Maximum Likelihood Decoding in the Surface Code

Sergey Bravyi

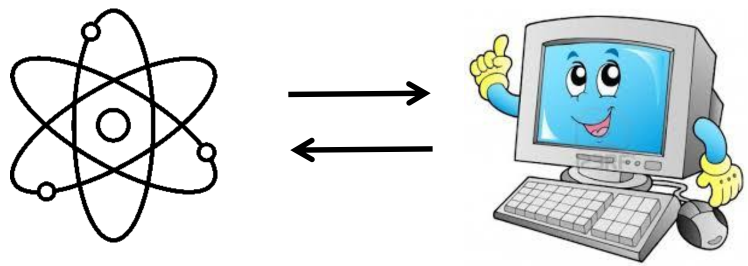
IBM Watson Research Center

SB, Suchara, and Vargo
arXiv:1405.4883

QEC 2014,
December 16, 2014

Motivation

Large-scale quantum computing is likely to require active error correction



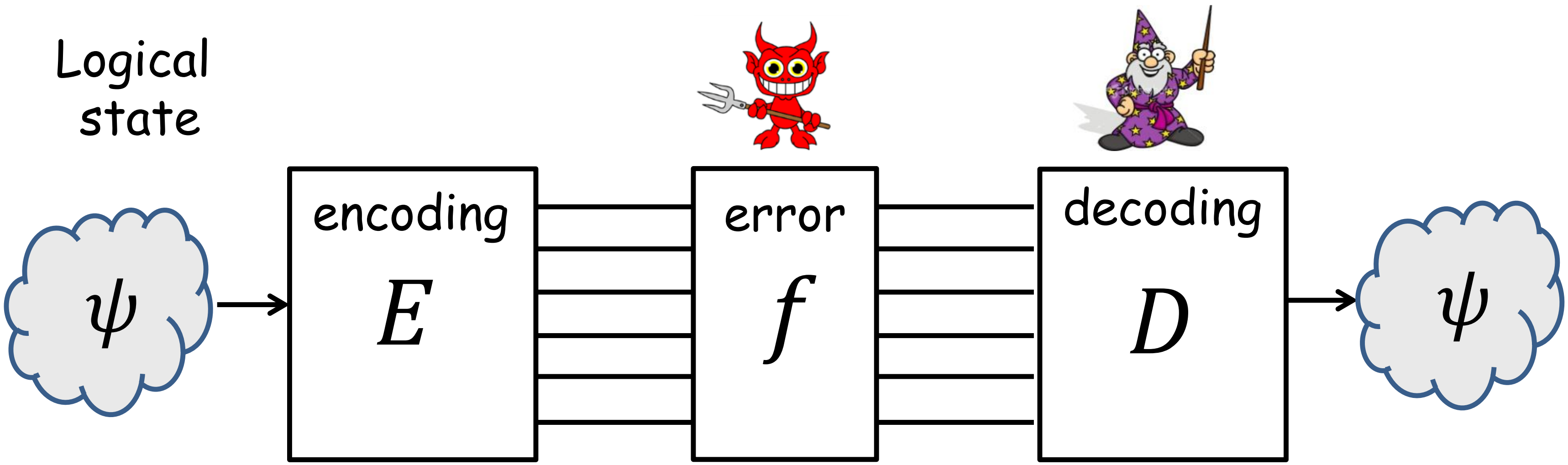
Timely problem: develop efficient algorithms for the optimal quantum error correction with the surface code

Our approach: use tensor network contraction algorithms

Outline

- Quantum error correction and surface codes
- Minimum weight matching decoder
- Maximum likelihood decoder (MLD)
- Approximate linear-time algorithm for MLD
- Exact quadratic-time algorithm for MLD

Quantum Error Correction



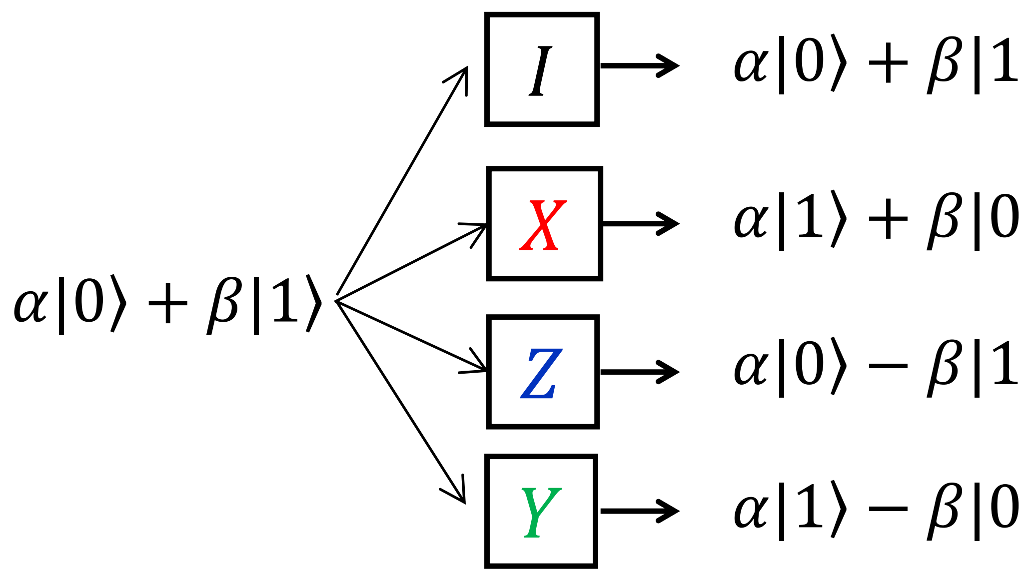
Pauli errors:

Perfect transmission I

Bit flip X

Phase flip Z

Bit and phase flip Y



Encoding: embed a logical qubit into a two-dimensional codespace C of n physical qubits

Stabilizer (additive) codes

The code is defined by parity check operators S_a called stabilizers:

$$S_a\psi = \psi \qquad \text{check passed}$$

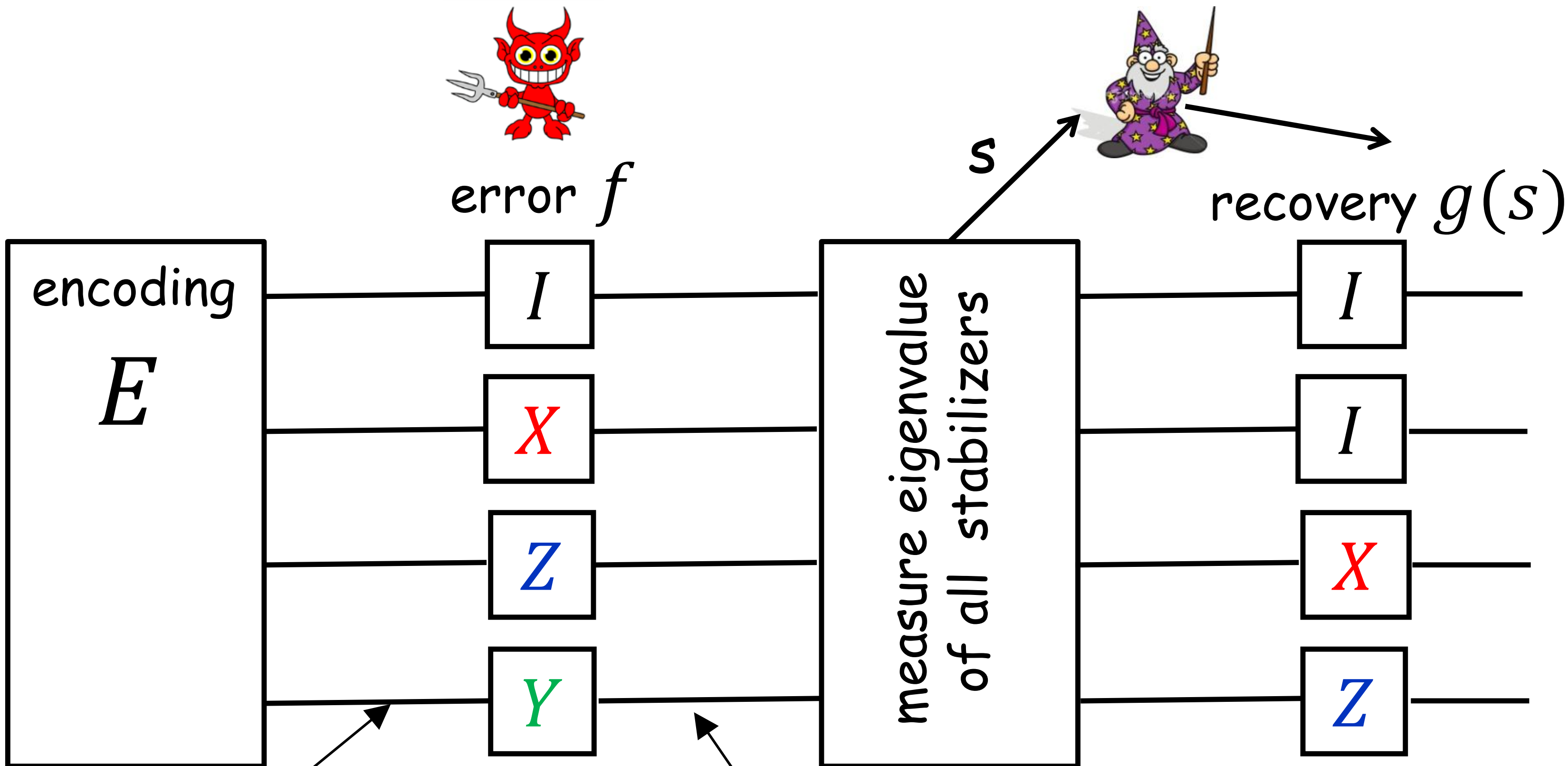
$$S_a\psi = -\psi \qquad \text{check failed}$$

Codespace: $C = \{\psi \in (\mathbf{C}^2)^{\otimes n} : \; S_a\psi = \psi \; \text{ for all } \; a\}$

All stabilizers S_a are multi-qubit Pauli operators

Stabilizers must pairwise commute, $S_aS_b = S_bS_a$

Decoding: syndrome measurement + recovery

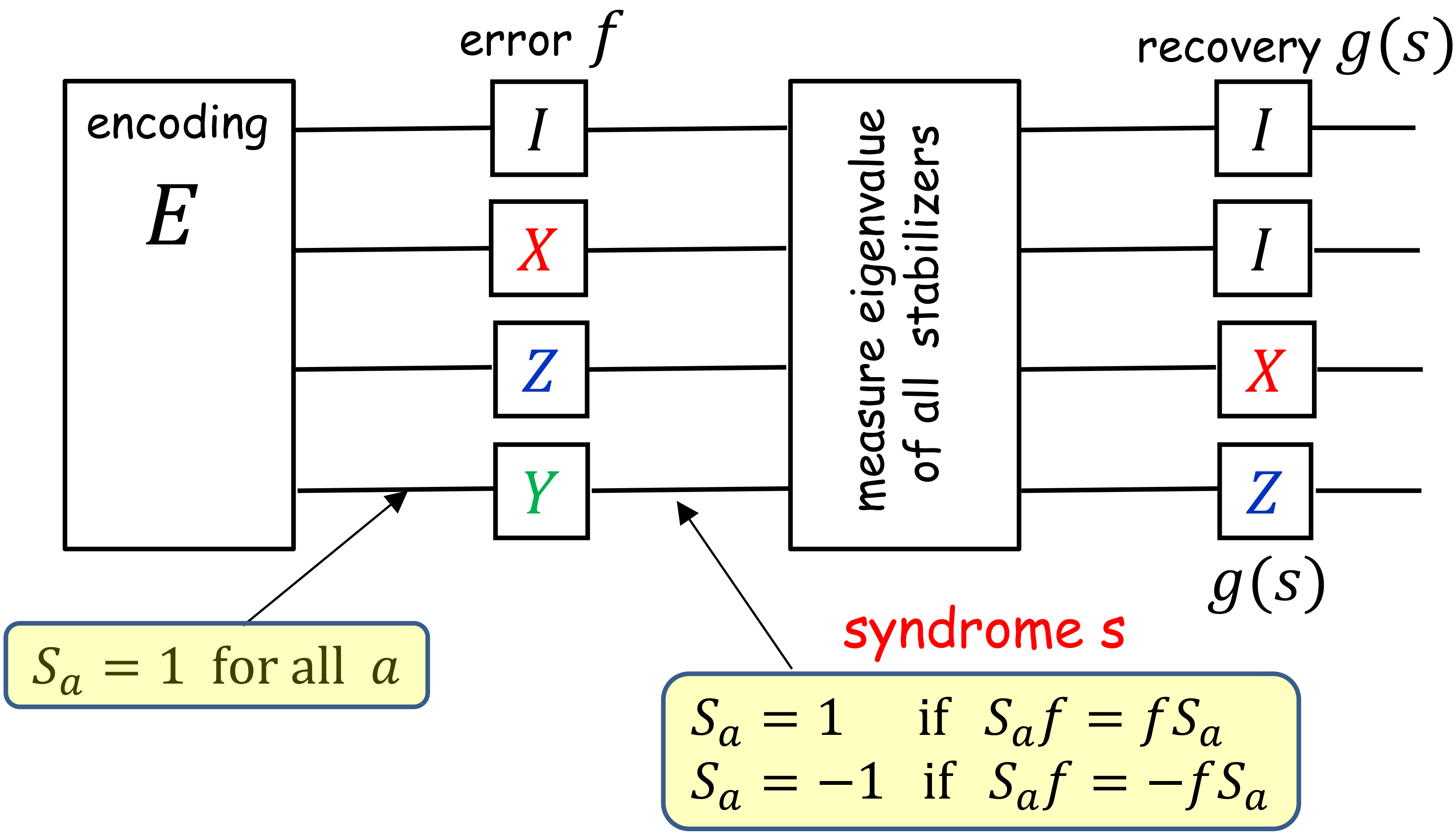


$S_a = 1$ for all a

syndrome s

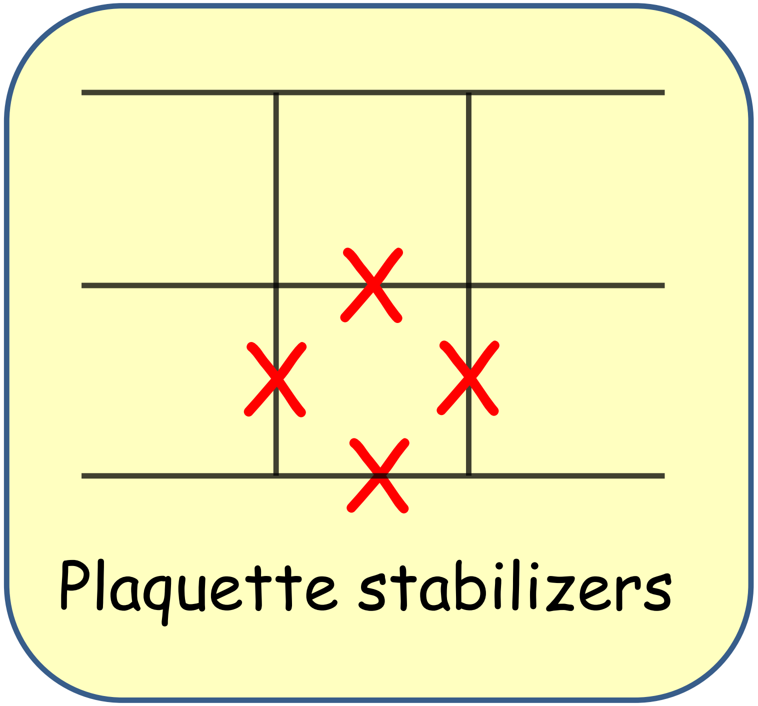
$$S_a = 1 \quad \text{if} \quad S_a f = f S_a$$
$$S_a = -1 \quad \text{if} \quad S_a f = -f S_a$$

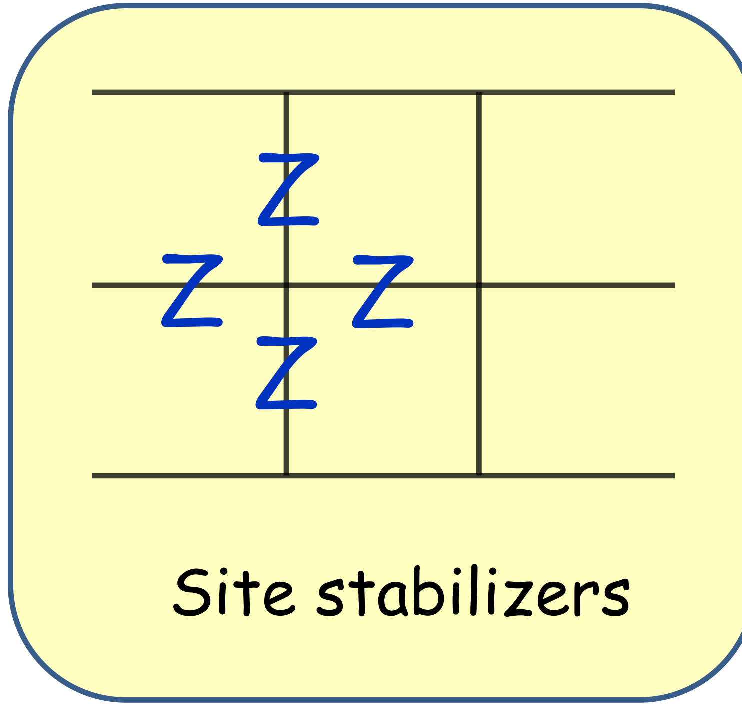
Decoding succeeds iff the recovery differs from the actual error by a product of stabilizers



Surface codes

Physical qubits live at edges of the 2D square lattice

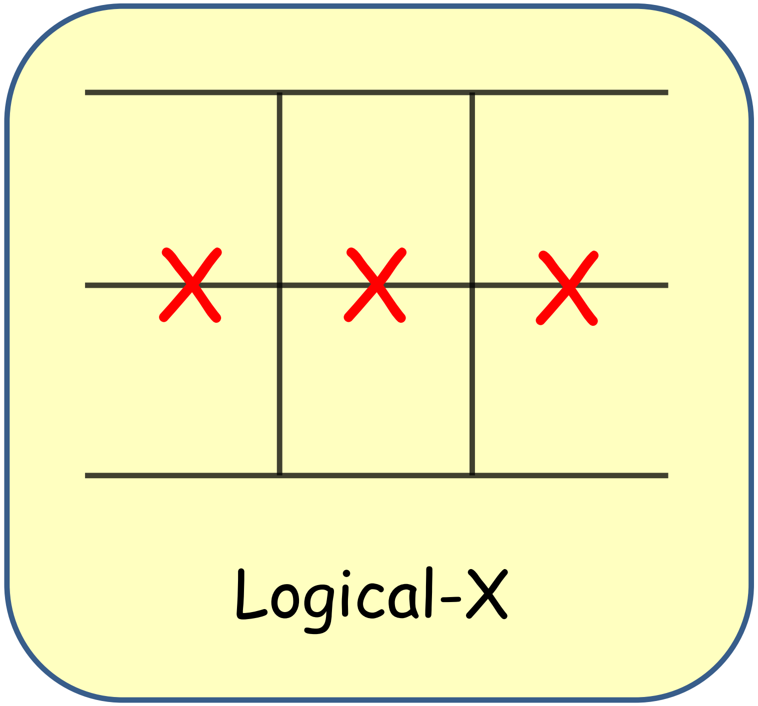


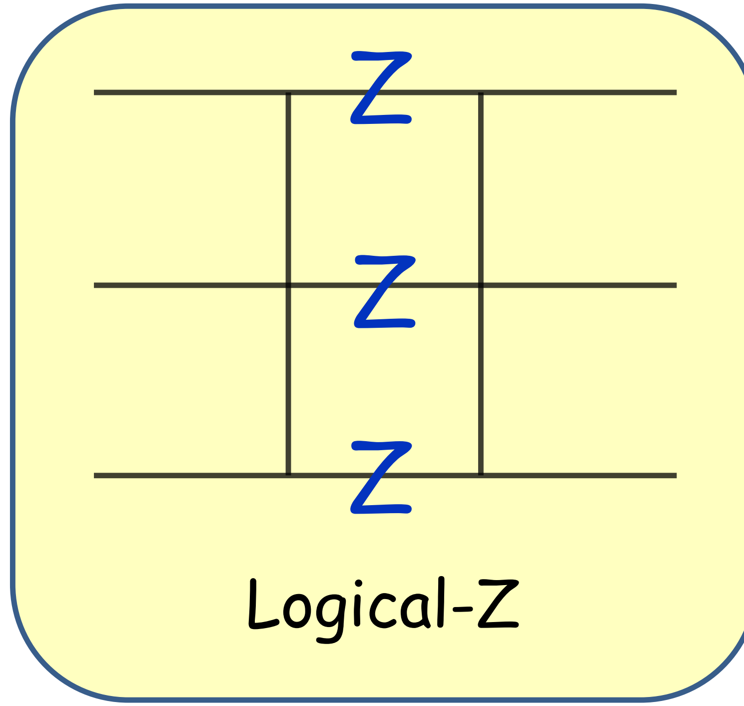


Kitaev (1997)
SB and Kitaev (1998)
Freedman and Meyer (1998)

Surface codes

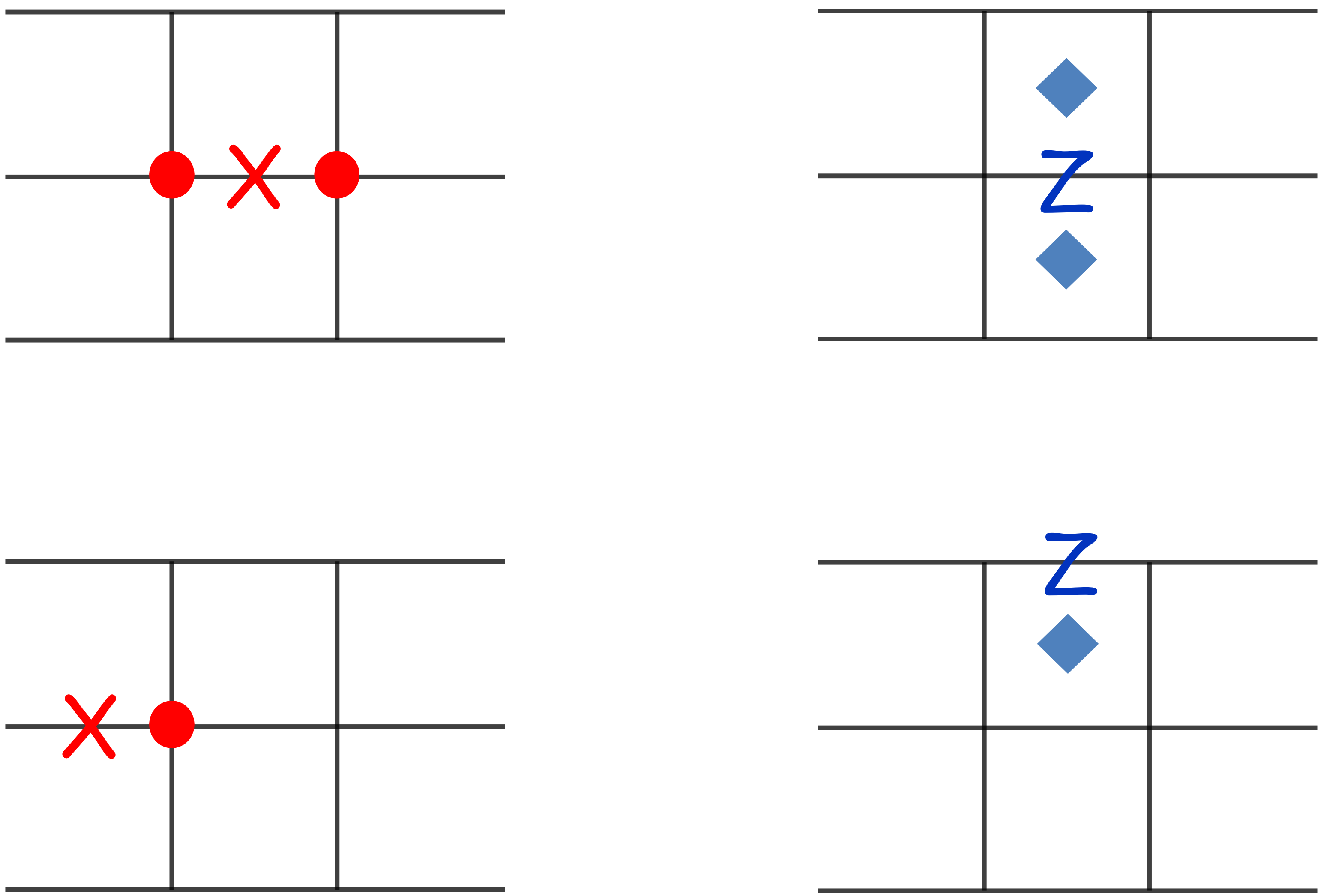
Physical qubits live at edges of the 2D square lattice



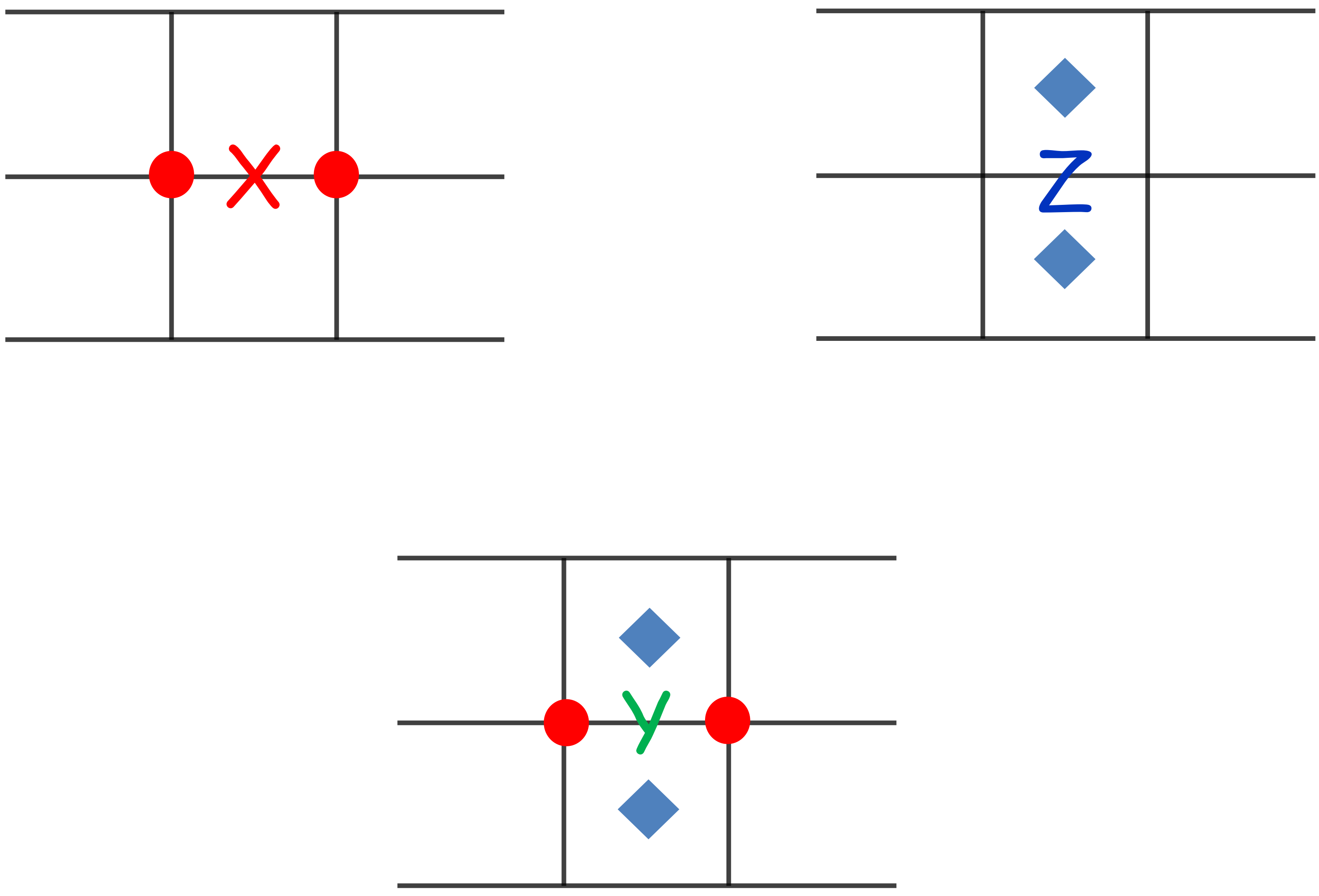


Kitaev (1997)
SB and Kitaev (1998)
Freedman and Meyer (1998)

Surface code: errors vs syndromes



Surface code: errors vs syndromes



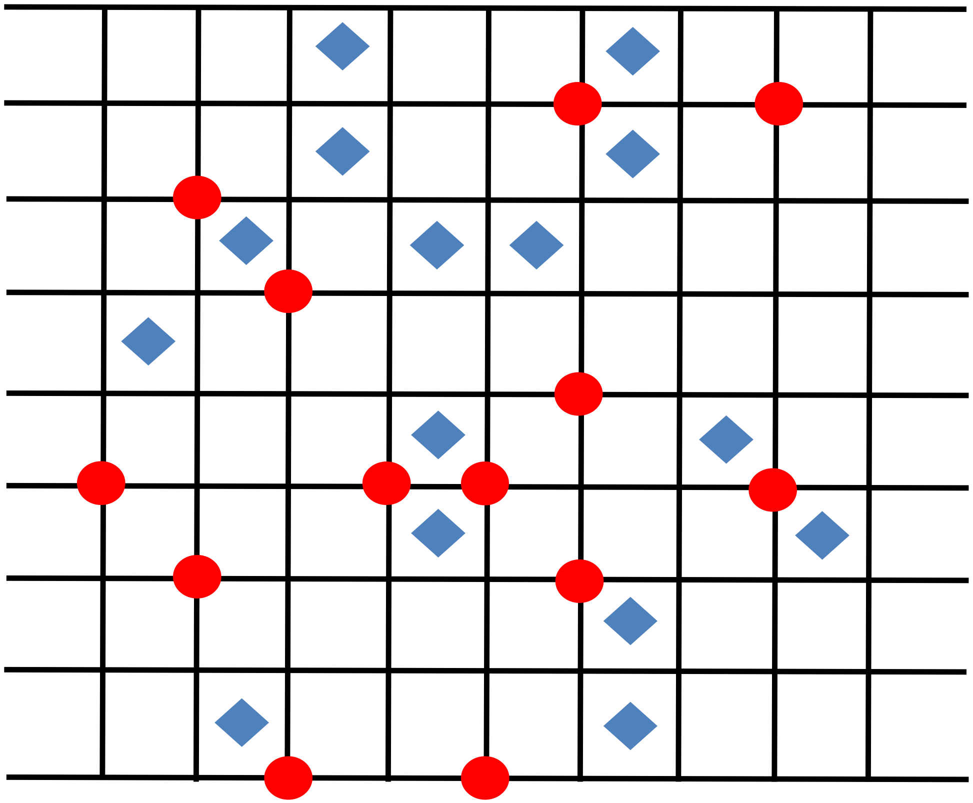
Depolarizing i.i.d. noise:

$$\Pr(X) = \Pr(Y) = \Pr(Z) = \epsilon/3$$

$$\Pr(I) = 1 - \epsilon$$

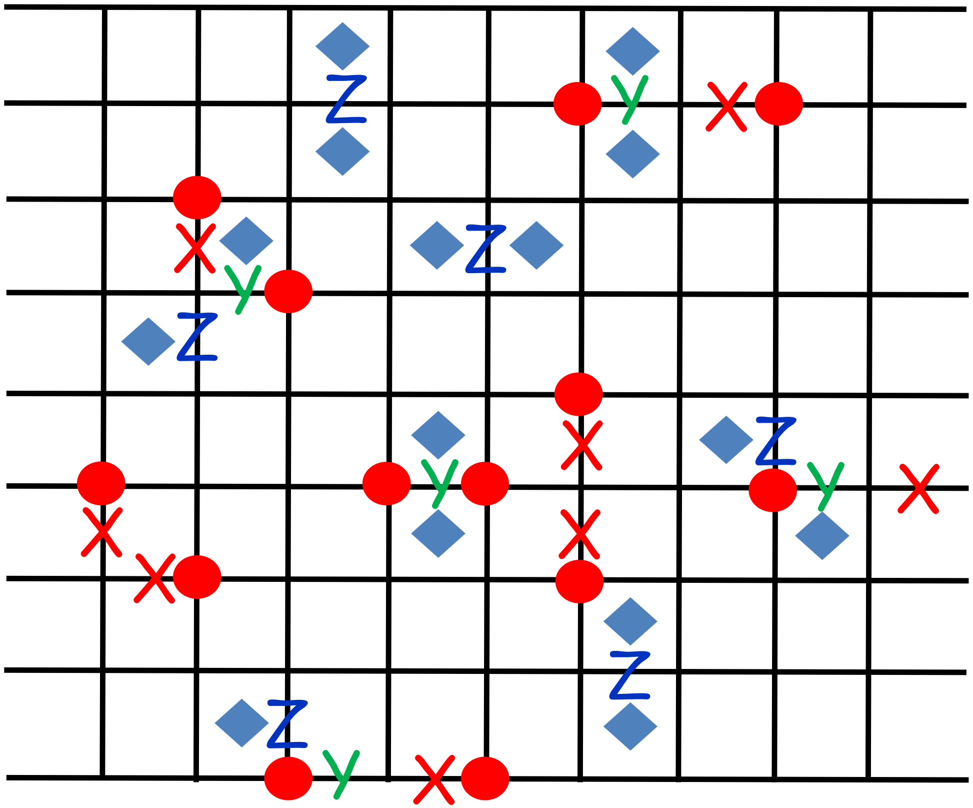
ϵ - error rate

Syndromes are measured perfectly



Decoding problem

Given error syndrome, guess which error has created it (modulo stabilizers)



Decoding problem

Given error syndrome, guess which error has created it (modulo stabilizers)

Minimum Weight Matching (MWM) decoder

1. Find a minimum weight X-error consistent with site-syndromes.

2. Find a minimum weight Z-error consistent with plaquette-syndromes.

3. Combine the X-error and the Z-error.

Motivation: for small error rate the actual error is likely to be among minimum weight errors consistent with the observed syndrome

---

Dennis, Kitaev, Landahl, Preskill (2001)
Wang, Fowler, Hollenberg (2011)

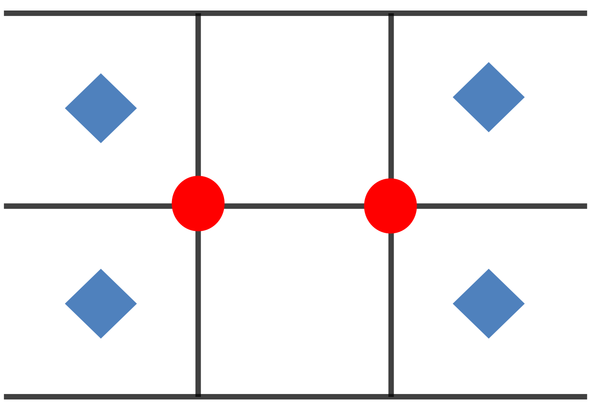
Minimum Weight Matching (MWM) decoder

1. Find a minimum weight X-error consistent with site-syndromes.

2. Find a minimum weight Z-error consistent with plaquette-syndromes.

3. Combine the X-error and the Z-error.

Syndrome:



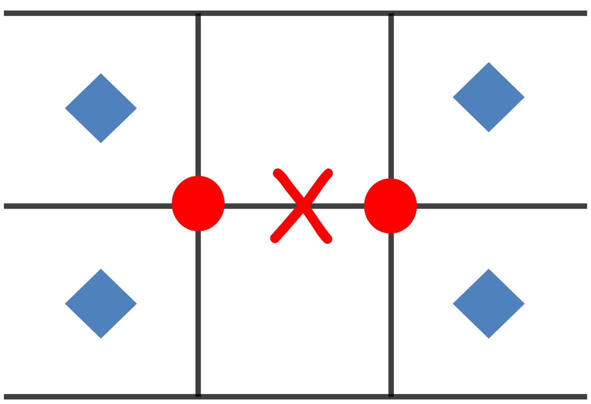
Minimum Weight Matching (MWM) decoder

1. Find a minimum weight X-error consistent with site-syndromes.

2. Find a minimum weight Z-error consistent with plaquette-syndromes.

3. Combine the X-error and the Z-error.

Step 1.



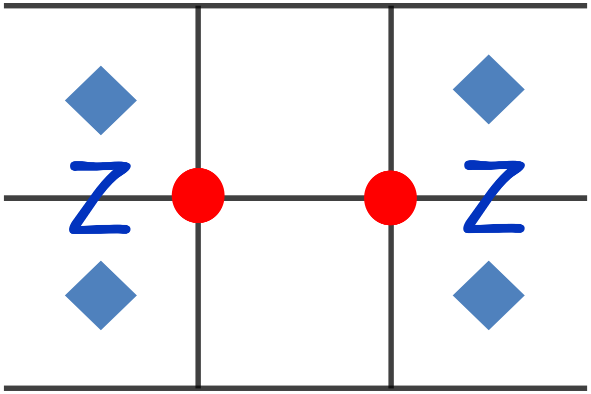
Minimum Weight Matching (MWM) decoder

1. Find a minimum weight X-error consistent with site-syndromes.

2. Find a minimum weight Z-error consistent with plaquette-syndromes.

3. Combine the X-error and the Z-error.

Step 2.



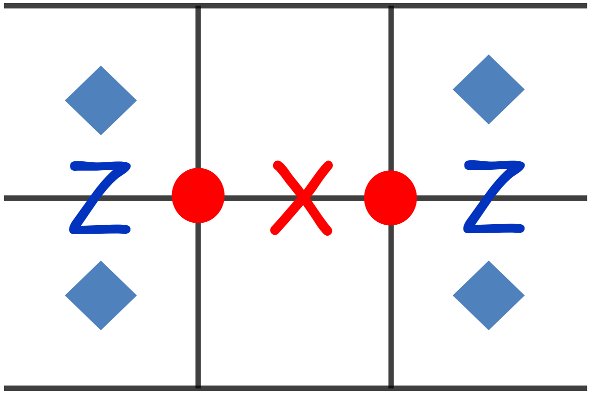
Minimum Weight Matching (MWM) decoder

1. Find a minimum weight X-error consistent with site-syndromes.

2. Find a minimum weight Z-error consistent with plaquette-syndromes.

3. Combine the X-error and the Z-error.

Step 3.



Minimum Weight Matching (MWM) decoder

Complexity

Worst-case running time: $O(n^3)$

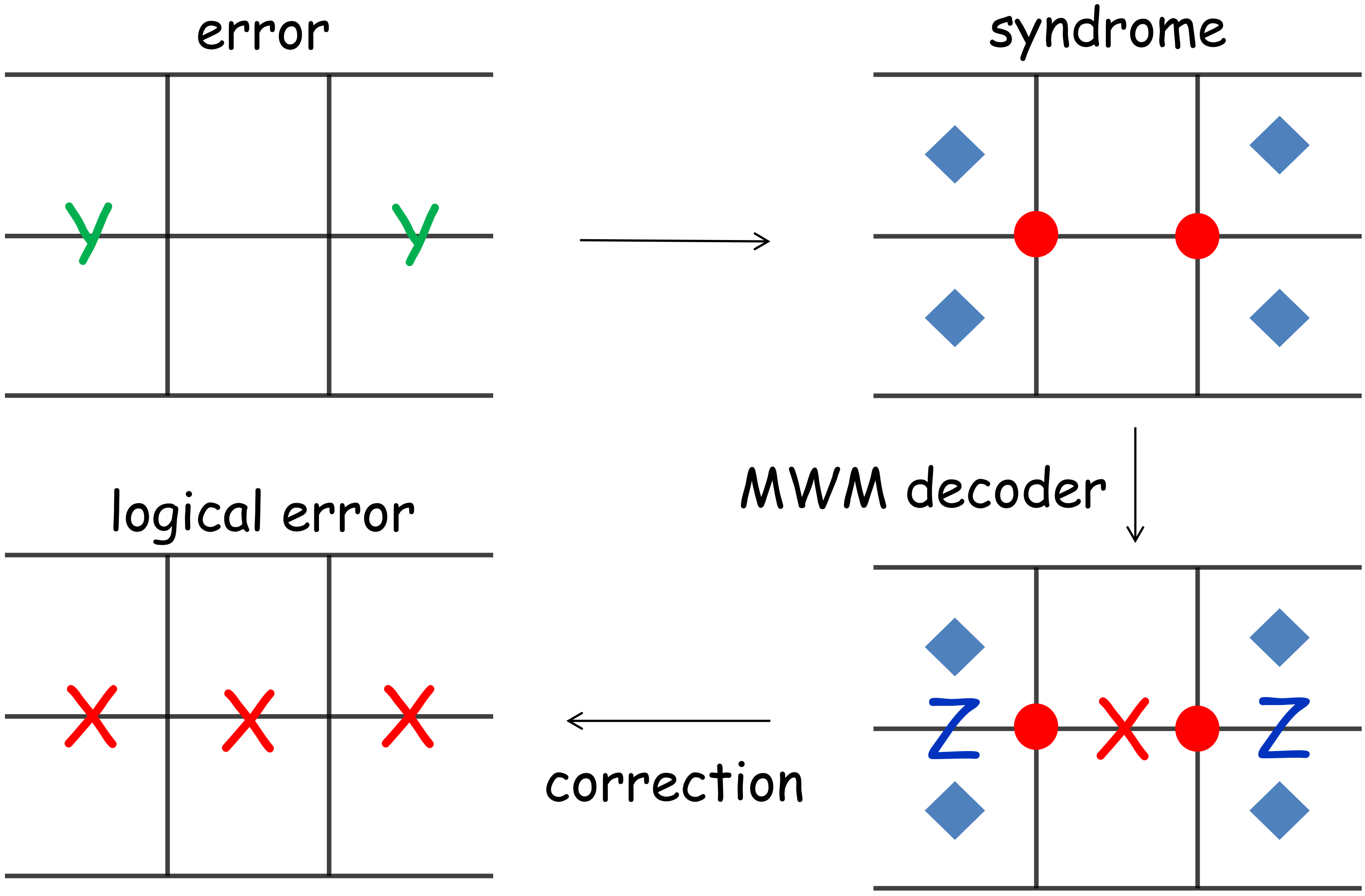
Edmonds (1965)
Gabov (1973)

Average-case running time: $O(n)$

Fowler, Whiteside, Hollenberg (2012)

Dennis, Kitaev, Landahl, Preskill (2001)
Wang, Fowler, Hollenberg (2011)

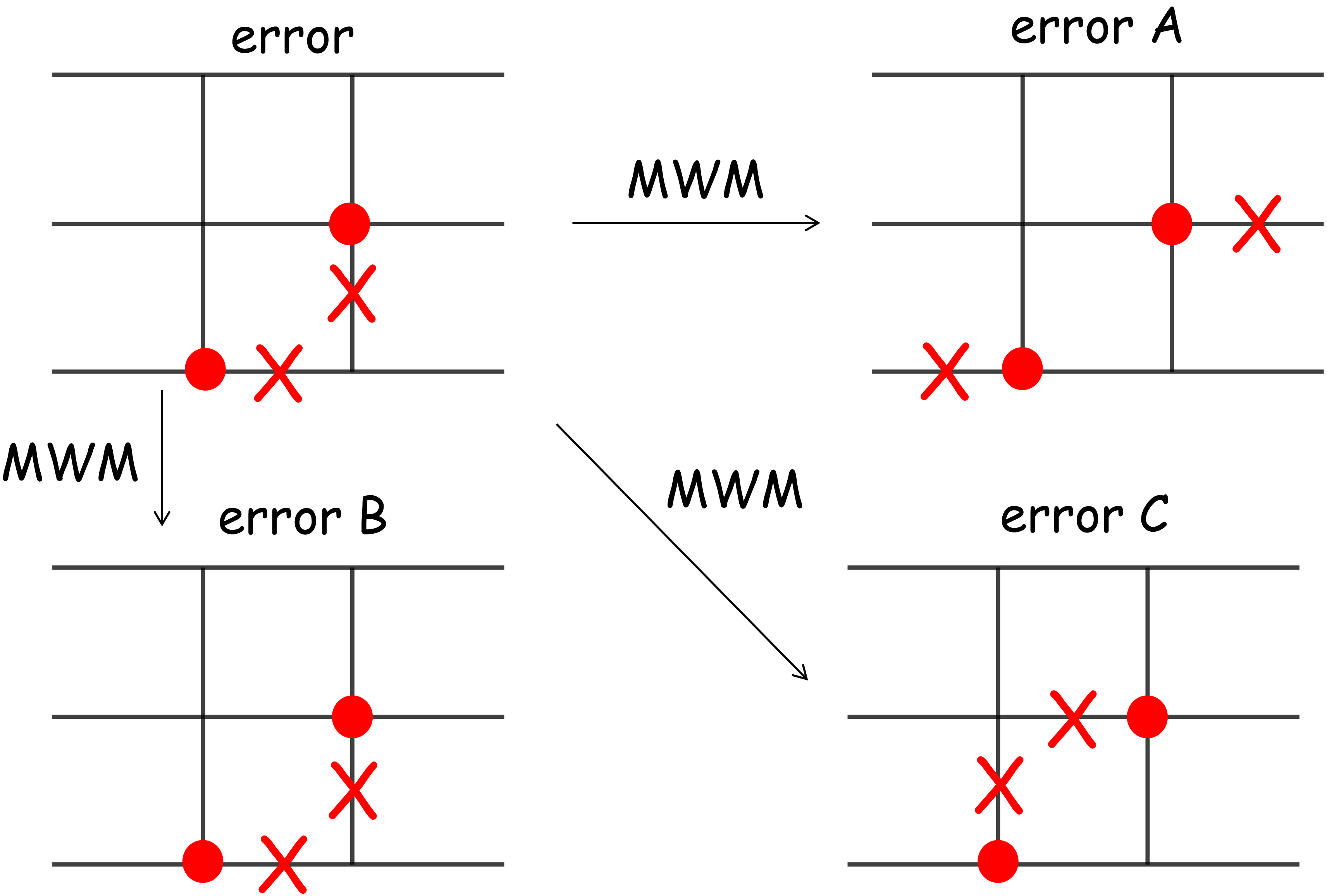
Why minimum matching is not good enough ?



Why Minimum Weight Matching is not good enough ?

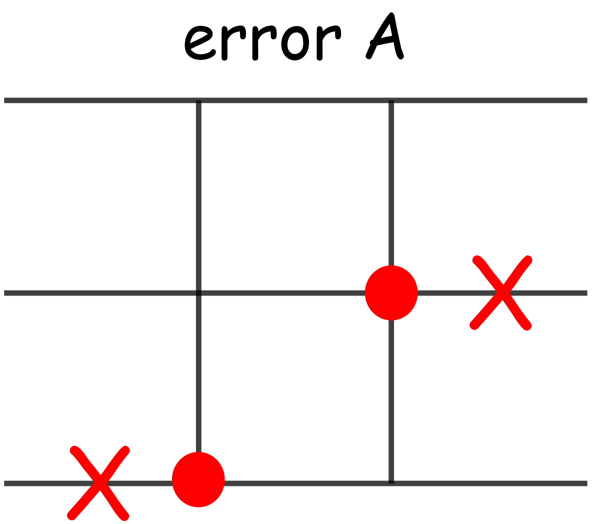
1. Minimum weight matching $\neq$ minimum weight error

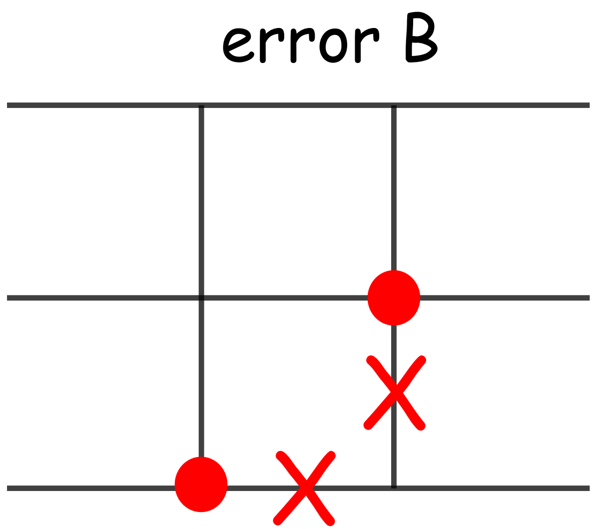
Why Minimum Weight Matching is not good enough ?

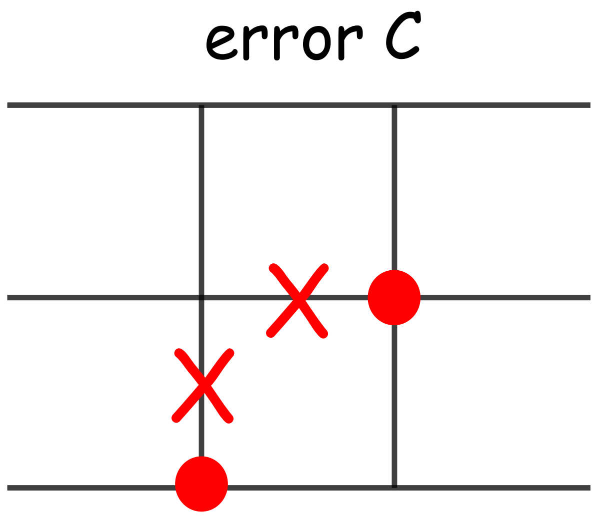


B and C have the same action on any encoded state.

It does not matter whether we choose B or C as a correction.





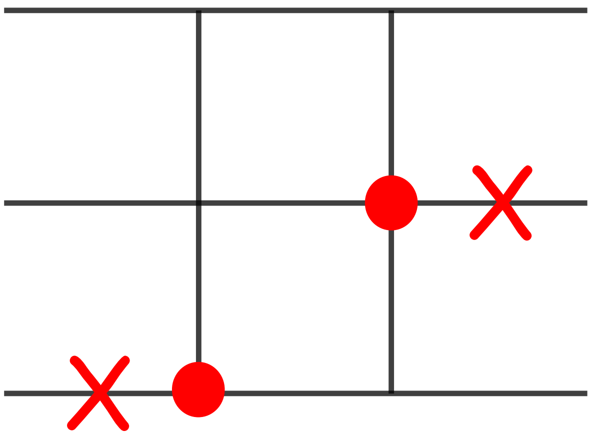


B and C have the same action on any encoded state

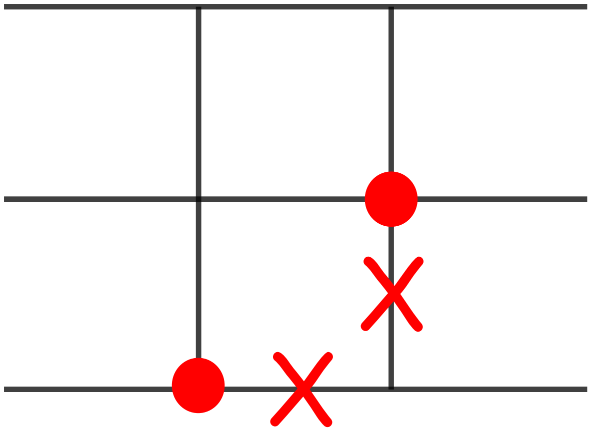
$$\Pr(B \text{ or } C) = 2\Pr(A)$$

Picking B or C is twice as likely to correct the error than picking A.

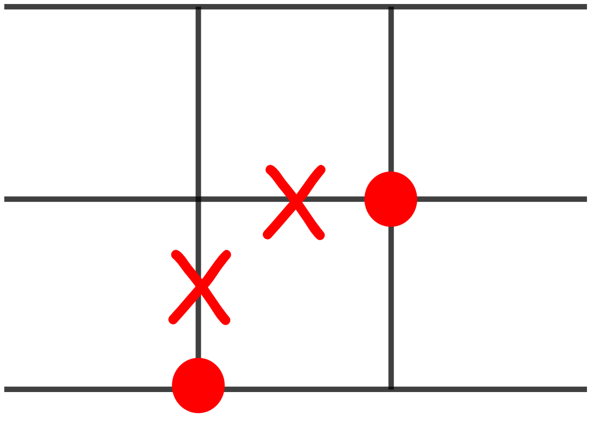
error A



error B



error C



Why Minimum Weight Matching is not good enough ?

1. Minimum weight matching $\neq$ minimum weight error

2. MWM fails to account equivalence between errors.

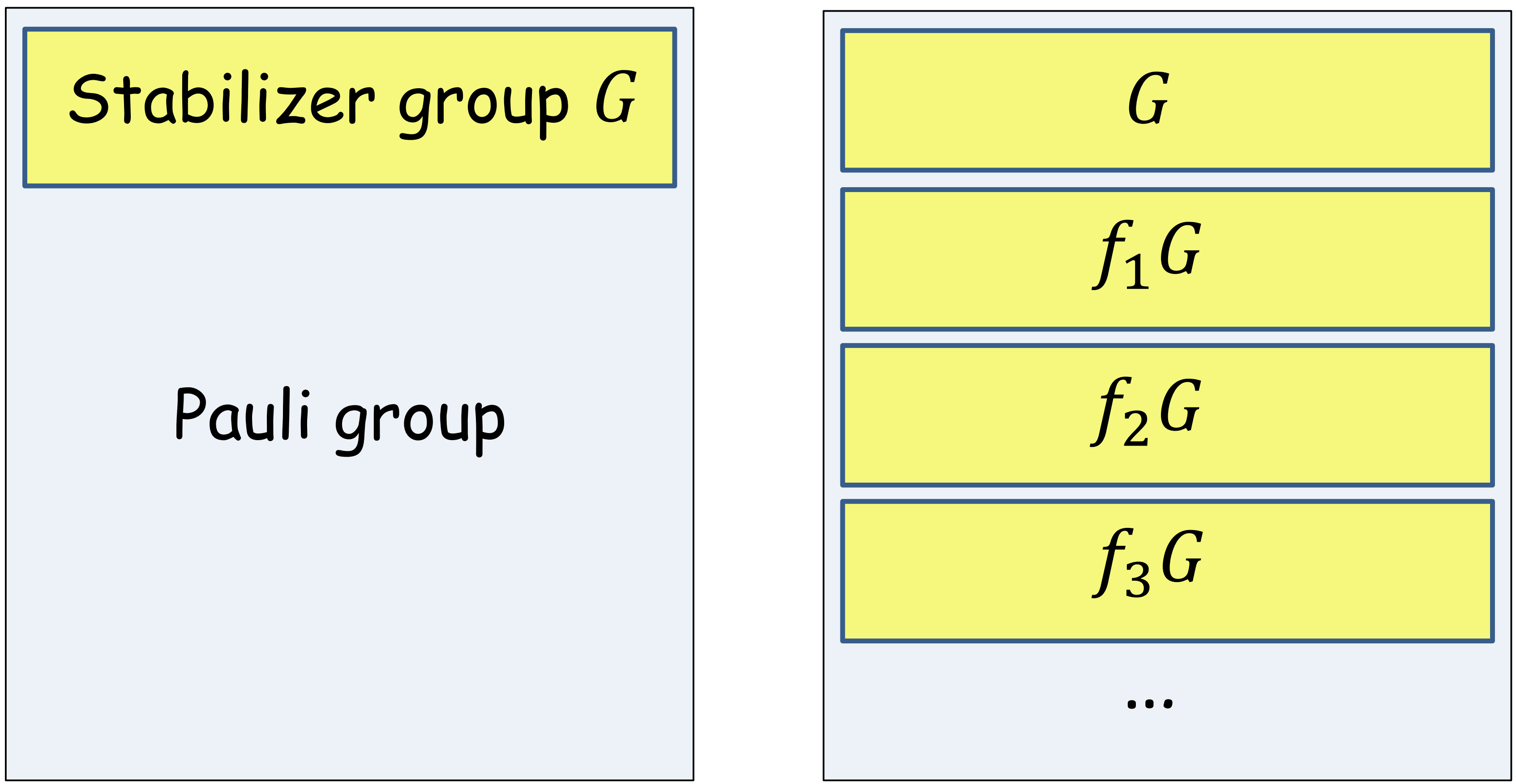
Beyond MWM: previous work

deterministic algorithms	randomized algorithms
Stace and Barrett, PRA 81, 022317 (2010) Tweak the weights in the MWM to favor chains with high entropy	Use Metropolis-type algorithms to sample errors conditioned on the observed syndrome.
Fowler, arXiv:1310.0863 X-MWM, update weights, Z-MWM	Wootton and Loss, PRL 109 160503 (2012) Parallel tempering
Duclos-Cianci and Poulin, PRL 104 050504 (2010) RG decoder: approximate surface code by a concatenated code.	Hutter, Wootton and Loss, PRA 89 022326 (2014) Faster heuristic version

Imagine unlimited computational power.

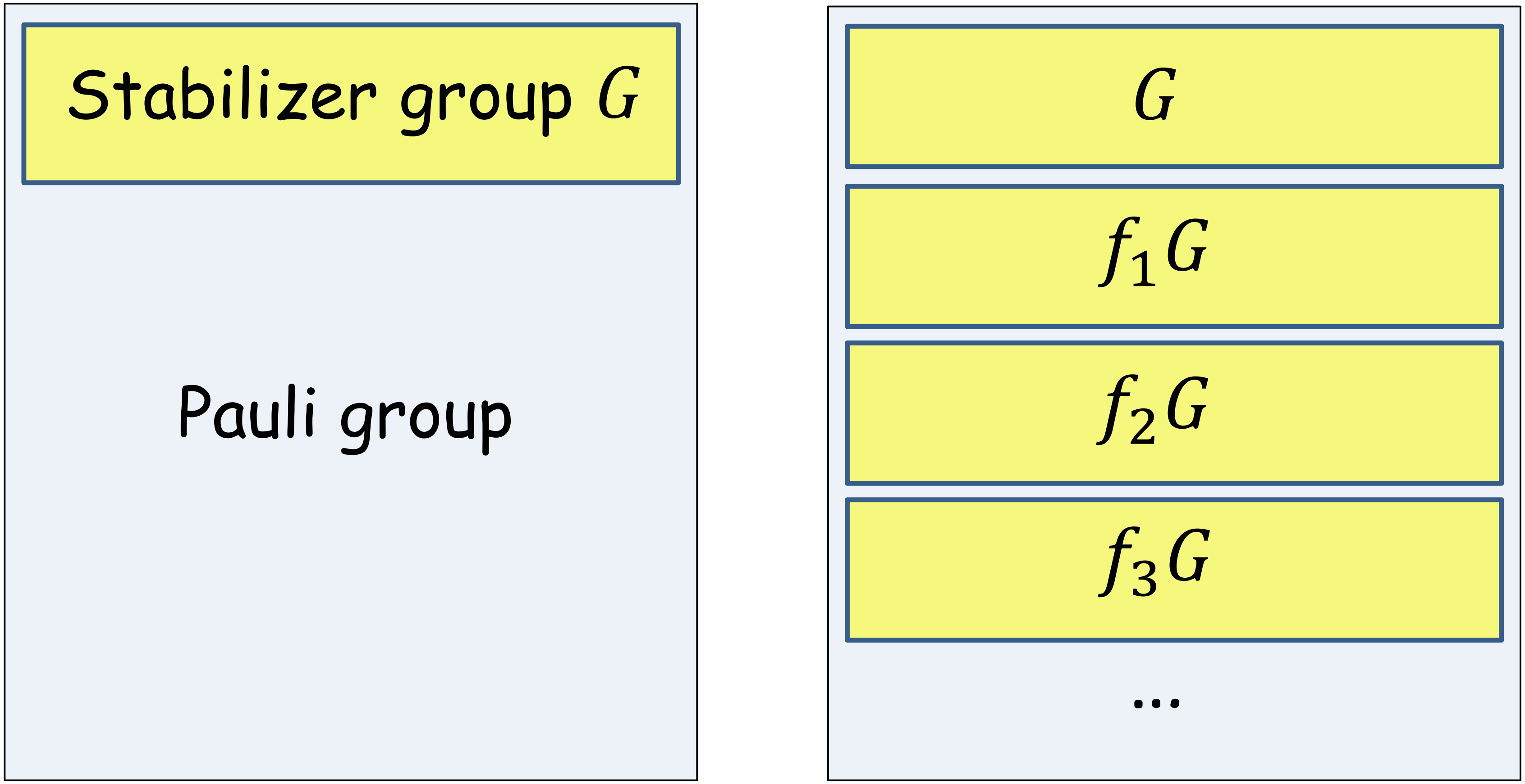
What decoding algorithm would we use ?

Some terminology:



$\{I, X, Y, Z\}^{\otimes n}$

cosets of the stabilizer group



Errors in the same coset have the same action on the codespace

Errors in the same coset have the same syndrome

The four cosets consistent with the syndrome s :

I-coset	X-coset
$f(s)G$	$f(s)\bar{X}G$

Y-coset	Z-coset
$f(s)\bar{Y}G$	$f(s)\bar{Z}G$

We fixed some canonical error $f(s)$ consistent with s

$\bar{X}$, $\bar{Y}$, $\bar{Z}$ are the logical operators

The four cosets consistent with the syndrome s :

I-coset	X-coset
$f(s)G$	$f(s)\bar{X}G$

Y-coset	Z-coset
$f(s)\bar{Y}G$	$f(s)\bar{Z}G$

Coset probability: $$\Pr(fG) = \sum_{g \in G} \Pr(fg)$$

The four cosets consistent with the syndrome s :

I-coset	X-coset
3.5e-249	4.5e-239
Y-coset	**Z-coset**
2.2e-263	7.9e-257

Real example for d=25, ϵ=10%

Coset probability: $$\Pr(fG) = \sum_{g \in G} \Pr(fg)$$

Most likely coset

I-coset	X-coset
3.5e-249	4.5e-239
Y-coset	**Z-coset**
2.2e-263	7.9e-257

All errors in the same coset have the same action on the codespace

The optimal decoding strategy is to pick the most likely coset.

Maximum Likelihood Decoder (MLD)

Input: syndrome s

Output: Pauli operator g consistent with s which is most likely to correct the error

1. Compute $\Pr(C)$ for the four cosets C consistent with the syndrome s.
2. $C^* \leftarrow \arg\max_{C} \Pr(C)$
3. Return any $g \in C^*$

Dennis, Kitaev, Landahl, Preskill (2001)
Poulin (2006)

Approximate algorithm for MLD:

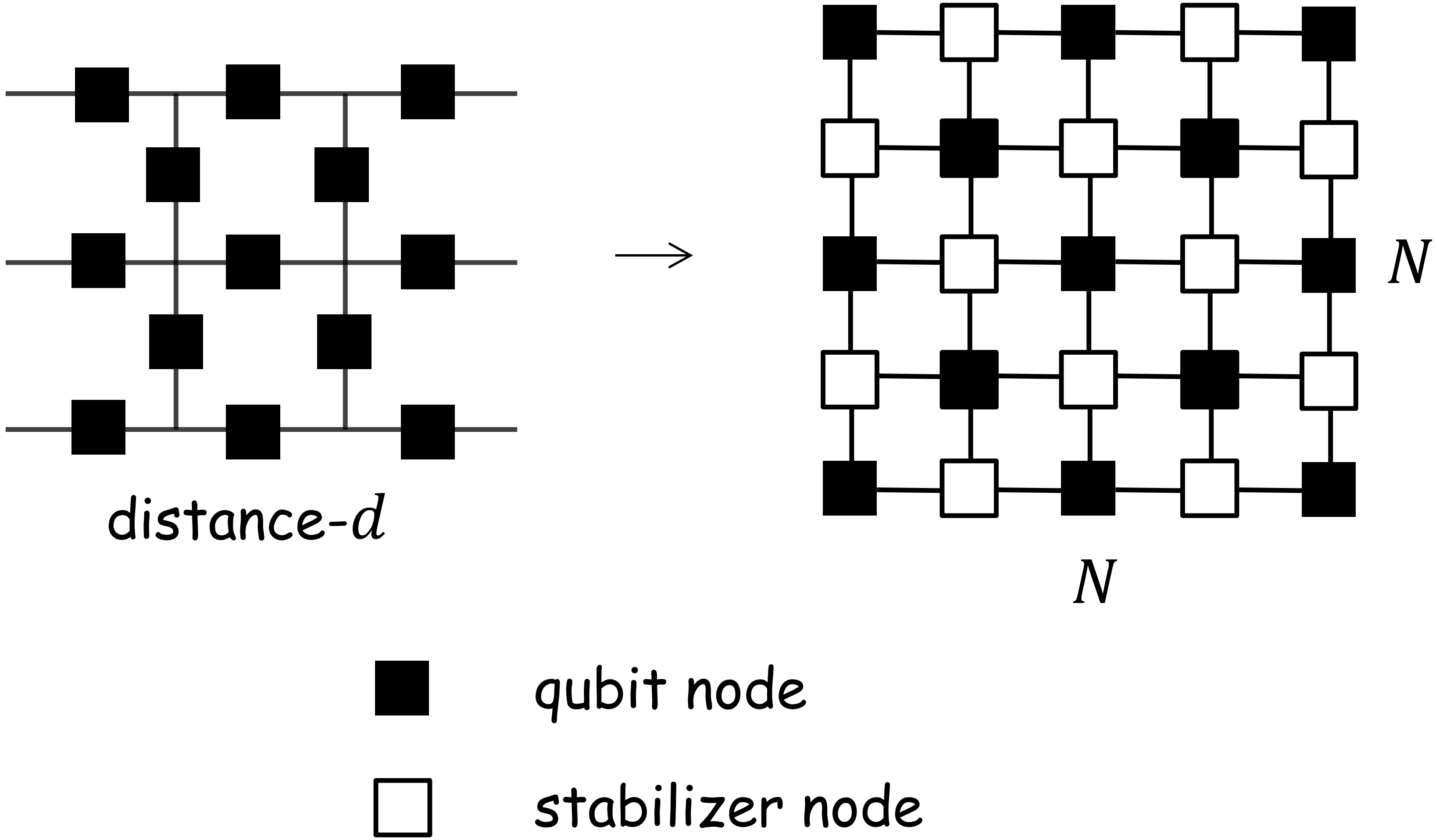
Step 1: express the coset probability as a contraction of a tensor network on a 2D grid.

Step 2: contract the network column by column using matrix product states

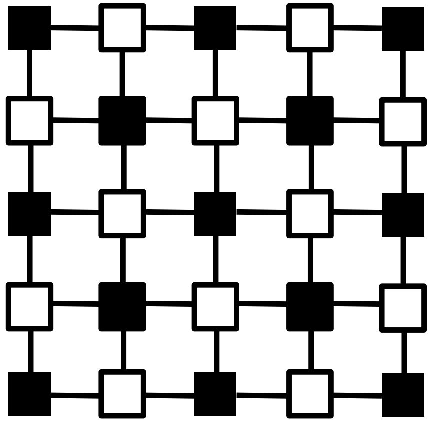
Illustrative example: the trivial coset

$$\Pr(G) = \sum_{g\in G} \Pr(g)$$

Tanner graph

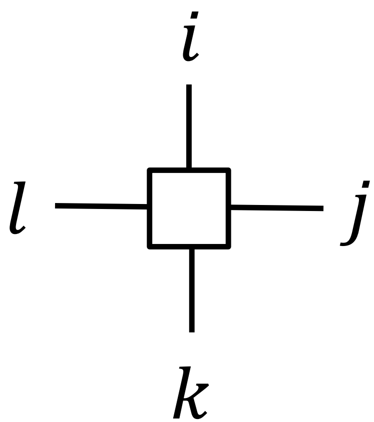


$$N = 2d - 1$$

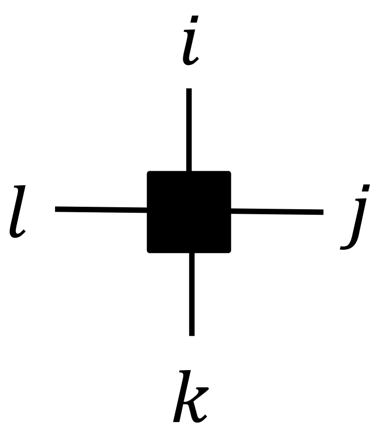


Nodes = tensors

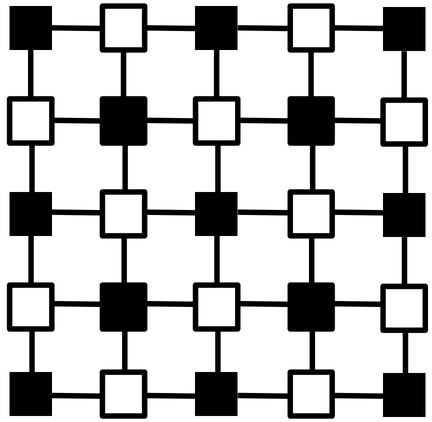
Edges = tensor indexes (0 or 1)



$$T_{i,j,k,l} = \begin{cases} 1 & \text{if} \quad i = j = k = l \\ 0 & \text{otherwise} \end{cases}$$



$$T_{i,j,k,l} = \begin{cases} 1 - \epsilon & \text{if} \quad i \oplus k = j \oplus l = 0 \\ \epsilon/3 & \text{otherwise} \end{cases}$$



Nodes = tensors

Edges = tensor indexes (0 or 1)

Contraction value of a tensor network :

$$c = \sum_{\gamma} \prod_{nodes} T(\gamma)$$

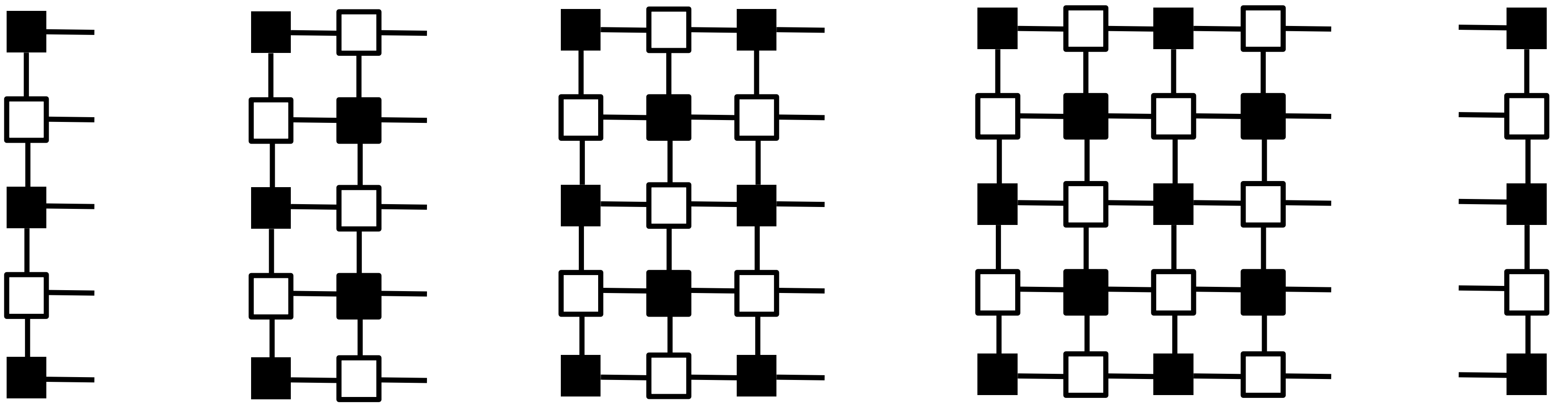
$\gamma =$ edge labeling by 0 and 1

$$\Pr(G) = c$$

Approximate contraction of 2D tensor networks

Murg, Verstraete, Cirac PRA 75, 033605 (2007)

Think of the contraction as a sequence of N-qubit states:



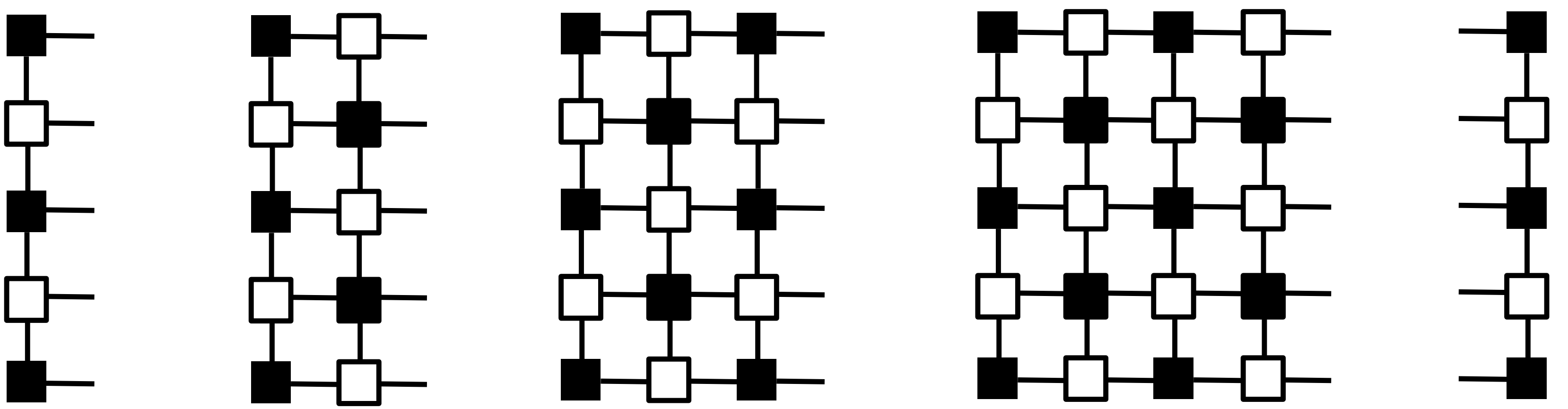
Ψ_0 Ψ_1 Ψ_2 Ψ_3 Ψ_4

$$\Pr(G) = \langle \Psi_3 | \Psi_4 \rangle$$

Approximate contraction of 2D tensor networks

Murg, Verstraete, Cirac PRA 75, 033605 (2007)

Think of the contraction as a sequence of N-qubit states:



Ψ_0 Ψ_1 Ψ_2 Ψ_3 Ψ_4

Let's hope that the time evolution is weakly-entangling.

Approximate Ψ's by matrix product states with a small bond dimension.

Matrix Product States (MPS)

$$\langle i_1 i_2 i_3 i_4 i_5 | \Psi \rangle = \underbrace{A_1(i_1)}_{1 \times \chi} \cdot \underbrace{A_2(i_2)}_{\chi \times \chi} \cdot \underbrace{A_3(i_3)}_{\chi \times \chi} \cdot \underbrace{A_4(i_4)}_{\chi \times \chi} \cdot \underbrace{A_5(i_5)}_{\chi \times 1}$$

χ - bond dimension

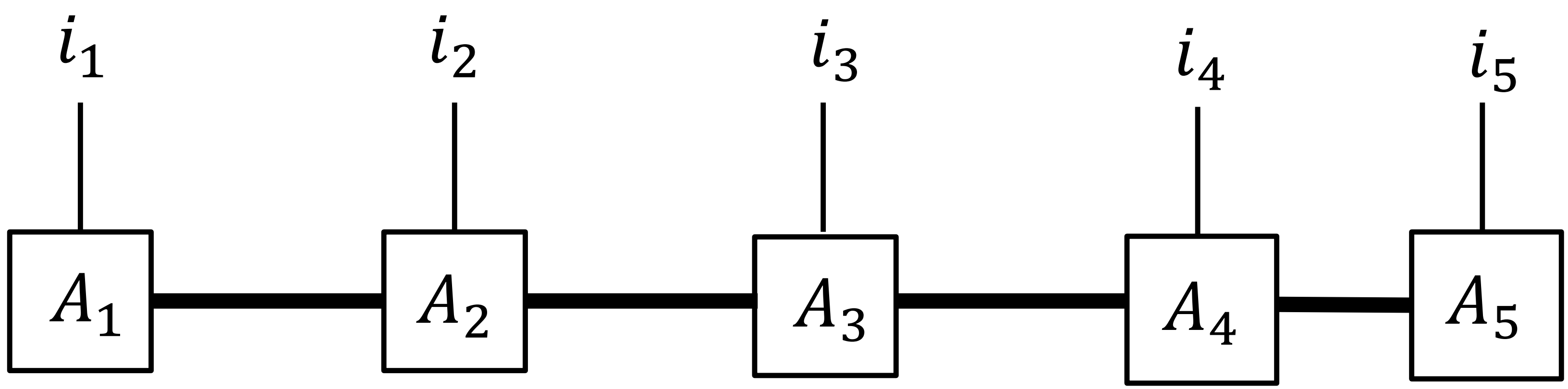
MPS admits a concise description as a list of matrices
($N\chi^2$ real parameters)

Matrix Product States (MPS)

$$\langle i_1 i_2 i_3 i_4 i_5 | \Psi \rangle =$$

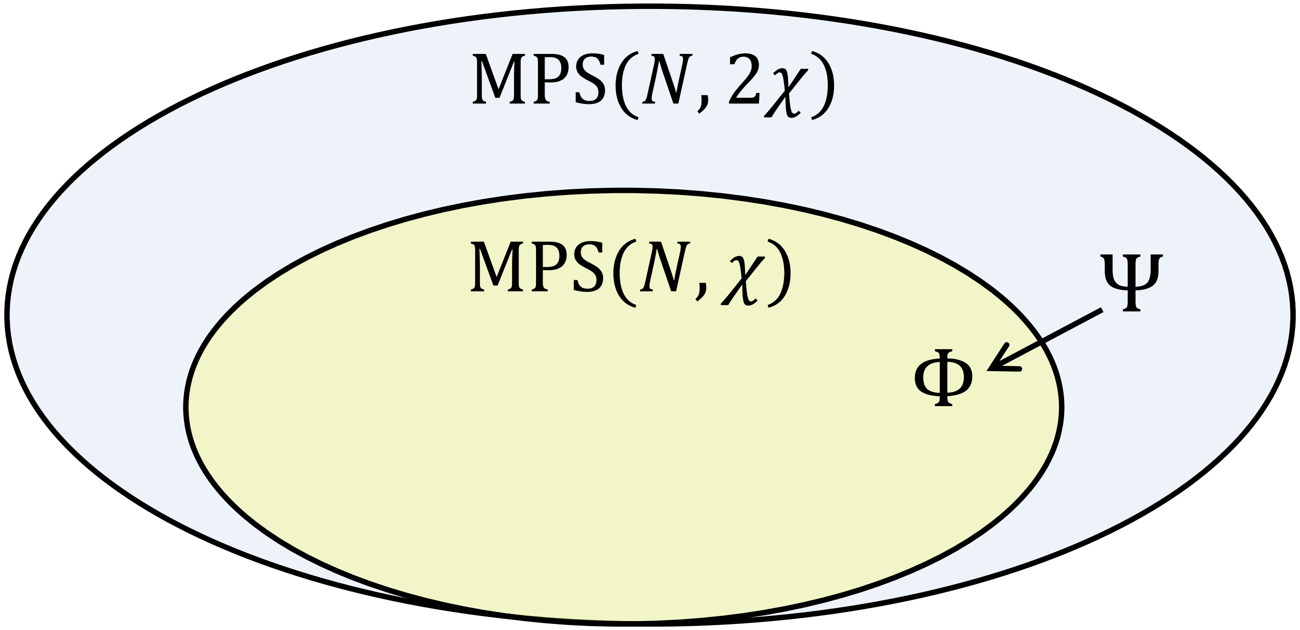
$$A_1(i_1) \cdot A_2(i_2) \cdot A_3(i_3) \cdot A_4(i_4) \cdot A_5(i_5)$$

$=$



Fact 1: Suppose $\Psi, \Phi \in \mathrm{MPS}(N, \chi)$. Then the inner product $\langle \Psi | \Phi \rangle$ can be computed in time $O(N\chi^3)$

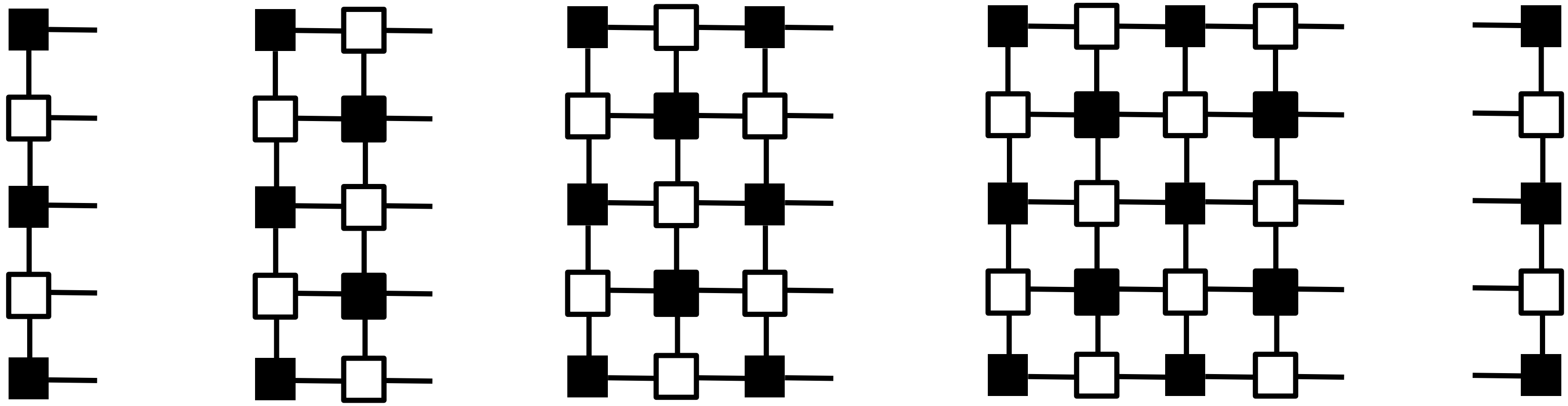
MPS compression



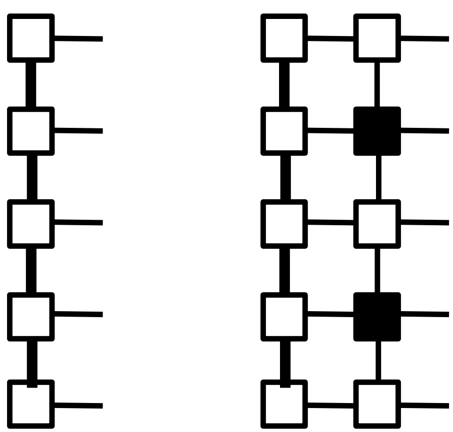
Efficient compression algorithm:
Schollwock, Ann. Phys. 326, 96 (2011)

Fact 2: MPS with a bond dimension 2χ can be approximated by an MPS with a bond dimension χ in time

$$N \cdot \mathrm{svd}(2\chi) + N \cdot \mathrm{qr}(2\chi) = O(N\chi^3)$$

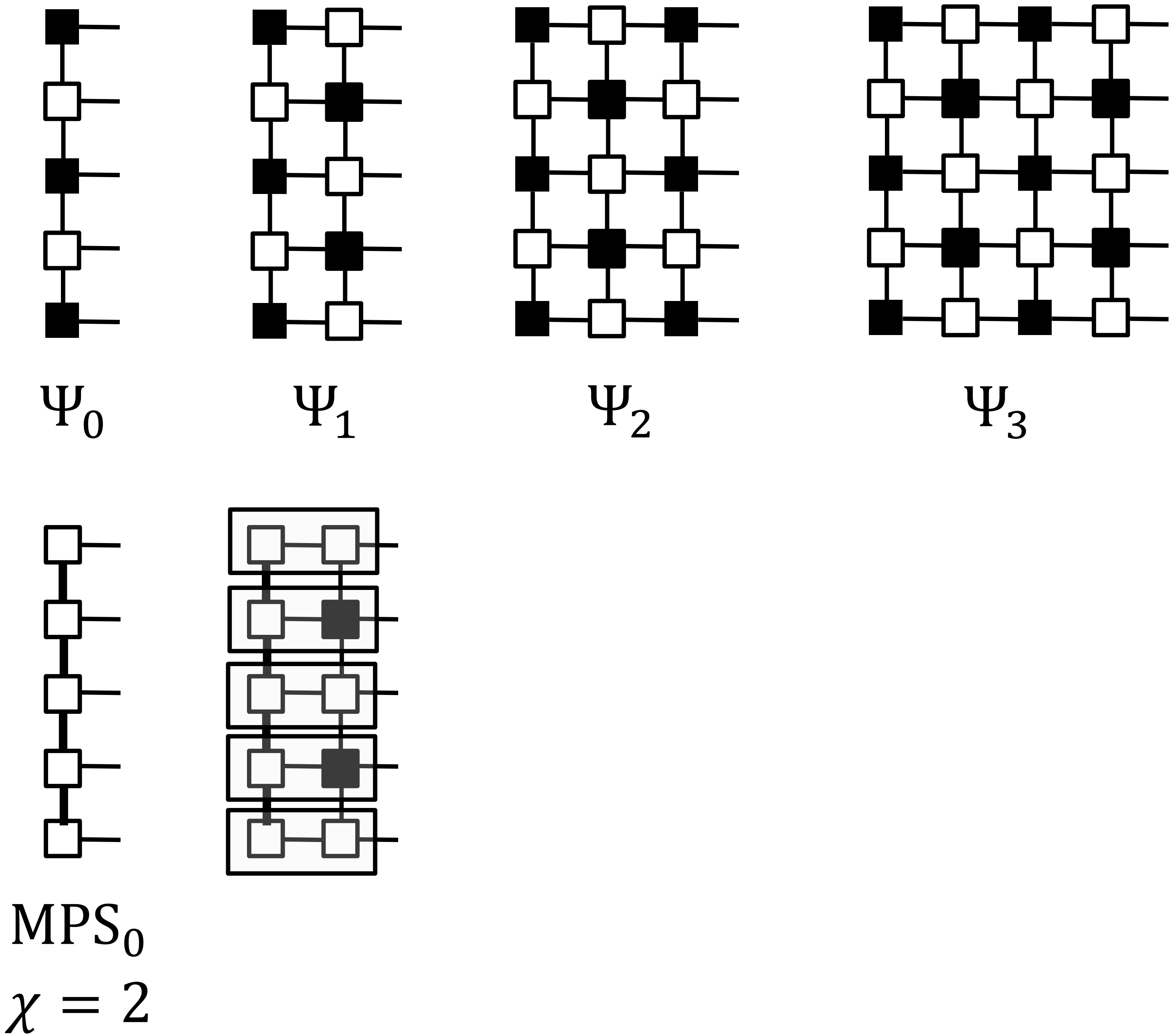


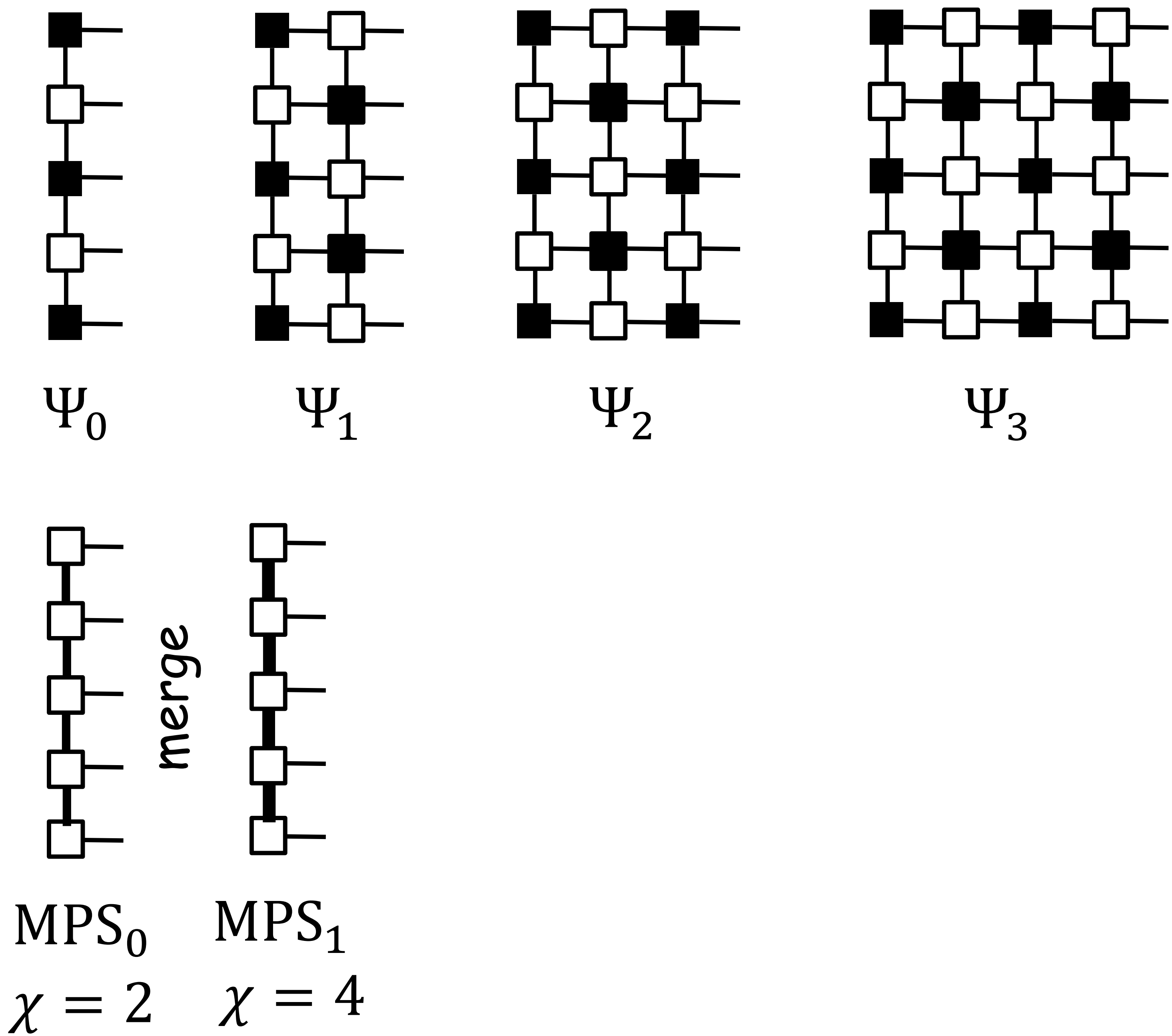
Ψ_0 Ψ_1 Ψ_2 Ψ_3 Ψ_4



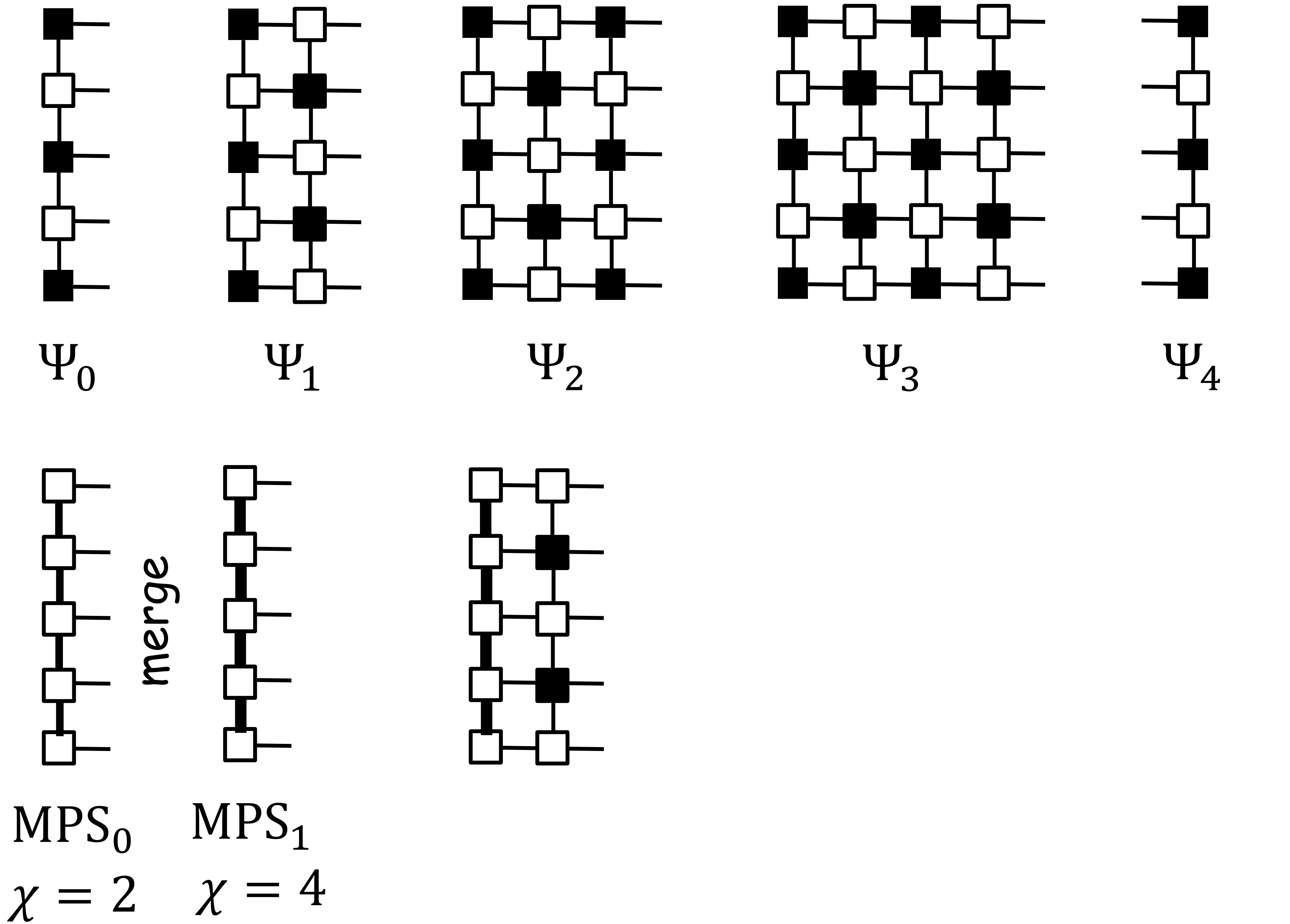
MPS_0

$\chi = 2$

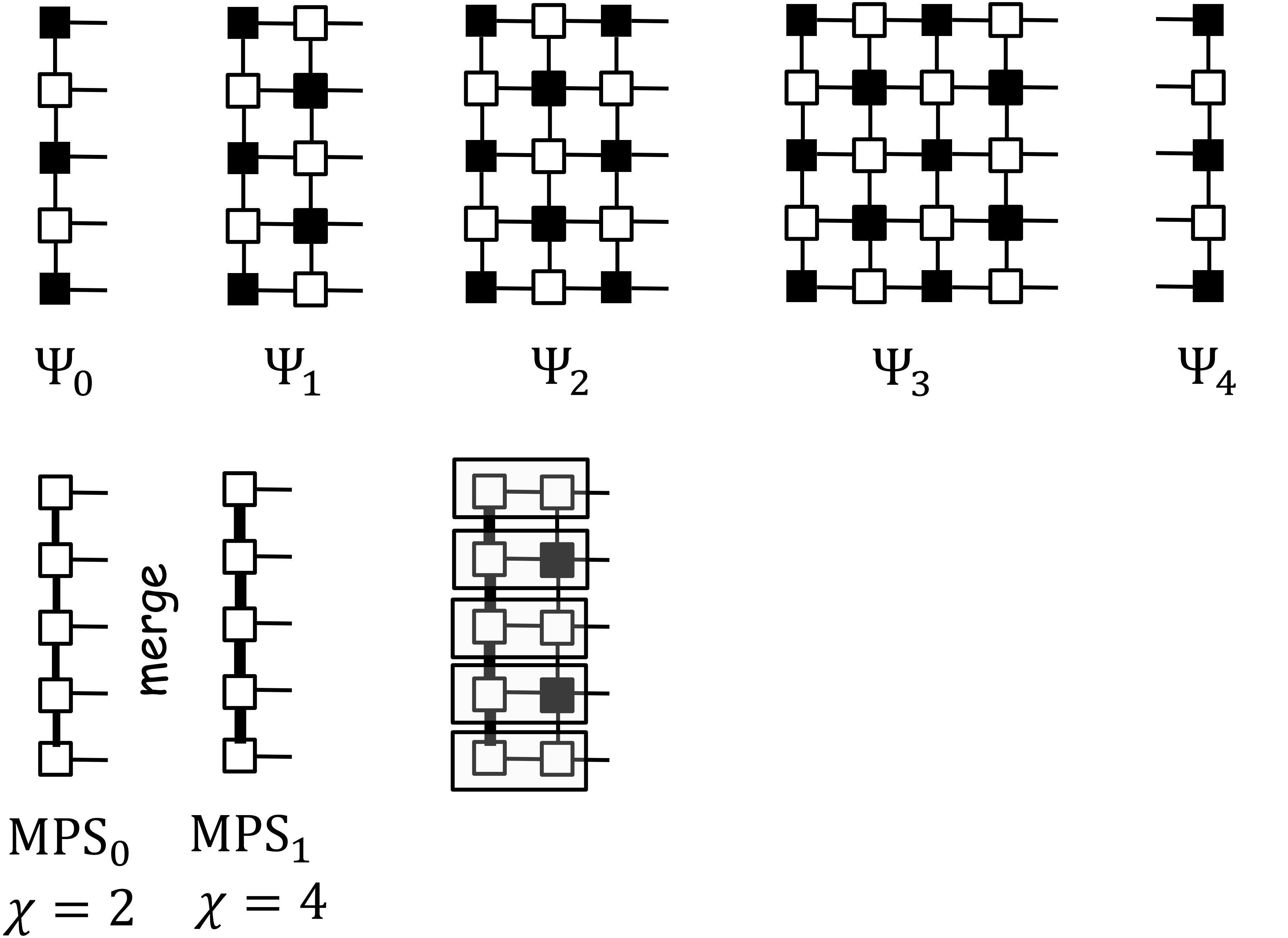

Ψ_0
Ψ_1
Ψ_2
Ψ_3
Ψ_4
MPS_0
$\chi = 2$



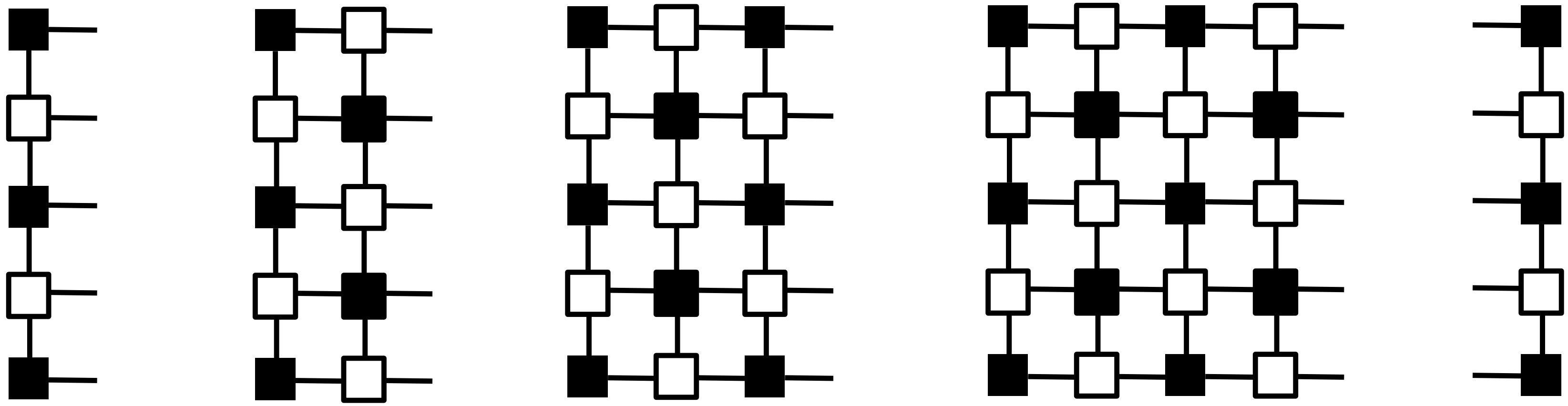
Ψ_0
Ψ_1
Ψ_2
Ψ_3
Ψ_4
merge
MPS_0
$\chi = 2$
MPS_1
$\chi = 4$



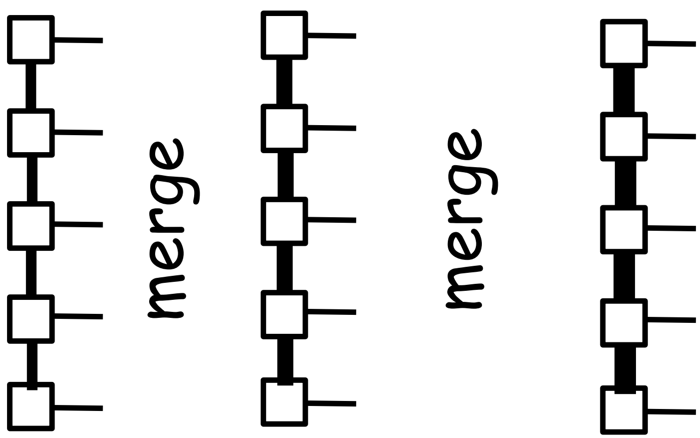
Ψ_0
Ψ_1
Ψ_2
Ψ_3
Ψ_4
merge
MPS_0
$\chi = 2$
MPS_1
$\chi = 4$



Ψ_0
Ψ_1
Ψ_2
Ψ_3
Ψ_4
merge
MPS_0
$\chi = 2$
MPS_1
$\chi = 4$



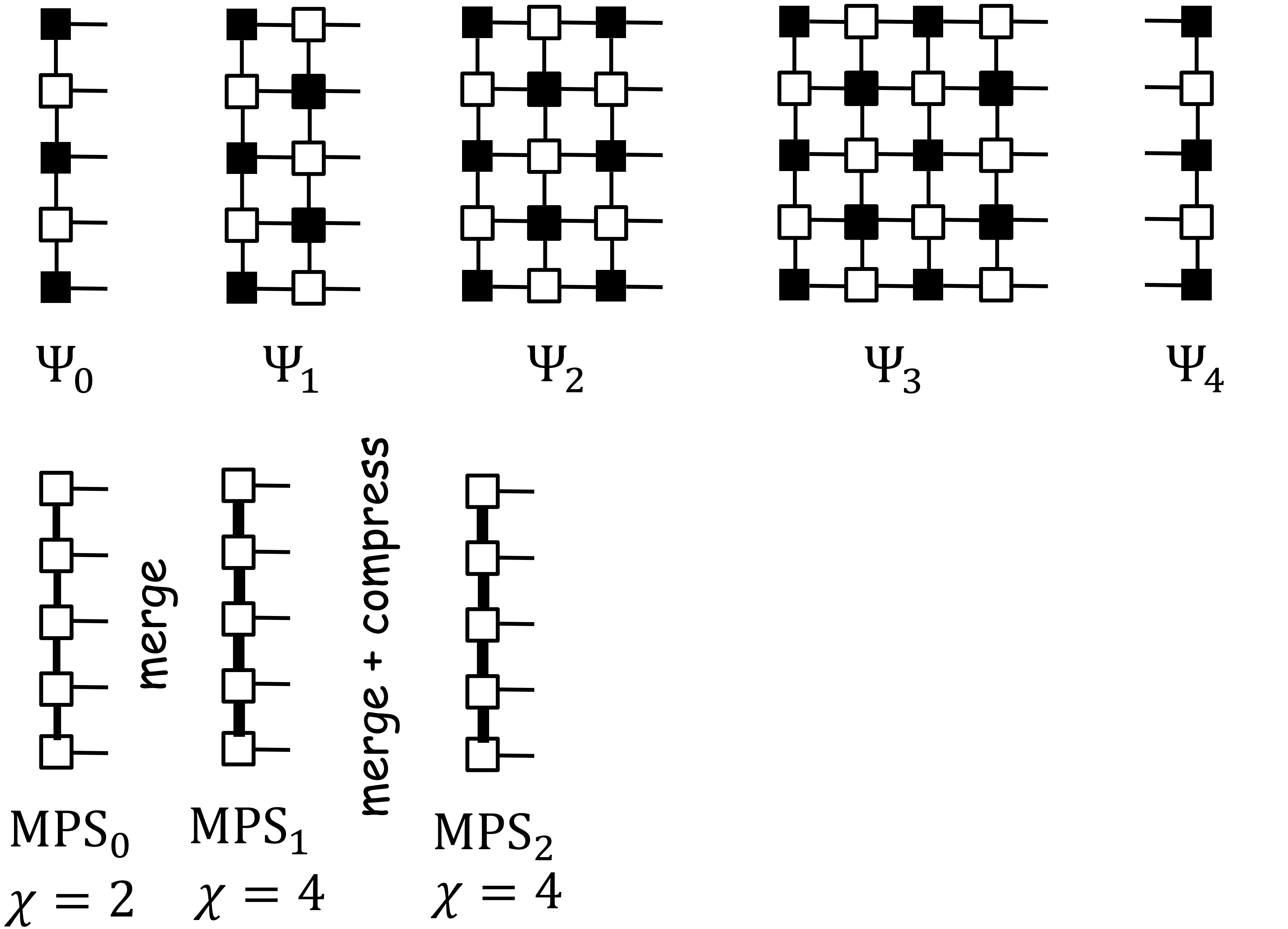
Ψ_0 Ψ_1 Ψ_2 Ψ_3 Ψ_4



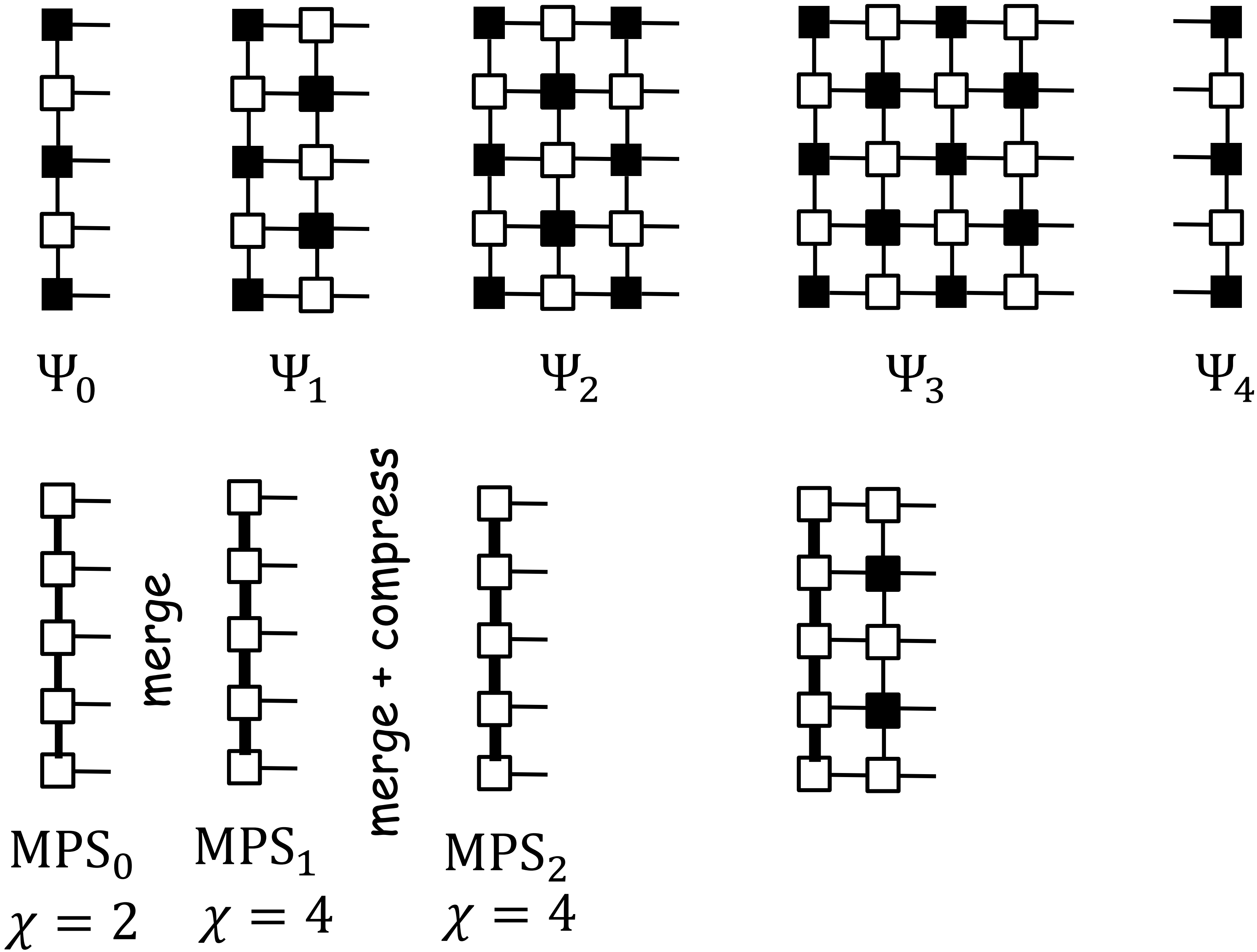
MPS_0 MPS_1 MPS_2

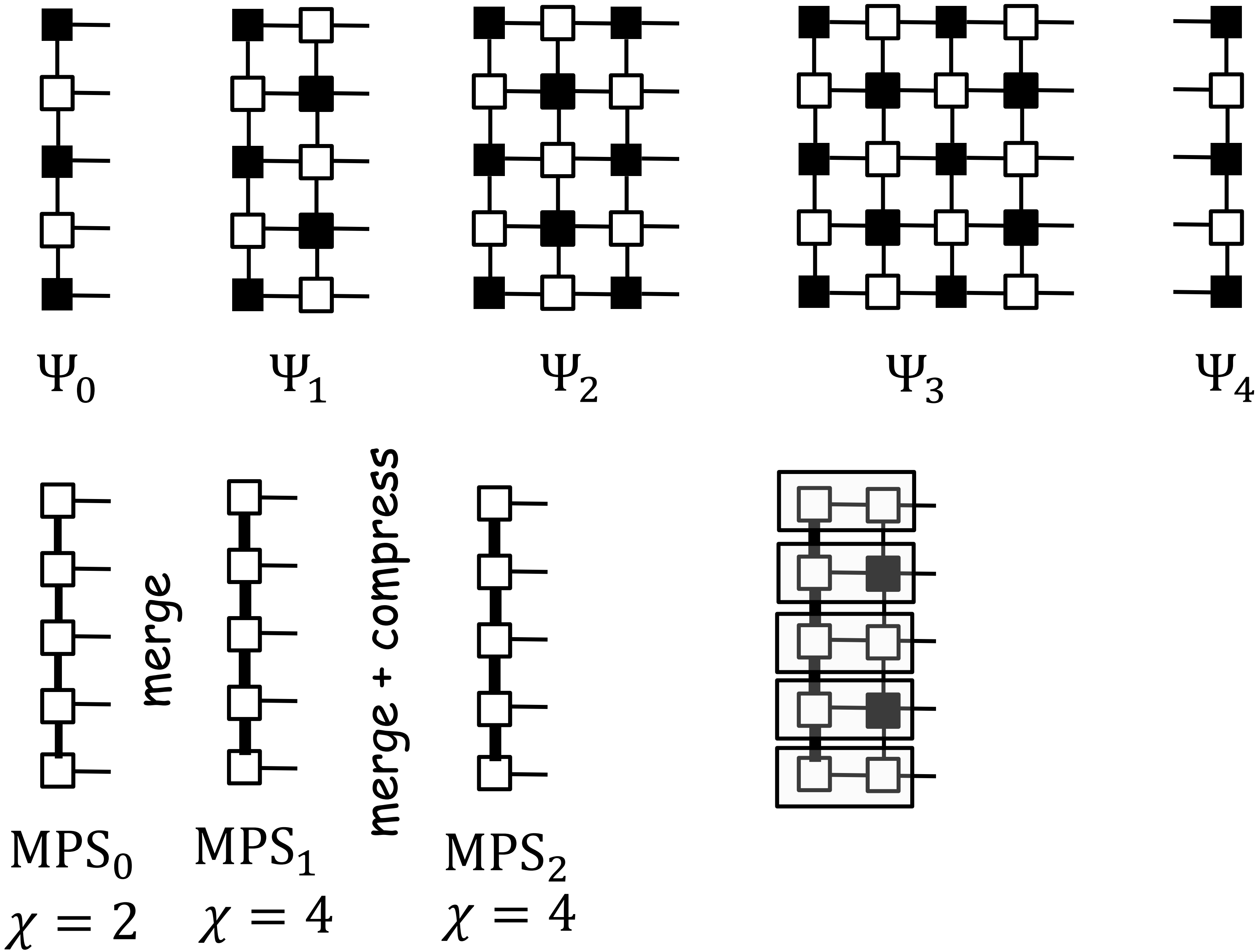
$\chi = 2$ $\chi = 4$ $\chi = 8$

bond dimension
is too large !

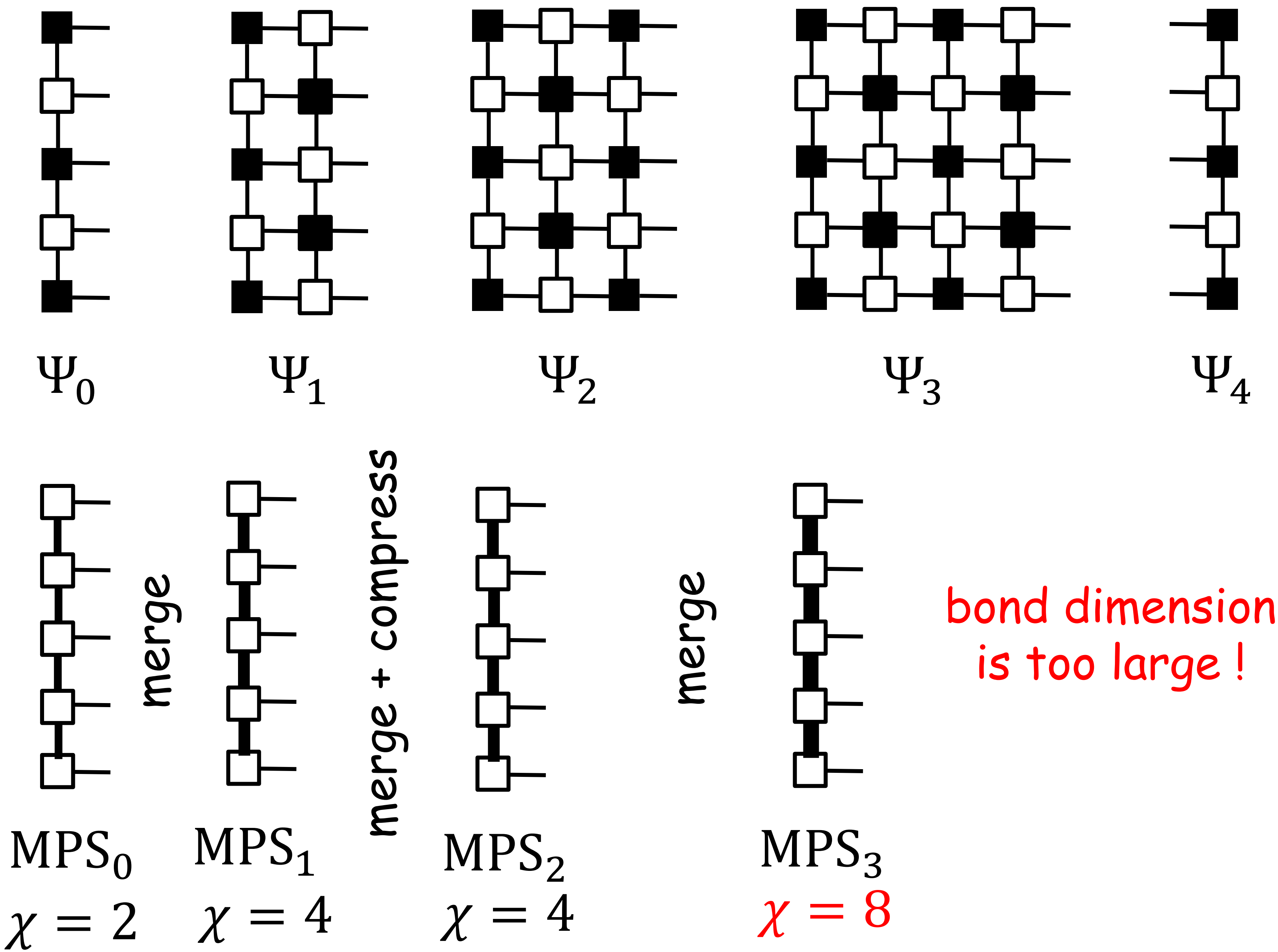


Ψ_0
Ψ_1
Ψ_2
Ψ_3
Ψ_4
merge
merge + compress
MPS_0
$\chi = 2$
MPS_1
$\chi = 4$
MPS_2
$\chi = 4$

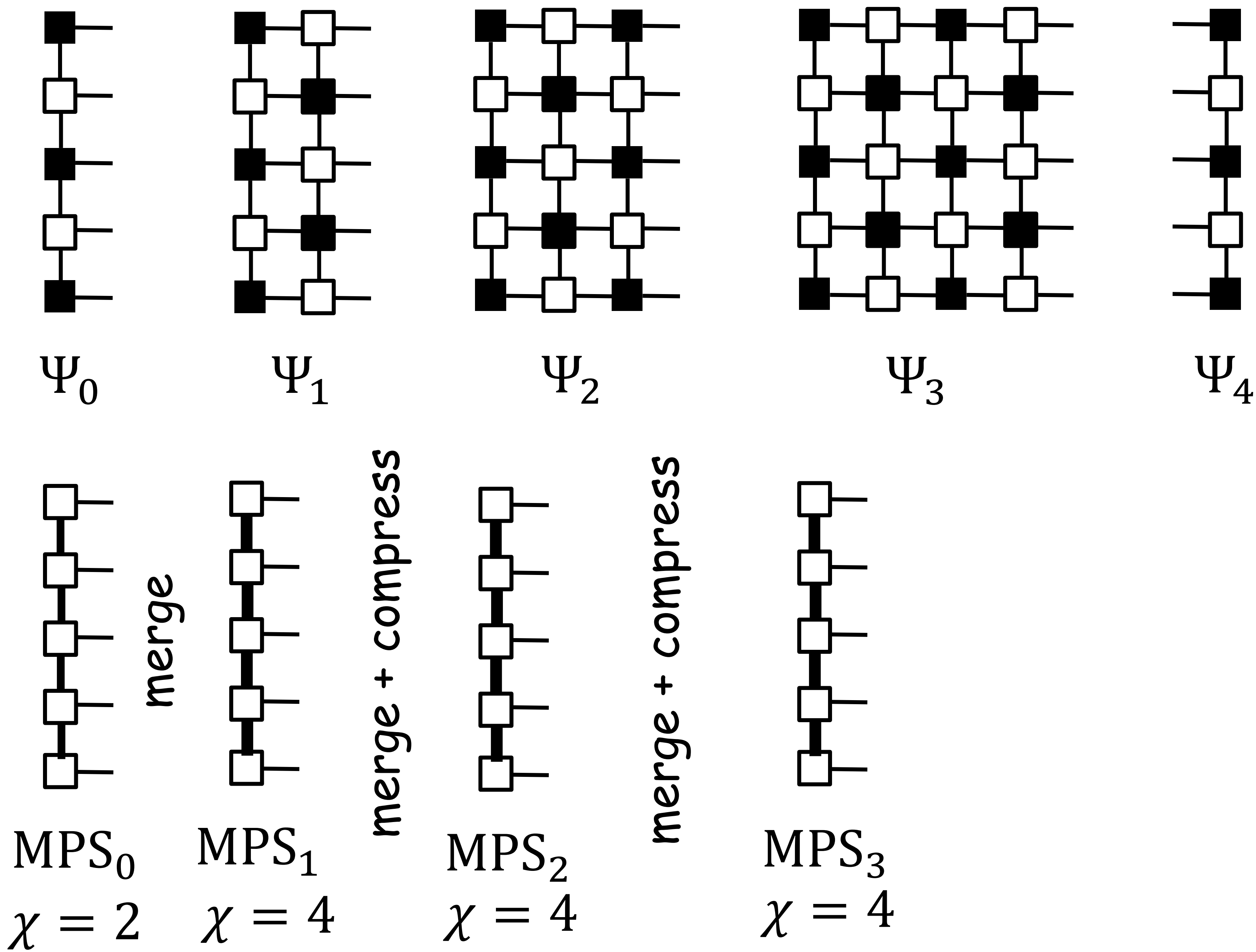

Ψ_0
Ψ_1
Ψ_2
Ψ_3
Ψ_4
merge
merge + compress
MPS_0
$\chi = 2$
MPS_1
$\chi = 4$
MPS_2
$\chi = 4$



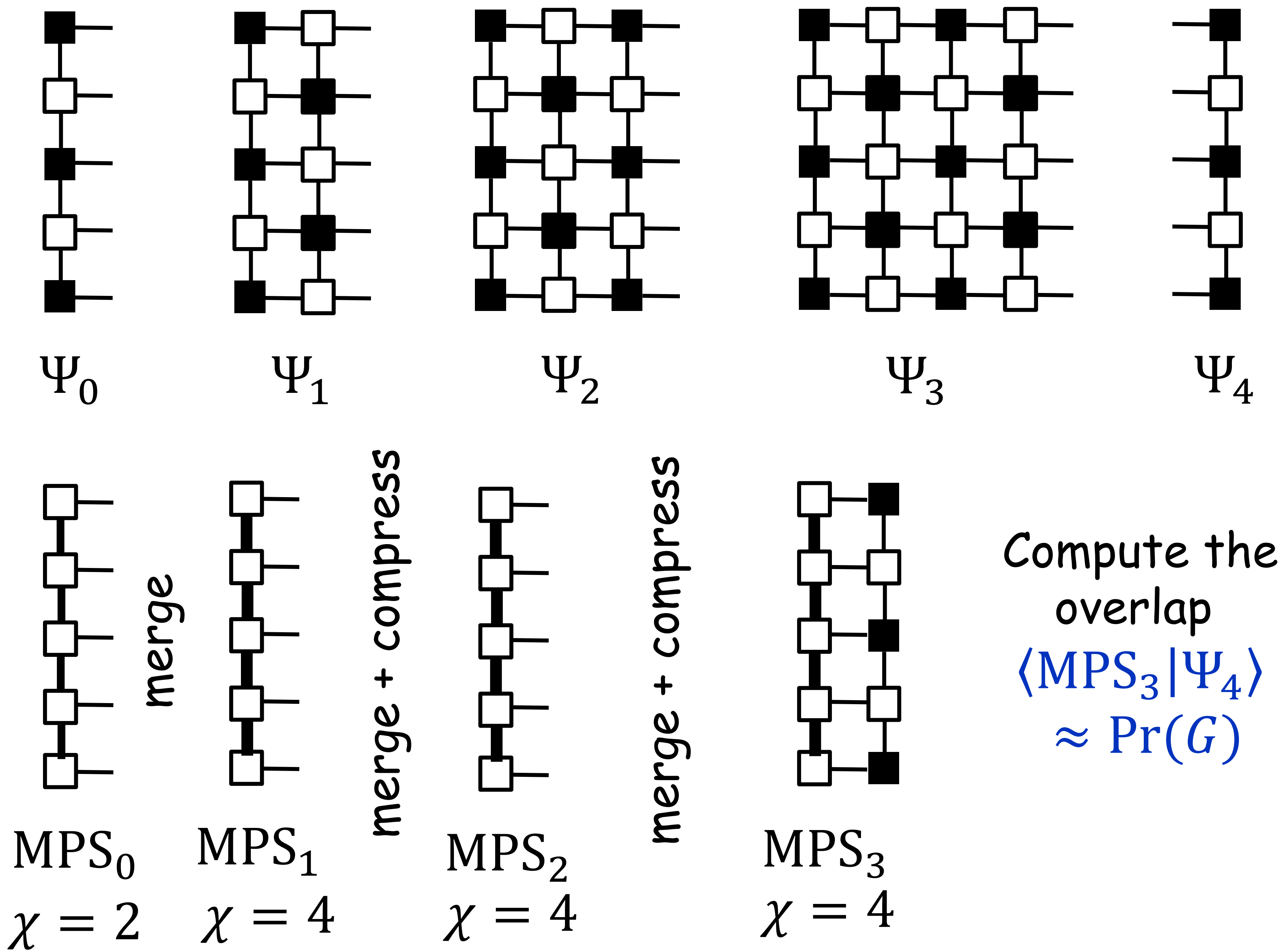
Ψ_0
Ψ_1
Ψ_2
Ψ_3
Ψ_4
merge
merge + compress
MPS_0
$\chi = 2$
MPS_1
$\chi = 4$
MPS_2
$\chi = 4$



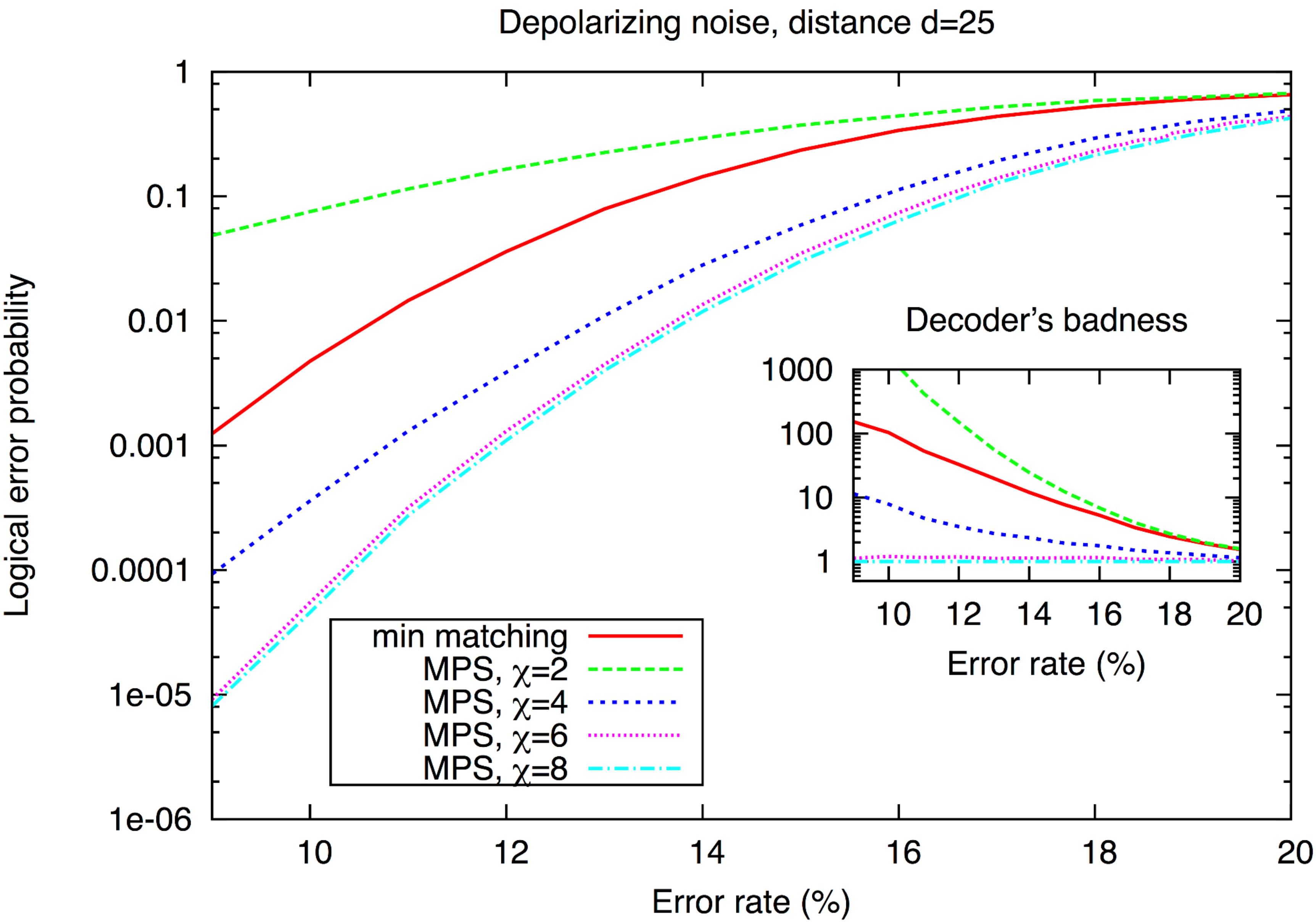
Ψ_0
Ψ_1
Ψ_2
Ψ_3
Ψ_4
merge
merge + compress
merge
bond dimension is too large !
MPS_0
$\chi = 2$
MPS_1
$\chi = 4$
MPS_2
$\chi = 4$
MPS_3
$\chi = 8$

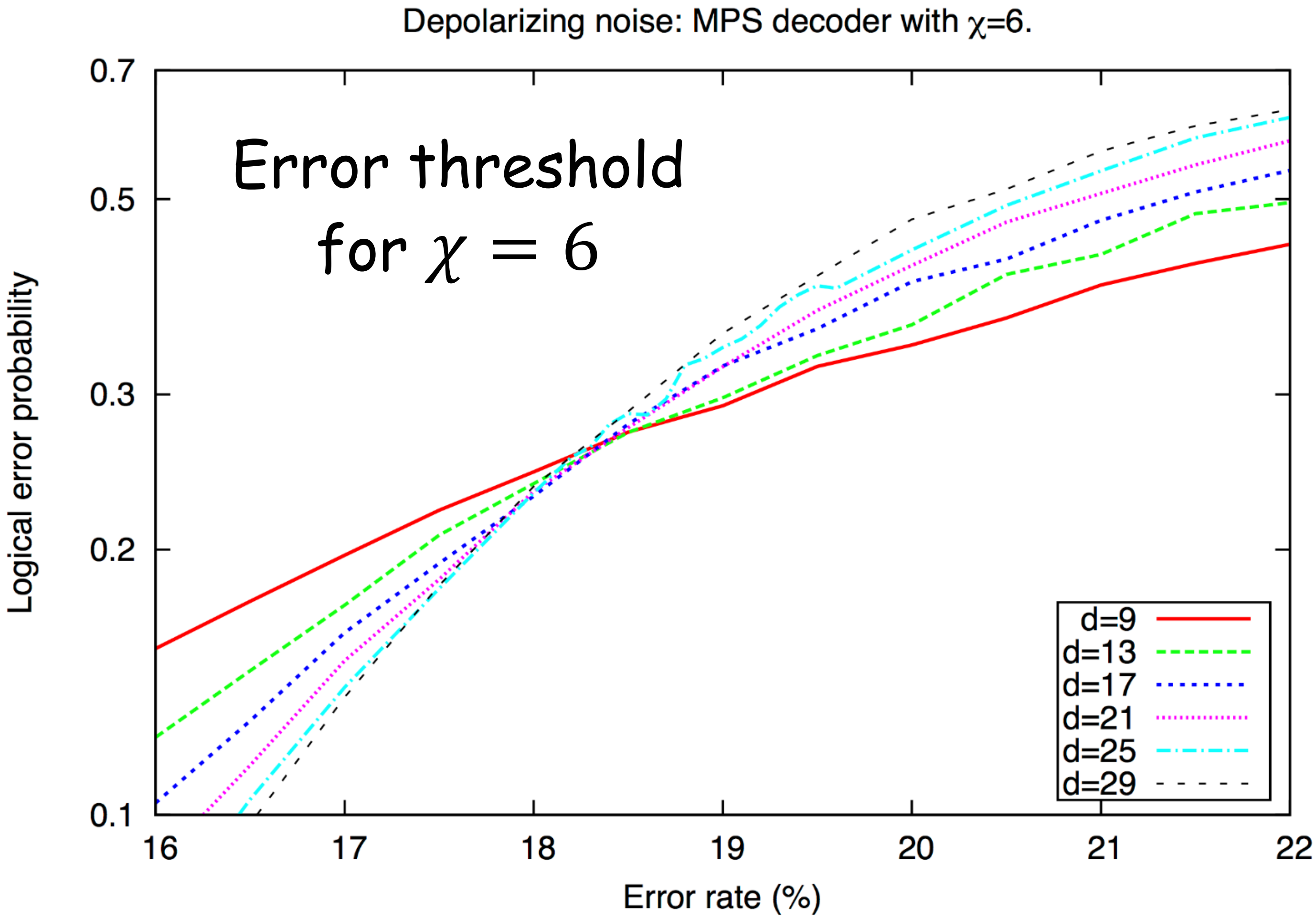


Ψ_0
Ψ_1
Ψ_2
Ψ_3
Ψ_4
merge
merge + compress
merge + compress
MPS_0
$\chi = 2$
MPS_1
$\chi = 4$
MPS_2
$\chi = 4$
MPS_3
$\chi = 4$


Ψ_0
Ψ_1
Ψ_2
Ψ_3
Ψ_4
merge
merge + compress
merge + compress
MPS_0
$\chi = 2$
MPS_1
$\chi = 4$
MPS_2
$\chi = 4$
MPS_3
$\chi = 4$
Compute the overlap
$\langle \mathrm{MPS}_3 | \Psi_4 \rangle$
$\approx \Pr(G)$

Comparison between MPS and MWM decoders





MWM threshold: 15%

Theoretical maximum: 18.9% Bombin et al, PRX 2 021004 (2012)

Markov chain algorithm: 16% Hutter et al, PRA 89 022326 (2014)

How good is the approximation ?

Example: d=25, $\epsilon = 10\%$

χ	$\Pr(G)$	$\Pr(\bar{X}G)$
2	1.11782e-55	2.81823e-89
3	1.11781e-55	2.81777e-89
4	1.11781e-55	2.81781e-89
5	1.11781e-55	2.81781e-89

X-noise:

$$\Pr(X) = \epsilon$$

$$\Pr(I) = 1 - \epsilon$$

$$\Pr(Y) = \Pr(Z) = 0$$

MLD can be implemented exactly in time $O(n^2)$ using a mapping to matchgate quantum circuits

Enables a direct comparison between the MPS-decoder and MLD.

Coset probability for the X-noise:

$$\Pr(fG^X) = \sum_{g\in G^X} \Pr(fg) = \Pr(f) \sum_{g\in G^X} \prod_{e\in g} w_e$$

G^X - subgroup generated by plaquette stabilizers

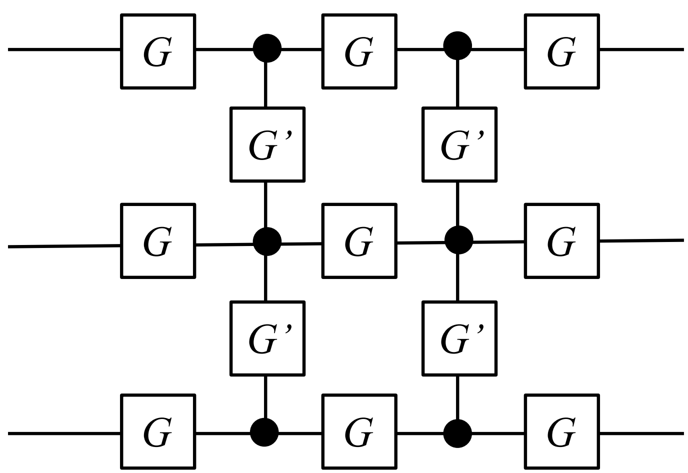
f - Pauli operator of X-type

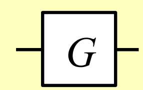
Edge weights: $$w_e = \begin{cases} \dfrac{\varepsilon}{1-\varepsilon} & \text{if } e \notin f \\ \dfrac{1-\varepsilon}{\varepsilon} & \text{if } e \in f \end{cases}$$

Reduction to a quantum circuit simulation

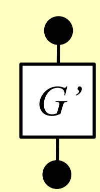
$$\Pr(fG^X) = \Pr(f)\langle\psi_0|U|\psi_0\rangle$$

$$|\psi_0\rangle = \sum_{even\ x} |x\rangle \in (\mathbf{C}^2)^{\otimes d}$$

$U =$ 



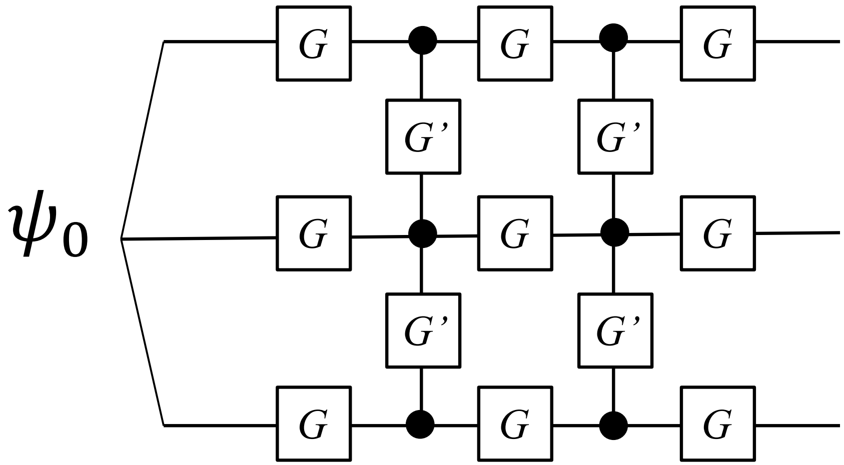
$$= \begin{pmatrix} 1 & 0 \\ 0 & w \end{pmatrix}$$



$$= \begin{pmatrix} 1 & & & w \\ & 1 & w & \\ & w & 1 & \\ w & & & 1 \end{pmatrix}$$

Matchgates

Valiant (2002)



$$\psi_0 \to \psi_1 \to \psi_2 \to \psi_3 \to \psi_4 \to \psi_5$$

Key insight: ψ_i are fermionic Gaussian states.

$$\psi = \mathrm{gauss}(\Gamma, M)$$

$\Gamma = \langle \psi | \psi \rangle$ - norm

$M = 2d \times 2d$ - covariance matrix

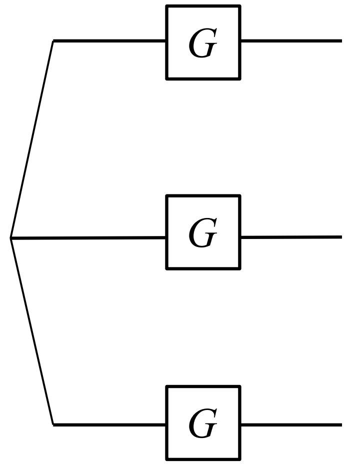
Initial state:

$$|\psi_0\rangle = \sum_{even\ x} |x\rangle = \mathrm{gauss}(\Gamma_0, M_0)$$

$$\Gamma_0 = 2^{d-1}$$

$$M_0 = \begin{array}{|c|c|c|c|c|c|}
\hline
 & & & & & 1 \\
\hline
 & & 1 & & & \\
\hline
 & -1 & & & & \\
\hline
 & & & & 1 & \\
\hline
 & & & -1 & & \\
\hline
-1 & & & & & \\
\hline
\end{array}$$

$$\psi_1 = \psi_0$$



$$= \mathrm{gauss}(\Gamma_1, M_1)$$

$$\Gamma_1 = \Gamma_0 \sqrt{\det(M_0 + A)}$$

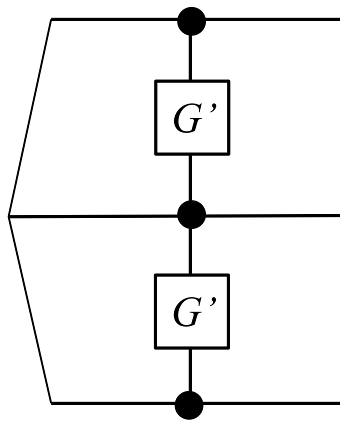
$$M_1 = A - B(M_0 + A)^{-1} B$$

$$A = \begin{bmatrix} & t_1 & & & & \\ -t_1 & & & & & \\ & & & t_2 & & \\ & & -t_2 & & & \\ & & & & & t_3 \\ & & & & -t_3 & \end{bmatrix}$$

$$B = diag(s_1, s_1, \ldots, s_3, s_3)$$

$$t = \frac{1 - w^2}{1 + w^2}$$

$$s = \frac{2w}{1 + w^2}$$

$\psi_2 = \psi_1$  $= \mathrm{gauss}(\Gamma_2, M_2)$

$$\Gamma_2 = \Gamma_1 \sqrt{\det(M_1 + A)} \qquad M_2 = A - B(M_1 + A)^{-1} B$$

$$A = \begin{bmatrix} & & & & & \\ & & s_1 & & & \\ & -s_1 & & & & \\ & & & & s_2 & \\ & & & -s_2 & & \\ & & & & & \end{bmatrix}$$

$$B = diag(1, t_1, t_1, t_2, t_3, 1)$$

$$t = \frac{1 - w^2}{1 + w^2}$$

$$s = \frac{2w}{1 + w^2}$$

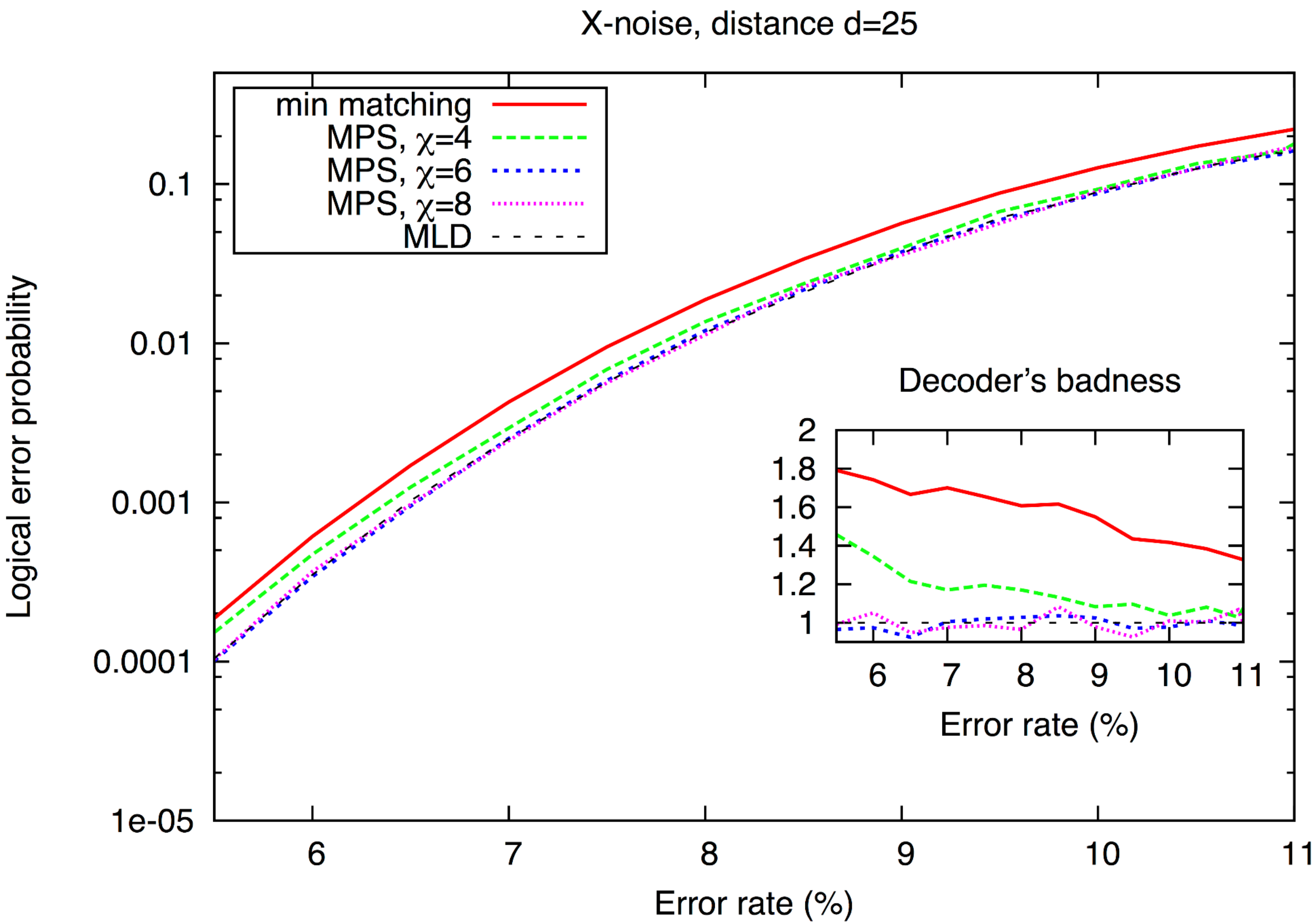
Last step: compute inner product between two Gaussian states $\psi_0 = \mathrm{gauss}(\Gamma_0, M_0)$ and $\psi_5 = \mathrm{gauss}(\Gamma_5, M_5)$

$$\mathrm{Pr}(f G^X) = \mathrm{Pr}(f)\langle\psi_5|\psi_0\rangle$$

$$= \mathrm{Pr}(f)\frac{\sqrt{\Gamma_0\Gamma_5}}{2^{d/2}}\det(M_0 + M_5)^{1/4}$$

Overall time complexity: $(2d-1) \times O(d^3) = O(n^2)$

Comparison between MLD, MPS and MWM decoders



Open problems

- Does the MPS decoder with $\chi = O(1)$ have a non-zero threshold ?

- Exploit parallel algorithms for 2D tensor network contraction to get a running time $poly(\chi)\log(n)$
Evenbly and Vidal, arXiv:1412.0732

- What is the analogue of the ML decoder for non-stabilizer codes/non-Pauli error models ?

- Generalize decoders based on TN contraction to the noisy syndrome readout