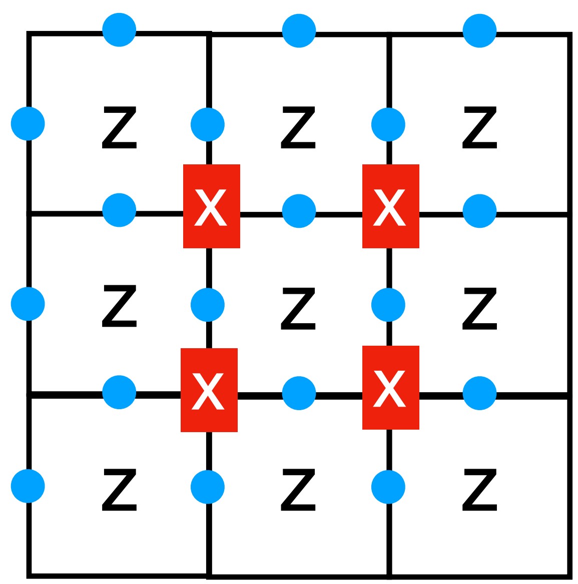
Designing neural decoders for large toric codes

Xiaotong Ni (TU Delft)

Erlangen, May 9

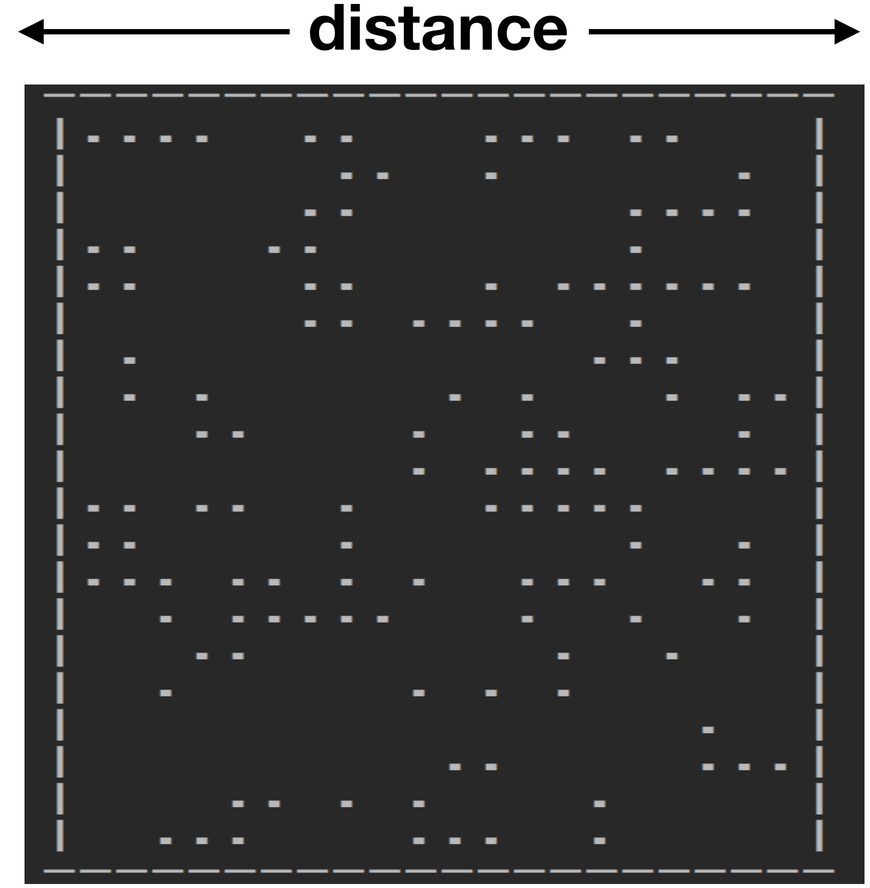
arXiv: 1809.06640

Toric code



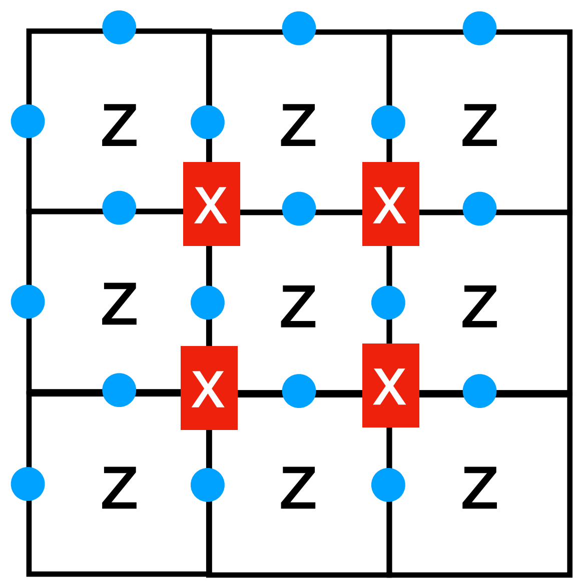
Assuming only Pauli-X error on qubits
Syndrome extraction is noiseless

code distance



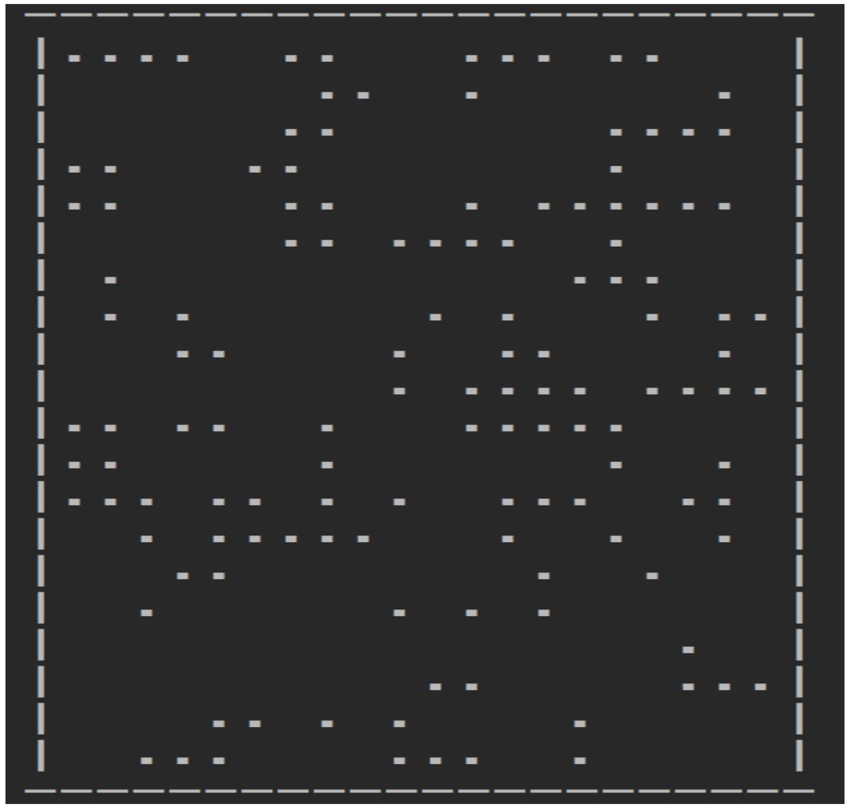
10% error rate

Toric code



Assuming only Pauli-X error on qubits
Syndrome extraction is noiseless

code distance



10% error rate

- Correction is sensitive to the small change of syndrome, and involves a lot of parity computation.

Some previous works of neural decoders

G. Torlai and R. G. Melko, Physical Review Letters (2017).

S. Krastanov and L. Jiang, Scientific Reports 7 (2017)

Repeatedly sample error configurations using RBM / NN

P. Baireuther, T. E. O'Brien, B. Tarasinski, and C. W. J. Beenakker, Quantum (2018)

Keep track of the Pauli frames in a simulated circuit level noise model

Main result

- Introduced a quite reliable way to build neural decoders for large distance toric code (and likely many other topological codes)
- Previous works: d~10 —> This work d=64,…
- Source code (and a Google Colab script) can be found at https://github.com/XiaotongNi/toric-code-neural-decoder

Motivation

Motivation

- Same as several previous works

Motivation

- Same as several previous works
  - A convenient way to adapt to experimental noise models

Motivation

- Same as several previous works
  - A convenient way to adapt to experimental noise models
  - Offer a nice combination of speed and accuracy for topological codes.

Motivation

- Same as several previous works
  - A convenient way to adapt to experimental noise models
  - Offer a nice combination of speed and accuracy for topological codes.
- Go to quite large input size / code distance

Motivation

- Same as several previous works
  - A convenient way to adapt to experimental noise models
  - Offer a nice combination of speed and accuracy for topological codes.
- Go to quite large input size / code distance
  - Traditional decoders works for arbitrary size.

Motivation

- Same as several previous works
  - A convenient way to adapt to experimental noise models
  - Offer a nice combination of speed and accuracy for topological codes.
- Go to quite large input size / code distance
  - Traditional decoders works for arbitrary size.
  - Test versatility of NN / Find ways to overcome difficulties

Implementation

“Imitation learning”

Fast Decoders for Topological Quantum Codes

Guillaume Duclos-Cianci and David Poulin
Département de Physique, Université de Sherbrooke, Québec, Canada
(Received 3 November 2009; published 5 February 2010)

We present a family of algorithms, combining real-space renormalization methods and belief propagation, to estimate the free energy of a topologically ordered system in the presence of defects. Such an algorithm is needed to preserve the quantum information stored in the ground space of a topologically ordered system and to decode topological error-correcting codes. For a system of linear size ℓ, our algorithm runs in time $\log \ell$ compared to ℓ^6 needed for the minimum-weight perfect matching algorithm previously used in this context and achieves a higher depolarizing error threshold.

DOI: 10.1103/PhysRevLett.104.050504 PACS numbers: 03.67.Ac, 03.65.Vf, 03.67.Pp, 05.50.+q

1. Approximate the decoder in the paper.

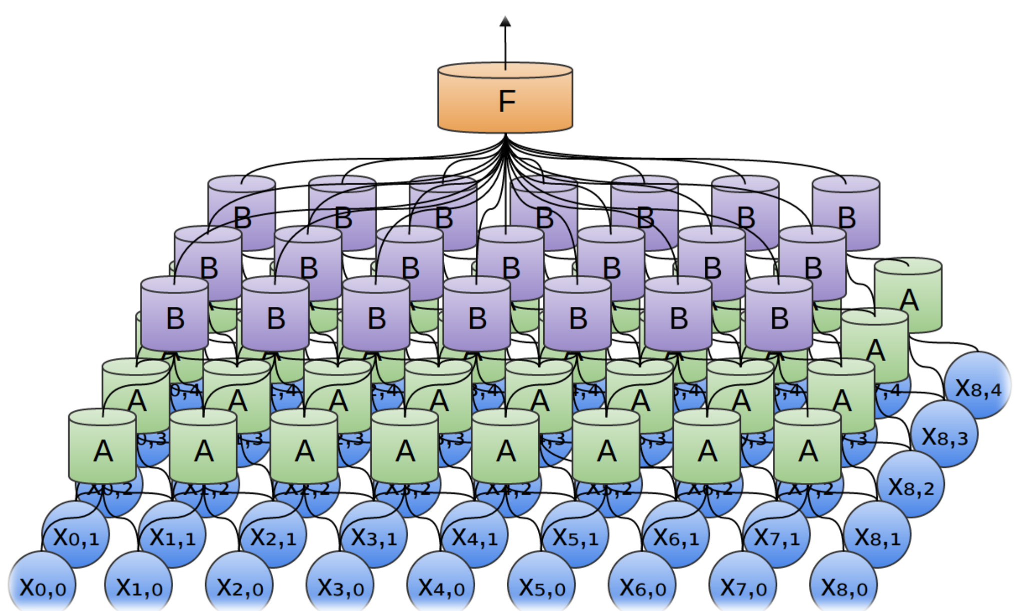


Image from Colah's blog

“Imitation learning”

Fast Decoders for Topological Quantum Codes

Guillaume Duclos-Cianci and David Poulin
Département de Physique, Université de Sherbrooke, Québec, Canada
(Received 3 November 2009; published 5 February 2010)

We present a family of algorithms, combining real-space renormalization methods and belief propagation, to estimate the free energy of a topologically ordered system in the presence of defects. Such an algorithm is needed to preserve the quantum information stored in the ground space of a topologically ordered system and to decode topological error-correcting codes. For a system of linear size ℓ, our algorithm runs in time $\log\ell$ compared to ℓ^6 needed for the minimum-weight perfect matching algorithm previously used in this context and achieves a higher depolarizing error threshold.

DOI: 10.1103/PhysRevLett.104.050504 PACS numbers: 03.67.Ac, 03.65.Vf, 03.67.Pp, 05.50.+q

A Better decoder

1. Approximate the decoder in the paper.

2. Further optimization with (syndrome, correction) pairs

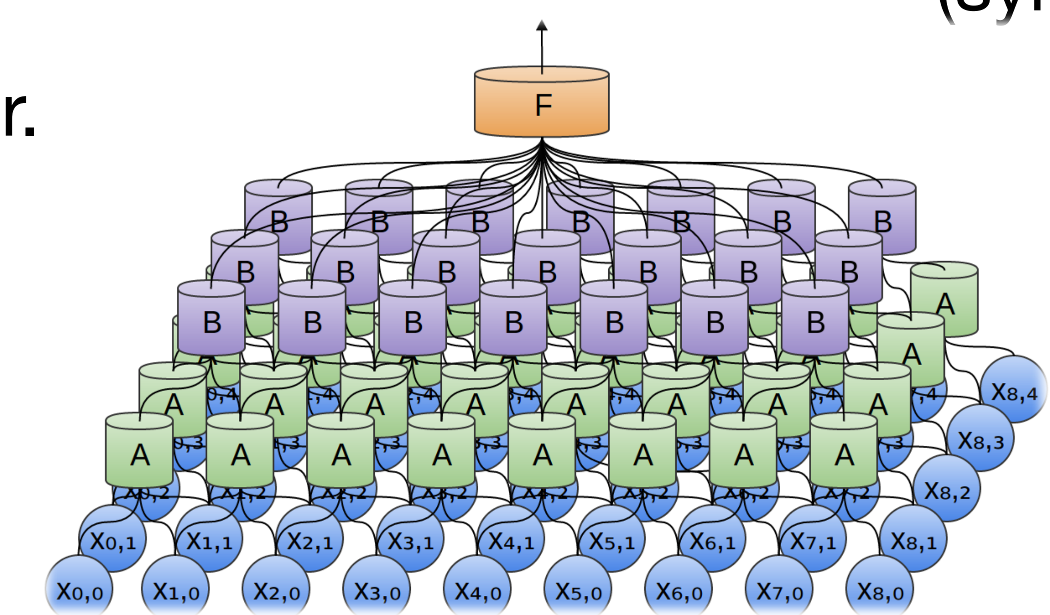
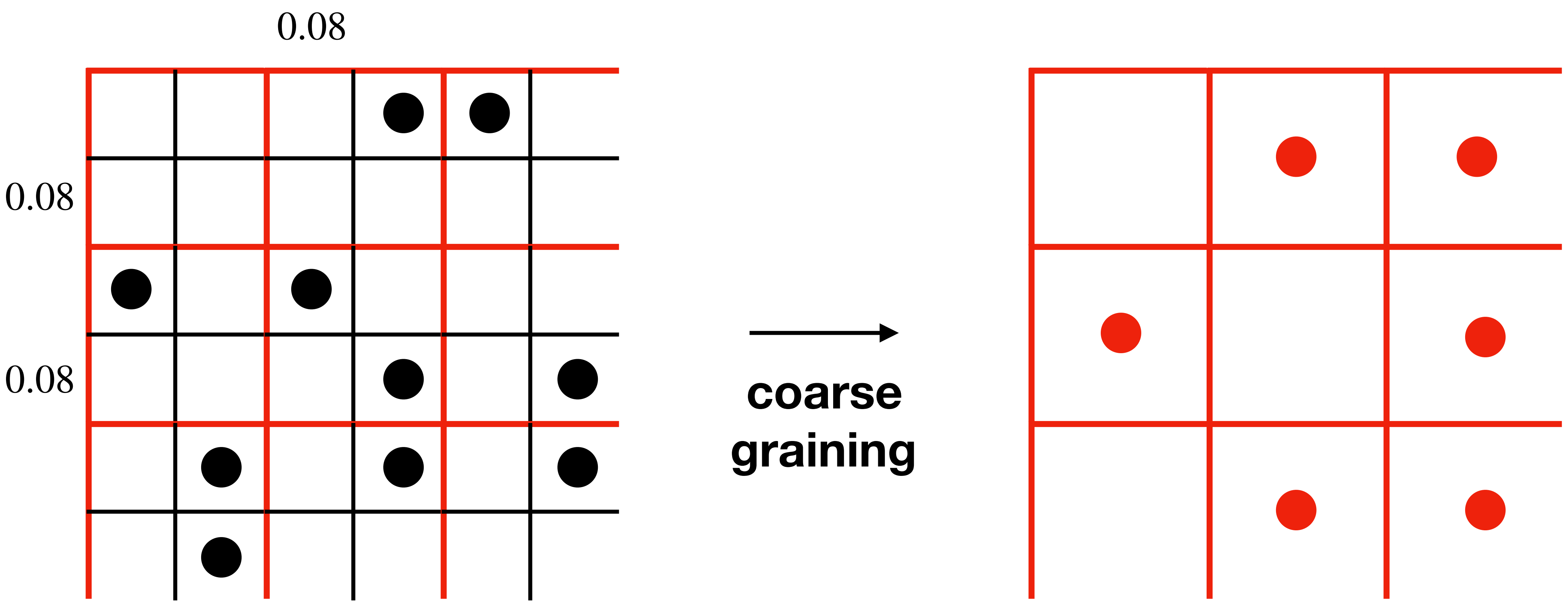
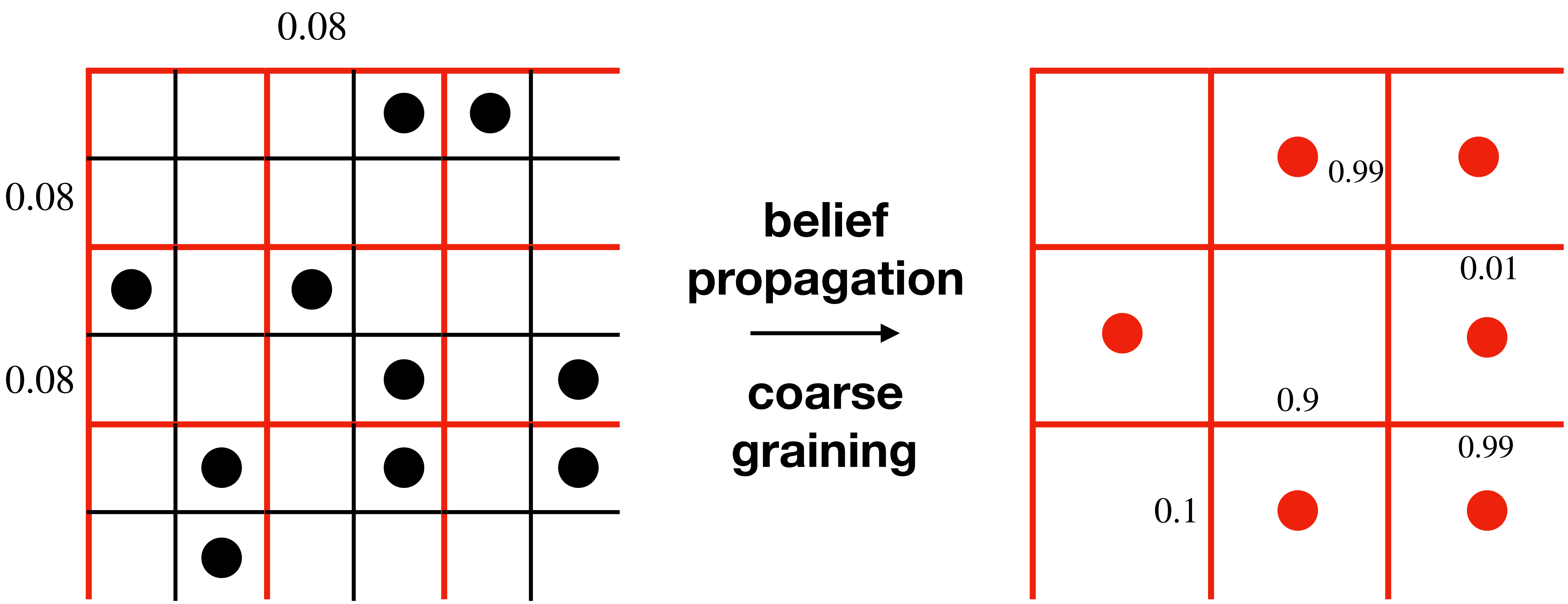


Image from Colah's blog

RG decoder

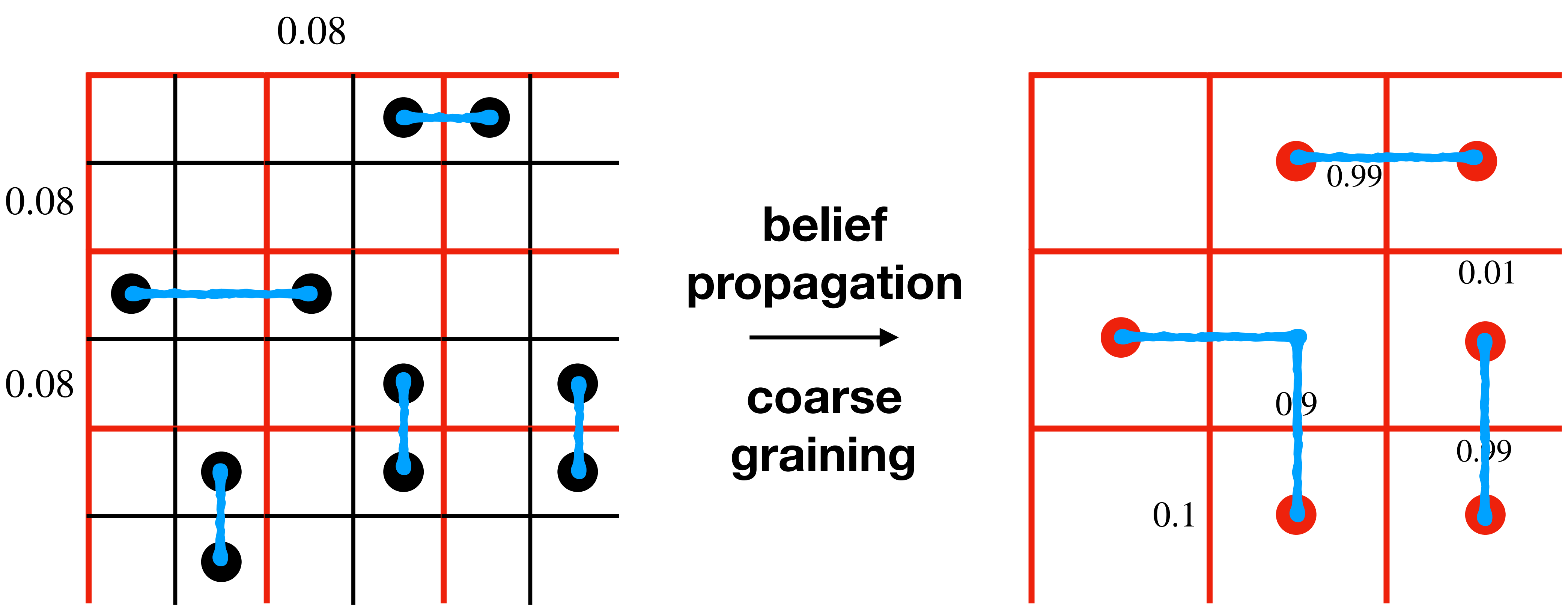


RG decoder



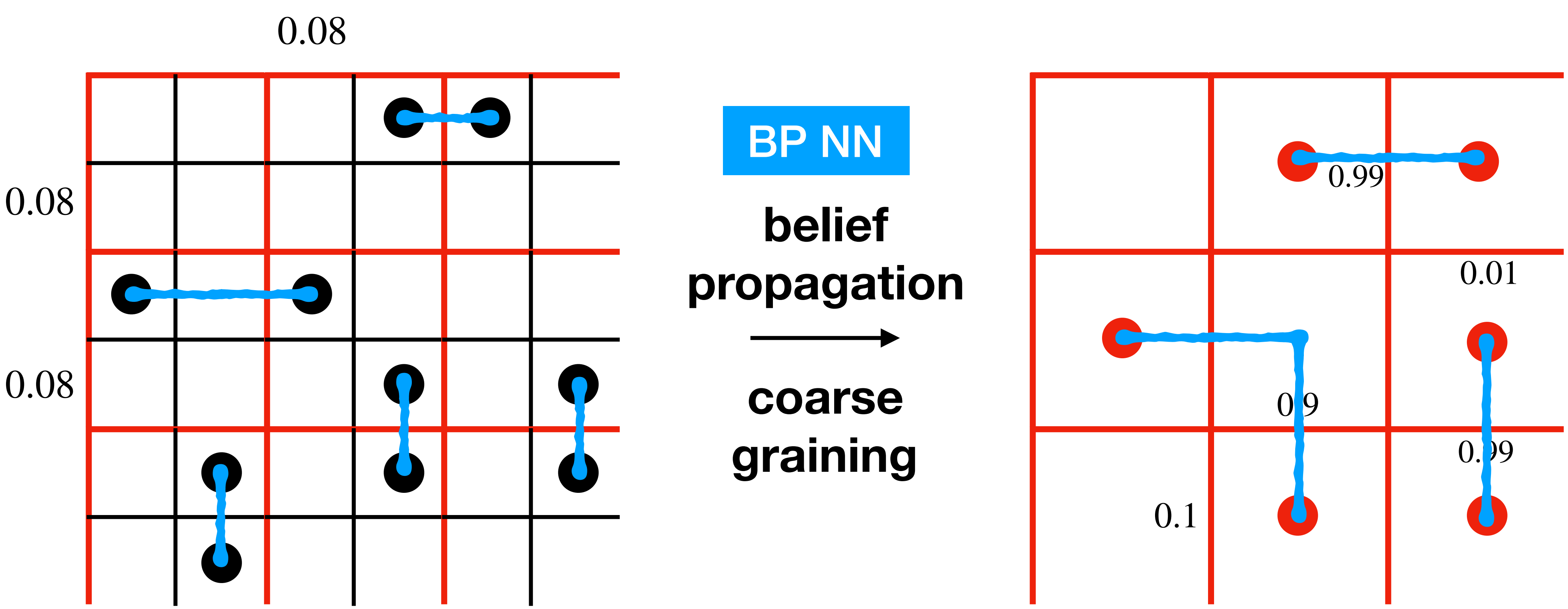
- Replace detailed syndrome information with educated guess on the error rates of “coarse-grained qubits”.
- Educated guess done by belief propagation.

RG decoder



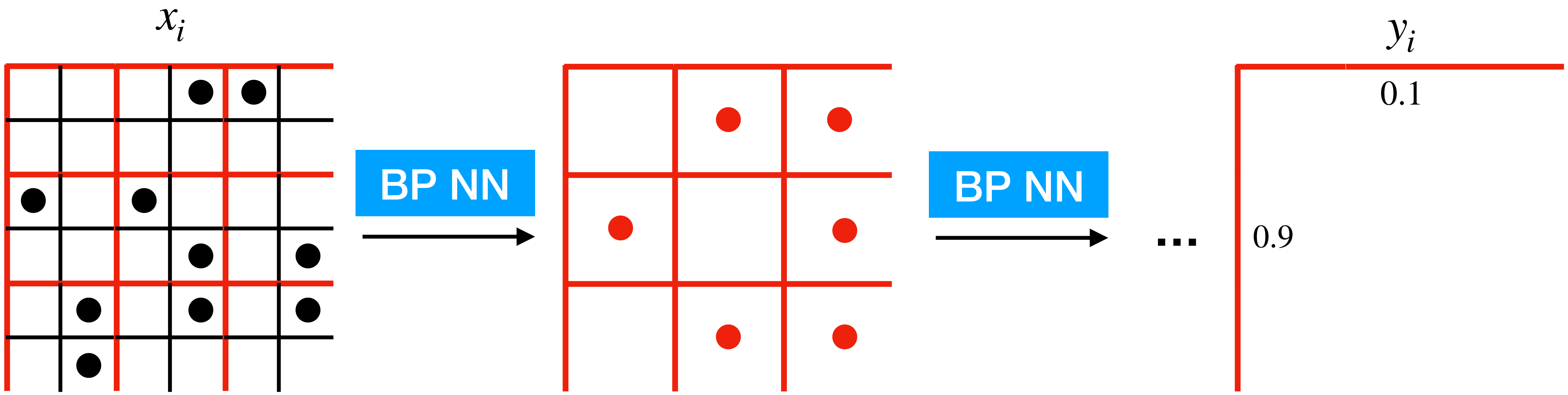
- Replace detailed syndrome information with educated guess on the error rates of “coarse-grained qubits”.
- Educated guess done by belief propagation.

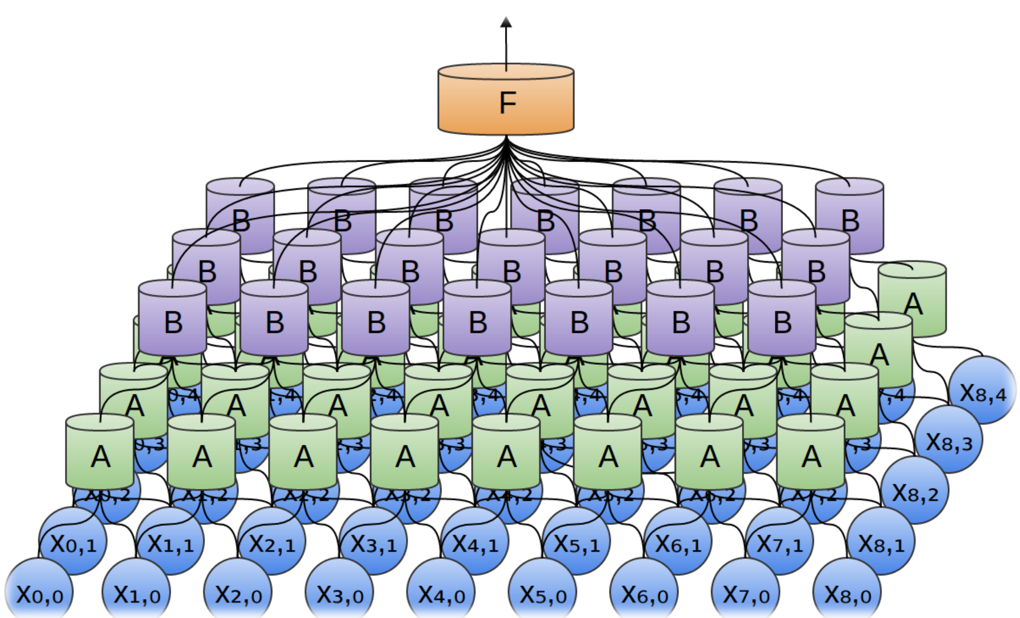
RG decoder



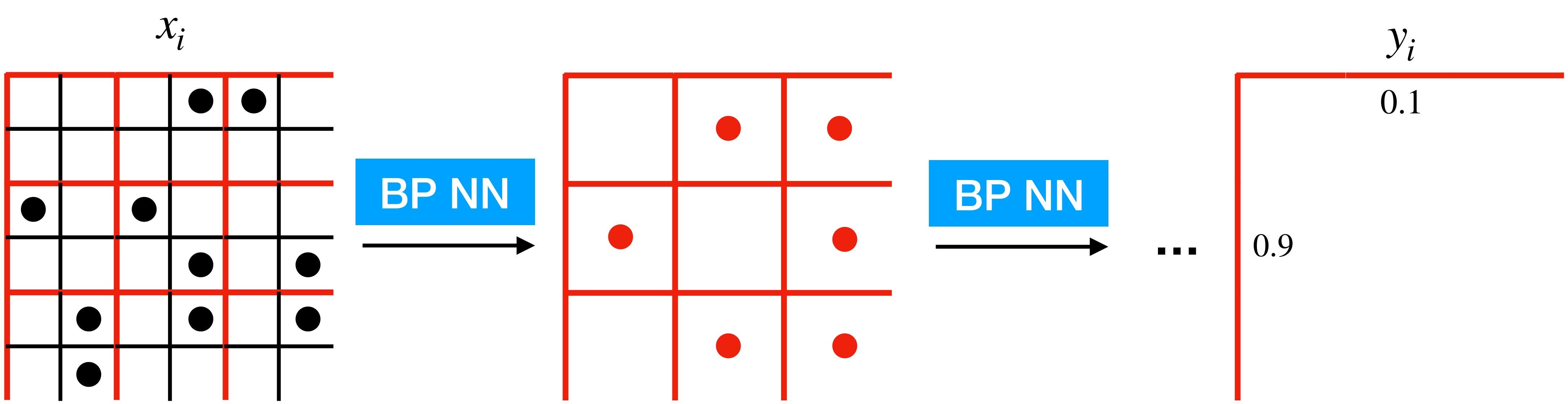
- Replace detailed syndrome information with educated guess on the error rates of "coarse-grained qubits".
- Educated guess done by belief propagation.

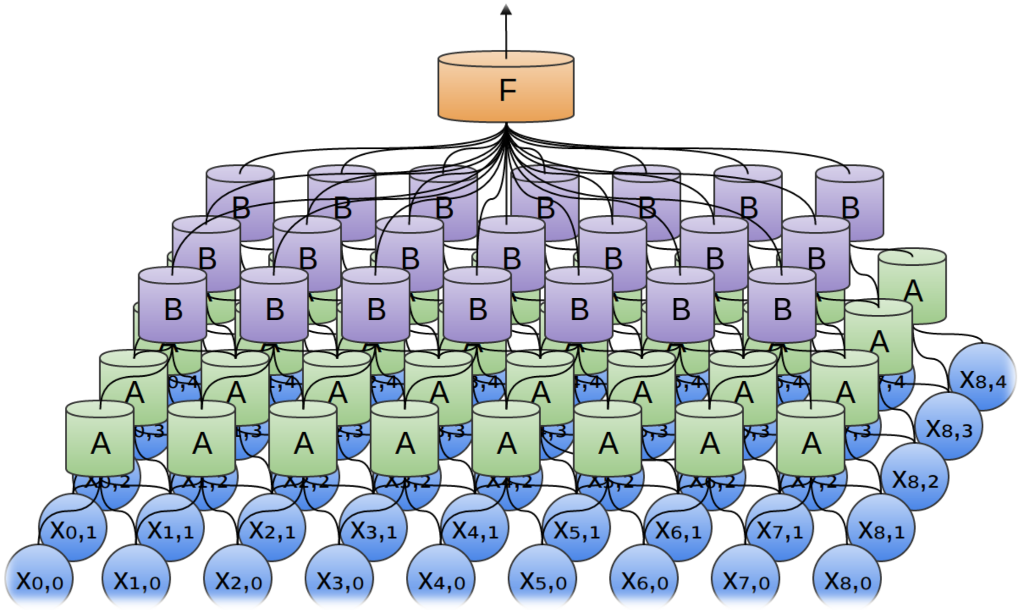
Stack NN together and train





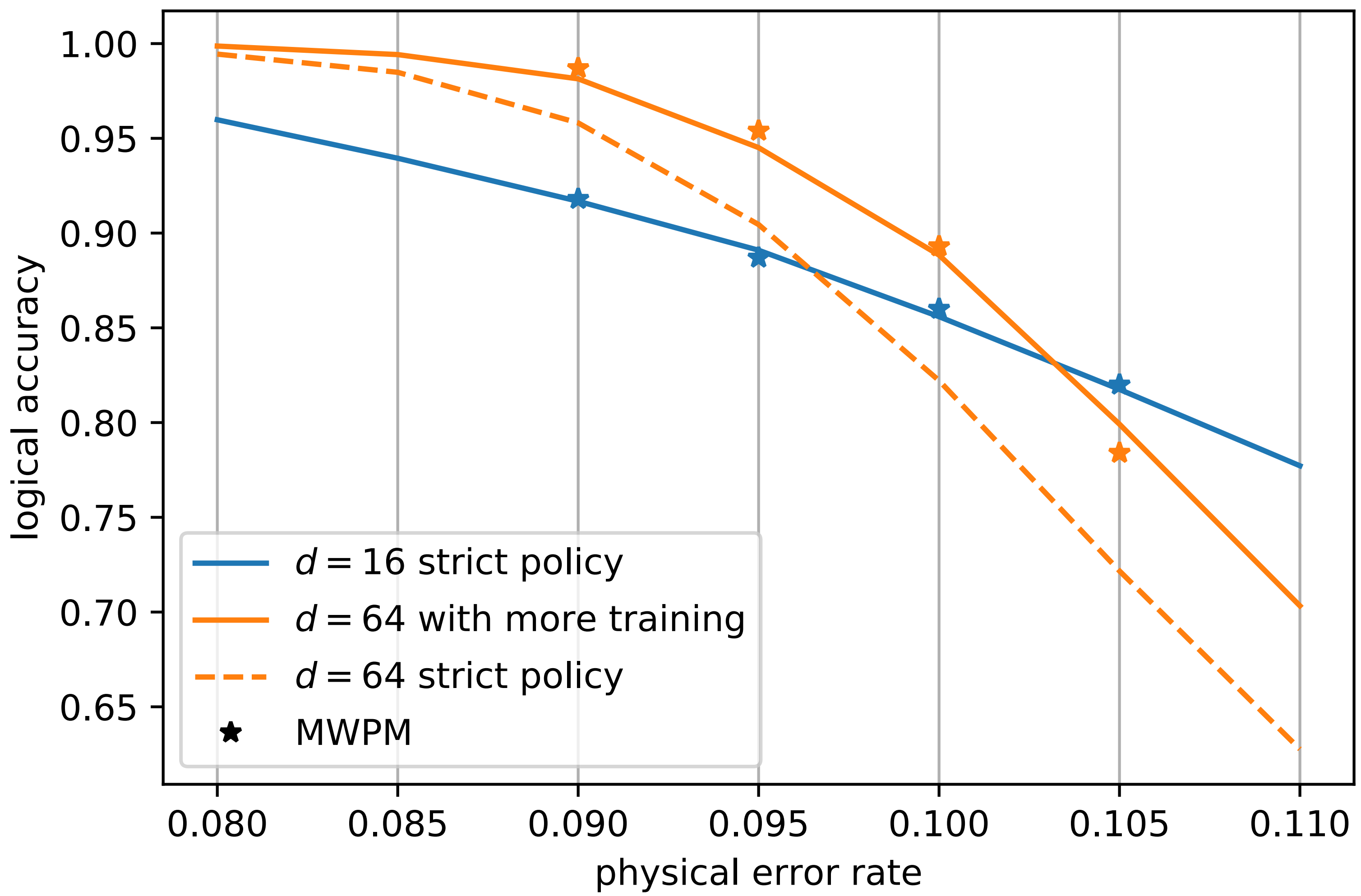
Stack NN together and train





We can then train all layers altogether, using (syndrome, correction) pairs.

Performance



With less than 2 hours of training time, we can get similar performance compared to minimum-weight perfect matching on bit-flip noise.

About training

Offline training

Neural decoder

Any reasonable amount of training data and time

Goal: good decoder on noise model similar to experiments, or decoders that can utilize error rate information

About training

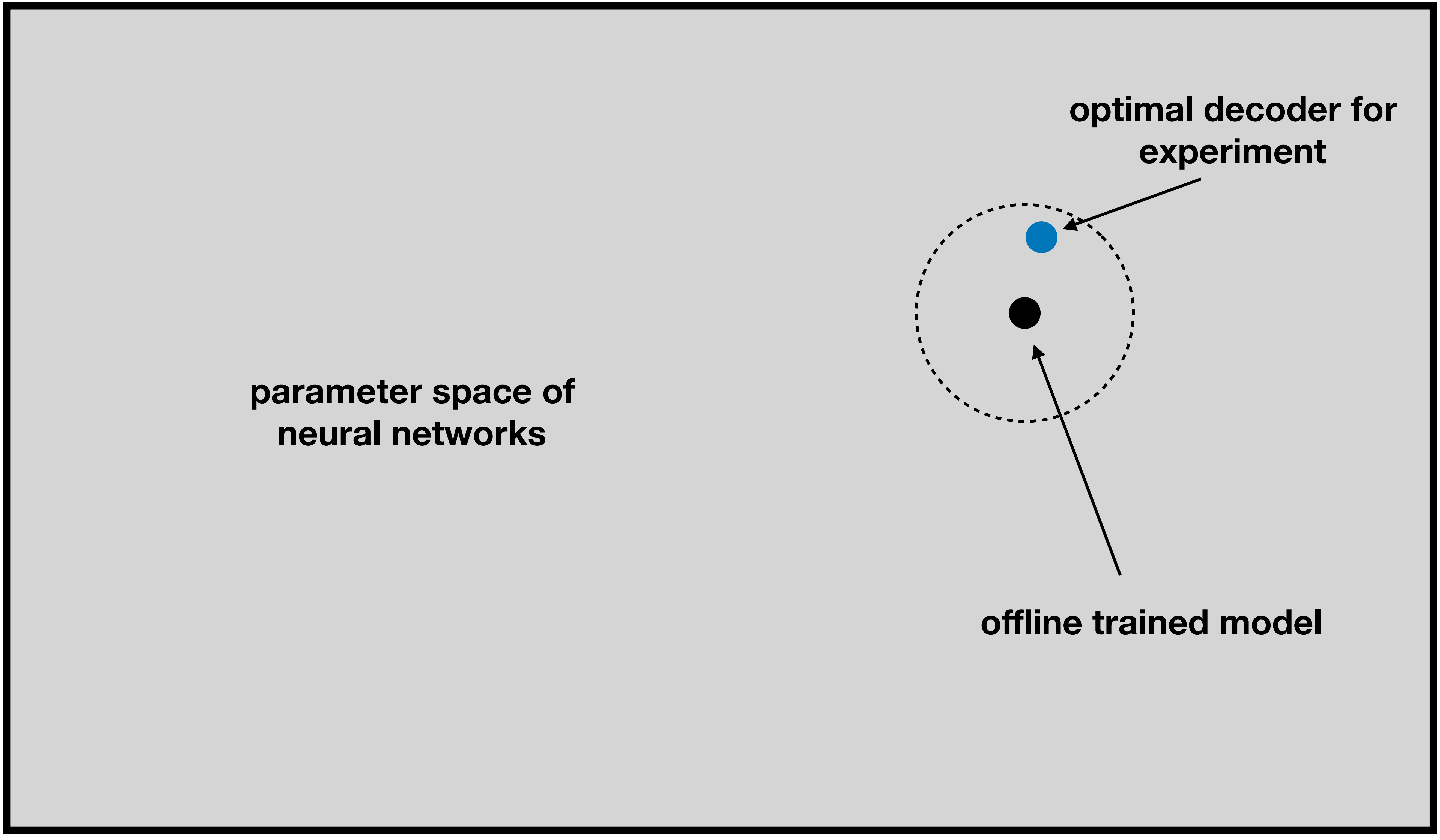
Offline training | **Training with experimental data**

Neural decoder → Neural decoder

Any reasonable amount of training data and time

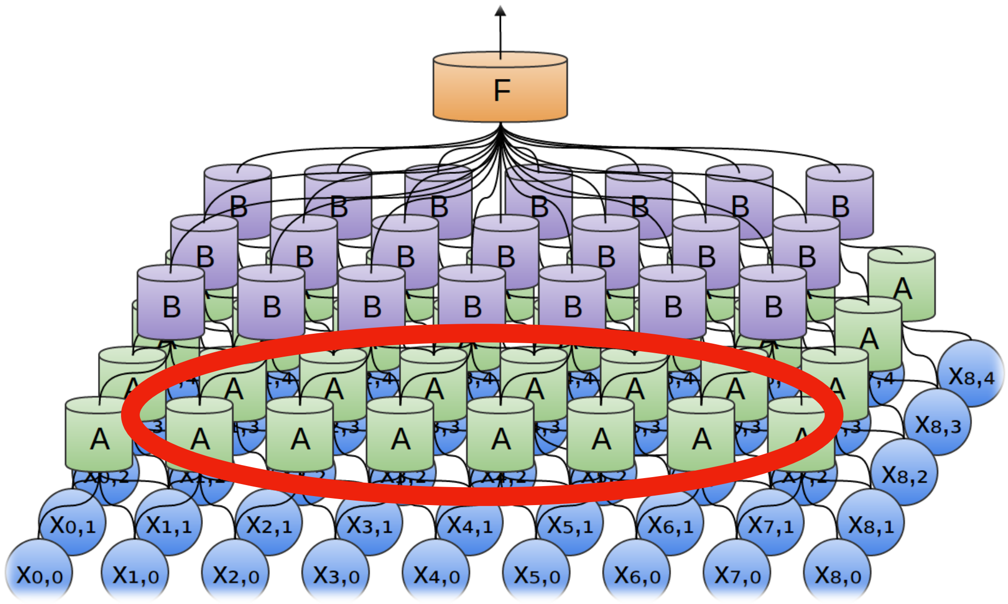
Limited amount of training data (not focused in this work)

Goal: good decoder on noise model similar to experiments, or decoders that can utilize error rate information


optimal decoder for experiment
parameter space of neural networks
offline trained model

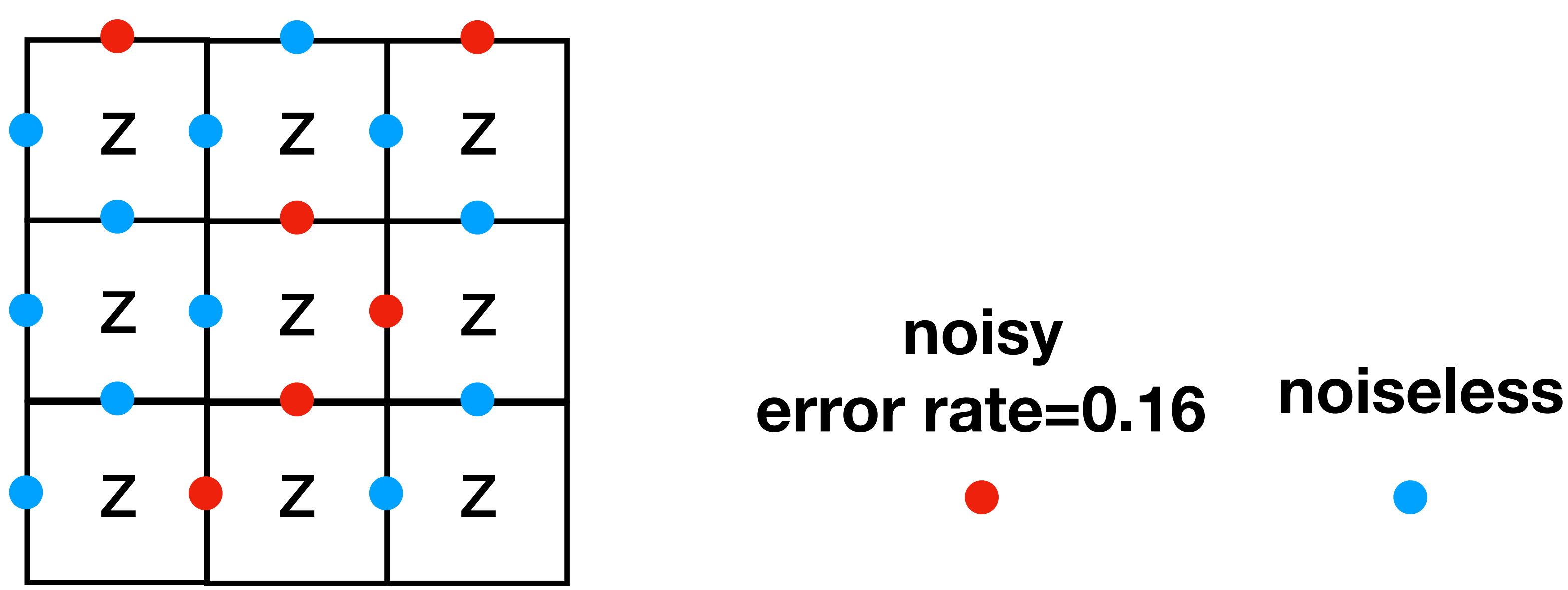
Adapt to other noise models

- Small changes in the noise model likely can be compensated by small changes in the early layers (need to break the translational symmetry in some way)

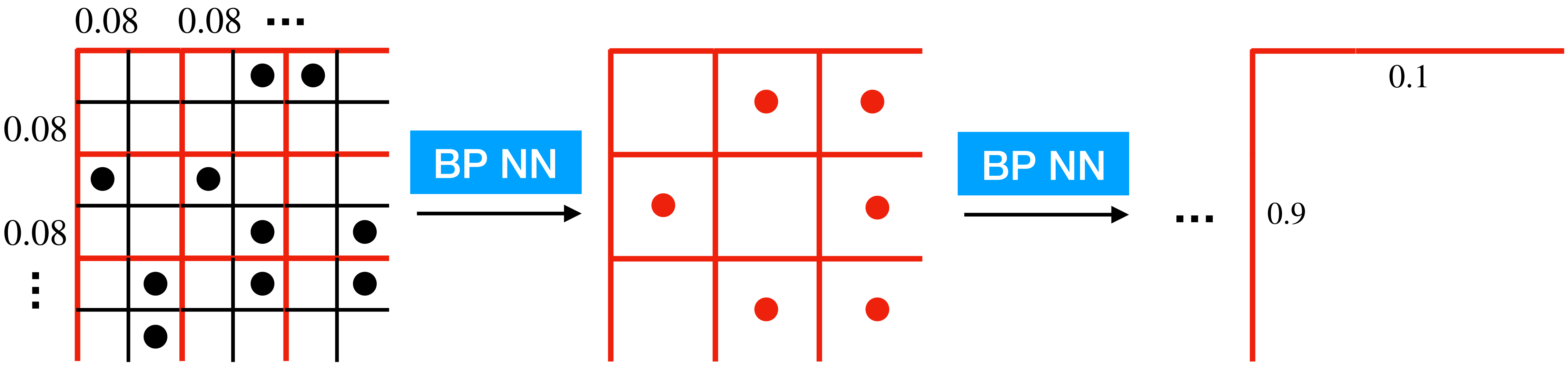


Spatially varying error rates

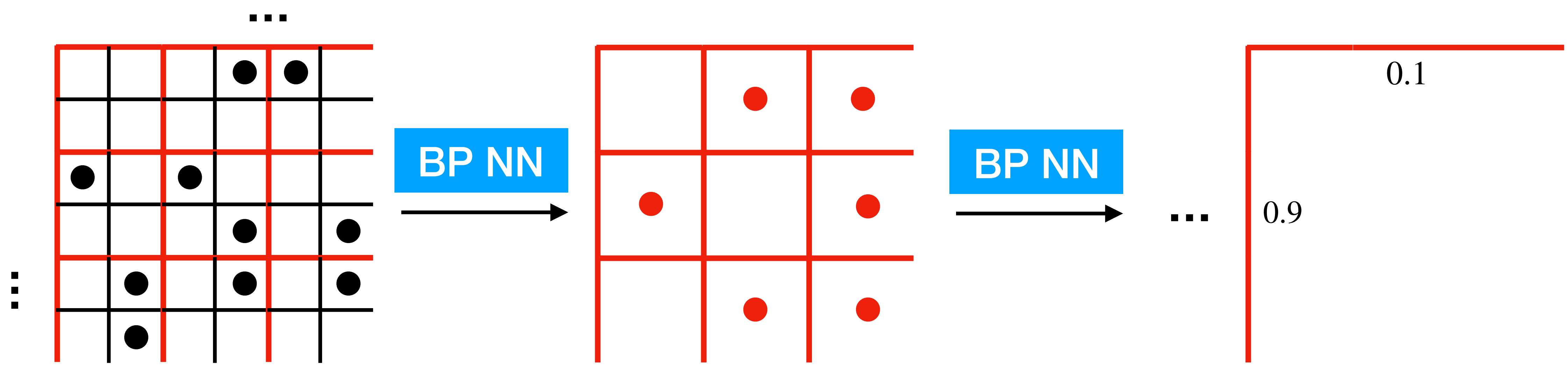
- Proof of principle for a toy noise model



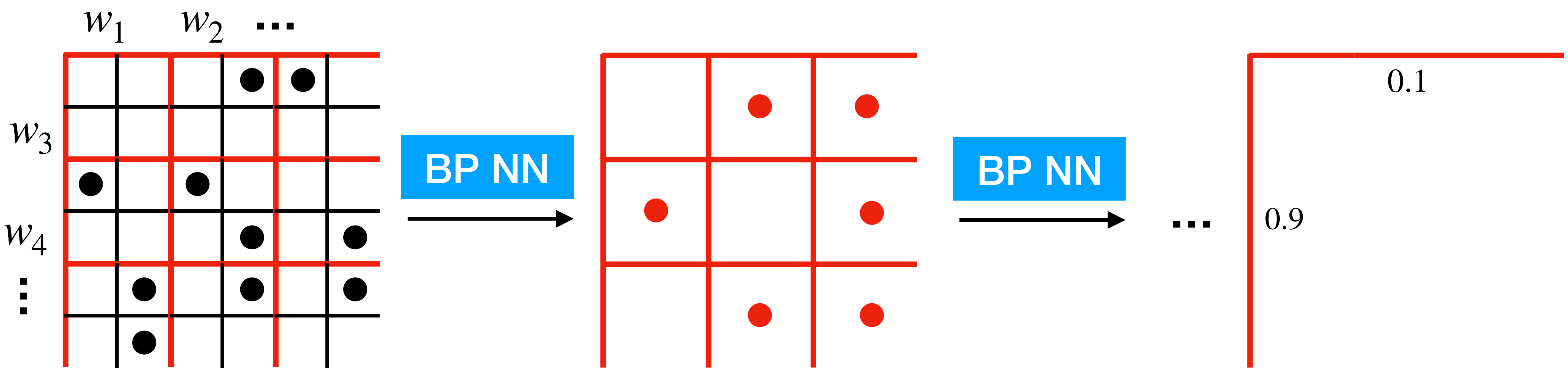
How it is done



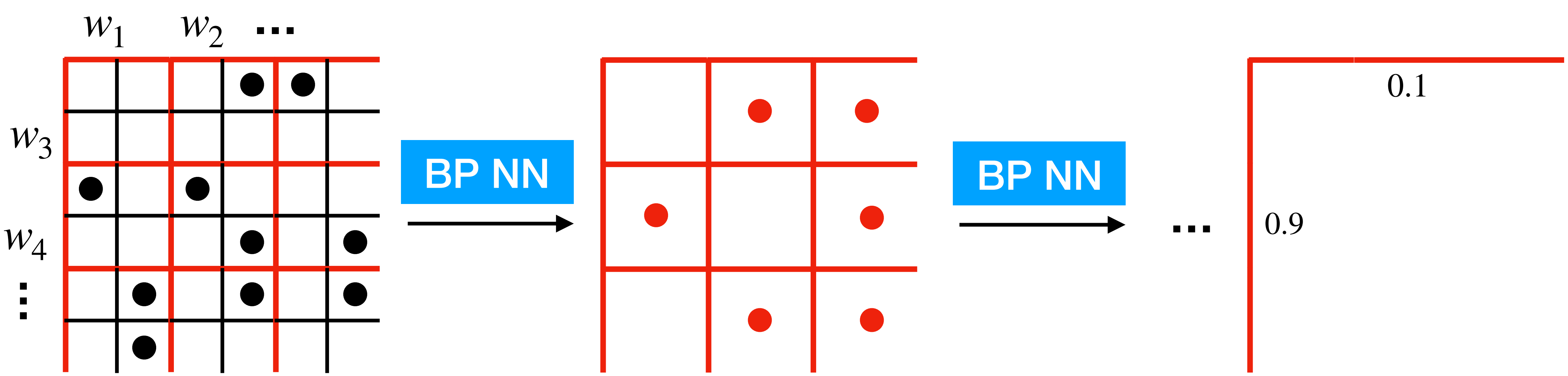
How it is done



How it is done

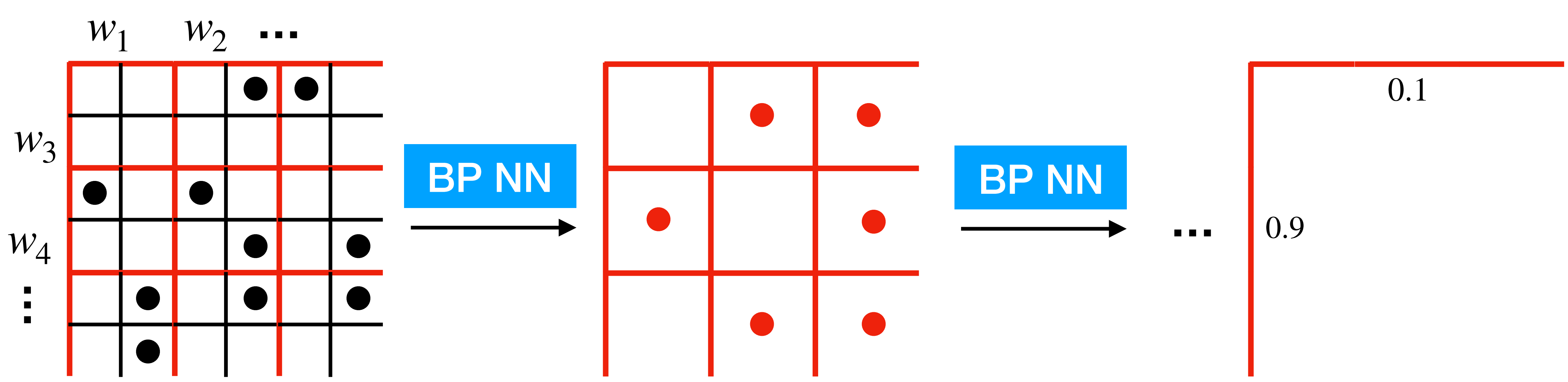


How it is done



Roughly speaking: fix parameters of the neural decoder and only train $\{w_i\}$

How it is done



Roughly speaking: fix parameters of the neural decoder and only train $\{w_i\}$

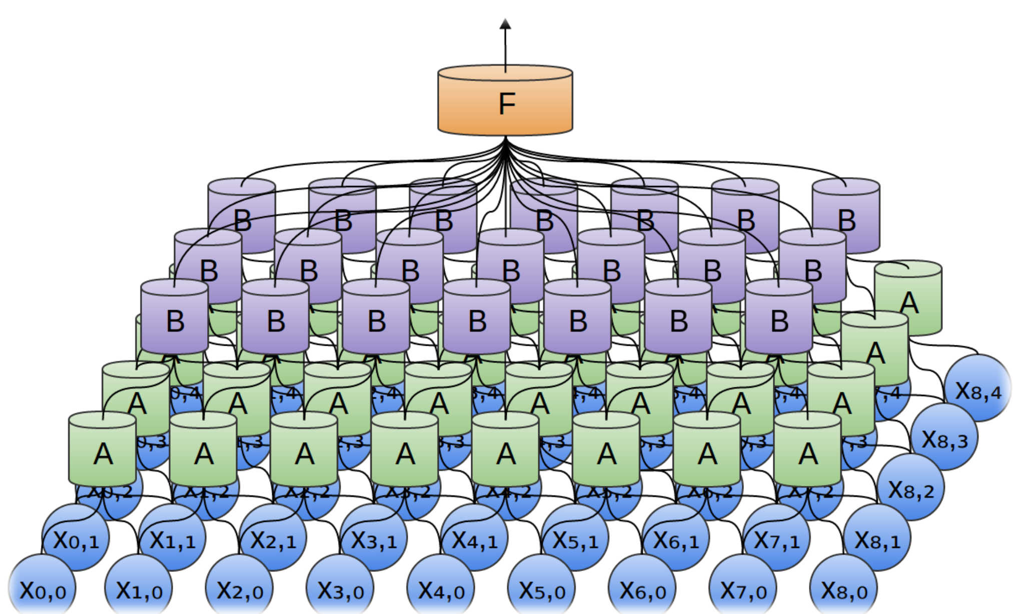
- Increase the logical accuracy from 0.967 to 0.993 after trained on this noise model (d=16)

General takeaway message

“Imitation learning”

Heuristic algorithm

1. Approximate



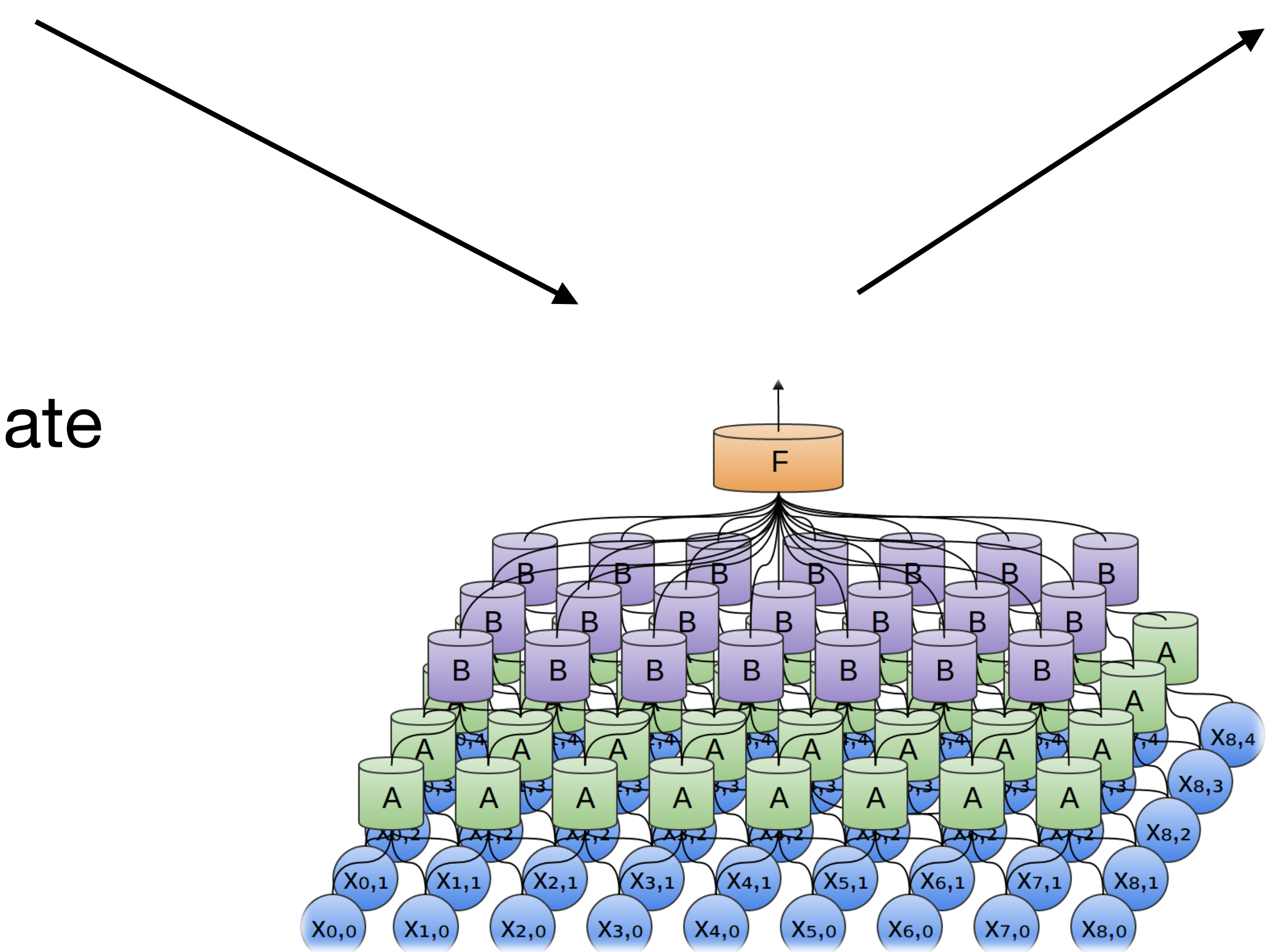
“Imitation learning”

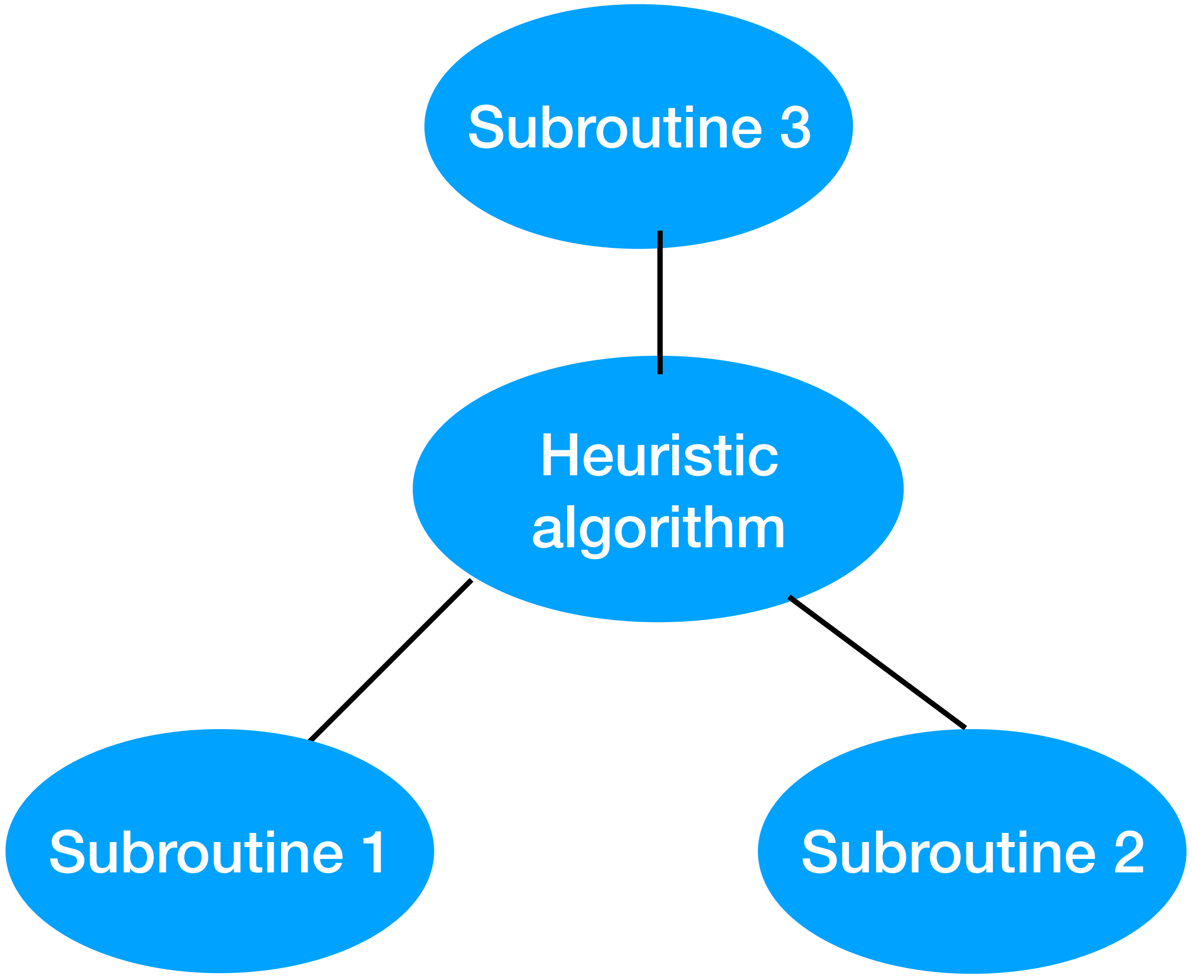
Heuristic algorithm

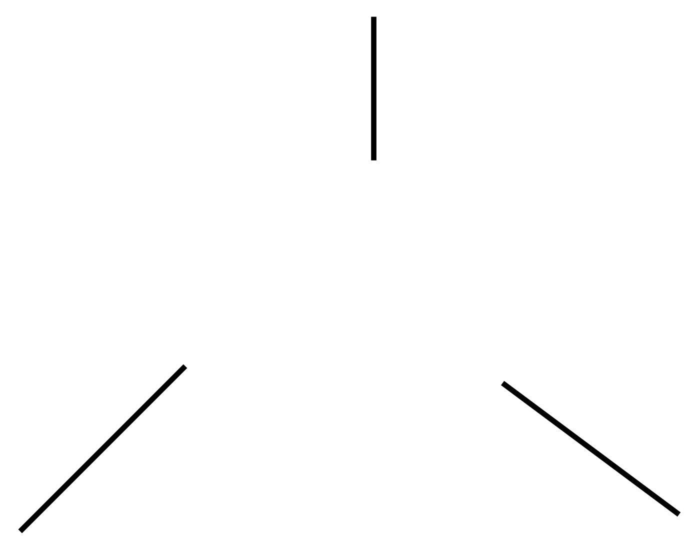
A Better heuristic algorithm

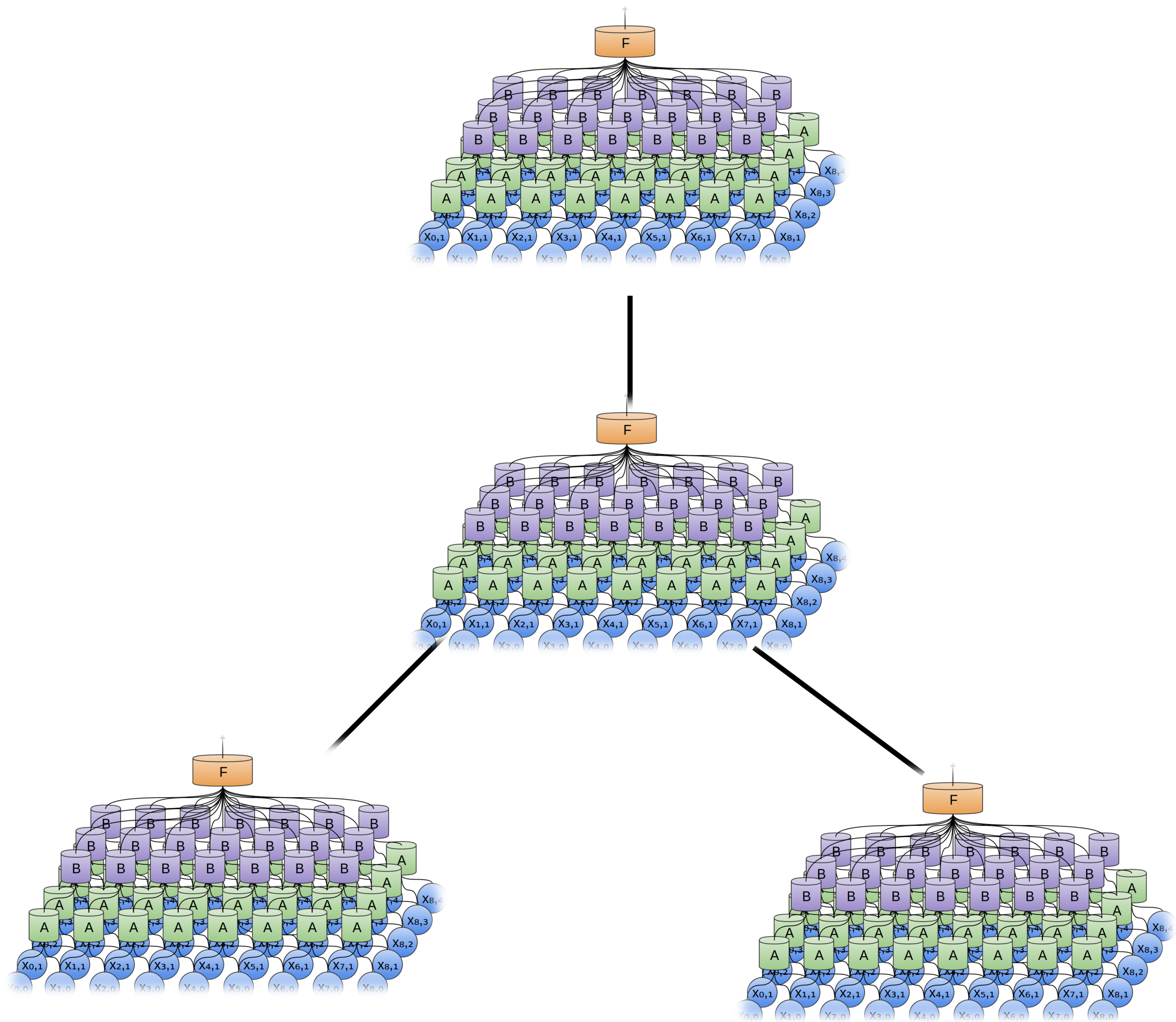
2. Further optimization with data

1. Approximate

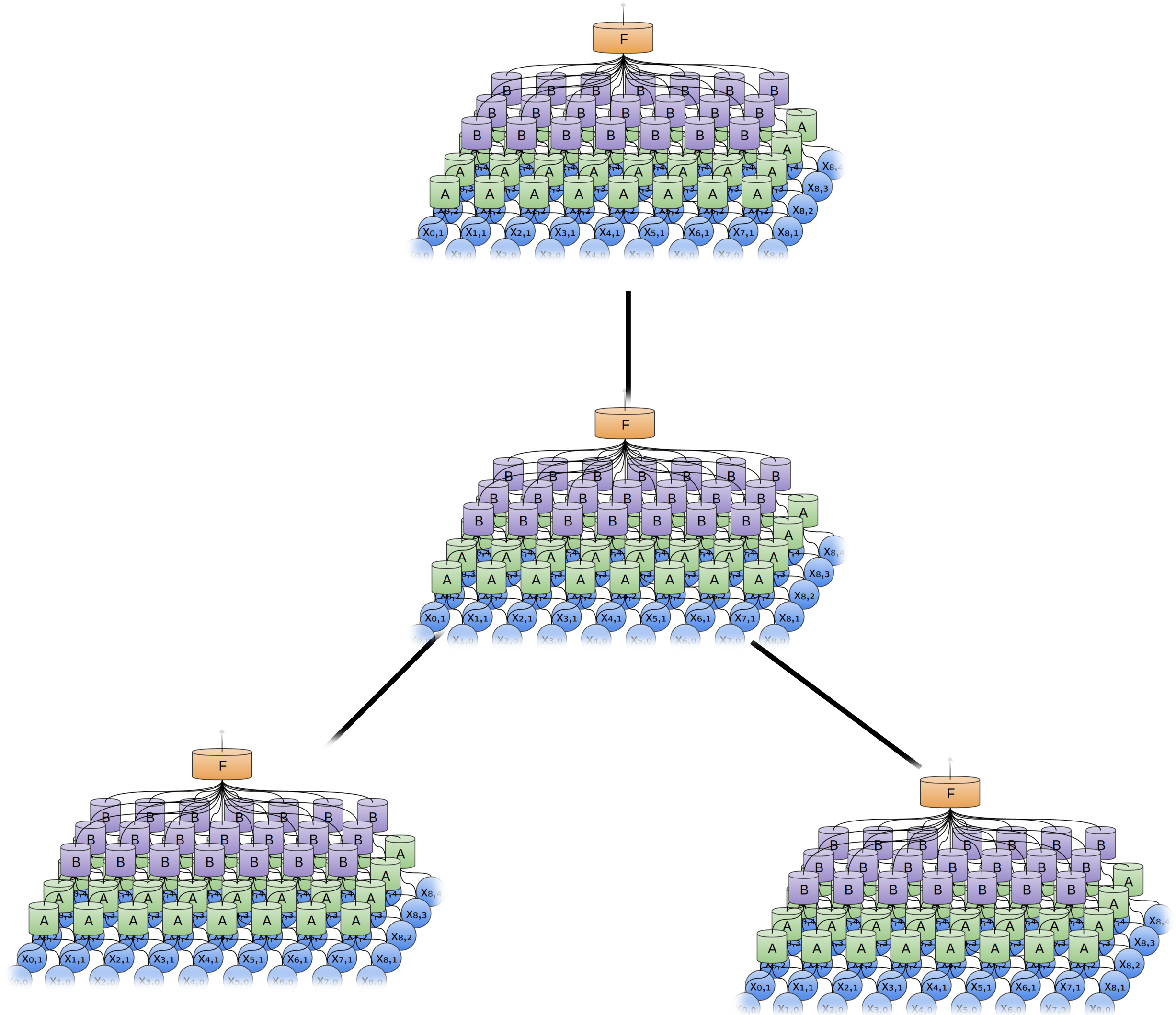



Subroutine 3
Heuristic
algorithm
Subroutine 1
Subroutine 2



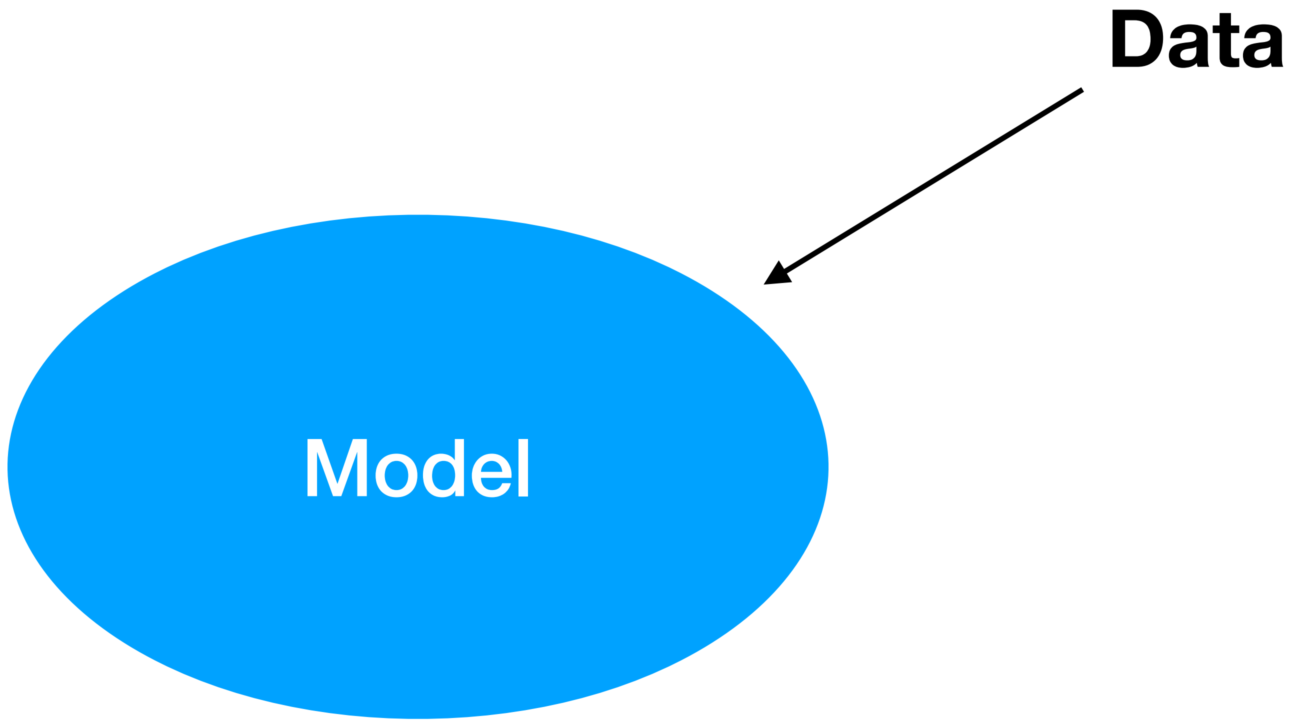


F
B
A
x0,1 x1,1 x2,1 x3,1 x4,1 x5,1 x6,1 x7,1 x8,1
x8,2
x8,3



Lazy way of injecting our knowledge about the problem to the machine learning model

Better way to utilize knowledge?



Better way to utilize knowledge?