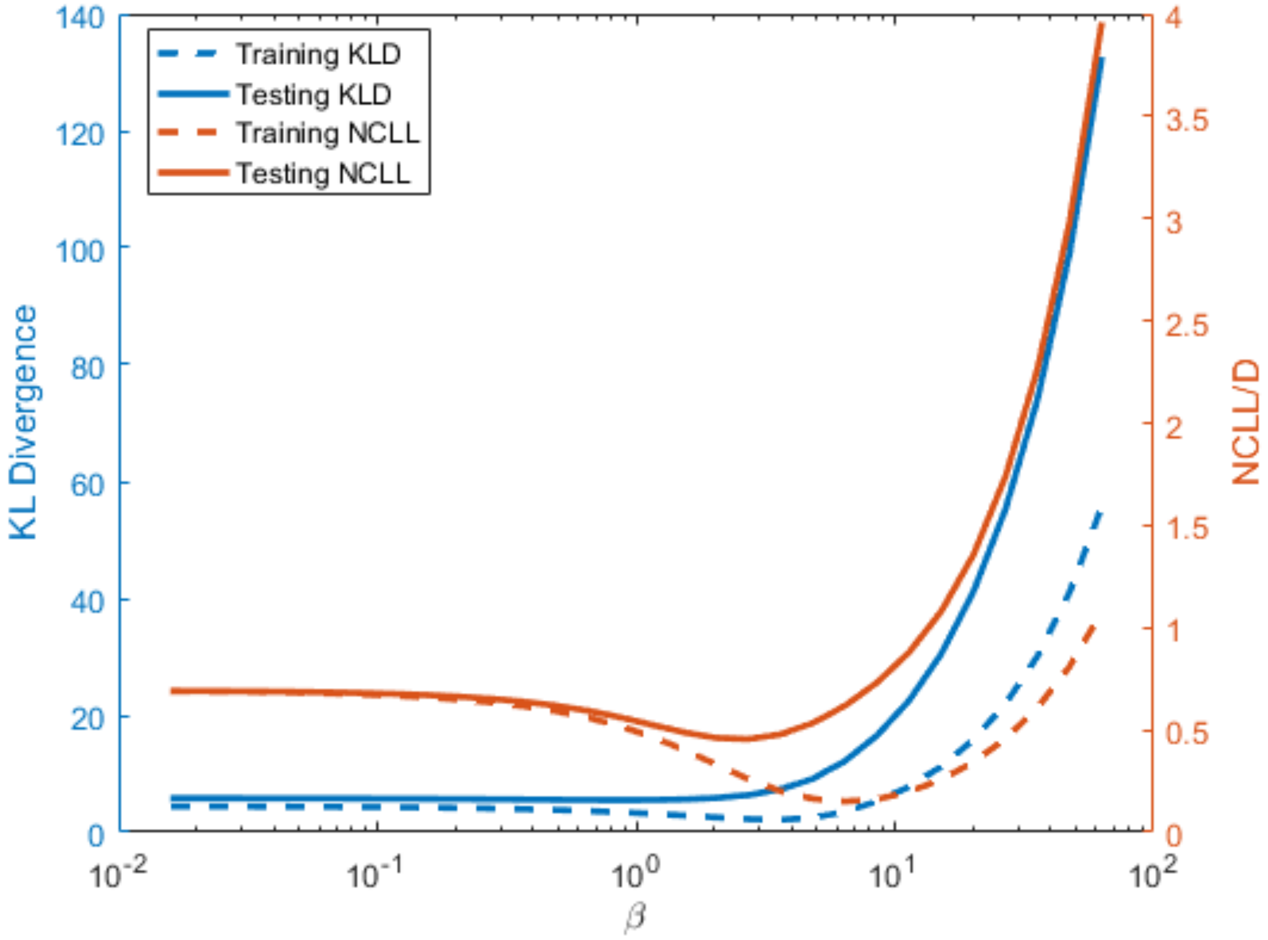


# Including classification cost




80/20 split of Training/Testing Data



# New challenge: Temperature ($\beta$) is unknown

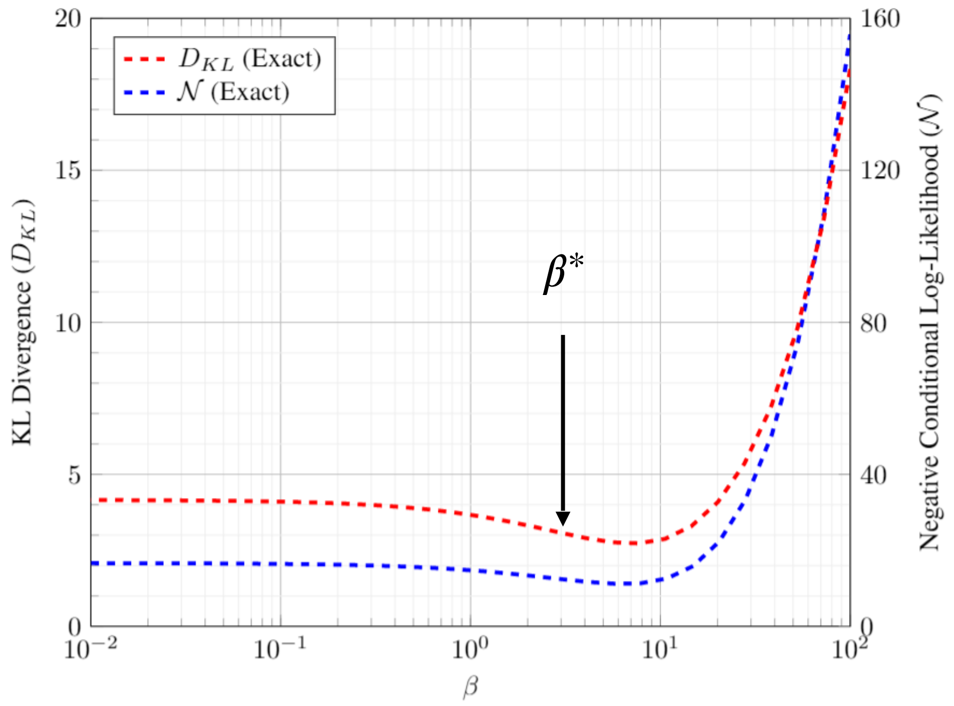
- Annealing temperature is unknown and dependent on the simulated graph.
- Need to evaluate $\beta$ to implement model in different machines

Observation: $\log p = -\beta E - \log Z$

Linear Regression:

$$\beta^* = -\frac{\sum (E - \mathbb{E}(E))\left(\log p - \mathbb{E}\left(\log p\right)\right)}{\sum (E - \mathbb{E}(E))^2}$$

Trained BM may not have the best performance at the Training temperature ($\beta^*$)



# Approximating the cost at different $\beta$

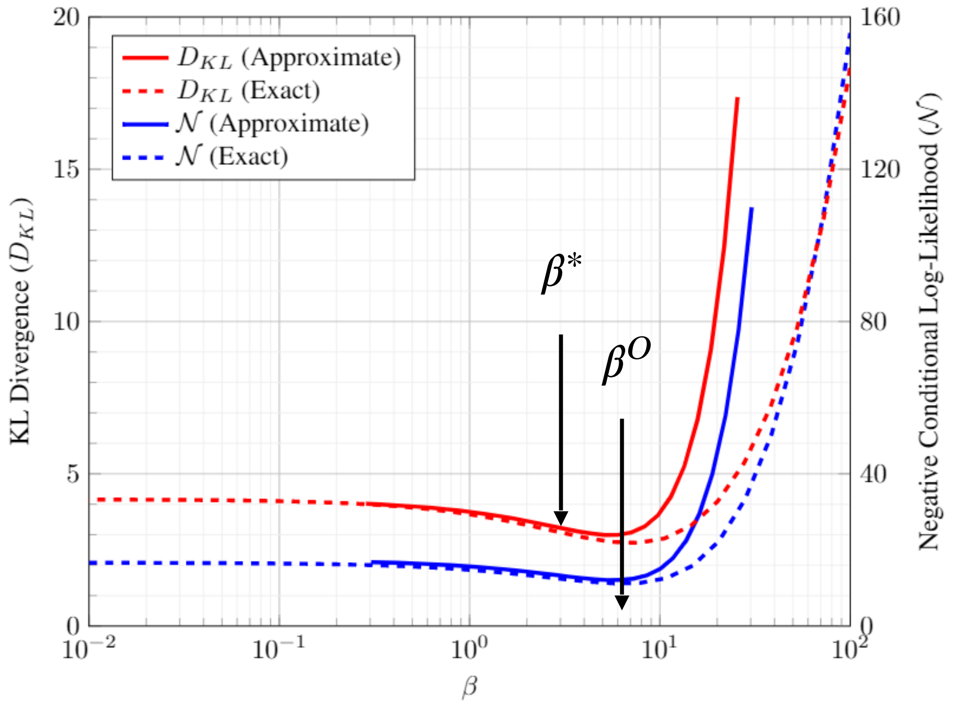
**Application:** Normalize parameters for best performance temperature

$$\theta \rightarrow \frac{\theta \beta^{O}}{\beta^{*}}, \qquad \beta^{O} = \text{optimal temperature}$$

Use Taylor expansion:

$$D_{KL}(\beta) = D_{KL}^{*} + \frac{\partial D_{KL}}{\partial \beta}\Big|_{\beta^{*}}(\beta - \beta^{*}) + \frac{1}{2}\frac{\partial^{2} D_{KL}}{\partial \beta^{2}}(\beta - \beta^{*}) + \ldots$$

- Coefficients estimated using sample statistics
- Similar results for NCLL cost




We have resolved the issue of transferability of the BM to different computing devices.

# Summary for Boltzmann Machine

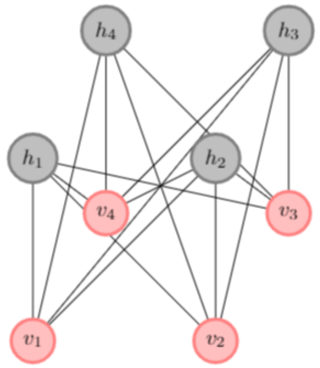
**State of the art:** Present training methods utilize topological features of a graph for reducing computational complexity

**Advantage of current work:** Training via QA samples works on a general BM. Sparse BMs enjoy additional computational advantages by allowing embedding of larger graphs in the hardware

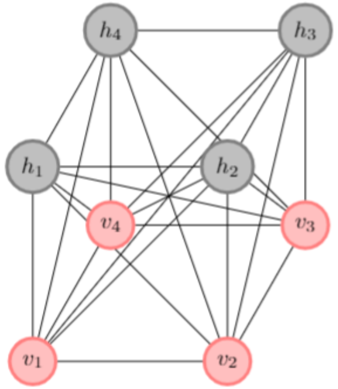
**Resolution of possible problems:** The issue of transferability of BM is resolved

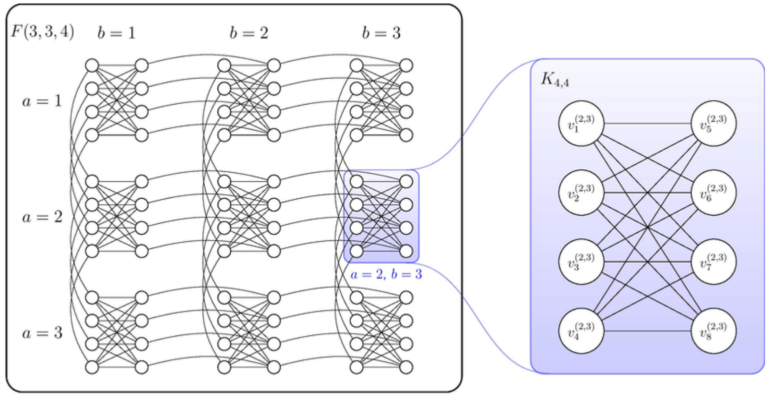
A MATLAB library is now available which implements this training method

**Future work:** As a next step, we will apply this method for problems concerning Process-Structure-Property (PSP) linkages in materials science




vs




| | Clique | NAE3SAT ($r = 3$) | NAE3SAT ($r = 2.1$) | 3-Regular | 3D Lattice w/defects | Native |
|---|---|---|---|---|---|---|
| 2000Q | 64 | 90 | 102 | 304 | 512 | 2030 |
| Advantage | 124 | 242 | 286 | 784 | 2354 | 5455 |

Thank you