

2023-12-06



テンソルネットワークと情報圧縮

藤堂眞治（東京大学大学院理学系研究科）

テンソルネットワークと情報圧縮

- ・テンソルネットワークの基礎

- ・次元の呪い
- ・計算科学におけるデータ圧縮
- ・テンソルとテンソルネットワーク
- ・テンソルネットワーク表現
- ・テンソルネットワークの縮約
- ・テンソルネットワークによる情報圧縮

- ・テンソルネットワークの応用

- ・テンソルネットワークくりこみ群
- ・量子系の時間発展
- ・画像データの圧縮
- ・テンソルネットワークによる機械学習
- ・階層構造の圧縮表現

参考文献

- ・ 日本語の文献
 - ・ 「テンソルネットワークの基礎と応用: 統計物理・量子情報・機械学習」(SGCライブラリ 169), 西野友年 サイエンス社 (2021)
 - ・ 数理科学 2022年2月号特集「テンソルネットワークの進展」
 - ・ 「テンソルネットワーク形式の進展と応用」西野友年, 大久保毅 日本物理学会誌 2017年10号
 - ・ 「テンソルネットワークによる情報圧縮とフラストレート磁性体への応用」大久保毅、物性研究 2018年
- ・ 英語の文献
 - ・ R. Orús, “A practical introduction to tensor networks: Matrix product states and projected entangled pair states”, Annals. of Physics 349, 117 (2014). (レビュー)
 - ・ Tao Xiang, “Density matrix and tensor matrix renormalization”, Cambridge University Press, 2023. (教科書)
 - ・ tensornetwork.org の “Review Articles and Learning Resources”に載っている文献・ビデオ

次元の呪い

次元の呪い (curse of dimensionality)

- n 元超立方体(1辺の長さ 2, 体積 2^n)に対する n 次元単位球の体積の割合

$$q = \frac{\pi^{n/2}/\Gamma(\frac{n}{2} + 1)}{2^n} \sim (\pi/n)^{n/2}$$

- $n = 10$ で 0.2%, $n = 20$ で 10^{-8} , $n = 100$ で 10^{-70}
- モンテカルロ積分で球の体積を計算しようとすると, 標準偏差に対する平均値の割合は指数関数的に小さい

$$\frac{q}{\sqrt{q(1-q)}} \sim \sqrt{q}$$

- 次元が高くなるにつれて指数関数的に大きな M が必要となる
- 通常の数値積分(台形公式等)でも同様

古典統計物理

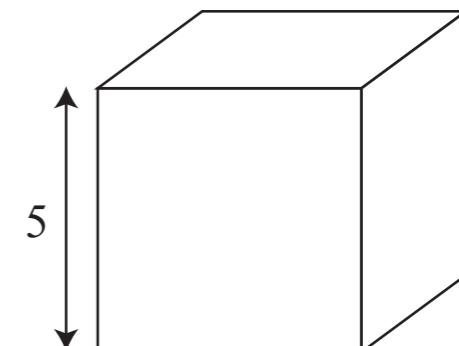
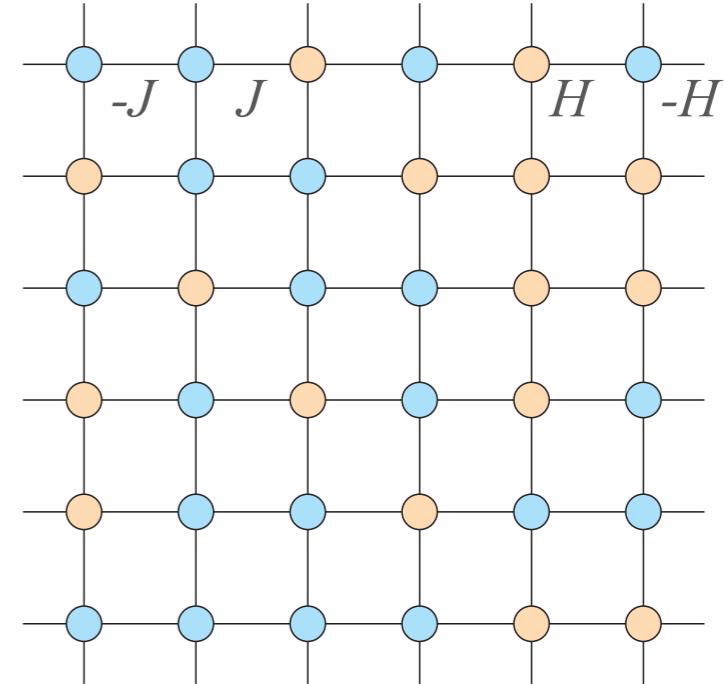
・イジング (Ising) 模型

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j - H \sum_i \sigma_i$$

- 物理量の期待値の計算(例: 磁化)

$$m = \frac{\text{Tr } \sigma_i \exp[-\mathcal{H}/kT]}{\text{Tr} \exp[-\mathcal{H}/kT]}$$

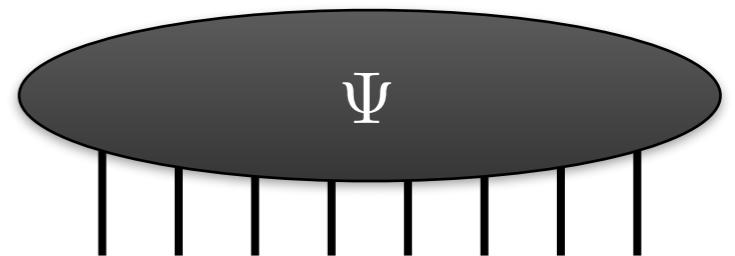
- 単純立方格子(一辺の長さ 5)
 - スピン数 $125 \rightarrow 2^{125} \approx 10^{38}$ の和
- 期待値の計算にかかる時間
 - $10^{38} \div (10^{18}/\text{sec}) \approx 3\text{兆年}$



量子多体系の波動関数

- 量子多体系状態 $\mathcal{H}|\Psi\rangle = E|\Psi\rangle$

$$|\Psi\rangle = \sum_{\{i_1, i_2, \dots, i_N\}} \Psi_{i_1 i_2 \dots i_N} |i_1 i_2 \dots i_N\rangle$$



- N スピン系
 - ハミルトニアン $\rightarrow 2^N \times 2^N$ の行列
 - 波動関数の係数 $\rightarrow 2^N$ の長さのベクトル (or N 本足のテンソル)
- N が大きくなると行列の次元、波動関数のサイズは指数関数的に増加
 - 量子多体系に共通する困難

量子回路

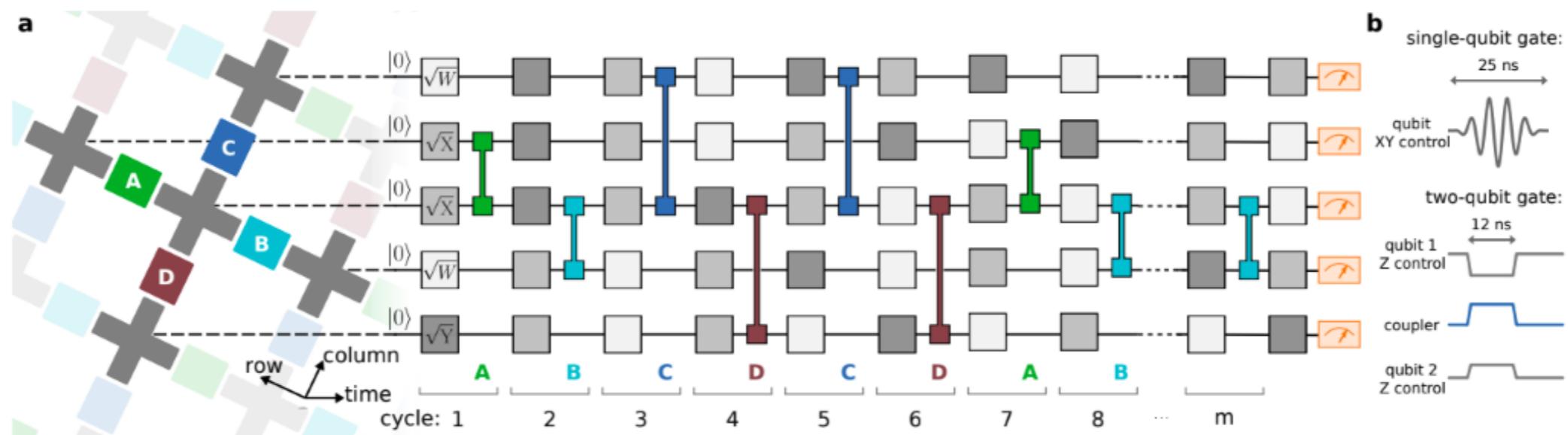
- 量子回路 = 量子ビットに演算するゲート操作の回路図

- 量子ビット(qubit)

- $|0\rangle$ と $|1\rangle$ の重ね合わせの状態をとる

- N -qubitの量子回路 $\rightarrow N$ 個の $S = 1/2$ スピン

- ヒルベルト空間の次元: 2^N



Arute, F., Arya, K., Babbush, R. et al. Nature 574, 505 (2019)

計算科学におけるデータ圧縮

クリロフ(Krylov)部分空間法

- 初期(ランダム)ベクトル v_1 に A を掛けて生成される

$$v_1, Av_1, A^2v_1, \dots, A^{m-1}v_1$$

を正規直交化して v_1, v_2, \dots, v_m を作る (Krylov 部分空間)

$$\mathcal{K}_m(A, v_1) = \text{span}\{v_1, Av_1, A^2v_1, \dots, A^{m-1}v_1\}$$

- 部分空間での Ritz 値を固有値の近似値とする
- $A^k v_1$ はどんどん最大固有ベクトルに近づいていくので、 $m \ll n$ でも良い近似固有値が得られると期待される
→ Lanczos 法 (疎行列・ベクトル積だけで実行可能)

特異値分解 (Singular Value Decomposition)

- 任意の $m \times n$ 實行列 A は

$$A = U\Lambda V^T$$

の形に (一意に) 分解できる ($k = \min(m, n)$)

- U : $(m \times k)$ 行列 (列ベクトルは互いに正規直交)
 V : $(n \times k)$ 行列 (列ベクトルは互いに正規直交)
 $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_k)$ ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k \geq 0$) 特異値
- ベクトル表示 (行列をランク 1 の行列で分解)

$$A = \sum_{i=1}^k \lambda_i u_i v_i^T$$

$k < m, n$ の時: $U^T U = V^T V = E_k$ だが $UU^T \neq E_n$, $VV^T \neq E_n$

特異値分解の例

$$\begin{pmatrix} 1 & 2 & 3 \\ 6 & 4 & 5 \\ 8 & 9 & 7 \\ 10 & 11 & 12 \end{pmatrix} = \begin{pmatrix} -0.14 & -0.62 & -0.05 \\ -0.34 & 0.37 & 0.81 \\ -0.55 & 0.54 & -0.58 \\ -0.75 & -0.44 & 0.06 \end{pmatrix} \times \begin{pmatrix} 25.35 & 0 & 0 \\ 0 & 2.15 & 0 \\ 0 & 0 & 1.71 \end{pmatrix} \begin{pmatrix} -0.56 & -0.59 & -0.59 \\ 0.68 & 0.09 & -0.73 \\ 0.48 & -0.81 & 0.35 \end{pmatrix}$$

行列の低ランク近似

- ランク r ($r < k$) の行列のうち、行列 A を「最も良く近似」する \tilde{A} を選ぶ
- 「最も良く近似」 = フロベニウスノルム $\|A - \tilde{A}\|_F$ を最小化

$$\|X\|_F^2 \equiv \sum_{ij} x_{ij}^2$$

- 特異値のうち大きなものから r 個とり、残りを零とした

$$\tilde{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_r, 0, \dots, 0)$$

を使い

$$\tilde{A} = U \tilde{\Lambda} V^T$$

を作れば良い (Eckart-Young の定理)

行列の低ランク近似の例

$$\begin{pmatrix} -0.14 & -0.62 & -0.05 \\ -0.34 & 0.37 & 0.81 \\ -0.55 & 0.54 & -0.58 \\ -0.75 & -0.44 & 0.06 \end{pmatrix} \begin{pmatrix} 25.35 & 0 & 0 \\ 0 & 2.15 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$
$$\times \begin{pmatrix} -0.56 & -0.59 & -0.59 \\ 0.68 & 0.09 & -0.73 \\ 0.48 & -0.81 & 0.35 \end{pmatrix} = \begin{pmatrix} 1.04 & 1.93 & 3.03 \\ 5.33 & 5.12 & 4.52 \\ 8.47 & 8.21 & 7.34 \\ 9.95 & 11.1 & 12.0 \end{pmatrix}$$

- それなりに悪くない近似が得られる
- 誤差(フロベニウスノルム) = 1.71 (無視した特異値の二乗和の平方根)

Eckart-Young の定理

- A を近似する行列 X (ランク $\leq r$) を考えると

$$\|A - X\|_{\text{F}}^2 = \sum_{ij} (a_{ij} - x_{ij})^2 = \text{tr}[(A - X)(A - X)^T]$$

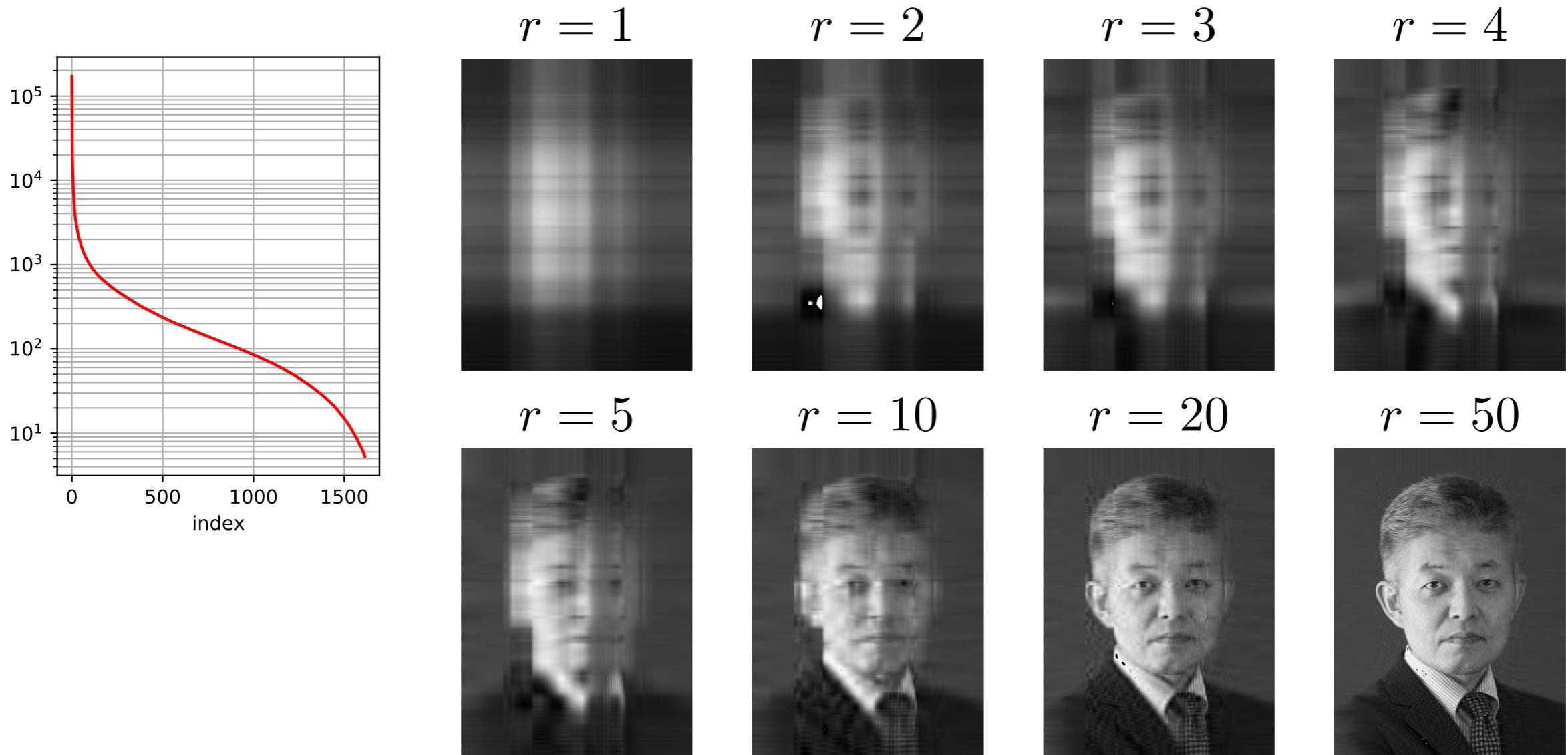
を最小化すればよい。完全 SVD について $UU^T = E_m$, $VV^T = E_n$ より

$$\begin{aligned}\|A - X\|_{\text{F}}^2 &= \text{tr}[UU^T(A - X)VV^T(A - X)^T] \\ &= \text{tr}[(\Lambda - G)(\Lambda - G)^T] \quad (G \equiv U^T X V) \\ &= \sum_i^k (\lambda_i - g_{ii})^2 + \sum_i \sum_{j \neq i} g_{ij}^2\end{aligned}$$

ランク r 以下で、 $\|A - X\|_{\text{F}}^2$ を最小化するには、 $g_{ii} = \lambda_i$ ($i = 1 \cdots r$)、それ以外は全て零とすればよい

特異値分解による画像圧縮

- 特異値の分布とランク r 近似 (1614×2178 グレイスケール写真)



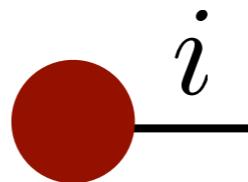
テンソルとテンソルネットワーク

テンソルとテンソルのダイアグラム表記

- ・テンソル: n 次元的な数字の並び

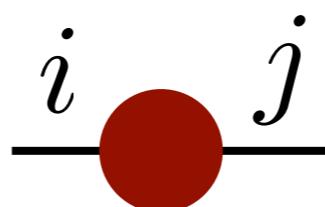
- ・ベクトル

$$\vec{v} : v_i$$



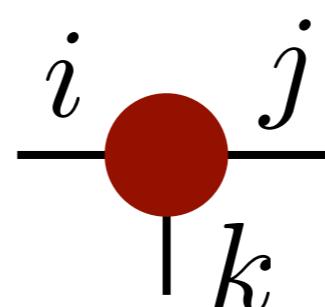
- ・行列

$$M : M_{i,j}$$



- ・3階テンソル

$$T : T_{i,j,k}$$



- ・ n 階テンソル $\rightarrow n$ 本の足

- ・スカラー(数字) \rightarrow 足なし

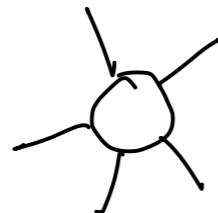


テンソルのデータ量

- 全ての足の次元を χ と仮定すると

- 2本足テンソル = $\chi \times \chi$ 行列
- 5階テンソル

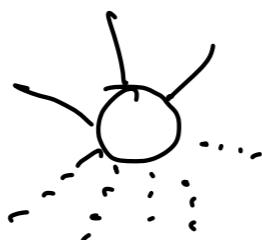
$$O(\chi^2)$$



$$T_{ijkln}$$

$$O(\chi^5)$$

- n 階テンソル



$$O(\chi^n)$$

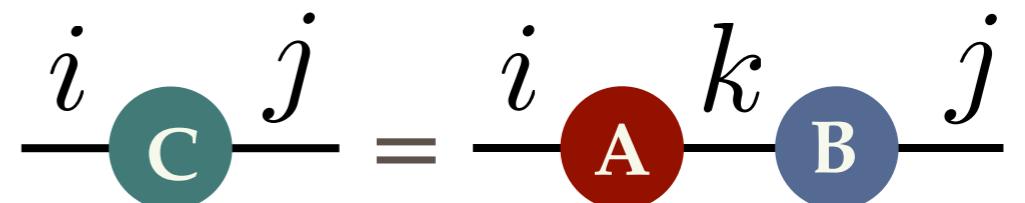
- 足の本数が増えるにつれて、データ量(=要素数)は指数関数的に増える
 - 注: 足の次元(ボンド次元と呼ぶ)は χ あるいは D あらわすことが多い

テンソルの縮約

- ・テンソルの縮約 (contraction)
 - ・共通する足について和を取る
- ・2階テンソル同士の縮約 → 結果は2階テンソル
 - ・行列行列積

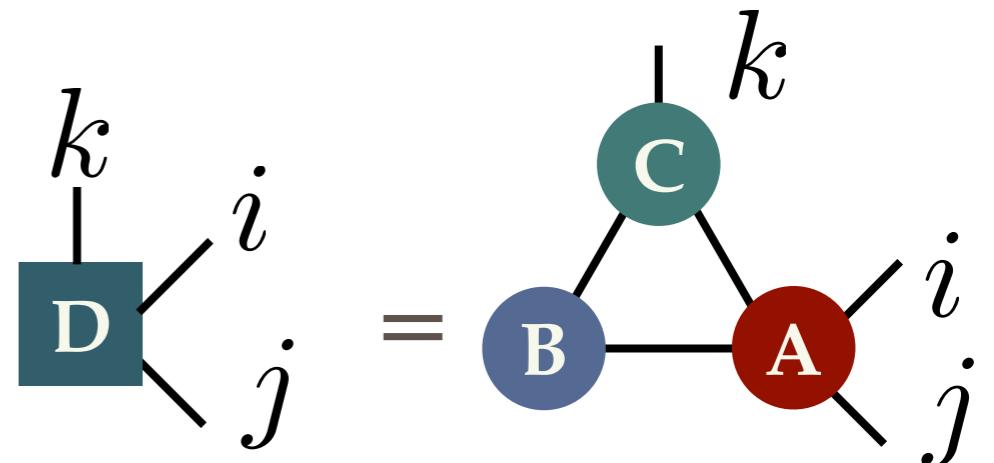
$$C_{i,j} = (AB)_{i,j} = \sum_k A_{i,k} B_{k,j}$$

$$C = AB$$



- ・一般のテンソルの縮約も同様に表現できる

$$D_{i,j,k} = \sum_{\alpha,\beta,\gamma} A_{i,j,\alpha,\beta} B_{\beta,\gamma} C_{\gamma,k,\alpha}$$



テンソル縮約の計算量

- ・行列ベクトル積



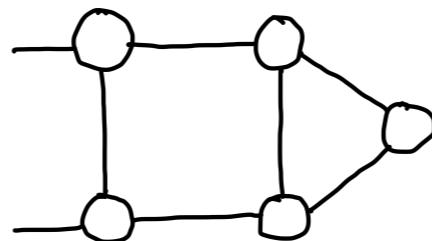
$$w_i = \sum_j T_{ij} v_j \quad O(\chi^2)$$

- ・行列行列積



$$A_{ij} = \sum_k T_{ik} R_{kj} \quad O(\chi^3)$$

- ・テンソルの縮約



$$O(\chi^?)$$

- ・縮約の計算量は縮約の順番に強く依存する

テンソルネットワーク表現

様々なテンソルネットワーク表現

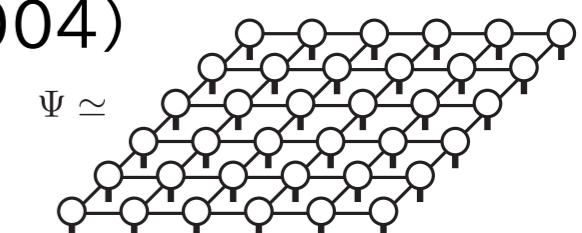
- 量子多体系の量子状態: $|\Psi\rangle = \sum \Psi_{s_1, s_2, \dots, s_N} |s_1, s_2, \dots, s_N\rangle$

• F. Verstraete, J. I. Cirac, arXiv, cond-mat/0407066 (2004)

• G. Vidal, Phys. Rev. Lett. 101, 110501 (2008)

• R. Orús, Ann. Phys. 349, 117 (2014)

TPS(PEPS)



- 統計力学模型の分配関数: $Z = \sum_{\{S_i\}} e^{-\beta \mathcal{H}(\{S_i\})}$

• M. Levin and C. P. Nave, Phys. Rev. Lett. 99, 120601 (2007)

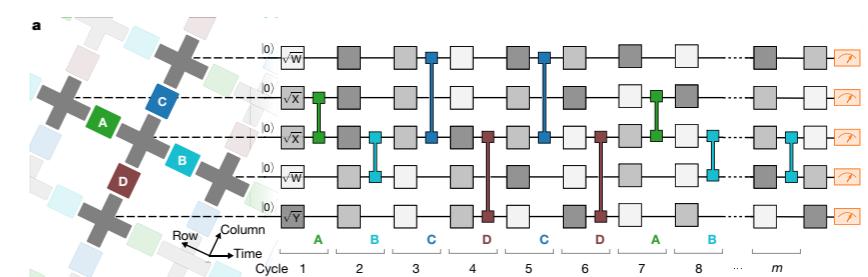
• G. Evenbly and G. Vidal, Phys. Rev. Lett. 115, 180405 (2015)

$$T_{ijkl}[s] = \begin{array}{c} i \\ l \\ \diagdown \\ \text{---} \\ j \\ k \\ \diagup \\ s \end{array}$$

量子回路

• F. Arute, et al., Nature 574, 505 (2019)

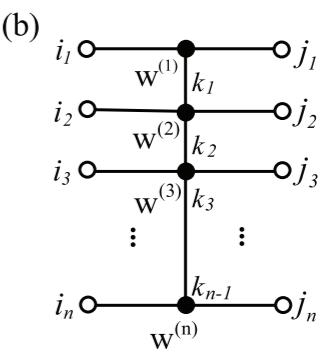
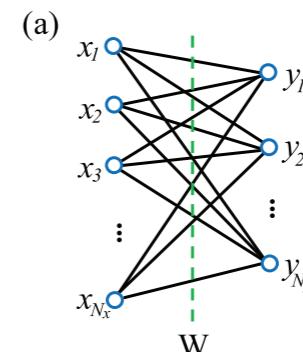
• Y. Lie , et al., SC '21 Proceedings, 3 (2021)



データ科学・機械学習

• E. Stoudenmire and D. J. Schwab, NIPS 29, 4799 (2016)

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



転送行列: 一次元イジング模型

- ハミルトニアン: $H = - \sum_i \sigma_i \sigma_{i+1}$

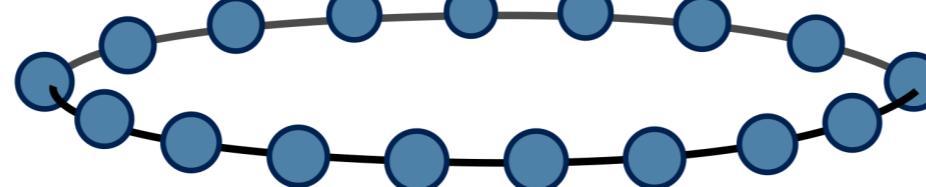
- 分配関数

$$Z = \sum_{\sigma_1} \sum_{\sigma_2} \cdots \sum_{\sigma_L} e^{\beta \sigma_1 \sigma_2} e^{\beta \sigma_2 \sigma_3} \cdots e^{\beta \sigma_L \sigma_1}$$

▶ $e^{\beta \sigma_1 \sigma_2}$ は 4 通りの値を持つ → 2×2 行列 $T_{\sigma_1 \sigma_2}$ の形に書くと

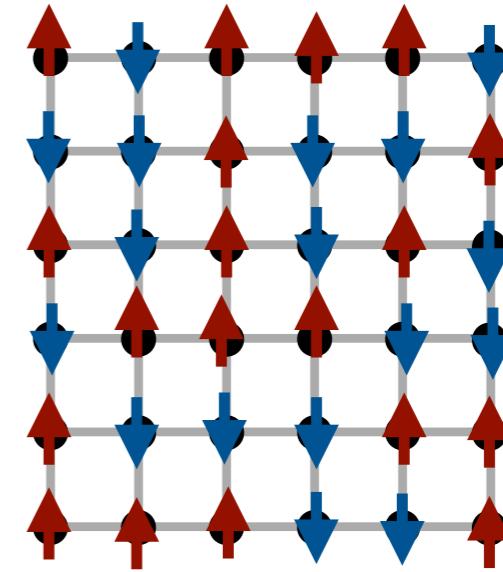
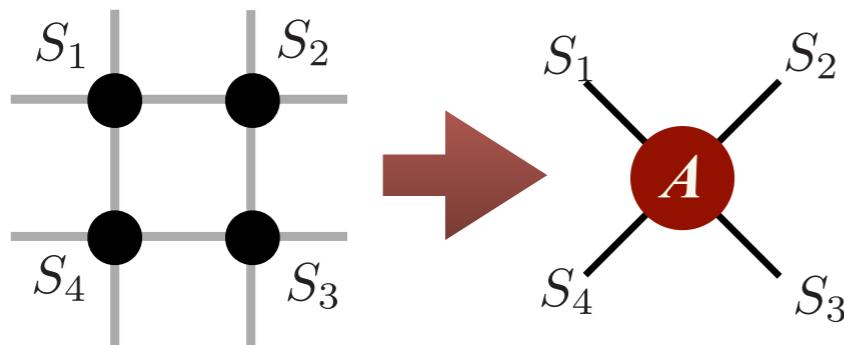
$$\sum_{\sigma_2} e^{\beta \sigma_1 \sigma_2} e^{\beta \sigma_2 \sigma_3} = \sum_{\sigma_2} T_{\sigma_1 \sigma_2} T_{\sigma_2 \sigma_3} = (T^2)_{\sigma_1 \sigma_3}$$

$$Z = \text{tr} T^L$$

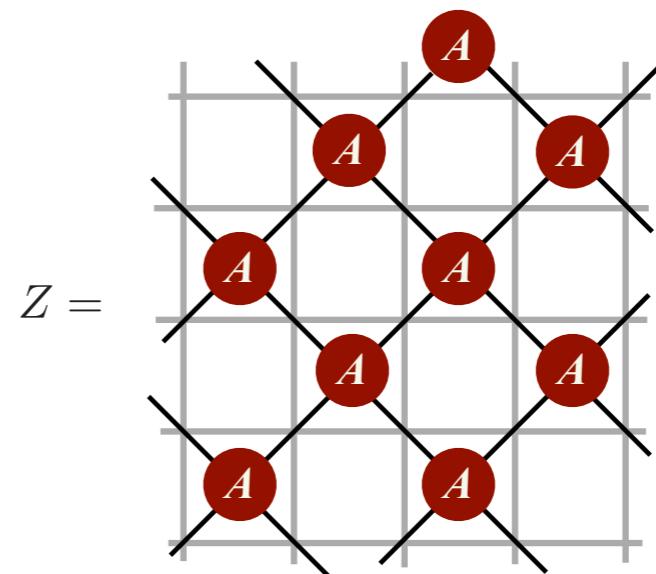


二次元イジング模型

- ・ 二次元正方格子
 - ・ ボルツマン重み $e^{\beta J(\sigma_1\sigma_2 + \sigma_2\sigma_3 + \sigma_3\sigma_4 + \sigma_4\sigma_1)}$



- ・ 分配関数は45度傾いた正方格子テンソルネットワークの縮約として表現できる
 - ・ もともとの正方格子と同じ形のネットワークで表すことも可能
 - ・ 任意の次元・任意の格子で可能

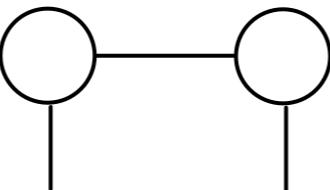


スピンシングレット状態

- ・波動関数

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

- ・テンソルネットワーク表現



$$\begin{array}{c} \textcircled{1} \\ \textcircled{0} \end{array} = \begin{array}{c} \textcircled{2} \\ \textcircled{1} \end{array} = \begin{array}{c} \textcircled{1} \\ \textcircled{4} \end{array} = \frac{1}{2^{1/4}}$$

$$\begin{array}{c} \textcircled{2} \\ \textcircled{0} \end{array} = -\frac{1}{2^{1/4}}$$

GHZ状態

- ・ N スピン系のGHZ (Greenberger–Horne–Zeilinger)状態

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|000\dots 0\rangle + |111\dots 1\rangle)$$

- ・テンソルネットワーク表現 ($\chi = 2$)

$$\begin{array}{c} 1 \\ - \end{array} \otimes \begin{array}{c} 1 \\ 0 \end{array} = \begin{array}{c} 2 \\ - \end{array} \otimes \begin{array}{c} 2 \\ 1 \end{array} = 2^{-1/(2N)}$$

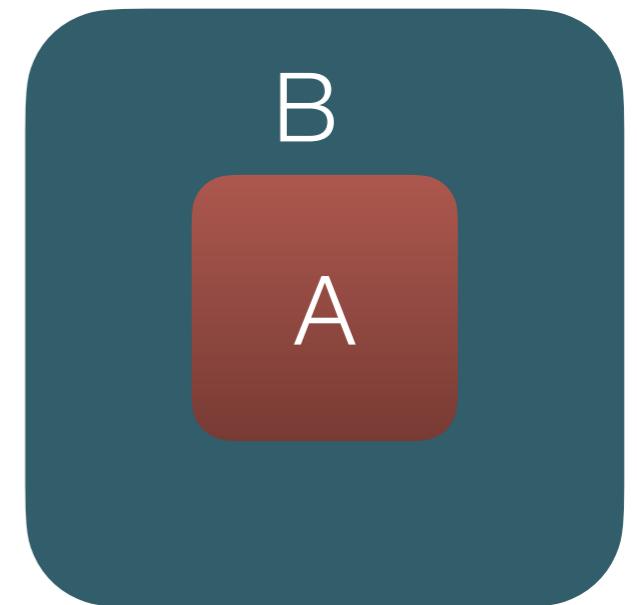
特異値とエンタングルメントエントロピー

- 量子状態のエンタングルメントエントロピー

- 部分系の縮約密度行列

$$\rho_A = \text{Tr}_B |\Psi\rangle\langle\Psi|$$

- エンタングルメントエントロピー
 - 縮約密度行列のvon Neumannエントロピー



$$S = - \text{Tr} \rho_A \log \rho_A$$

- Schmidt分解（量子状態のSVD）

$$|\Psi\rangle = \sum \lambda_i |\alpha_i\rangle |\beta_i\rangle$$

- エンタングルメントエントロピーはSchmidt係数(特異値)を用いて

$$S = - \sum_i \lambda_i^2 \log \lambda_i^2$$

エンタングルメントエントロピーの例

- ・ 2スピン系

- ・ 積状態・古典的な状態

$$|\Psi\rangle = |0\rangle|0\rangle$$

$$\rho_A = |0\rangle\langle 0| \quad (\text{純粹状態})$$

$$S = -\text{Tr}\rho_A \log \rho_A = 0$$

- ・ エンタングルした状態(スピンシングレット状態)

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle|1\rangle - |1\rangle|0\rangle)$$

$$\rho_A = \frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|) \quad (\text{混合状態})$$

$$S = -\text{Tr}\rho_A \log \rho_A = \log 2$$

エンタングルメントエントロピーの面積則

- 一般の量子状態

- ・エンタングルメントエントロピーは部分系の体積に比例(体積則)

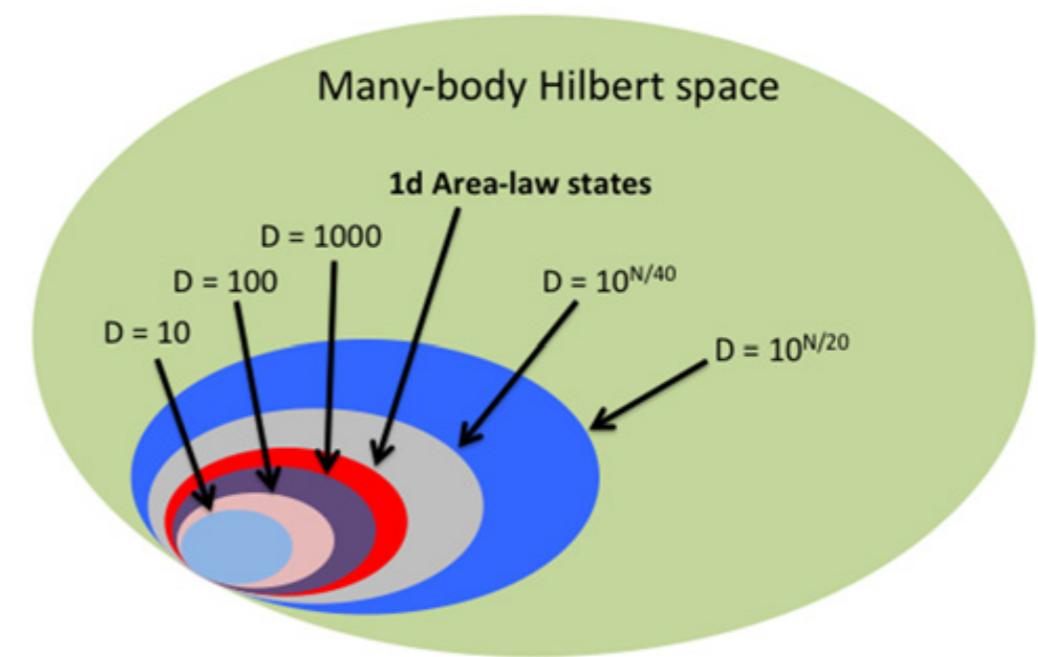
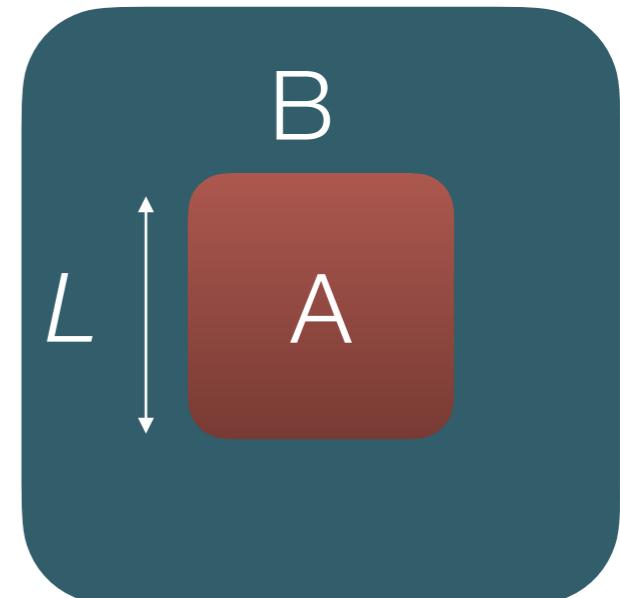
$$S = -\text{Tr}\rho_A \log \rho_A \sim L^d$$

- 基底状態

- ・多くの場合、エンタングルメントエントロピーは体積ではなく部分系の面積に比例(面積則)

$$S \sim L^{d-1}$$

- ・基底状態はヒルベルト空間の狭い部分空間で表現可能



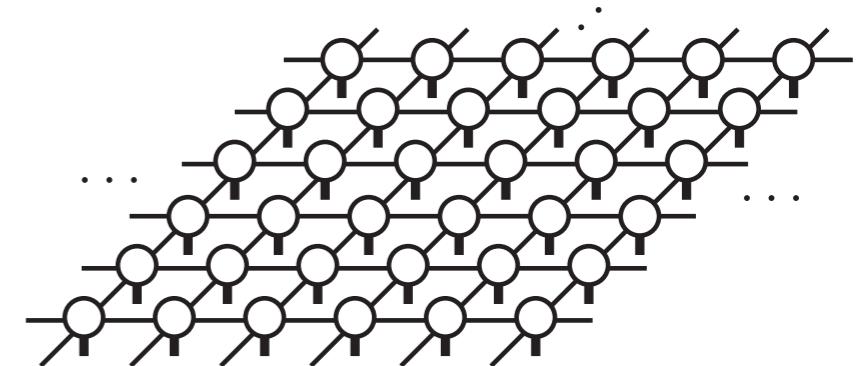
テンソルネットワーク状態のEE

- ・ **上限値**

- ・ 系を半分に分けた時に切れるボンド次元の積(=行列として見たときのランクの最大値)の対数

- ・ **面積則を満たすテンソルネットワーク状態**

- ・ 1次元系: 行列積状態 (MPS)
- ・ 2次元系: PEPS (Projected Entangled Pair State)
 - ・ (4+1)階のテンソルが敷き詰められたネットワーク



- ・ **面積則を満たすテンソルネットワーク変分関数**

- ・ 面積則を満たすため、有限 χ でも精度の良い近似（情報圧縮）
- ・ ボンド次元を増やすことで、系統的に精度を改善できる

$$T_{ijkl}[s] = \begin{array}{c} i \\ | \\ \text{---} \\ | \\ l \end{array} \begin{array}{c} j \\ | \\ \text{---} \\ | \\ k \end{array} \begin{array}{c} s \\ | \\ \text{---} \\ | \\ s \end{array}$$

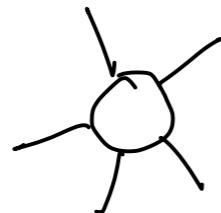
テンソルネットワークの縮約

テンソルのデータ量

- 全ての足の次元を χ と仮定すると

- 2本足テンソル = $\chi \times \chi$ 行列
- 5階テンソル

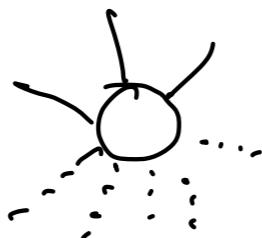
$$O(\chi^2)$$



$$T_{ijkln}$$

$$O(\chi^5)$$

- n 階テンソル



$$O(\chi^n)$$

- 足の本数が増えるにつれて、データ量(=要素数)は指数関数的に増える
 - 注: 足の次元(ボンド次元と呼ぶ)は χ あるいは D あらわすことが多い

テンソル縮約の計算量

- ・行列ベクトル積



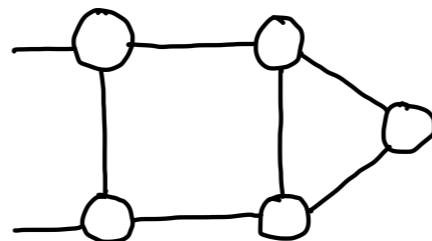
$$w_i = \sum_j T_{ij} v_j \quad O(\chi^2)$$

- ・行列行列積



$$A_{ij} = \sum_k T_{ik} R_{kj} \quad O(\chi^3)$$

- ・テンソルの縮約



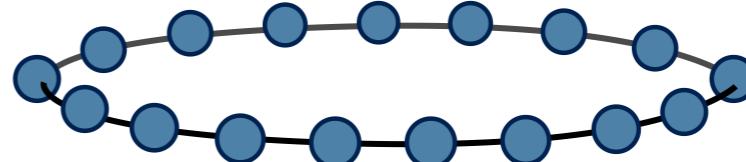
$$O(\chi^?)$$

- ・縮約の計算量は縮約の順番に強く依存する

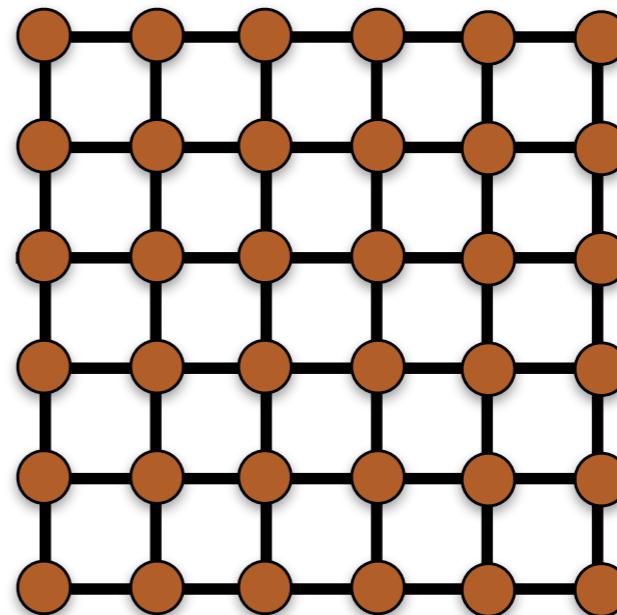
テンソル縮約の計算量

- d -次元の超立方格子 ($N = L^d$) の場合

- 1次元鎖 ($d = 1$)
 - 開境界条件: $O(N\chi^2)$
 - 周期境界条件: $O(N\chi^3)$



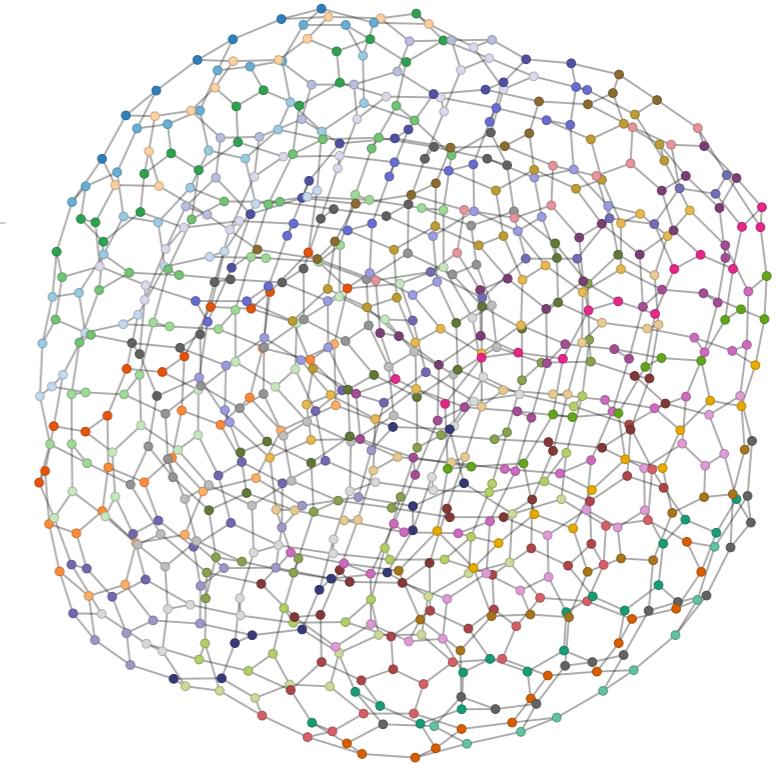
- 2次元正方格子
 - 開境界条件: $O(\chi^L)$
 - 周期境界条件: $O(\chi^{2L})$



- d -次元
 - $O(\chi^{L^{d-1}})$

縮約順序の最適化

- ・一般的なルール
 - ・たくさんの足をもつテンソルが途中で現れないように
- ・最適な縮約順序を見つける問題は(少なくとも)
「#P困難」であることが知られている
 - ・大きなテンソルネットワークの場合、最適解を見つけるのは事実上不可能
- ・さまざまな経験的な方法が提案されている
 - ・R. Schutski, T. Khakhulin, I. Oseledets, D. Kolmakov, Phys. Rev. A, 102, 1 (2020)
 - ・J. Gray, S. Kourtis, Quantum, 5, 1 (2021)



テンソルネットワークの「スライシング」

- ・ **スライシング(Slicing)**

- ・ テンソルネットワークの中のボンドの部分集合について
 - ・ それらのボンド変数をある値に固定する
 - ・ 固定されたボンドはテンソルネットワークから消える
- ・ 固定された値の全ての組み合わせについて
 - ・ 独立に取り足し合わせる

- ・ **スライシングの利点**

- ・ メモリコストも減らすことが可能
- ・ それぞれの組み合わせに関する縮約は(並列コンピュータ上で)並列に計算可能

最先端のテンソルネットワーク縮約手法

- ・2021年ACMゴードンベル賞
 - ・Y. A. Liu et al., Closing the “quantum supremacy” gap: Achieving real-Time simulation of a random quantum circuit using a new sunway supercomputer. International Conference for High Performance Computing, Networking, Storage and Analysis, SC (2021)
- ・しかしながらテンソルネットワーク規模が大きくなると厳密な縮約は結局不可能に
 - ・情報圧縮(SVD)を使った高精度近似
 - ・テンソルネットワーク繰り込み群

The screenshot shows a web browser displaying the ACM awards page. The main content is about the "Real-Time Simulation of Random Quantum Circuit". It highlights a 14-member team from Chinese institutions that received the 2021 ACM Gordon Bell Prize. The team members listed are Yong (Alexander) Liu, Xin (Lucy) Liu, Fang (Nancy) Li, Yuling Yang, Jiawei Song, Pengpeng Zhao, Zhen Wang, Dajia Peng, and Huarong Chen from Zhejiang Lab, Hangzhou; and the National Supercomputing Center in Wuxi; Haohuan Fu and Dexun Chen from Tsinghua University, Beijing; and the National Supercomputing Center in Wuxi; Wenzhao Wu from the National Supercomputing Center in Wuxi; and Heliang Huang and Chu Guo from the Shanghai Research Center for Quantum Sciences. The text explains that quantum supremacy is a term for when a quantum device can solve a problem a classical computer can't in a reasonable time. The Chinese team's work involved simulating a 10x10x(1+40+1) random quantum circuit using a Sunway supercomputer, achieving a performance of 1.2 Eflops for classical simulation. Their approach took only 304 seconds. The page also mentions the 2021 ACM Gordon Bell Special Prize for COVID-19 research on digital transformation and droplet/aerosol infection risk assessment using Fugaku supercomputer.

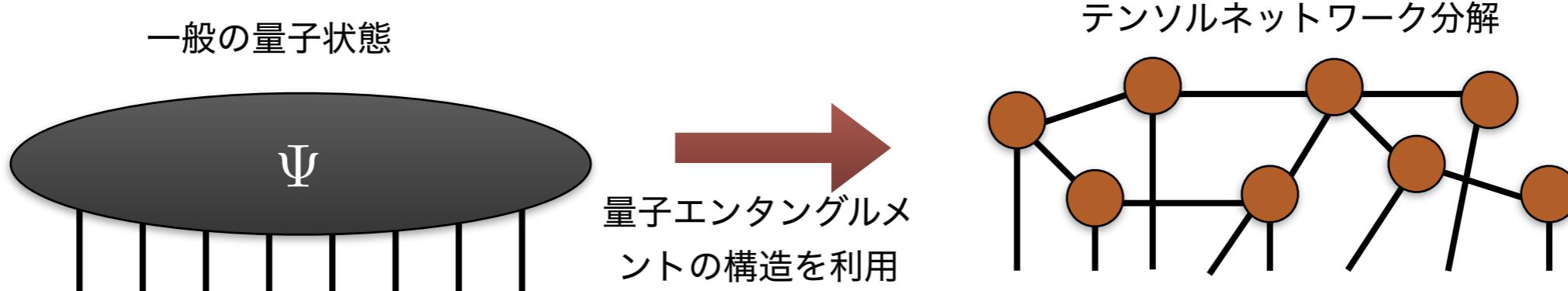
<https://awards.acm.org/bell>

テンソルネットワークによる情報圧縮

テンソルネットワーク分解

- ・指数関数的に大きな状態空間を古典計算機で全て扱うことは不可能
 - ・実効的な次元を減らしたい

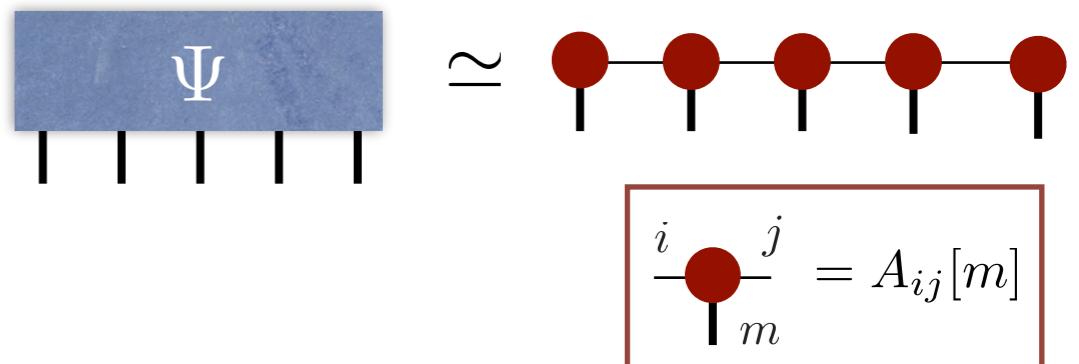
- ・テンソルネットワーク分解
 - ・情報のエンタングルメントに注目することで、適切な部分空間を構成



行列積状態 (MPS)

- N 本足のテンソル(ベクトル)を行列の積で表現

$$\Psi_{i_1 i_2 \dots i_N} \simeq A_1[i_1] A_2[i_2] \cdots A_N[i_N]$$



- MPS は応用数理では”tensor train decomposition”とも呼ばれている
- 直積状態は 1×1 “行列” (スカラー) のMPSで表現できる

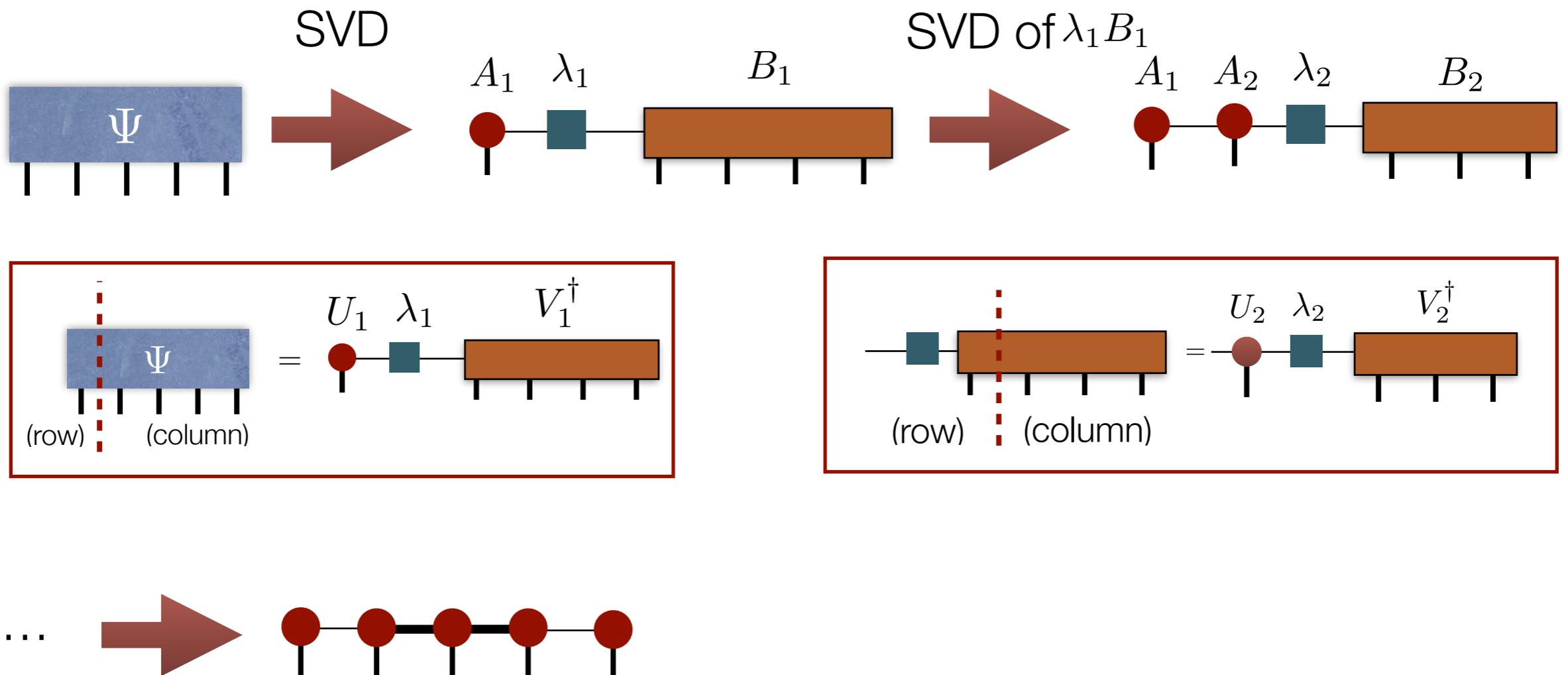
$$|\Psi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots$$

$$\Psi_{i_1 i_2 \dots i_N} = \phi_1[i_1] \phi_2[i_2] \cdots \phi_N[i_N]$$

$$\phi_n[i] \equiv \langle i | \phi_i \rangle$$

行列積状態への変換

- 任意のテンソル(ベクトル)は特異値分解を繰り返すことで厳密なMPS表現に常に変換できる



- ボンド次元は場所に依存 \rightarrow 最大のボンド次元 $= O(a^{N/2})$

行列積状態における低ランク近似

$$\Psi = \begin{array}{c} \text{---} \\ | \\ | \\ | \\ | \\ a^N \end{array} = \begin{array}{c} \bullet-\bullet-\bullet-\bullet-\bullet \\ | | | | \\ \text{---} \end{array} \quad \boxed{\text{最大ボンド次元} = a^{N/2}}$$
$$\simeq \begin{array}{c} \bullet-\bullet-\bullet-\bullet-\bullet \\ | | | | \\ \text{---} \end{array} \quad \boxed{\text{最大ボンド次元} = \chi} \\ \propto N a \chi^2$$

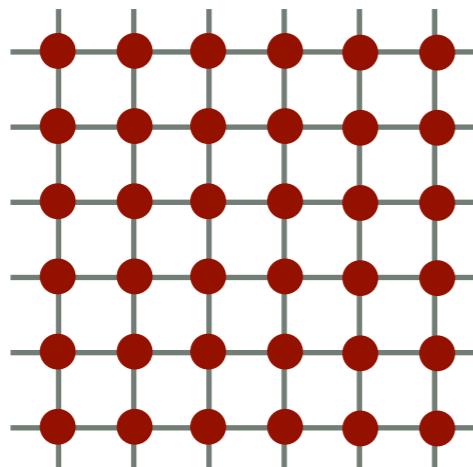
- ・もとのテンソルがボンド次元 χ の行列積状態で精度良く近似できれば、 N の指
数関数のデータ量を N の多項式にまで大幅に減らせる！
 - ・もし、“エンタングルメントエントロピー”が N に依存しない ($S \sim O(1)$) ならば、ボンド次
元 χ が N に依存しないようにできる
 - ・逆に、EEが N と共に増大する場合、(同じ近似精度を保つには)、 χ も N と共に(指
数
関数的に?)増やす必要がある

テンソルネットワークくりこみ群

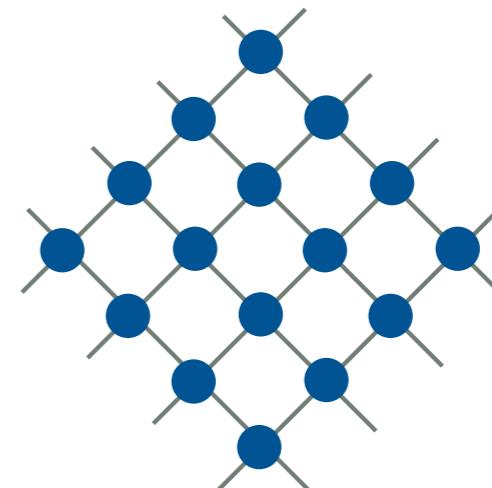
テンソルネットワークくりこみ群

- ・テンソルネットワークを近似しながら少しづつ縮約
 - ・ネットワークの「粗視化」 \Leftrightarrow 「実空間くりこみ群」
- ・テンソルのボンド次元やつながり形を変えずに、ネットワークのサイズを小さく

$L \times L$ 個のテンソル



$(L \times L)/2$ 個のテンソル

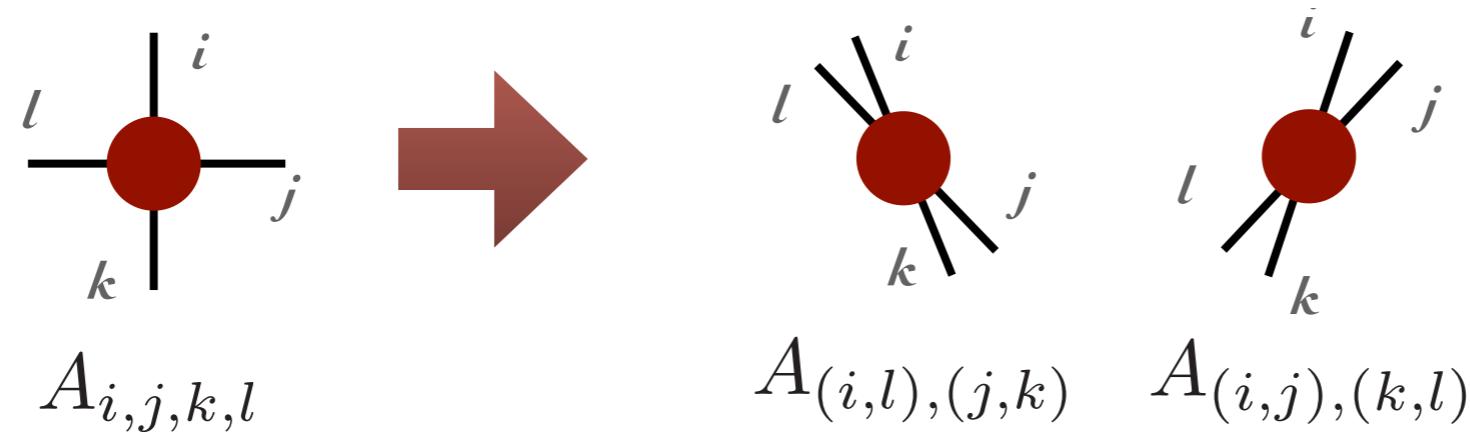


2個のテンソルを1個に

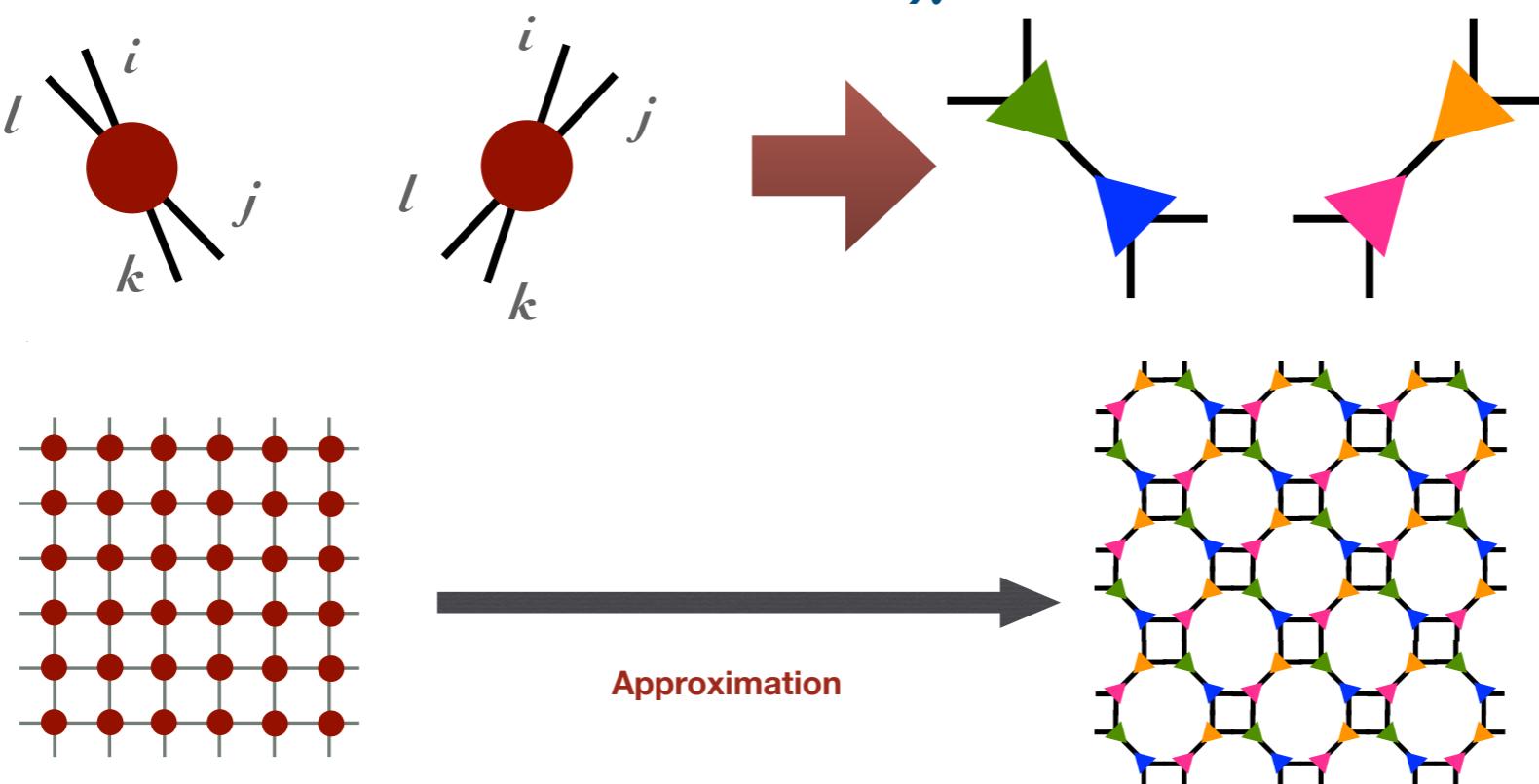
Levin and Nave, Phys. Rev. Lett. 99, 120601 (2007)

テンソルネットワークくりこみ群の手順

- ・ステップ1: 4階テンソルを2階テンソル(行列)に変形

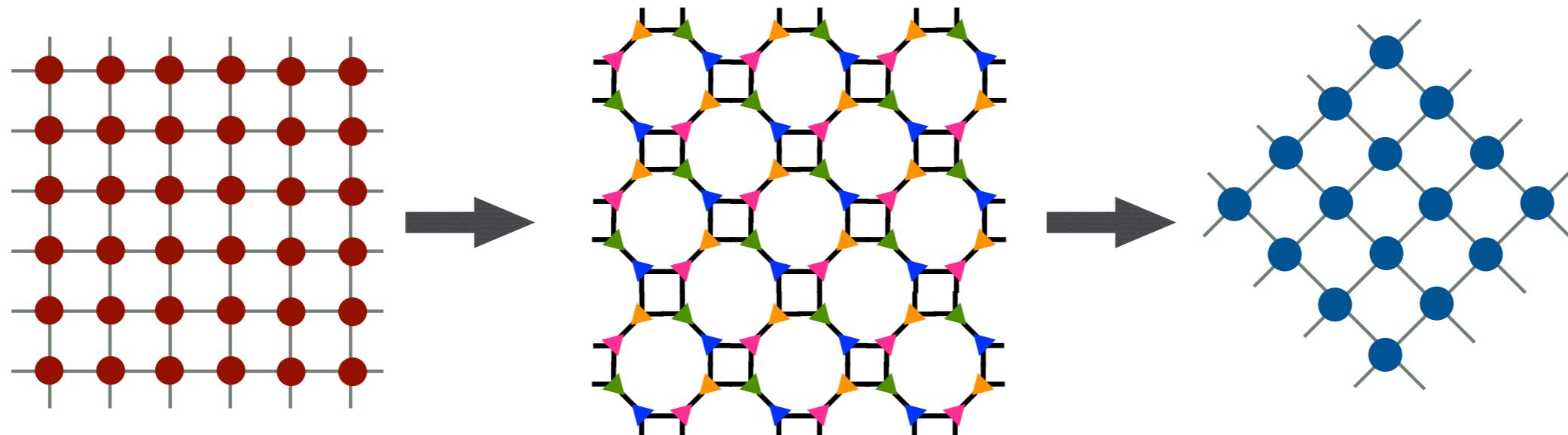


- ・ステップ2: 特異値分解を使いランク- χ 近似



テンソルネットワークくりこみ群の手順

- ・ステップ3: 4つのテンソルを縮約し4階テンソルに

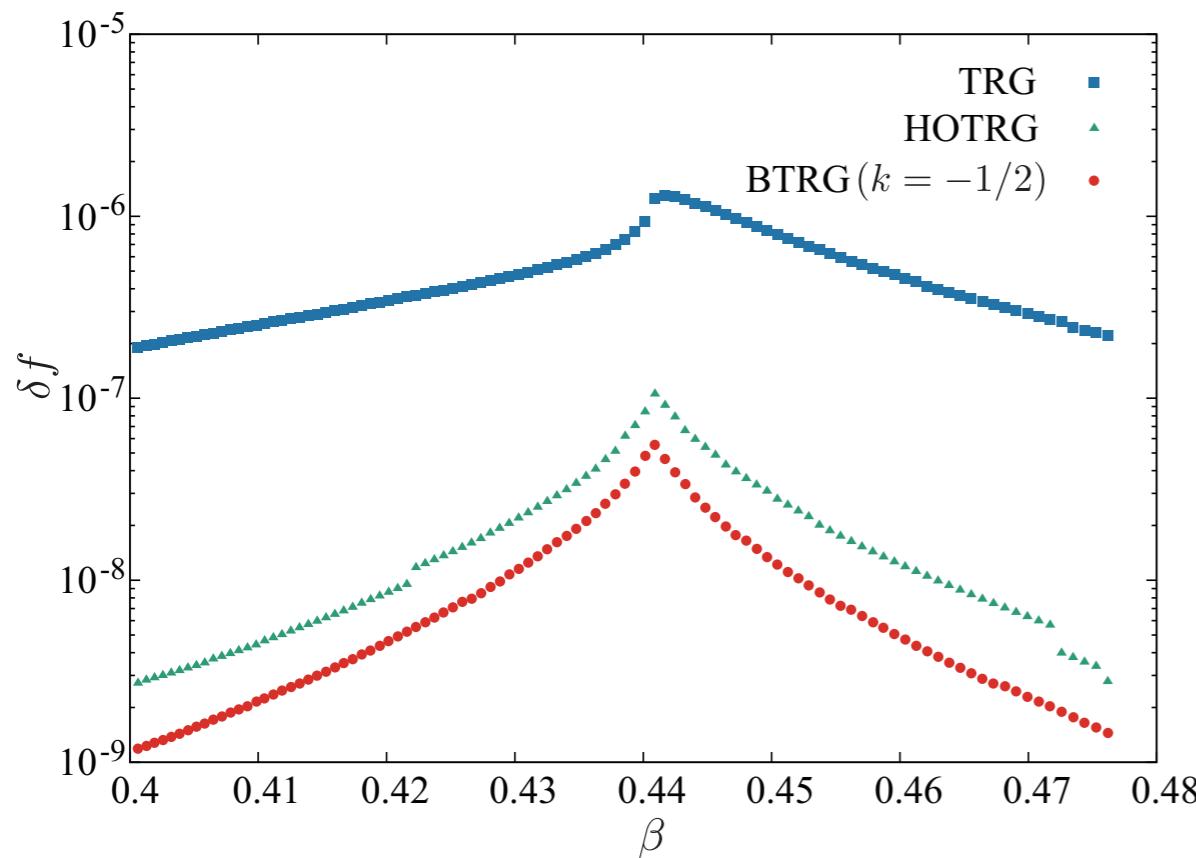


- ・1回のくりこみとテンソルの数は半分に
 - ・テンソル1個になるまで繰り込み→トレース→分配関数
- ・1回のくりこみの計算コスト
 - ・SVD: $O(\chi^6)$
 - ・縮約: $O(\chi^6)$

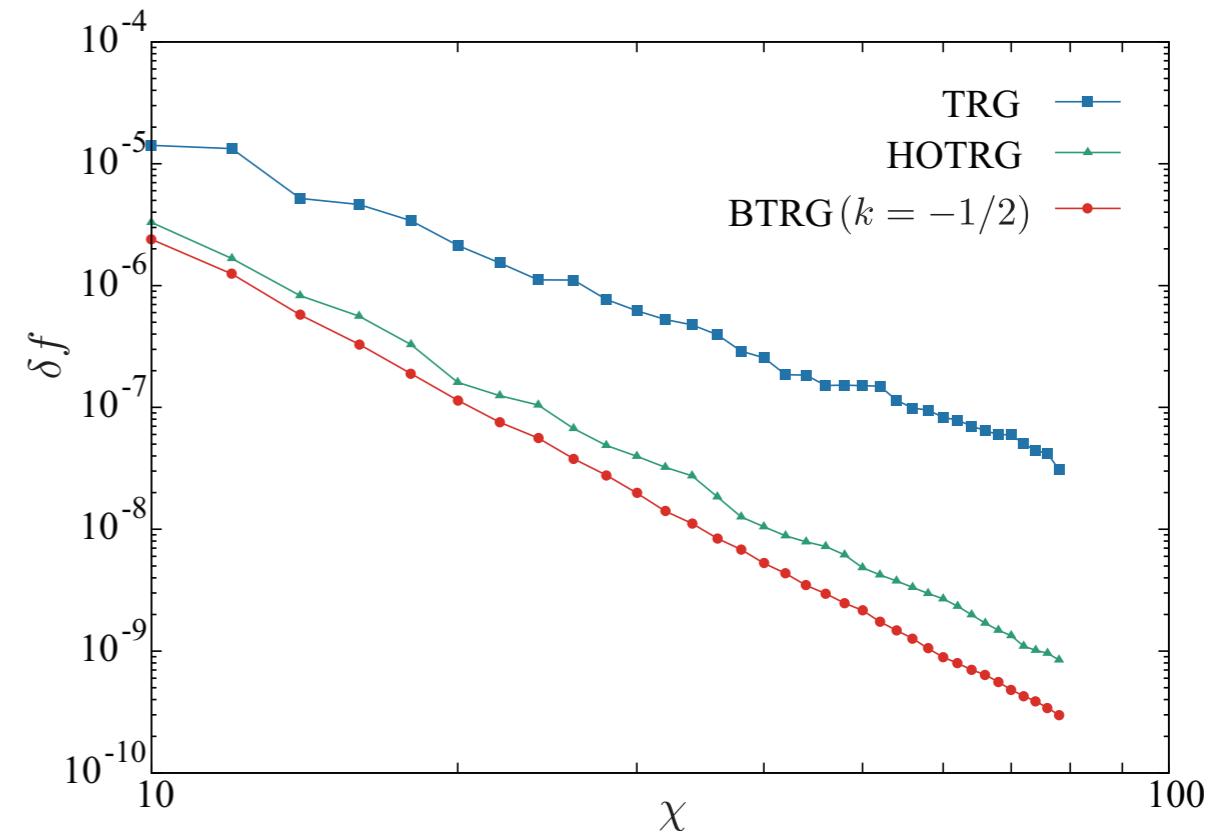
テンソルネットワークくりこみ群の精度

- 二次元イジング模型の自由エネルギー $f = -k_B T \log Z$ の誤差

$$N = 2^{16} \times 2^{16}, \chi = 24$$



$$N = 2^{16} \times 2^{16}, T = T_c$$



量子系の時間発展

横磁場イジング模型

- ハミルトニアン ($2^N \times 2^N$ 行列)

$$H = H_z + H_x = -J \sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z - h \sum_i \sigma_i^z - \Gamma \sum_i \sigma_i^x$$

- σ_i^x 、 σ_i^z : パウリ行列 (2×2 行列)

$$\begin{aligned} (\sigma_i^z)^2 &= (\sigma_i^x)^2 = I \\ [\sigma_i^z, \sigma_i^x] &\neq 0 \end{aligned}$$

- J : スピン間の相互作用 ($J > 0$: 強磁性、 $J < 0$: 反強磁性)
- h : 縦磁場 (準位間のエネルギー差 = $2h$)
- Γ : 横磁場 (トンネリング)
- 以降、 σ_i^z を対角化する基底 ($|\uparrow\rangle_i$, $|\downarrow\rangle_i$) で考える

横磁場イジング模型

■ 2 サイト系

$$H = -J\sigma_1^z\sigma_2^z - h(\sigma_1^z + \sigma_2^z) - \Gamma(\sigma_1^x + \sigma_2^x)$$

■ 行列要素

$$\langle \uparrow\uparrow | H | \uparrow\uparrow \rangle = -J - 2h$$

$$\langle \uparrow\uparrow | H | \uparrow\downarrow \rangle = -\Gamma$$

$$\langle \uparrow\uparrow | H | \downarrow\downarrow \rangle = 0$$

⋮

横磁場イジング模型の時間発展

- 時間依存シュレディンガー方程式の形式解

$$\Psi(t) = e^{-iHt}\Psi(0)$$

- ▶ 有限差分法、クランク・ニコルソン法
- 鈴木・トロッターディスクリプション ($\Delta t = t/M$)

$$\begin{aligned} e^{-iHt} &= [e^{-iH\Delta t}]^M \approx [e^{-iH_z\Delta t} e^{-iH_x\Delta t}]^M \\ &= [e^{i\Delta t J \sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z} e^{i\Gamma \sigma_1^x \Delta t} e^{i\Gamma \sigma_2^x \Delta t} \dots e^{i\Gamma \sigma_N^x \Delta t}]^M \end{aligned}$$

(さらに $e^{-iH\Delta t} \approx e^{-iH_z\Delta t/2} e^{-iH_x\Delta t} e^{-iH_z\Delta t/2}$ と対称に分解すると近似の次数が上がる)

- $[\sigma_i^x]^2 = I$ より

$$e^{i\Gamma \sigma_i^x \Delta t} = \cos(\Gamma \Delta t) + i\sigma_i^x \sin(\Gamma \Delta t)$$

初期状態と時間発展演算子

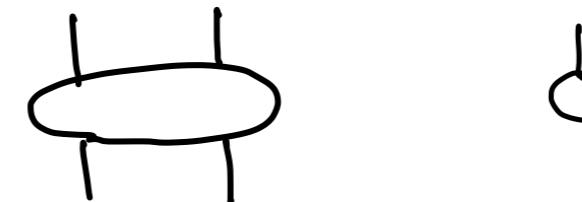
- 初期状態の例: スピンが全て上向きの状態
 - $\chi = 1$ の行列積状態 (MPS) で表すことができる

$$|\Psi\rangle = \text{---} \quad \text{---} \quad \text{---} \quad \text{---} \quad \text{---} \quad \text{---} \quad \text{---}$$

$$\begin{array}{ccc} \text{---} & = & 0 \\ |0\rangle & & |1\rangle \\ \text{---} & = & 1 \end{array}$$

時間発展演算子

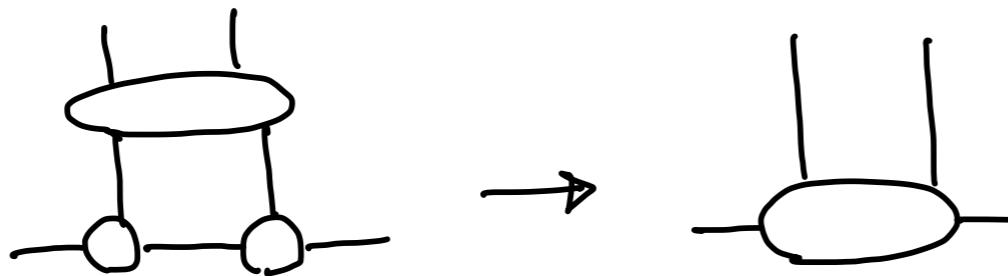
- 相互作用項 \rightarrow 4階テンソル
- 横磁場項 \rightarrow 2階テンソル



$$e^{iJ\sigma_i^z\sigma_j^z\Delta t} \quad e^{i\Gamma\sigma_i^x\Delta t}$$

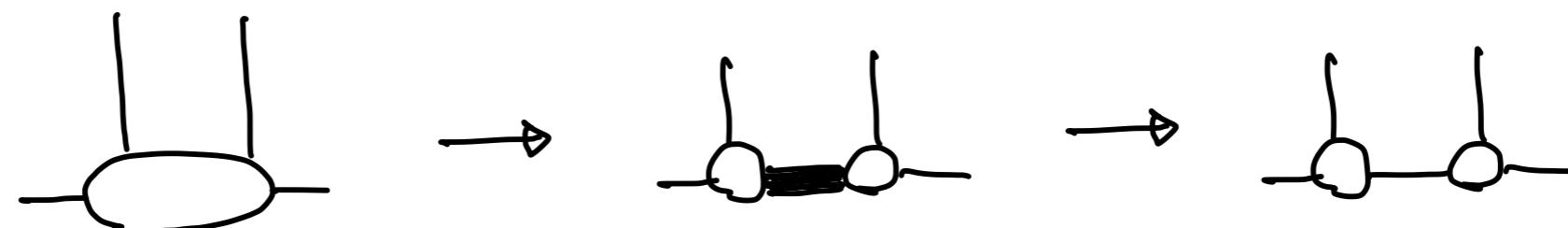
時間発展

- MPSに時間発展演算子を順番に掛けていく(縮約)
 - 相互作用項をかけると形が変わる



- テンソルを分解してもとのMPSの形にもどす

- 分解するだけではボンド次元が増加してしまうので、間をつなぐボンド次元が χ を超えないように低ランク近似する



より高度なテンソルネットワーク法

- ・より計算時間をかけて精度を向上

- ・SRG [Xie et al, Phys. Rev. Lett. 103, 160601 (2009)], TNR [Evenbly, Vidal, Phys. Rev. Lett. 115, 180405 (2015)], Loop TNR [Yang et al, Phys. Rev. Lett. 118, 110504 (2017)], Gilt [Hauru, Delcamp, Mizera, Phys. Rev. B 97, 045111 (2018)]

- ・計算時間を変えずに精度を向上

- ・BTRG [Adachi, Okubo, Todo, Phys. Rev. B 105, L060402 (2022)]

- ・高次元への拡張・コストの削減

- ・HOTRG [Xie et al., Phys. Rev. B 86, 045139 (2012)], ATRG [Adachi, Okubo, Todo, Phys. Rev. B 102 054432 (2020)]

- ・任意形状のテンソルネットワークのくりこみ

- ・CATN [Pan et al., Phys. Rev. Lett. 125, 060503 (2020)]

- ・時間発展

- ・TEBD [Vidal, Phys. Rev. Lett. 93, 040502 (2004)]

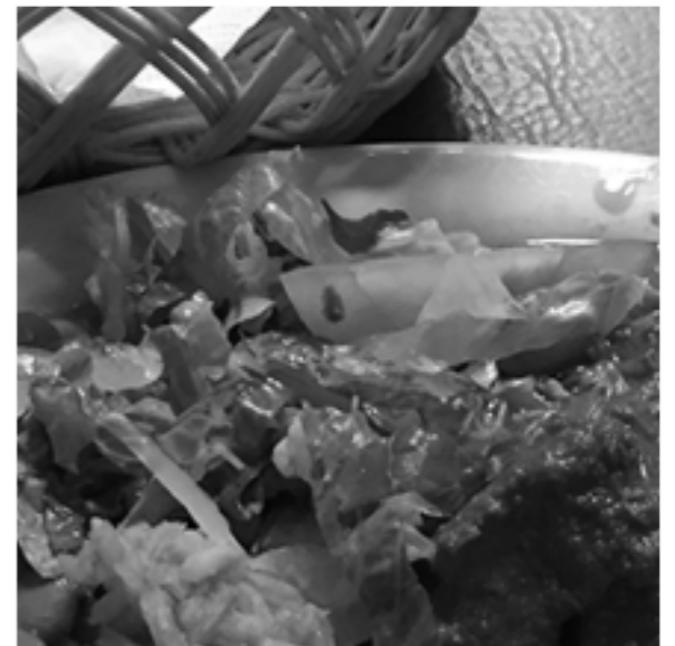
画像データの圧縮

画像データの圧縮

- 256×256 ピクセルの画像
 - 2種類のデータ構造
 - 256×256 の行列(画像見たまま)
 - 単純な特異値分解を適用可能

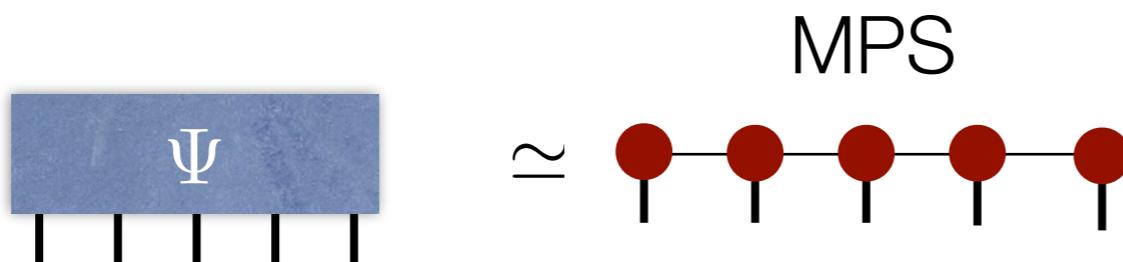
$$A \approx U \Sigma V^\dagger$$

$$256 = 2^8$$



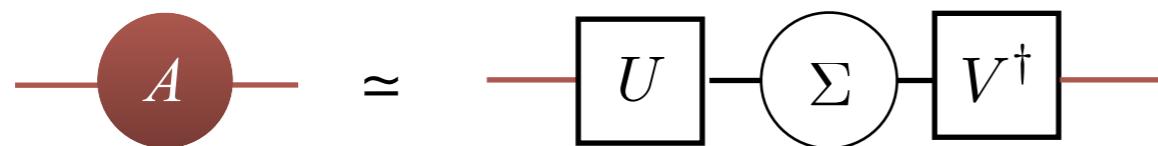
$$256 = 2^8$$

- $2 \times 2 \times \dots \times 2$ の 16 本足のテンソル
 - MPSによる近似を適用できる形
 - どのように変形するか、足を並べるのかの自由度がある

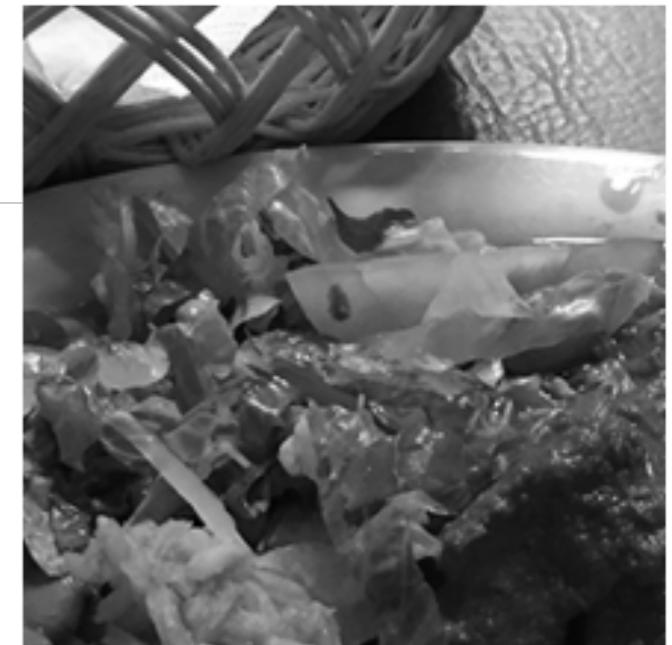


特異値分解による画像の圧縮

- 256×256 の行列(画像見たまま)



$256=2^8$



$$\chi = 10$$

$$\begin{aligned}\Delta &= 0.2025 \\ D &= 5130\end{aligned}$$



$$\chi = 20$$

$$\begin{aligned}\Delta &= 0.1529 \\ D &= 10260\end{aligned}$$

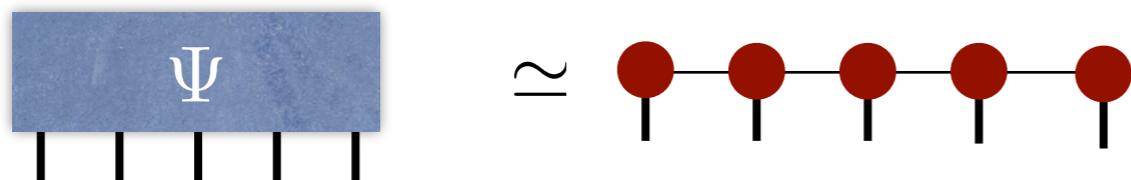


$$\chi = 50$$

$$\begin{aligned}\Delta &= 0.0885 \\ D &= 25650\end{aligned}$$

MPSによる画像の圧縮

- ・ $2 \times 2 \times \dots \times 2$ の16本足のテンソル



- ・近似精度: $\Delta \equiv \|A - \tilde{A}\|/\|A\|$
 - ・データ量: $D \simeq N\chi_{max}^2$



$$\chi_{max} = 10$$

$$\Delta = 0.2405$$

$$D = 2088$$

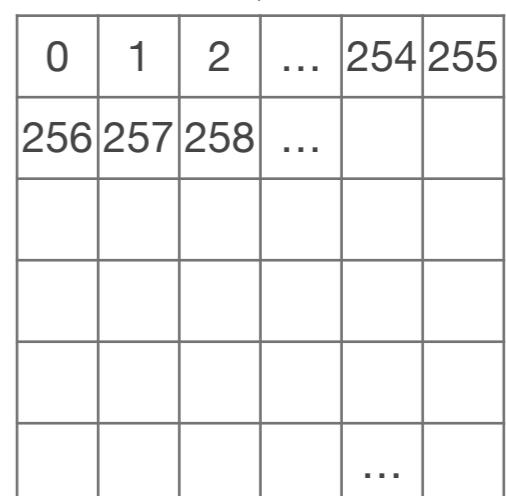
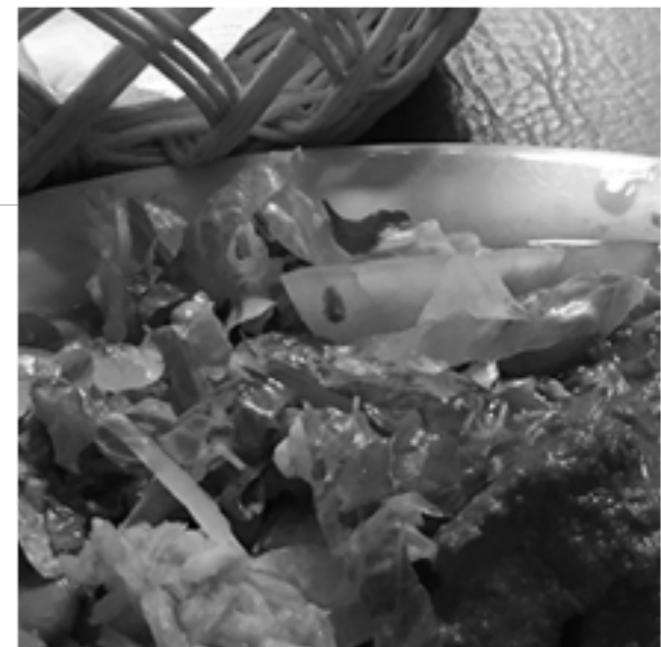


$$\chi_{max} = 20$$

$$\Delta = 0.1873$$

$$D = 6760$$

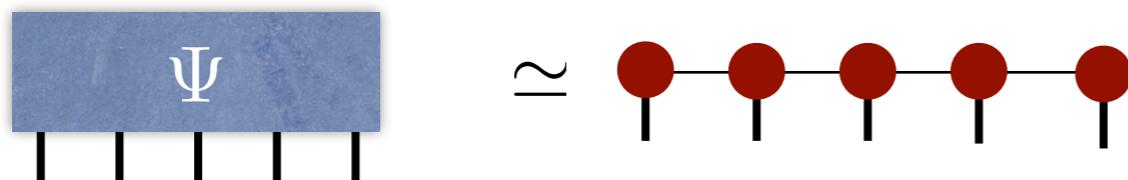
256 = 28



数字を二進数表記して 16本足のテンソル化

画像の圧縮: テンソル化依存性

- $2 \times 2 \times \dots \times 2$ の 16 本足のテンソル



- 近似精度: $\Delta \equiv \|A - \tilde{A}\|/\|A\|$
- データ量: $\chi_{max} = 50$ $D = 29128$



連續

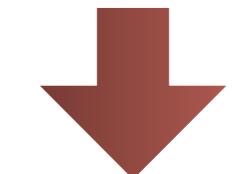
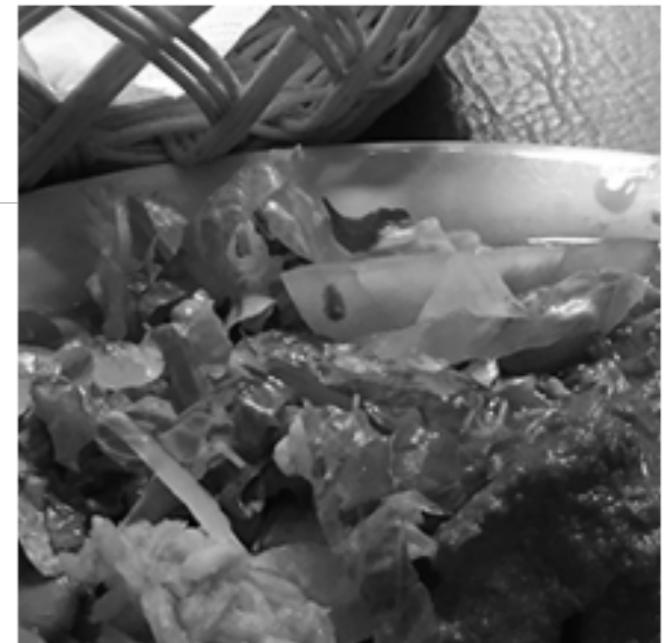
$$\Delta = 0.1122$$



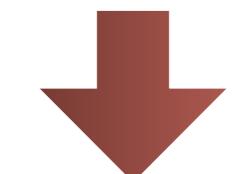
ランダム

$$\Delta = 0.4227$$

$256=2^8$



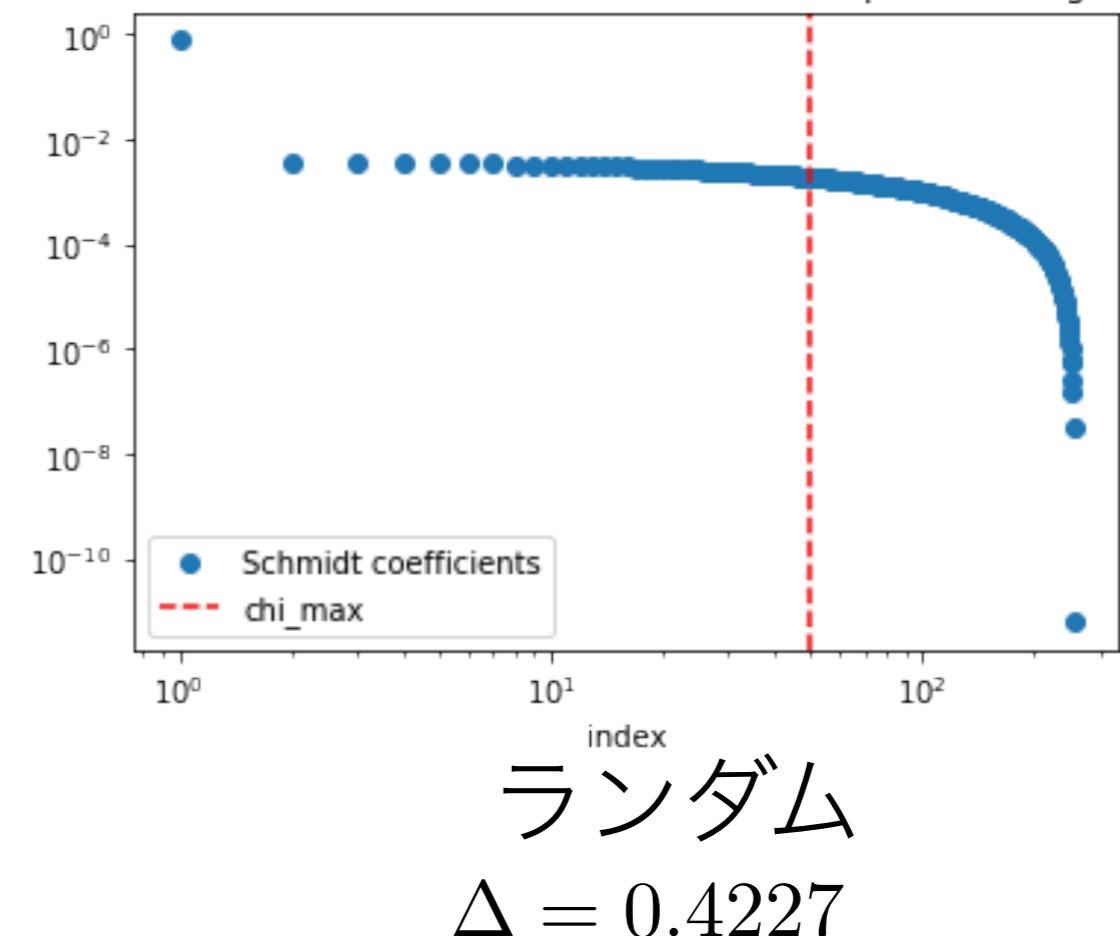
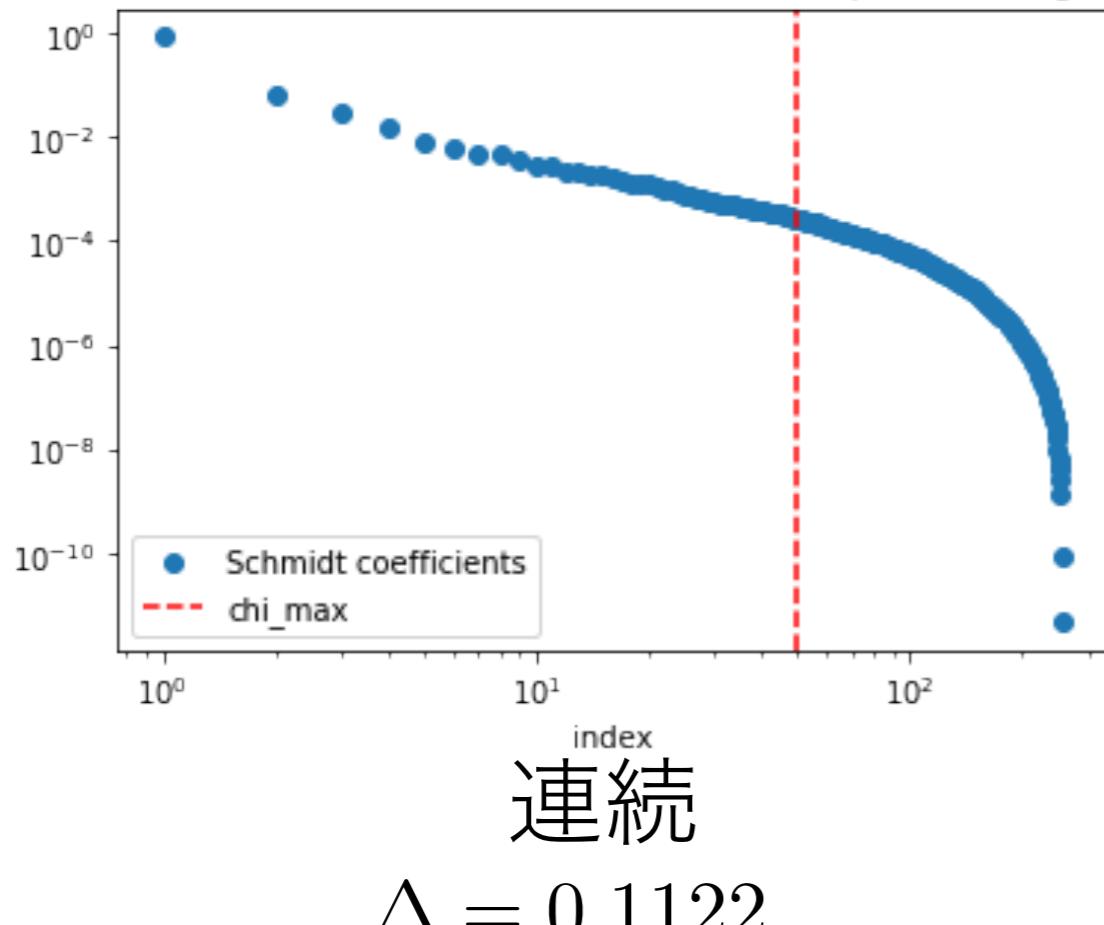
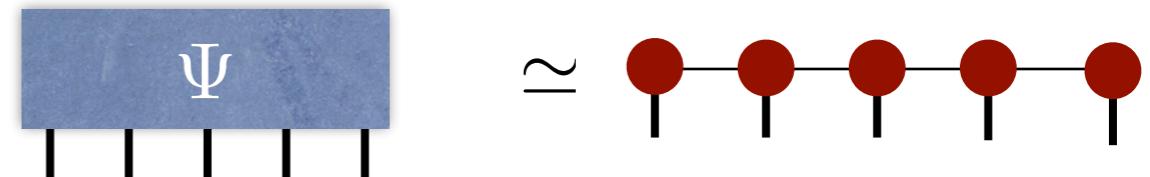
1			
2			
3			
0			



数字を二進数表記して
16本足のテンソル化

低ランク近似と特異値スペクトル

- $2 \times 2 \times \dots \times 2$ の 16 本足のテンソル
 - 中央で分割した場合の特異値



- テンソル化の方法によって“エンタングルメント”が異なる

テンソルネットワークによる機械学習

教師なしの生成モデル

- ・ N ピクセルの白黒画像



$$\text{バイナリデータ } \vec{v} \in \mathbb{V} = \{0, 1\}^{\otimes N}$$

$$\vec{v} \sim (0, 1, 0, 1, \dots, 0, 0)$$

- ・ 画像の確率分布: $P(\vec{v})$ 量子ビットの古典状態と同じ構造

・ ボルンマシン表現

- ・ 確率分布を量子状態と対応づけて考える

$$|\Psi\rangle = \sum_i \Psi(\vec{v}_i) |\vec{v}_i\rangle \quad P(\vec{v}) = \frac{|\Psi(\vec{v})|^2}{Z}$$

- ・ 量子状態 $|\Psi\rangle$ を測定すると確率 $P(\vec{v})$ で状態 $|\vec{v}\rangle$ が観測される

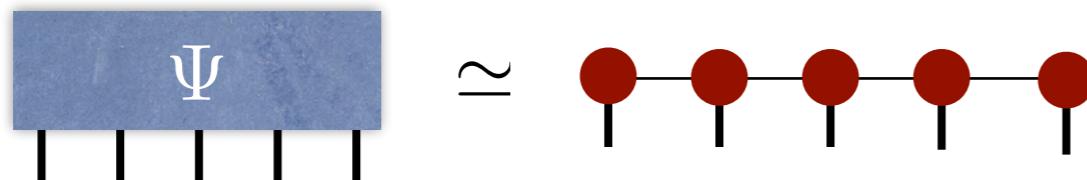
・

教師なしの生成モデル

- ・ボルンマシン

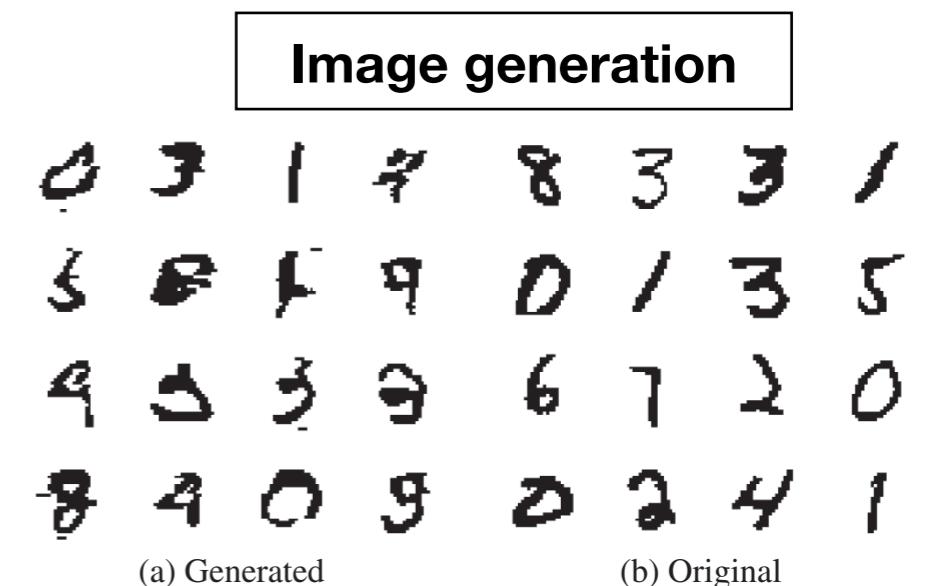
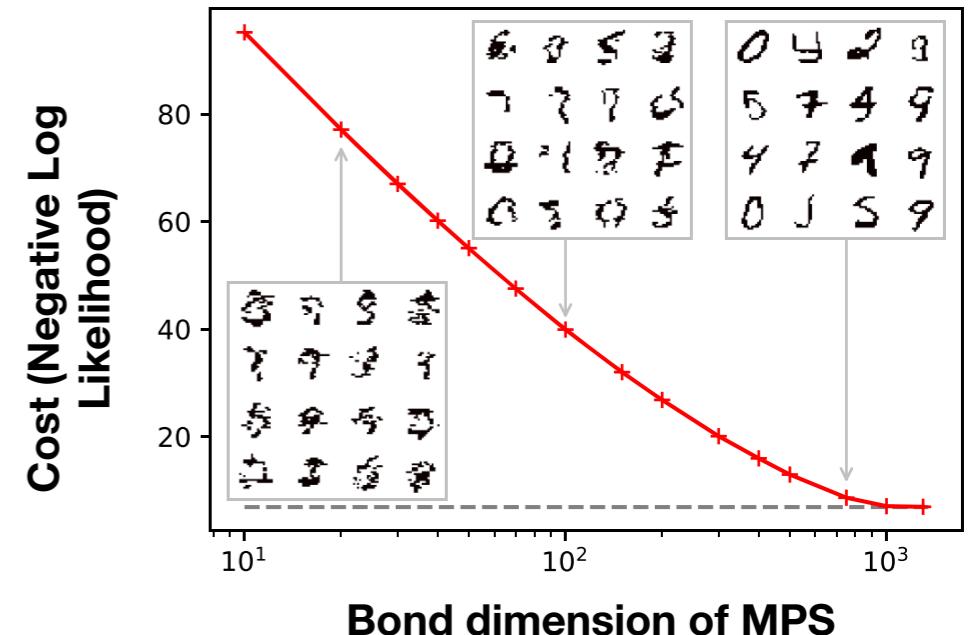
$$P(\vec{v}) = \frac{|\Psi(\vec{v})|^2}{Z} \quad (Z = \sum_{\vec{v}} |\Psi(\vec{v}_i)|^2)$$

- ・対応する量子状態をテンソルネットワークで表現する



- ・負の対数尤度を最小化することでMPSを最適化する

$$F = -\frac{1}{T} \sum_{\vec{v} \in T} \ln P(\vec{v})$$



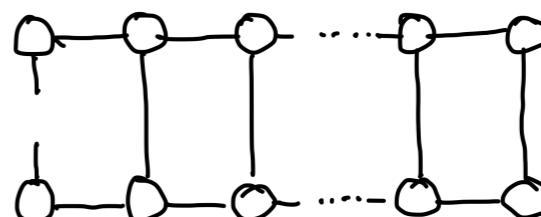
MPS波動関数からのサンプリング

- 2^N の状態空間からのサンプリングは困難？

$$P(s_1, s_2, \dots, s_n) = |\langle s_1 s_2 \dots s_n | \Psi \rangle|^2 / \langle \Psi | \Psi \rangle$$

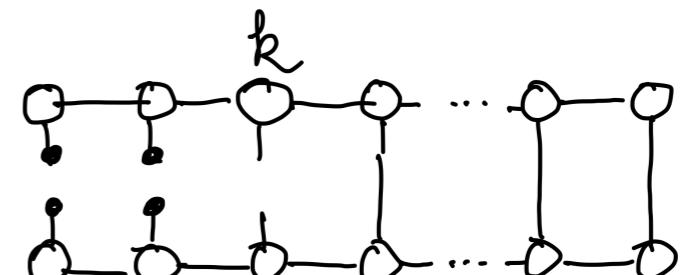
- タワーサンプリング: メモリ $\sim O(2^n)$, 計算量 $\sim O(2^n)$
- 波動関数がMPSの形で得られていれば、そこからのサンプリングは容易
 - 周辺分布

$$P(s_1) = \sum_{s_2, \dots, s_n} P(s_1, s_2, \dots, s_n)$$



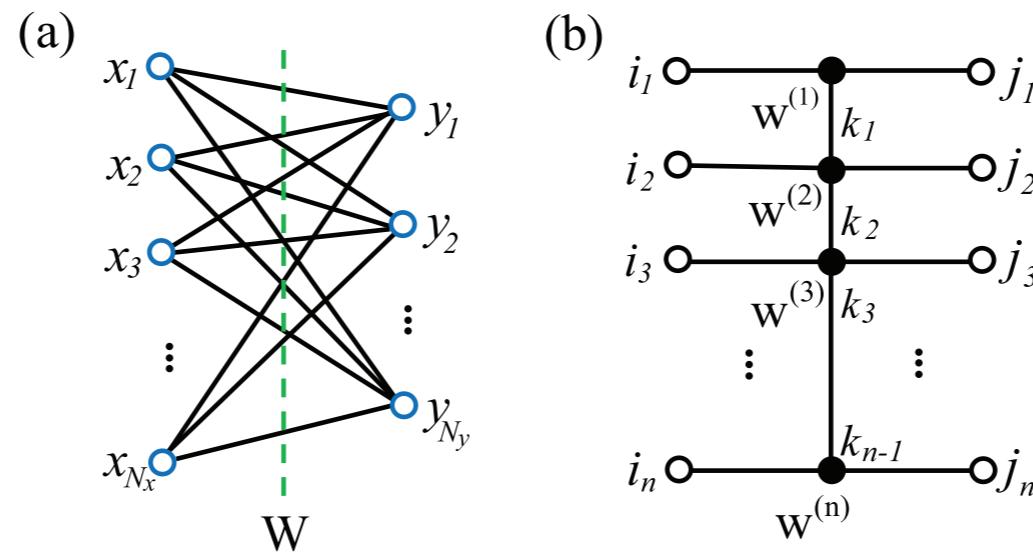
- 条件付き分布

$$P(s_k | s_1, \dots, s_{k-1}) = \sum_{s_{k+1}, \dots, s_n} P(s_1, s_2, \dots, s_n)$$



深層学習の情報圧縮

- 重み行列のテンソルネットワークによる圧縮

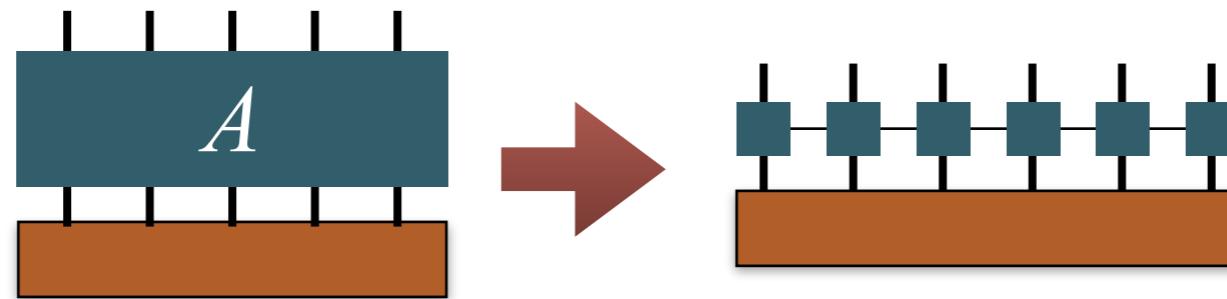


- W をMPS形式(MPO)で近似する

Data set	Network	Original Rep a (%)	MPO-Net	
			a (%)	ρ
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	~ 0.0005
	VGG-19	93.36 ± 0.26	93.80 ± 0.09	~ 0.0005

機械学習における行列圧縮

- ・ 行列の情報を蓄えるためのデータ量を大幅に圧縮できる
 - ・ テンソルの行列積分解と同様
- ・ 行列の演算コストの低下
 - ・ 大きな行列を直接かける代わりに、小さいテンソルを順番にかけてコストが低下



- ・ 訓練・予測時間の低減
- ・ 無駄な自由度の削減による最適化(訓練)の効率アップ
 - ・ テンソルネットワーク近似で性能が変わらないことは、元の表現には「無駄」が多いことを示唆
 - ・ 過学習、汎化性能への影響も期待？

階層構造の圧縮表現

微分方程式への応用

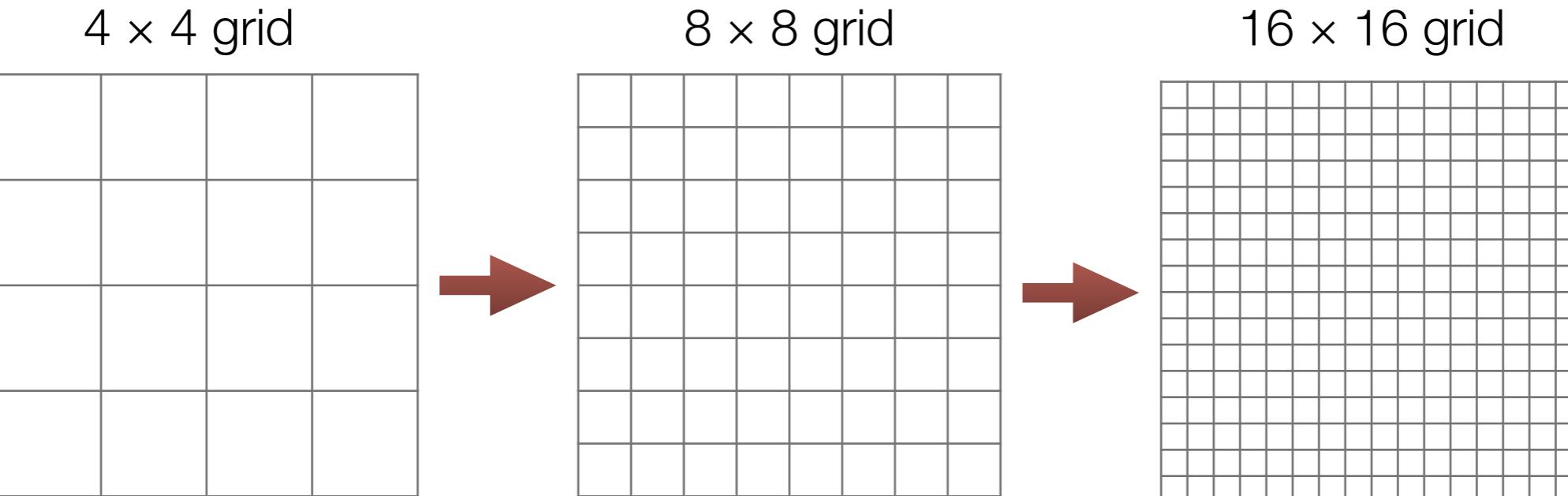
- ・非圧縮のNaiver-Stokes 方程式

$$\nabla \cdot V = 0$$

$$\frac{\partial V}{\partial t} + (V \cdot \nabla) V = -\nabla p + \nu \nabla^2 V,$$

- ・直接法によるシミュレーション

- ・離散化された格子グリッド上で微分方程式を解く
- ・乱流のような複雑な問題を扱う場合、無数の長さスケールが混合するため、非常に小さなメッシュが必要

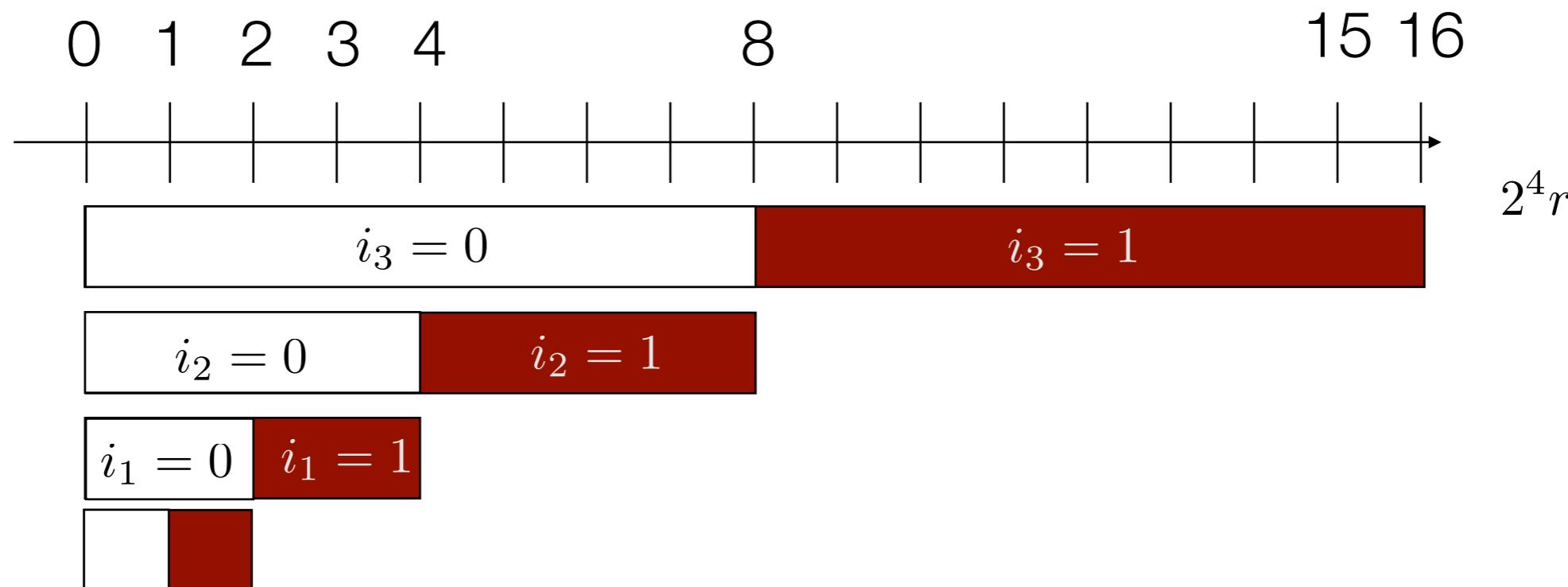


同次多項式表現(quantics representation)

- 実数の N ビット二進数表現

$$r_q \in [0, 1) \rightarrow 2^N r_q = i_0 + 2^1 i_1 + 2^2 i_2 + \cdots + 2^{N-1} i_{N-1}$$

- 各ビットが“長さスケール”に対応している

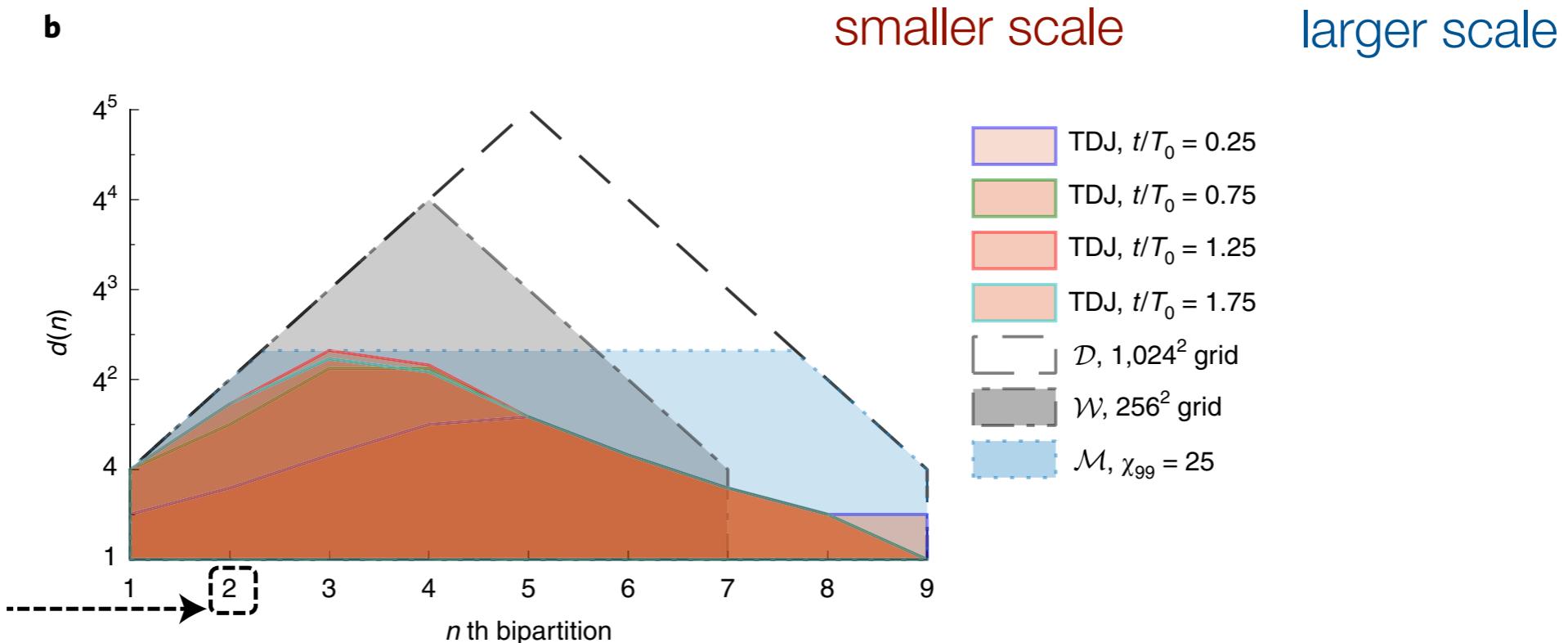


同次多項式表現(quantics representation)

- ・実数の N ビット二進数表現
 - ・テンソル(ネットワーク)形式で表した場合の各足は異なる長さスケールに対応
- ・長い空間スケールと短い空間スケールの間の“特異値分解”

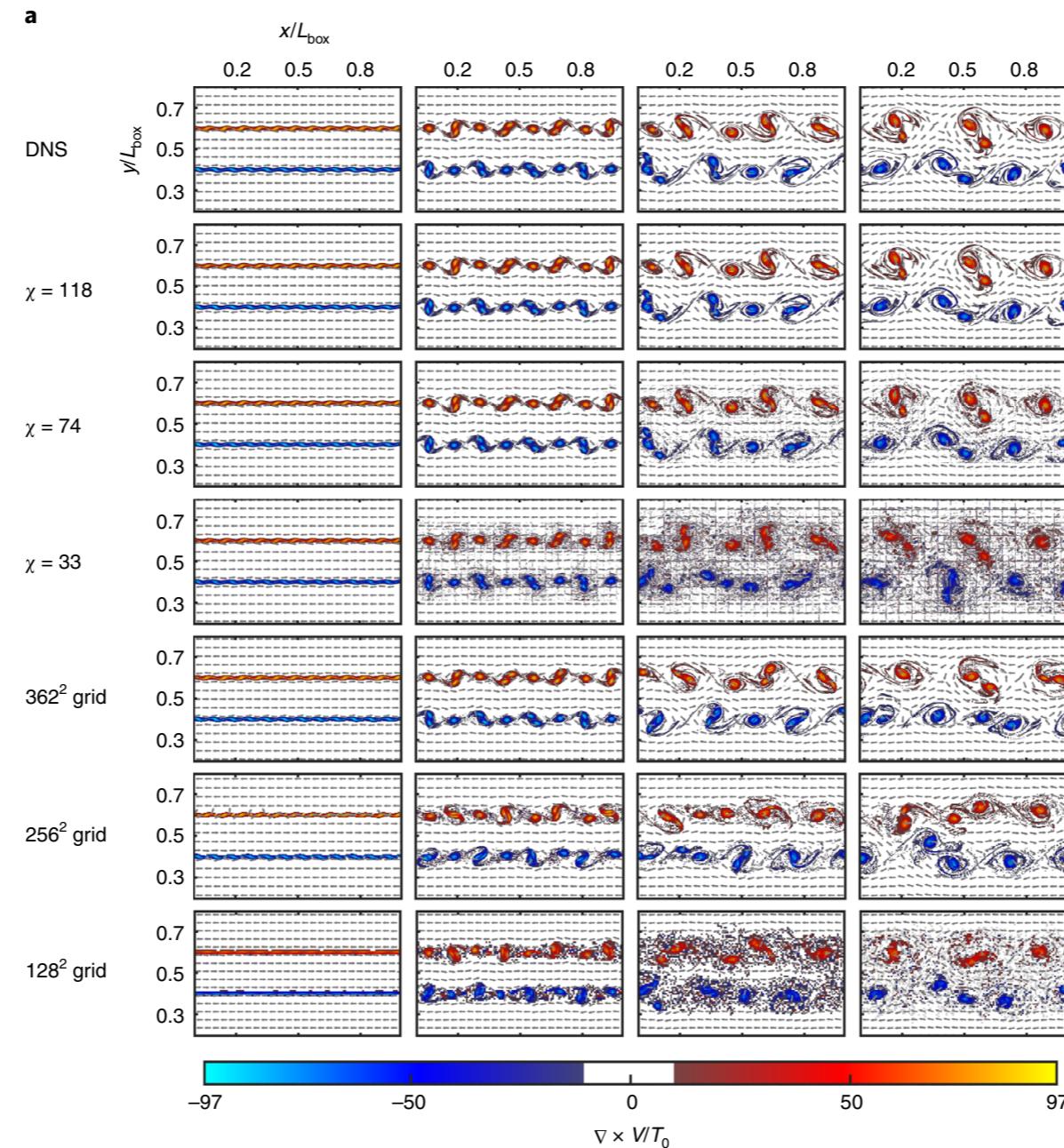
$$V(t, r_q) = \sum_{i=1}^K u_i(t, r_q) \hat{\mathbf{e}}_i \quad u_i(t, r_q) = \sum_{\alpha=1}^{d(n)} \lambda_\alpha(t) R_\alpha(t, X_k) f_\alpha(t, x_l), \quad r_q = X_k + x_l.$$

$$2^N r_q = i_0 + 2^1 i_1 + 2^2 i_2 + \cdots + 2^{N-1} i_{N-1}$$



微分方程式への応用

- ・MPSを使うことで、乱流を正しく捉えたまま、効率的なシミュレーションが可能



Nikita Gourianov et. al., Nat. Comput. Sci. 2, 20 (2022)

まとめ

- ・ テンソルネットワークはいたるところに現れる
 - ・ 様々なデータをテンソルネットワークで厳密あるいは近似的に表現できる
 - ・ 特異値分解を使うことで最適な近似が可能に
- ・ テンソルネットワーク形式
 - ・ データ(=要素数)の削減
 - ・ 計算量の削減
- ・ ネットワークの選択
 - ・ テンソルネットワーク表現は一意ではない
 - ・ どのような形のテンソルネットワークで表すかにより効率が劇的に変わる
- ・ テンソルネットワークの可能性
 - ・ 量子状態、深い階層にまたがる現象、深層学習などにおいて、情報を効率的に圧縮し、大規模な計算が可能に