Compressing neural networks by tensor networks

Kenji Harada

24 Nov. 2022

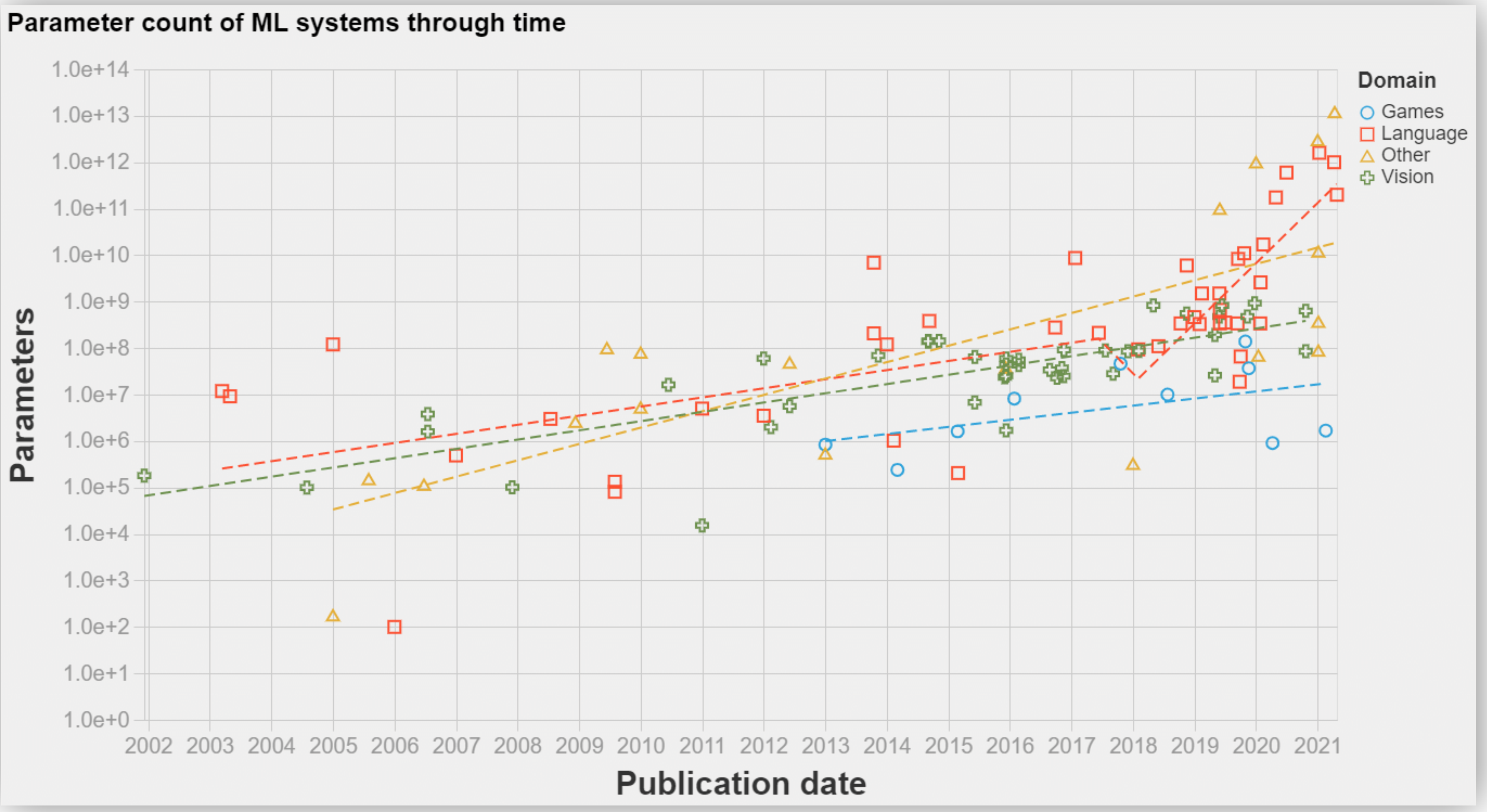
Graduate School of Informatics, Kyoto Univ.

Large scale AI models

AI models are rapidly developing

- scale of models
- computational power

Jaime Sevilla, Pablo Villalobos (2021): Parameter counts in Machine Learning



The number of parameters in AI models becomes a large scale

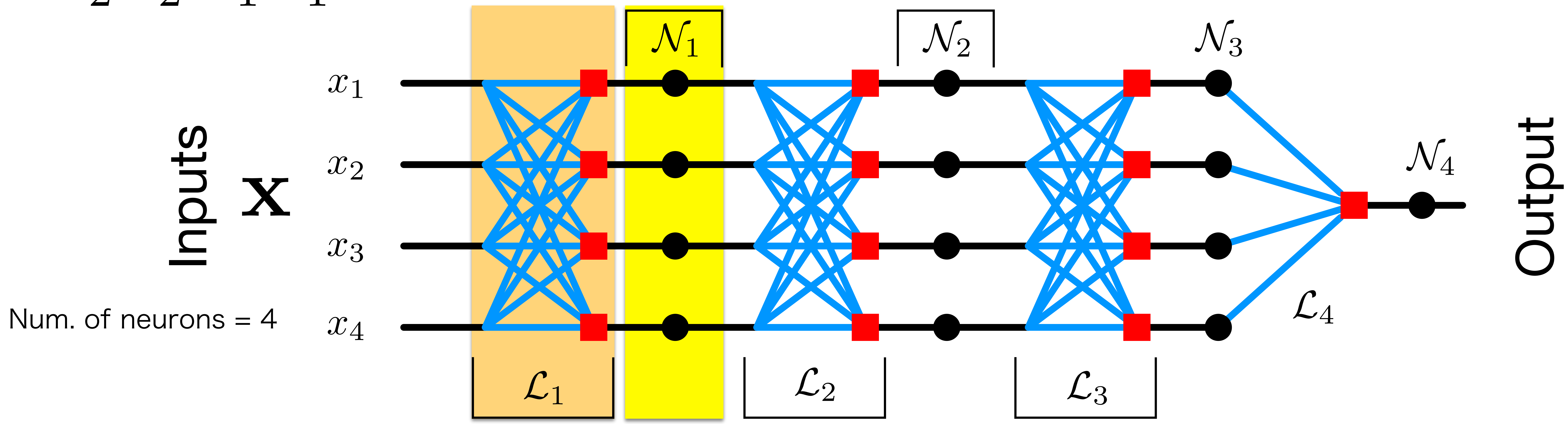
Feed-forward neural network

Layer structure

$$\mathcal{F} = \mathcal{N}_n\mathcal{L}_n \cdots \mathcal{N}_2\mathcal{L}_2\mathcal{N}_1\mathcal{L}_1$$

Non-linear layer : sigmoid, ReLU

$$\mathbf{y} = f(\mathbf{x})$$



$$\mathbf{y} = W\mathbf{x} + \mathbf{b}$$

Linear layer

Num. of parameters in a weight matrix $\#(W) \propto N^2$

➡ **Can we compress the weight matrix?**

A. Novikov, D. Podoprikhin, A. Osokin, and D. Vetrov, “Tensorizing Neural Networks,” NIPS (2016).

Matrix decomposition and compression

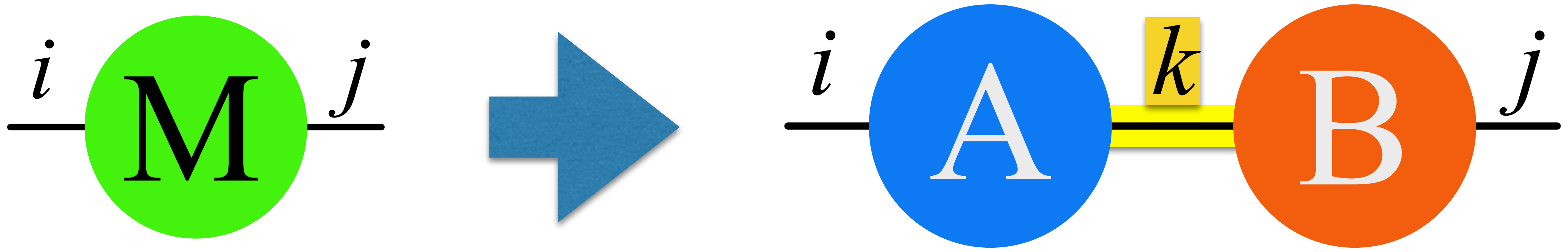
$$M = U\Lambda V^\dagger \approx U\tilde{\Lambda} V^\dagger = \left(U\sqrt{\tilde{\Lambda}}\right)\left(\sqrt{\tilde{\Lambda}}V^\dagger\right) = AB$$

Keep larger singular values

$$M_{ij} = \sum_{k=1}^{N} U_{ik}\Lambda_k \bar{V}_{jk} \approx \sum_{k=1}^{D} U_{ik}\Lambda_k \bar{V}_{jk} = \sum_{k=1}^{D} (U_{ik}\sqrt{\Lambda_k})(\sqrt{\Lambda_k}\bar{V}_{jk}) = \sum_{k=1}^{D} A_{ik}B_{kj}$$

Total number of parameters decreases $N^2 \to 2ND$ **Compression!**

Diagram notation



M_{ij}

$$\sum_{k=1}^{D} A_{ik}B_{kj}$$

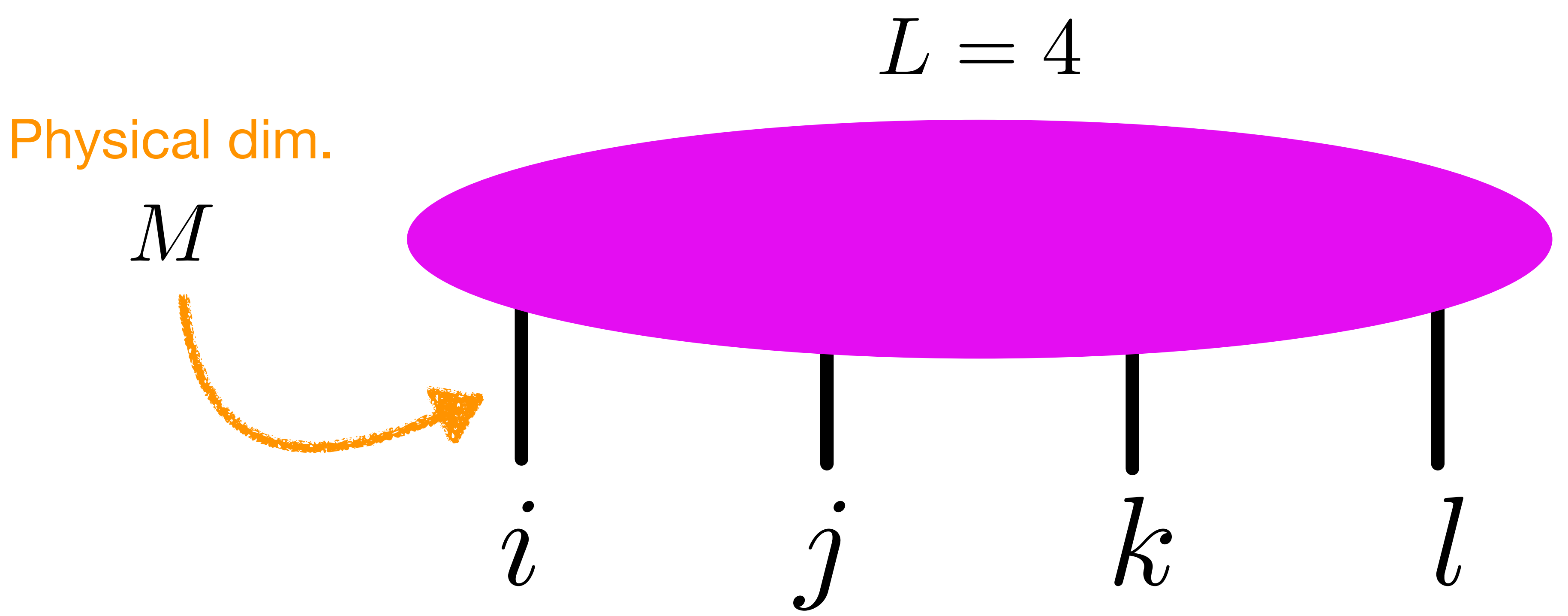
Node: multidimensional array

Edge: index

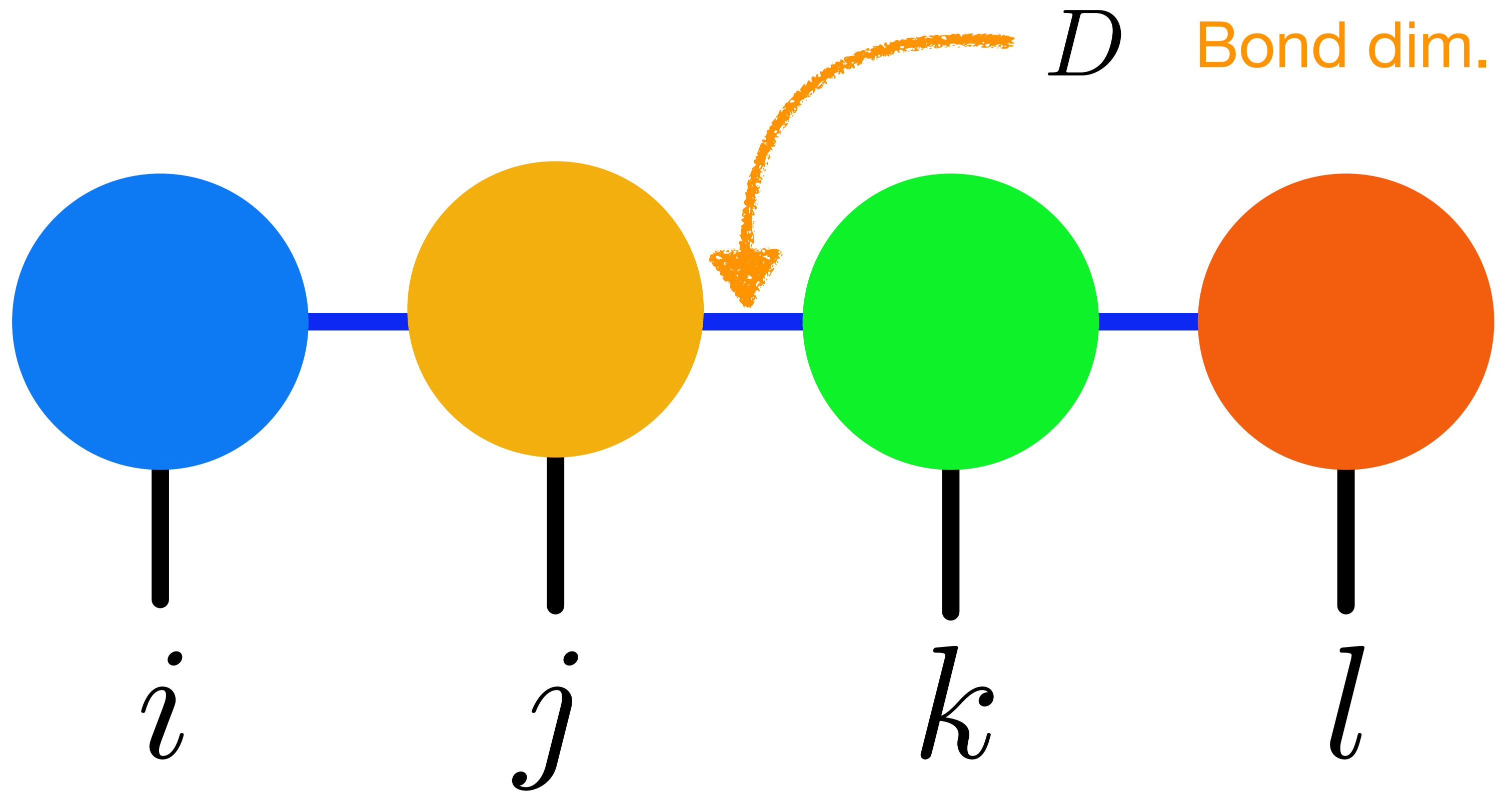
If two edges are connected,
it means the contraction of two indexes.

Tensor decomposition and compression by MPS

Four-dimensional array



MPS (matrix product state)



Number of parameters: $M^L \to L \times M D^2$

From exponential to linear for *L*

Compression!

Wave function defined by MPS

$$|\psi\rangle = \sum_{i,j,k,l} \sum_{\alpha,\beta,\gamma} L^i_\alpha T^j_{\alpha\beta} T^k_{\beta\gamma} R^l_\gamma |ijkl\rangle$$

$$\langle\psi|\psi\rangle = 1$$

Density matrix | Reduced density matrix

$$\rho = |\psi\rangle\langle\psi|, \quad \rho_{(ij)} = \mathrm{Tr}_{(kl)}[\rho]$$

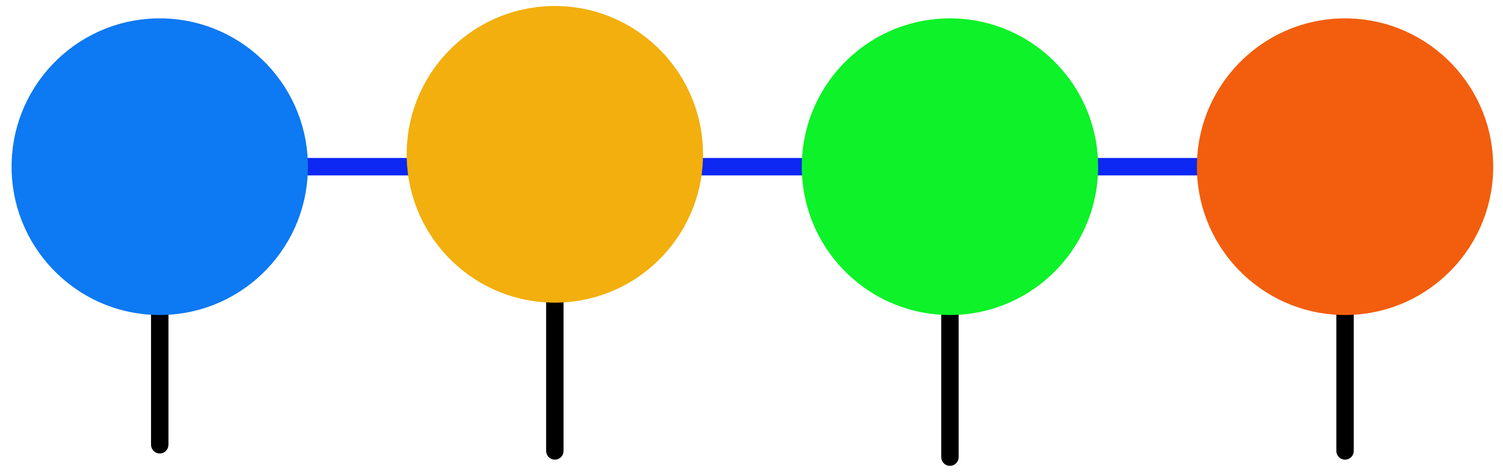
Entanglement entropy

$$S = -\mathrm{Tr}\left[\rho_{(ij)} \ln \rho_{(ij)}\right] \leq \ln D$$

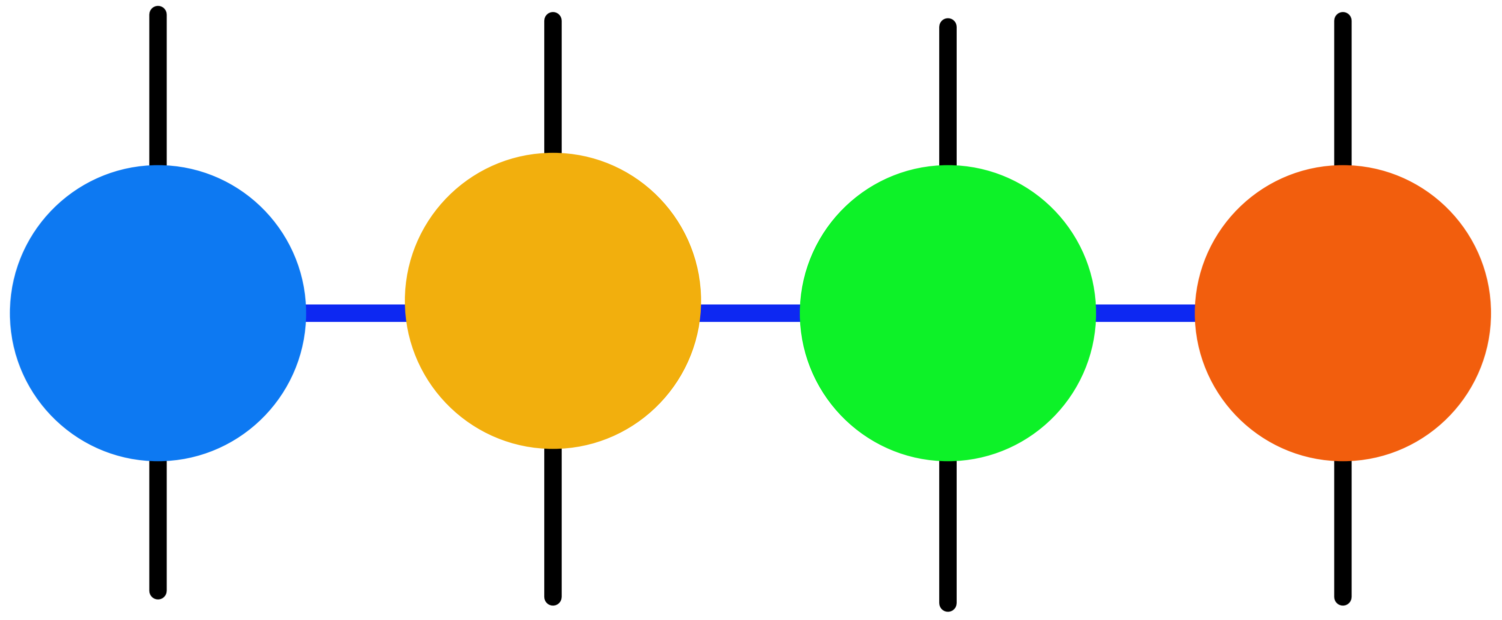
$$A_{(ij),(kl)} = \langle (ij)(kl)|\psi\rangle = \left(U\Lambda V^\dagger\right)_{(ij),(kl)} \Rightarrow S = -\sum_{m=1}^{D} \lambda_m \ln \lambda_m$$

Area law in tensor networks

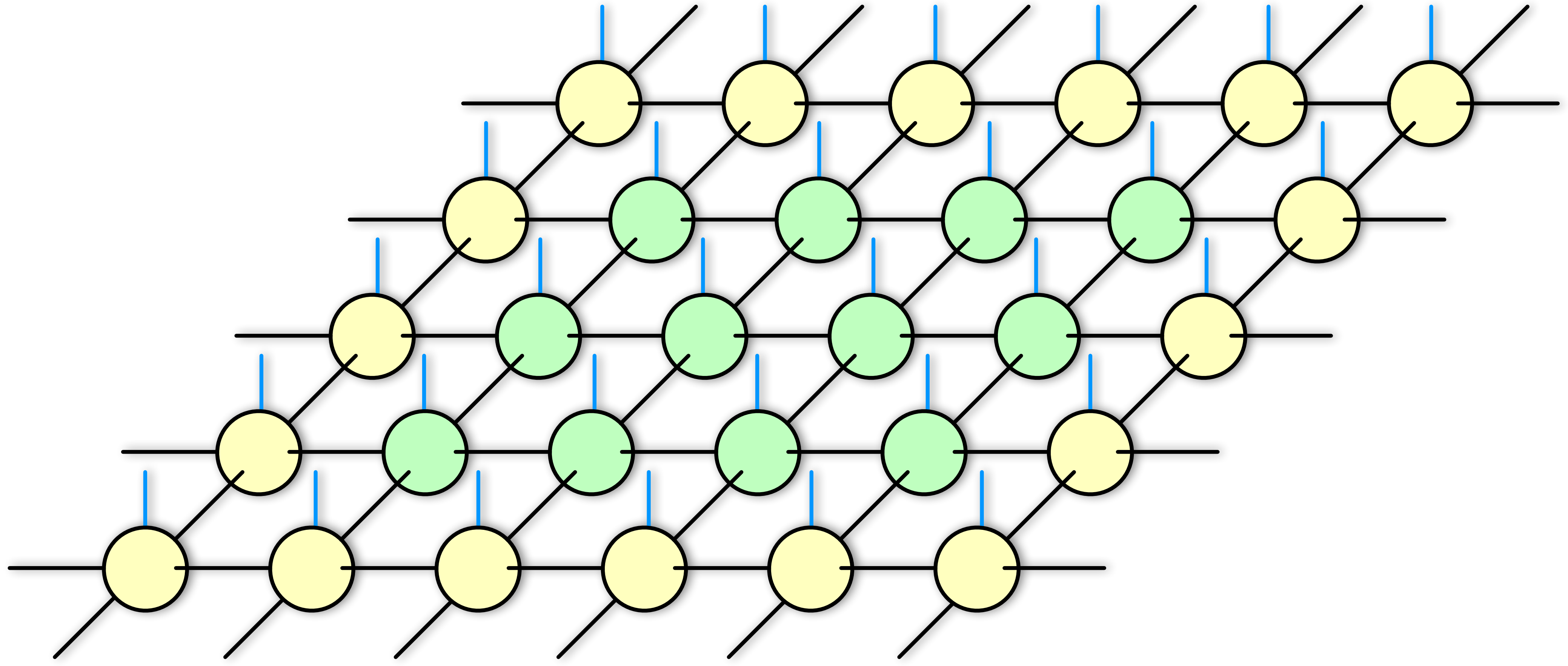
MPS (matrix product state)



MPO (matrix product operator)

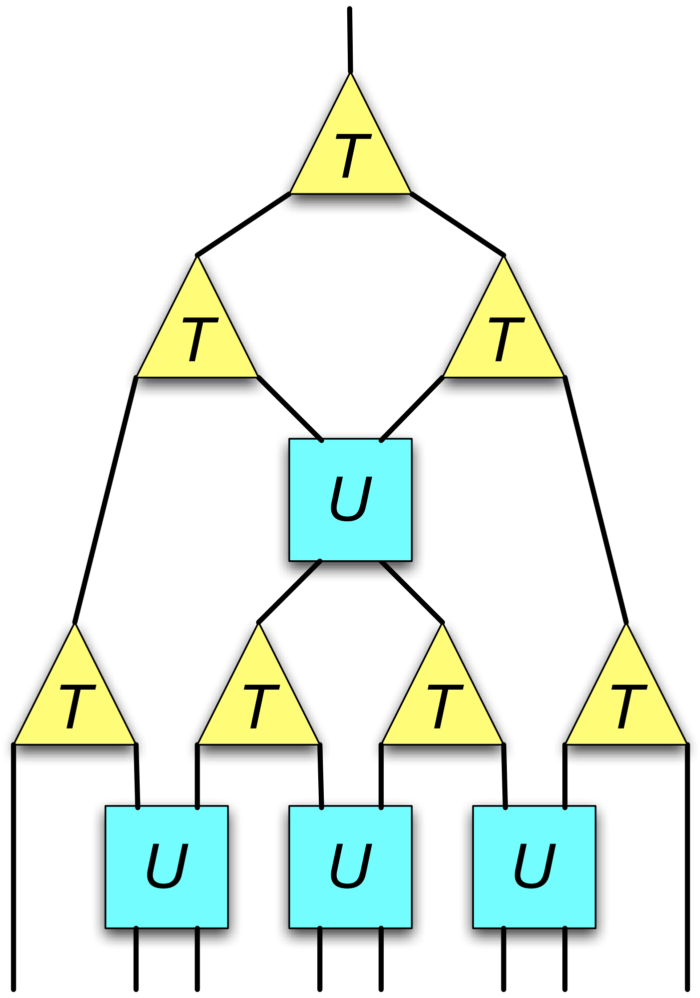


TPS (tensor product state),
PEPS (projected entangled pair state)



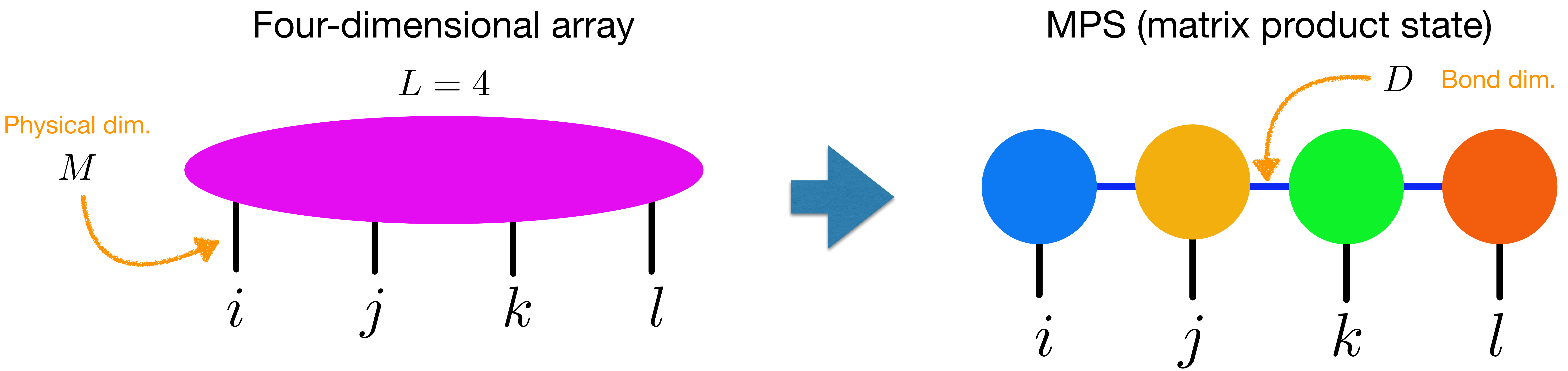
$$S_A \propto \partial A$$

MERA (multiscale entanglement renormalization ansatz)



Various tensor network states hold the area law of entanglement entropy

Entanglement in MPS



Entanglement entropy

$$S = -\mathrm{Tr}\left[\rho_{(ij)} \ln \rho_{(ij)}\right] = -\sum_{m=1}^{D} \lambda_m \ln \lambda_m \leq \ln D$$

MPS represents a class of quantum states in which entanglement is limited

Note. MPS = **Tensor train** (Oseledets, SIAM J. Sci. Comput. 2011)

Tensorization of a weight matrix

Weight matrix in a linear layer: $\mathbf{y} = W\mathbf{x} + \mathbf{b}$

$$W = (W_i^j)$$ Weight between input neuron *i* and output neuron *j*

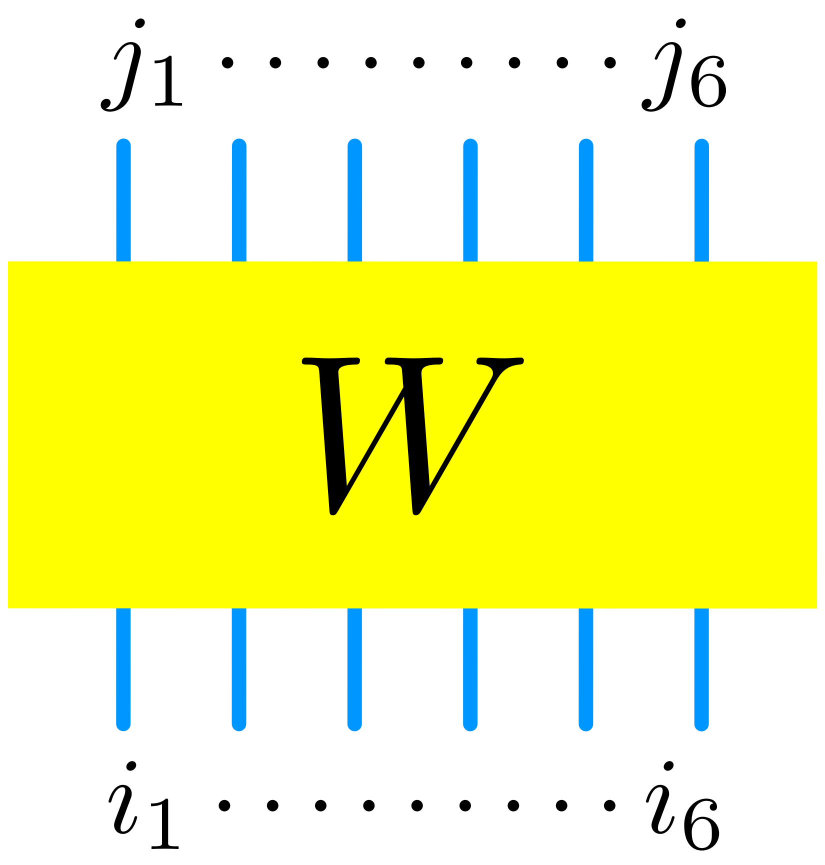
Tensorization

Num. of input neurons	$N_x = a_1 \times a_2 \times \cdots \times a_n$
Num. of output neurons	$N_y = b_1 \times b_2 \times \cdots \times b_n$

$$i \to (i_1, \cdots, i_n) \qquad j \to (j_1, \cdots, j_n)$$

$$i = \sum_{l=1}^{n} a_l \cdot i_l \qquad j = \sum_{l=1}^{n} b_l \cdot j_l$$

$$W_i^j = W_{i_1 \cdots i_n}^{j_1 \cdots j_n}$$

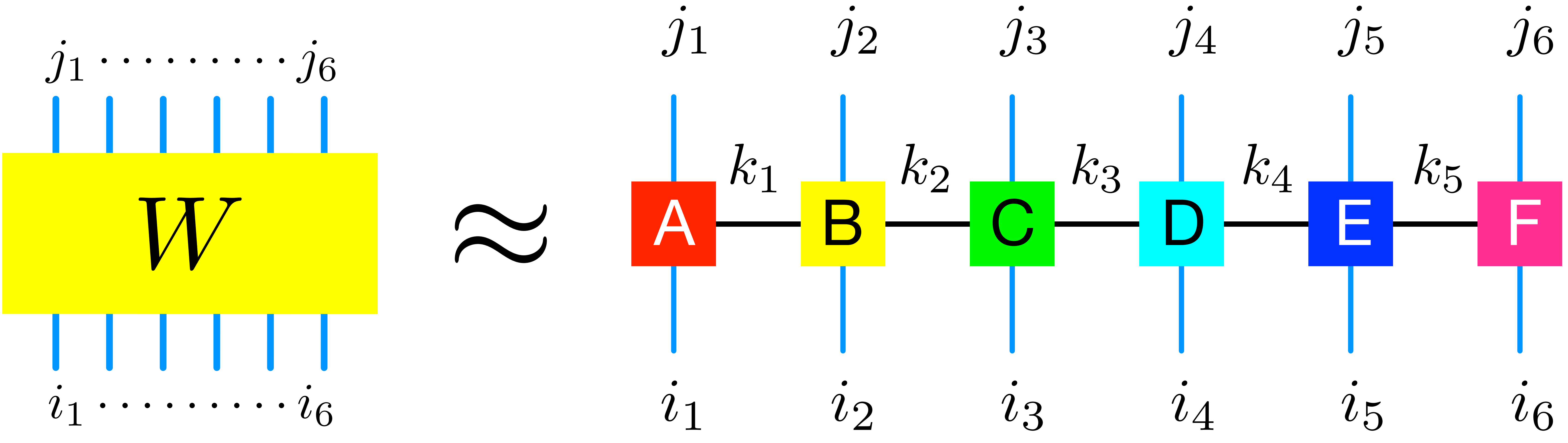


Note. The num. of elements does not change

$$N = N_x \times N_y$$

MPO representation of a tensorized weight matrix

$$W_{i_1\cdots i_n}^{j_1\cdots j_n} \approx \sum_{k_1\cdots k_5} A_{i_1 j_1 k_1} B_{i_2 j_2 k_1 k_2} C_{i_3 j_3 k_2 k_3} D_{i_4 j_4 k_3 k_4} E_{i_5 j_5 k_4 k_5} F_{i_6 j_6 k_5}$$



Bond dim.
$$\chi_l = \dim(k_l)$$

Matrix Product Operator (MPO)

Number of elements

Compression!

$$N_x N_y = \prod_l (a_l b_l) \sim O(a^n b^n) \quad \Leftrightarrow \quad a_1 b_1 \chi_1 + \sum_{l=2}^{5} a_l b_l \chi_{l-1} \chi_l + a_6 b_6 \chi_5 \sim O(nab\chi^2)$$

Proportional to the length of MPO

in an original weight matrix

in a MPO

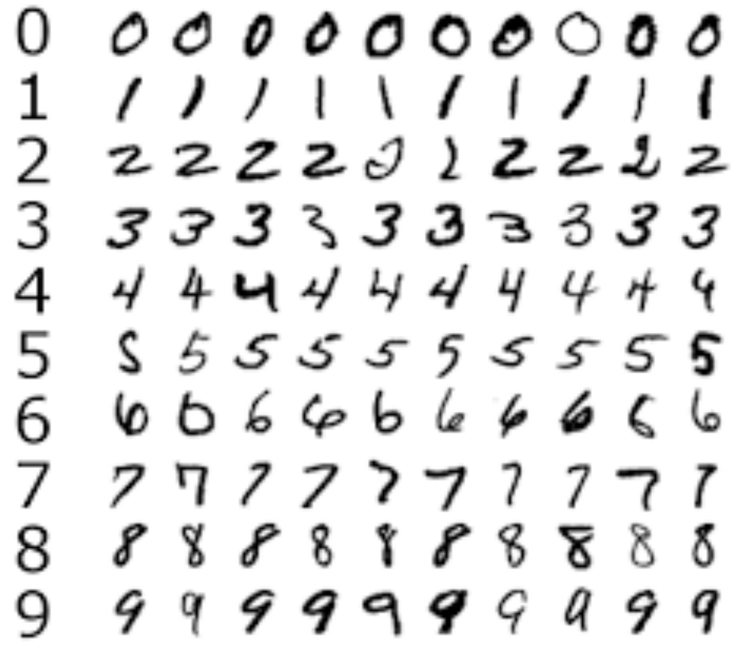
Performance of a tensorized neural network

Z.-F. Gao, et al., "Compressing deep neural networks by matrix product operators," Phys. Rev. Research, vol.2, 023300 (2020).

FC2 network for MNIST

Network structure : two fully connected layers

No.	Layer name	Input size	Output size	Comment	N_{para}	Represented
1	FC	28×28	256		200704	Yes
	ReLu					
2	FC	256	10		2560	Yes
	Softmax					



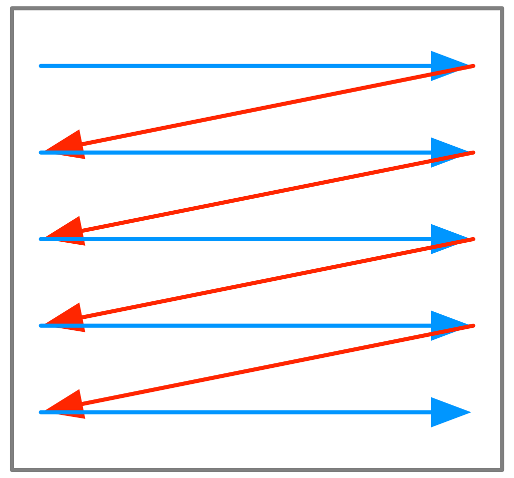
$$\text{Compression ratio} = \frac{\text{Num. of parameters in MPO}}{\text{Num. of parameters in an original weight matrix}}$$

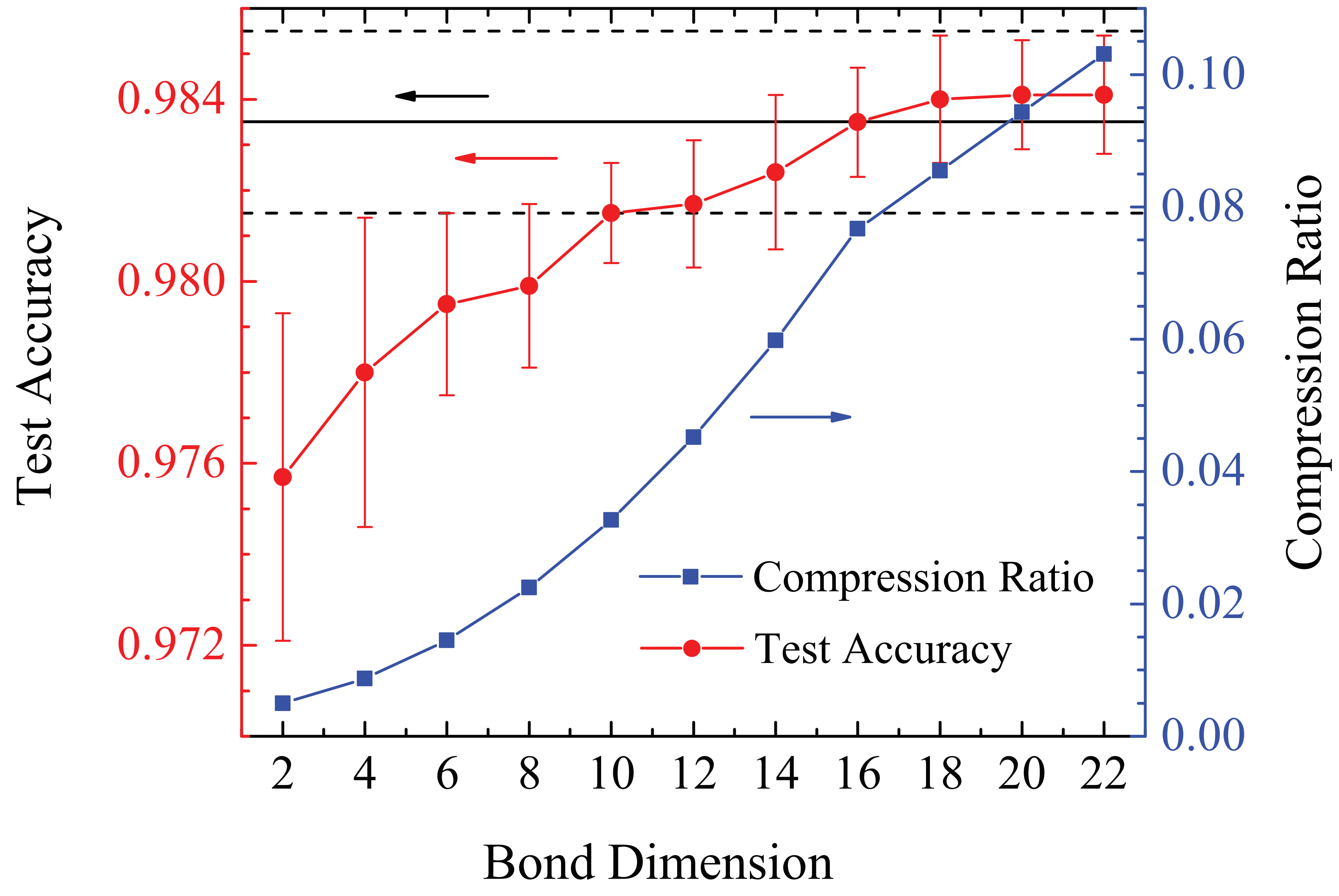
MPO rep. of weight matrix

The first layer $W^{4,4,4,4}_{4,7,7,4} : \chi = D$

The second layer $W^{1,1,10,1}_{4,4,4,4} : \chi = 4$

Ordering of neurons in an image





Compression ratio in other cases

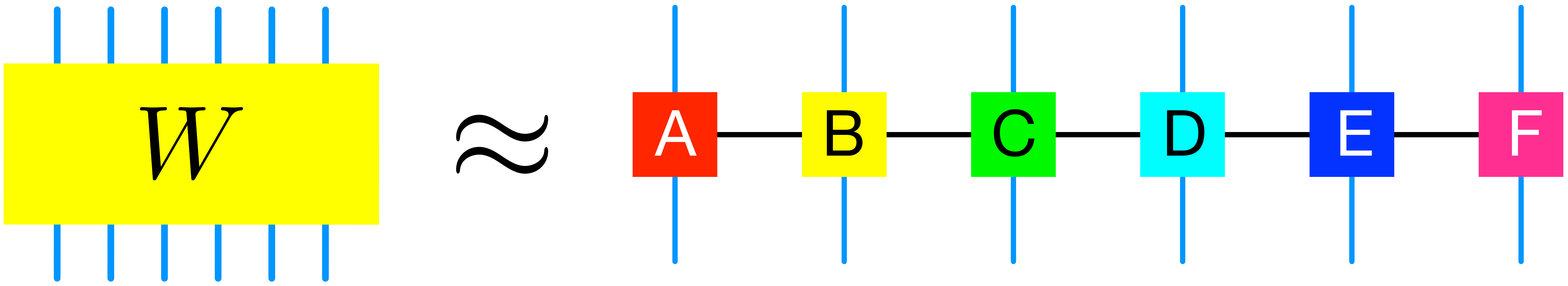
Data set	Network	Original Rep Accuracy	MPO-Net Accuracy	MPO-Net Compression ratio
MNIST	LeNet-5	99.17 ± 0.04	99.17 ± 0.08	0.05
CIFAR-10	VGG-16	93.13 ± 0.39	93.76 ± 0.16	$\sim$0.0005
	VGG-19	93.36 ± 0.26	93.80 ± 0.09	$\sim$0.0005

The compression ratio of MPO-Net is small!

Features of tensorized neural networks

A. Novikov, D. Podoprikhin, A. Osokin, and D. Vetrov, “Tensorizing Neural Networks,” NIPS (2016).
Z.-F. Gao, et al., “Compressing deep neural networks by matrix product operators,” Phys. Rev. Research, vol.2, 023300 (2020).

- Compression of weight matrixes by MPO in a neural network
    - Low computational cost
    - Applicability to any NN: FC2, VGG, ResNet, DenseNet
    - High compression rate: MNIST, CIFAR-10, Fashion-MNIST



Why is the weight matrix effectively compressed?

Observation of effective components in weight matrix

“Entanglement in MPO”

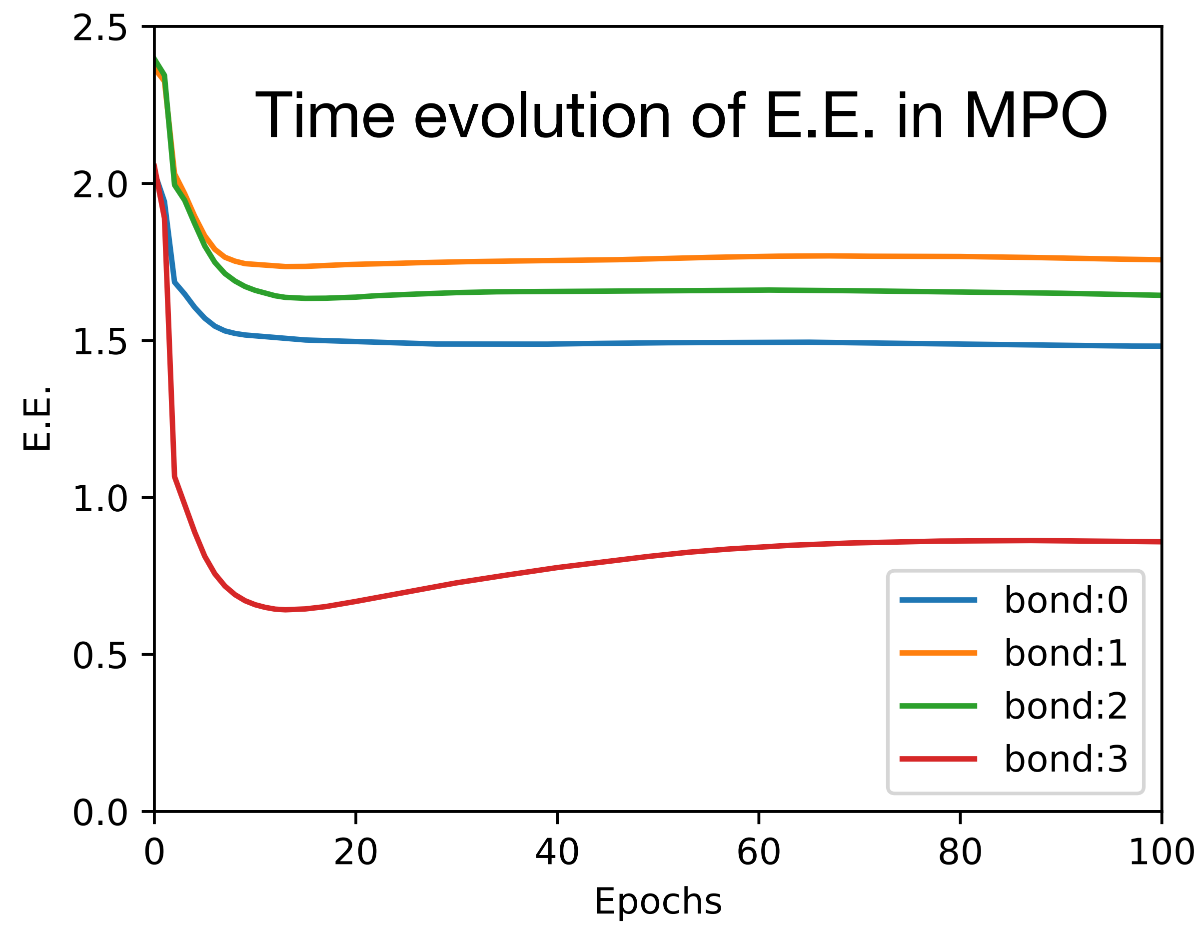
Asoshina and Harada: “Entanglement analysis of neural networks with MPS,”
JPS Autumn Meeting 2021 (22pL4-9).
Asoshina and Harada: “Automatic rank optimization of MPO in tensorized neural networks,”
JPS Autumn Meeting 2022 (14pH112-1)

(in preparation)

MPO + FC

MPO	4,4,4,4,4 $\to$ 4,4,4,4,4, χ=12
ReLU	
FC	1024 $\to$ 10
Softmax	

Dataset: MNIST (32x32)

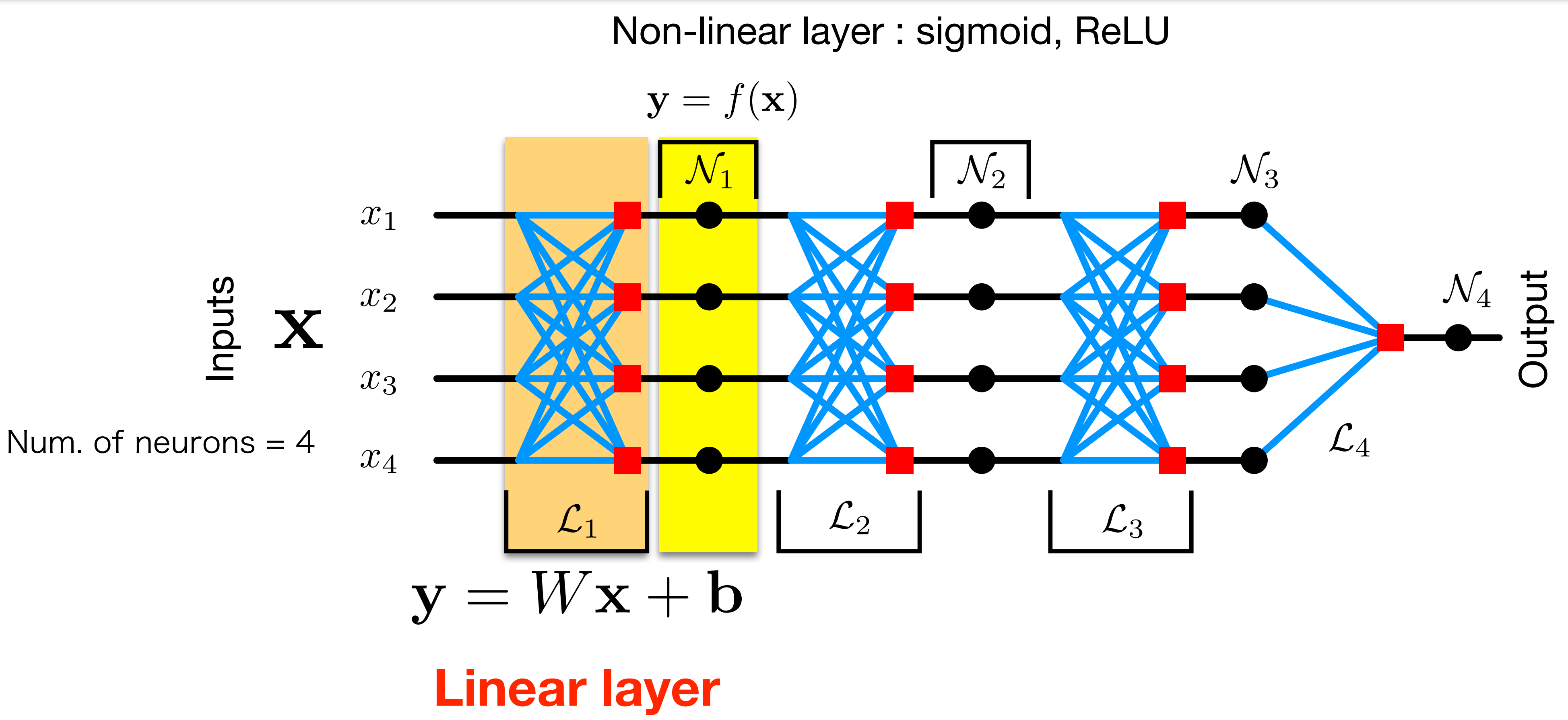


Supported by TOYOTA-Kyoto Univ. joint project

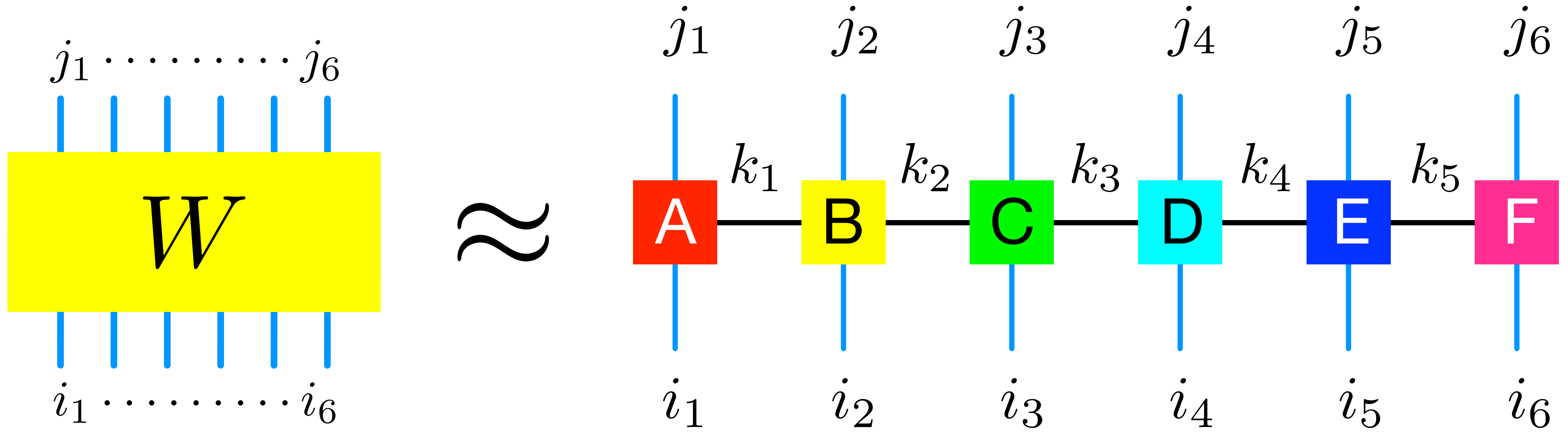
Summary and discussion

Compression of neural networks

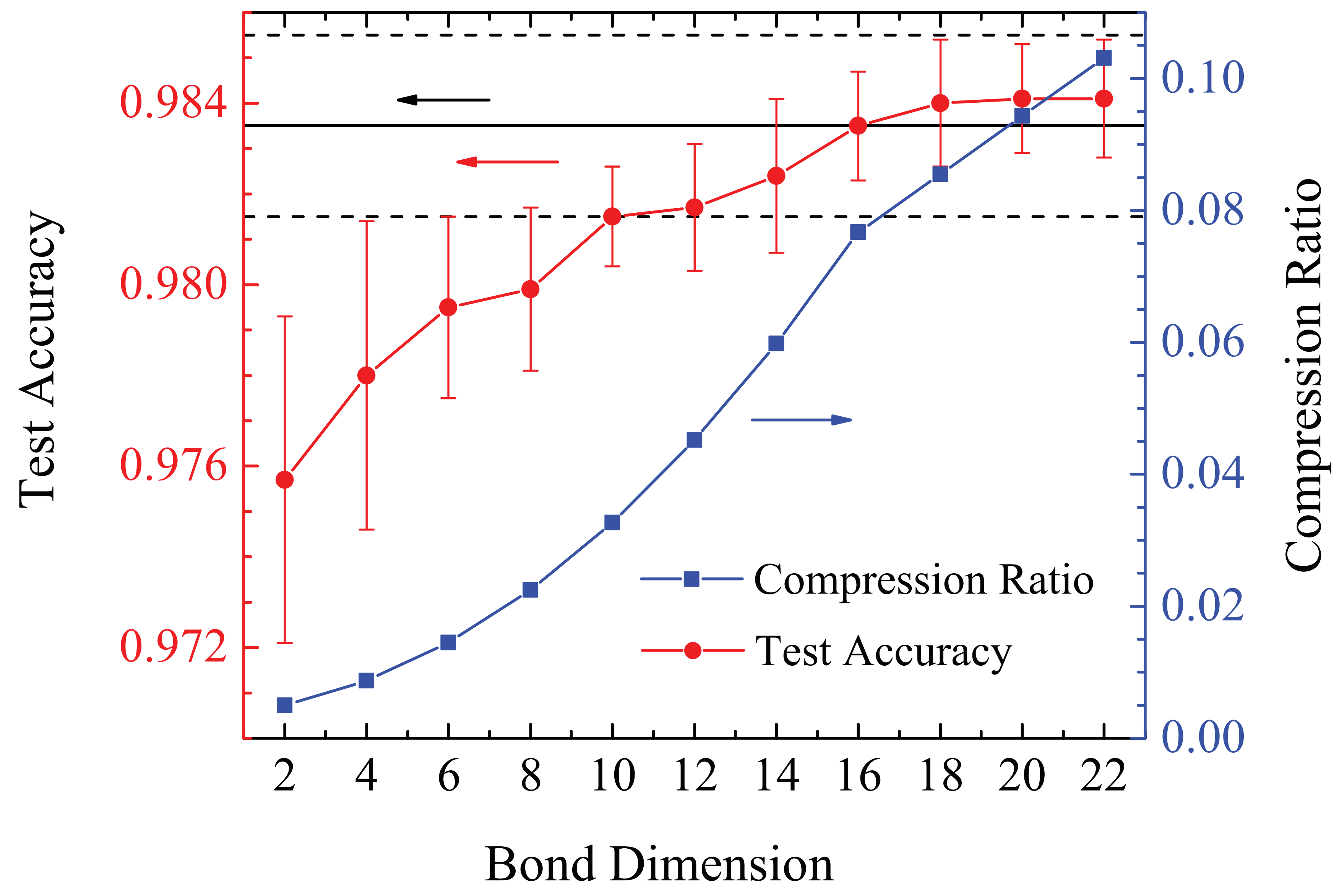
A. Novikov, D. Podoprikhin, A. Osokin, and D. Vetrov, “Tensorizing Neural Networks,” NIPS 2016.



MPO (matrix product operator)



Z.-F. Gao, et al. Phys. Rev. Research, vol.2, 023300 (2020).



Why is the weight matrix effectively compressed?