

2026-09-09 離散的手法による場と時空のダイナミクス



量子多体系計算のためのテンソルネットワーク: 量子計算から機械学習まで

藤堂眞治

東京大学大学院理学系研究科

自己紹介 - 藤堂眞治



• 経歴

- 1996 東京大学大学院理学系研究科博士課程修了 博士(理学)
- 1996 東京大学物性研究所（六本木→柏）共同利用スパコンの運用
- 2000 スイス連邦工科大学理論物理学研究所（Zurich）
- 2002 東京大学大学院工学系研究科物理工学専攻
- 2011 東京大学物性研究所（神戸AICS(当時)に常駐）
- 2014 東京大学大学院理学系研究科物理学専攻



• 興味

- 計算物理、統計物理、固体物理、データ科学、量子コンピューティング
- モンテカルロ、テンソルネットワーク、最適化、機械学習
- オープンソースソフトウェア開発・公開

• 趣味

- 学生のプログラムのデバッグ
- オープンソースソフトウェアのインストール

ALPS

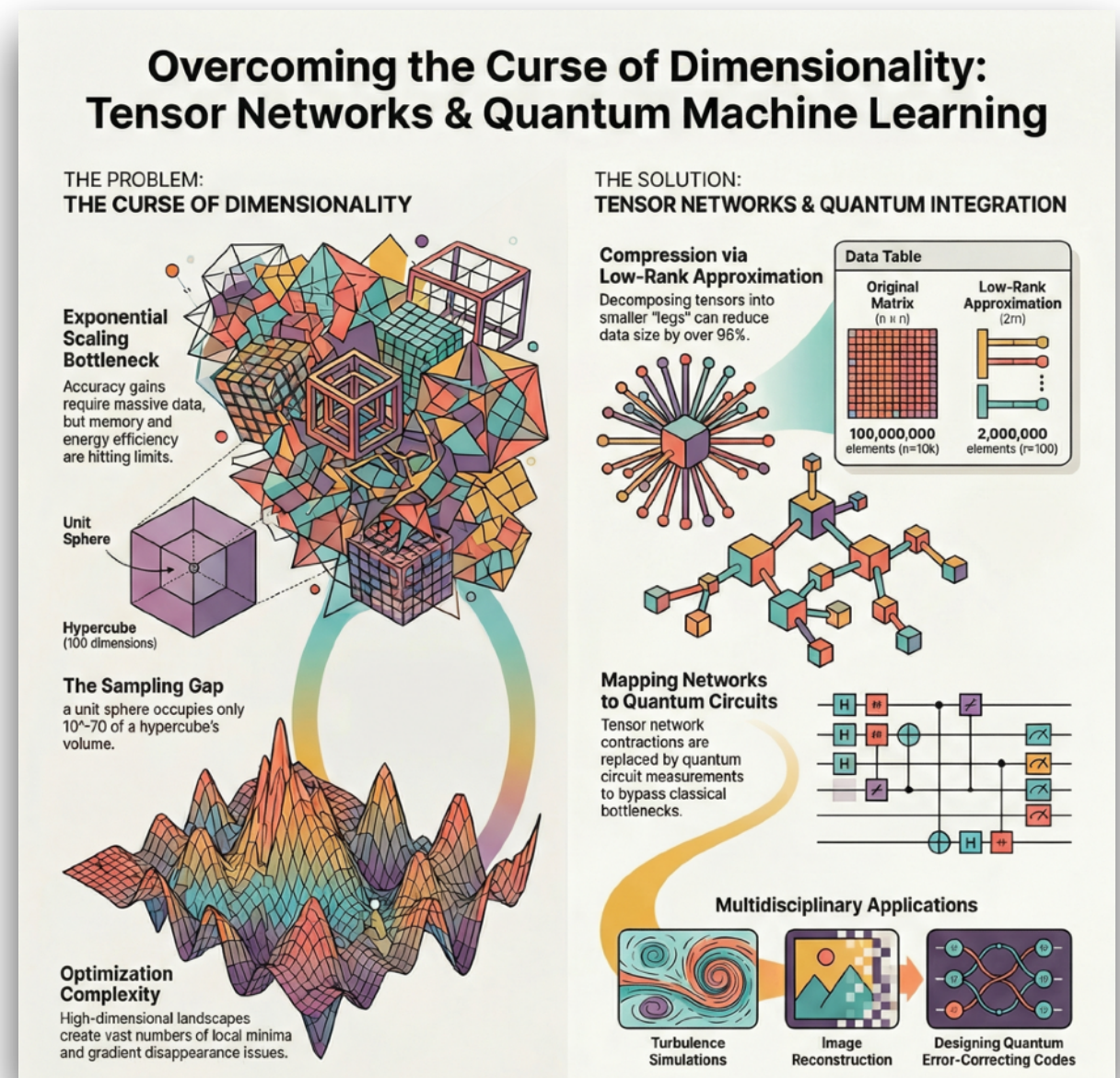


MateriApps

MateriApps
LIVE!

Today's Agenda

- 量子多体系計算とテンソルネットワーク
- テンソルネットワークの基礎
- 縮約と繰り込み群
- サンプリングとテンソルネットワーク
- テンソルネットワークから見た量子計算
- テンソルネットワークによる関数表現と機械学習

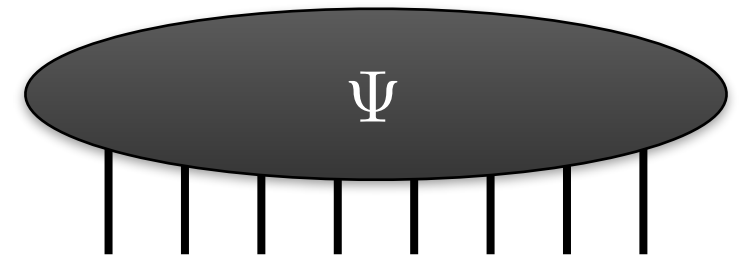


量子多体系計算とテンソルネットワーク

量子系の波動関数

- 量子状態

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



- 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

古典統計物理

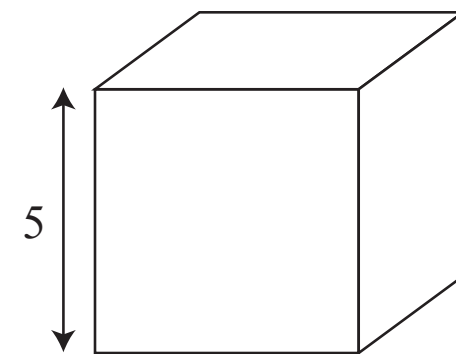
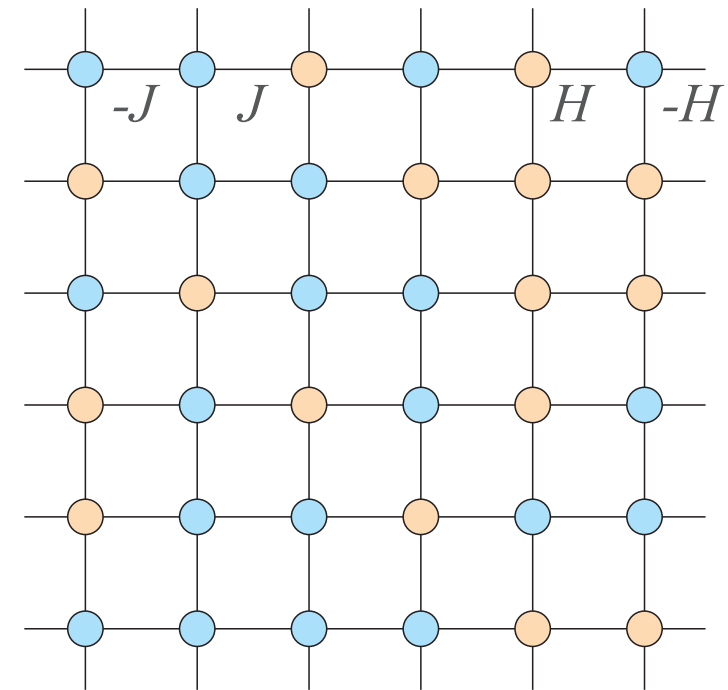
・イジング (Ising) 模型

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

- ・ 物理量の期待値の計算(例: 磁化)

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

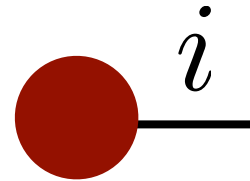
- ・ 単純立方格子(一辺の長さ 5)
 - ・ スピン数 125 $\rightarrow 2^{125} \approx 10^{38}$ の和
- ・ 期待値の計算にかかる時間
 - ・ $10^{38} \div (10^{18}/\text{sec}) \approx 3$ 兆年



テンソルとテンソルのダイアグラム表記

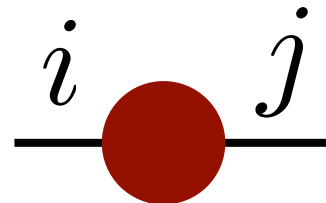
- テンソル: 多次元的な数字の並び (= 多次元配列)
- ベクトル

$$\vec{v} : v_i$$



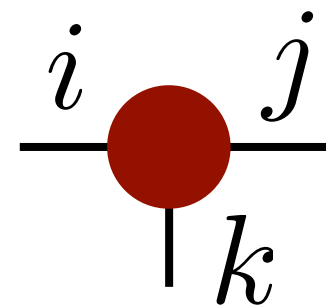
- 行列

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



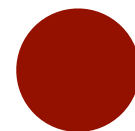
- 3階テンソル

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



- n 階テンソル $\rightarrow n$ 本の足

- スカラー(数字) $\rightarrow$ 足なし

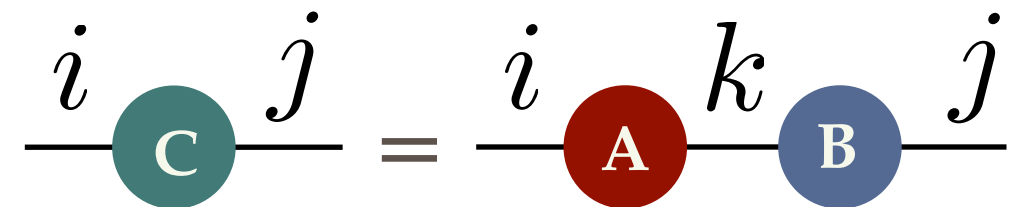


テンソルの縮約

- 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

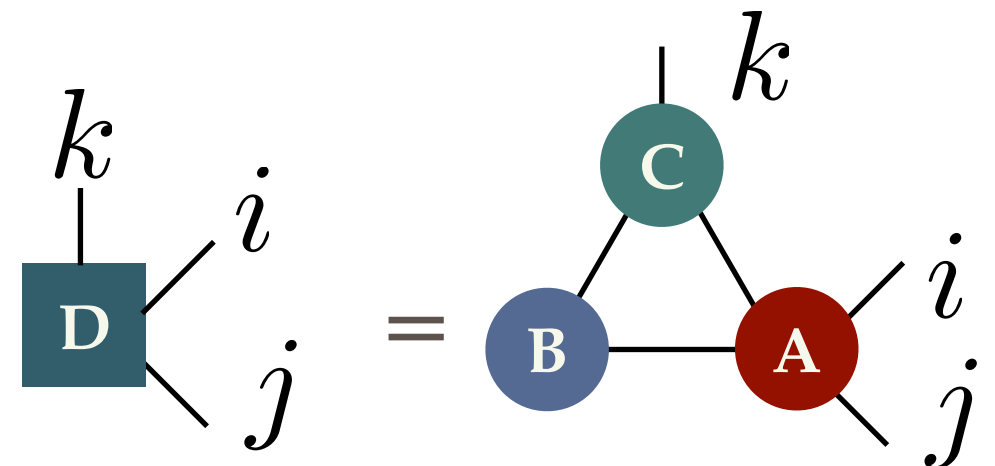
$$C_{i,j} = (AB)_{i,j} = \sum_k A_{i,k} B_{k,j}$$

$$C = AB$$



- 一般のテンソルの縮約も同様に表現できる

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



Ising分配関数からテンソルネットワークへ

- 局所Boltzmann因子は 2×2 の行列で表される。この行列を分解し、各サイトのスピンを足し上げると局所テンソルができる

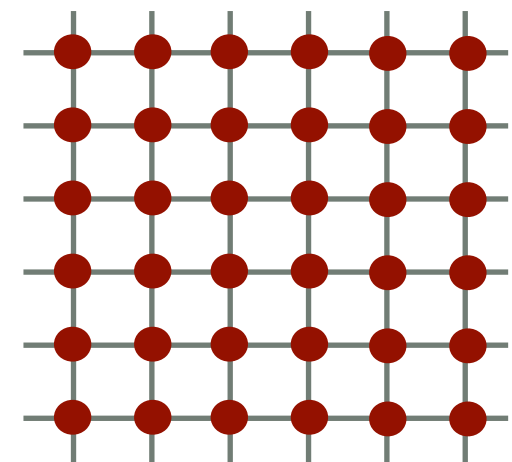
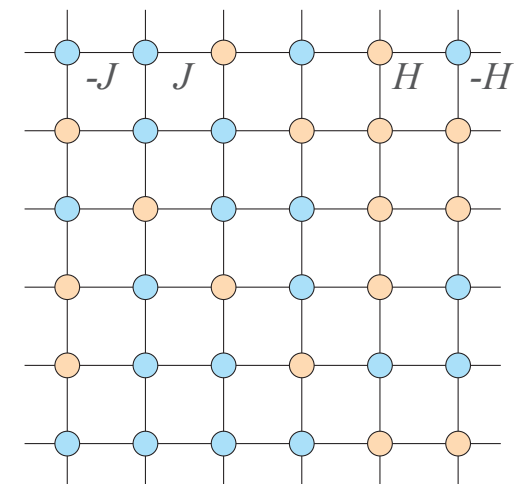
$$e^{-\beta J \sigma \sigma'} = e^{K \sigma \sigma'} = \sum_{a=0,1} W_{\sigma a} W_{\sigma' a}, \quad W_{\sigma 0} = \sqrt{\cosh K}, \quad W_{\sigma 1} = \sigma \sqrt{\sinh K}.$$

- 正方格子の各サイトに

$$T_{abcd} = \sum_{\sigma=\pm 1} e^{\beta h \sigma} W_{\sigma a} W_{\sigma b} W_{\sigma c} W_{\sigma d}$$

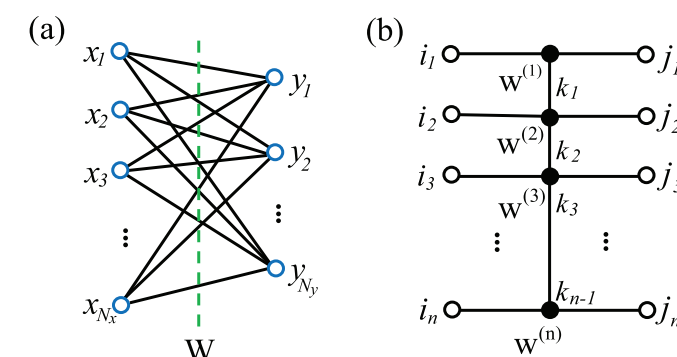
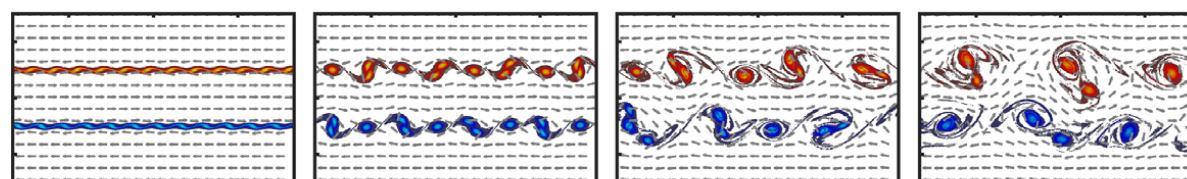
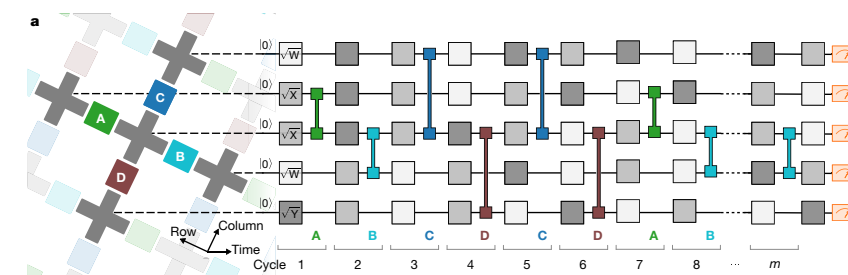
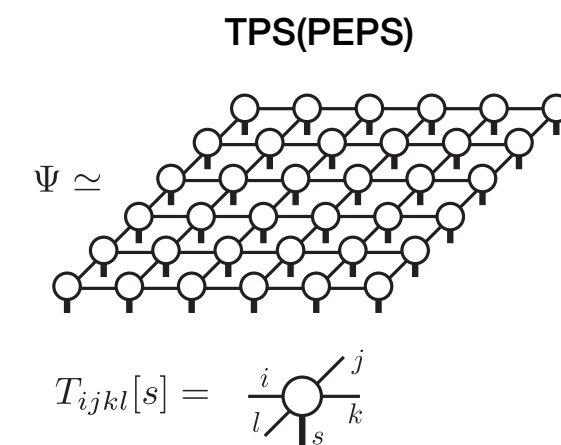
を置くと、共有ボンドの添字を足し合わせたものが分配関数になる

$$Z = \text{tTr} \prod_i T^{(i)}$$



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)
- 統計力学モデルの分配関数: $Z = \sum_{\{S_i\}} e^{-\beta \mathcal{H}(\{S_i\})}$
 - M. Levin and C. P. Nave, Phys. Rev. Lett. 99, 120601 (2007)
 - G. Evenbly and G. Vidal, Phys. Rev. Lett. 115, 180405 (2015)
- 量子回路
 - F. Arute, et al., Nature 574, 505 (2019)
 - Y. Liu, 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)
 - Gourianov et al., Nat. Comput. Sci. 2, 30 (2022)



テンソルネットワークの基礎

特異値分解 (Singular Value Decomposition)

- 任意の $m \times n$ 実行列 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 $U U^T \neq E_n$, $V V^T \neq E_n$

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

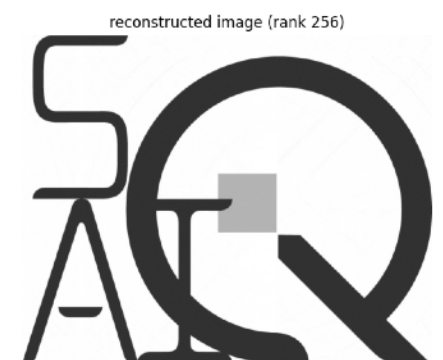
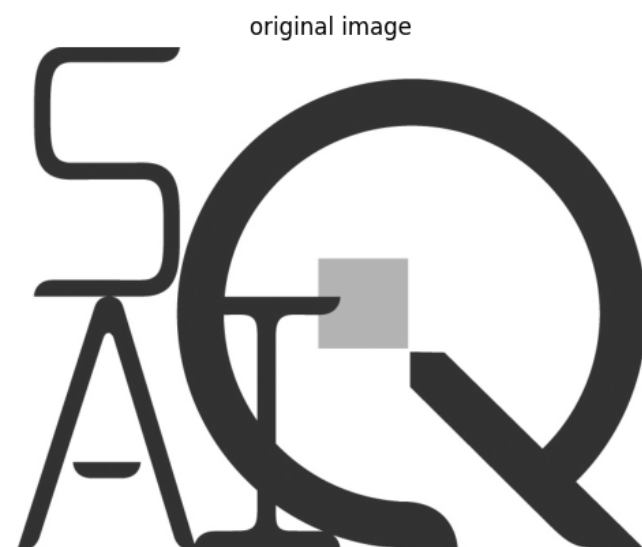
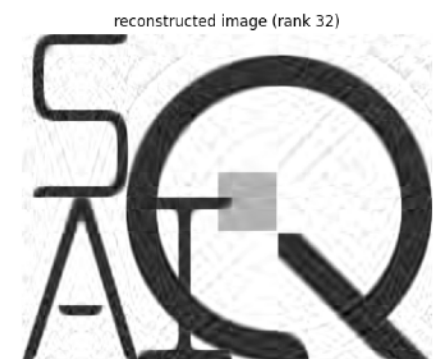
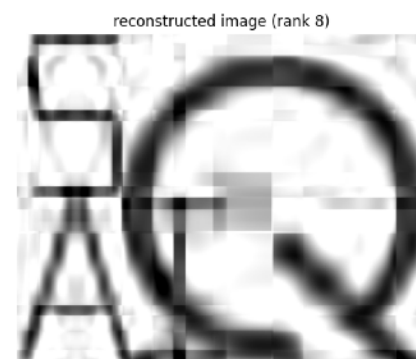
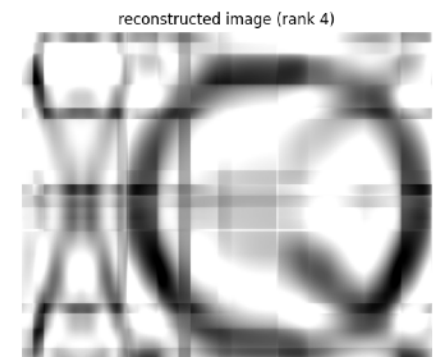
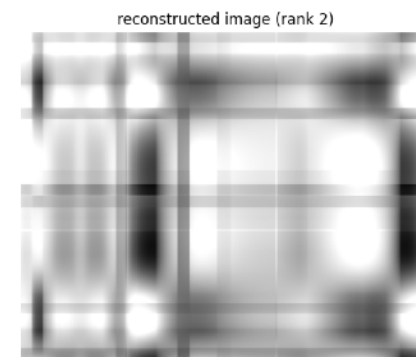
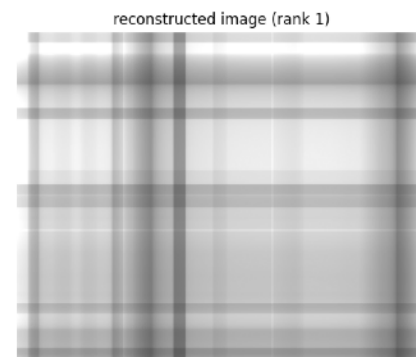
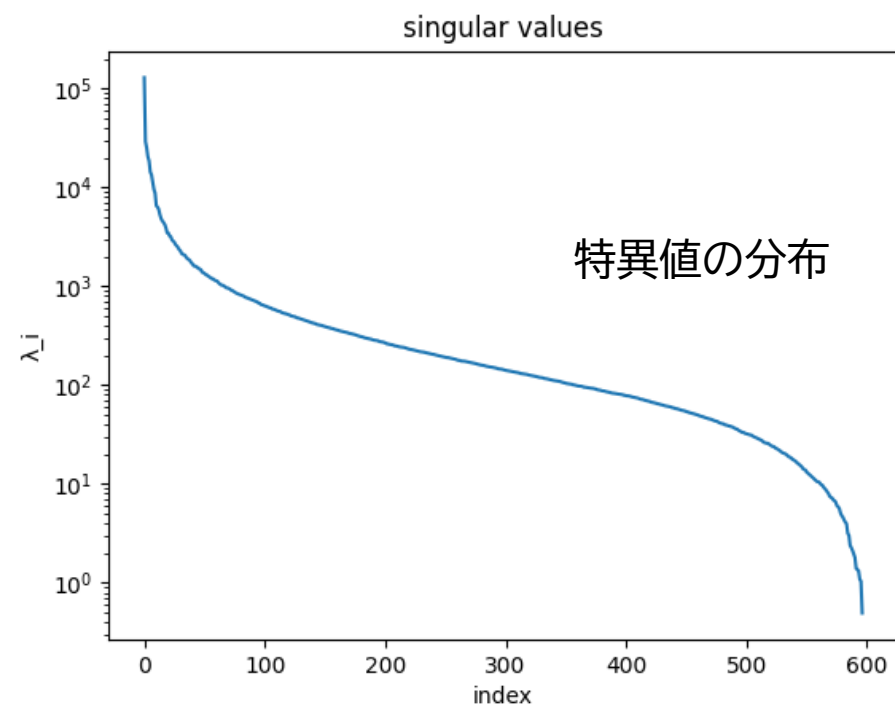
$$\tilde{A} = U \tilde{\Lambda} V^T$$

を作れば良い (Eckart-Young の定理)

特異値分解による画像圧縮

$$\begin{matrix} i & & j \\ & \text{C} & \\ \hline \end{matrix} = \begin{matrix} i & & k \\ & \text{A} & \\ \hline \end{matrix} \begin{matrix} k & & j \\ & \text{B} & \\ \hline \end{matrix}$$

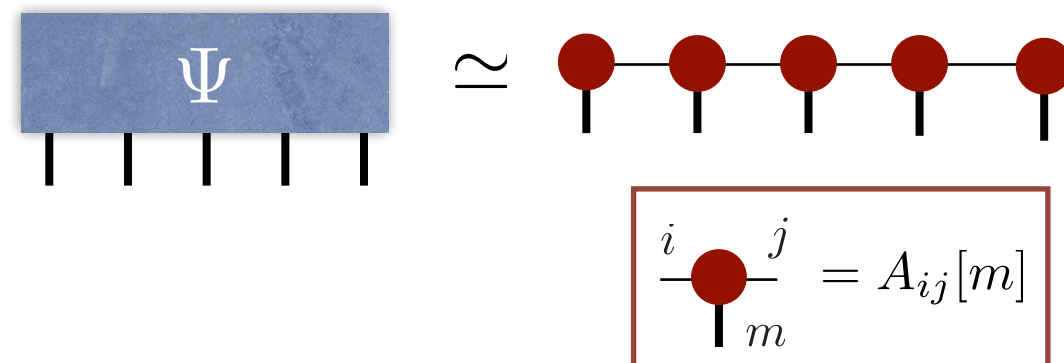
- グレースケール画像(597 x 744)を行列とみなして低ランク近似



MPS (Matrix Product State - 行列積状態)

- 一次元的につながったテンソルネットワーク
 - 応用数理分野では「Tensor Train」と呼ばれる
- N 本足のテンソル(ベクトル)を行列の積で表現

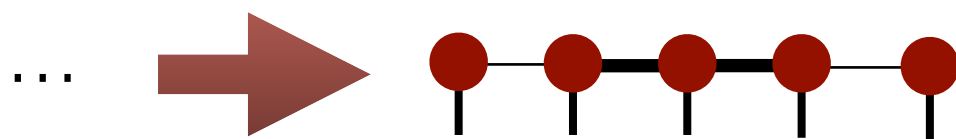
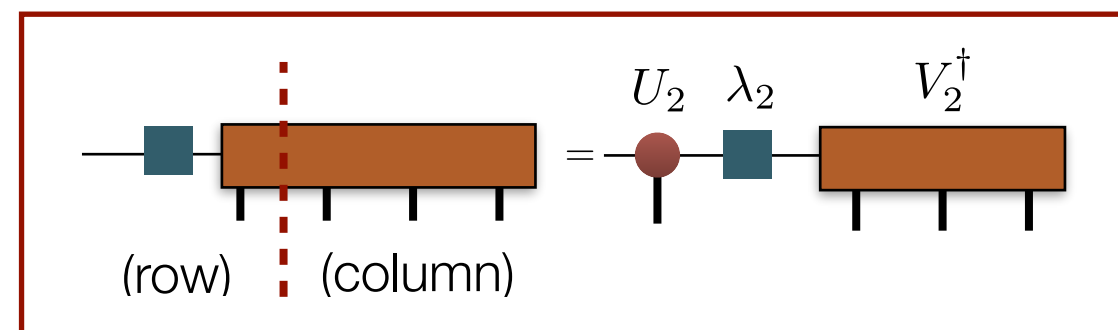
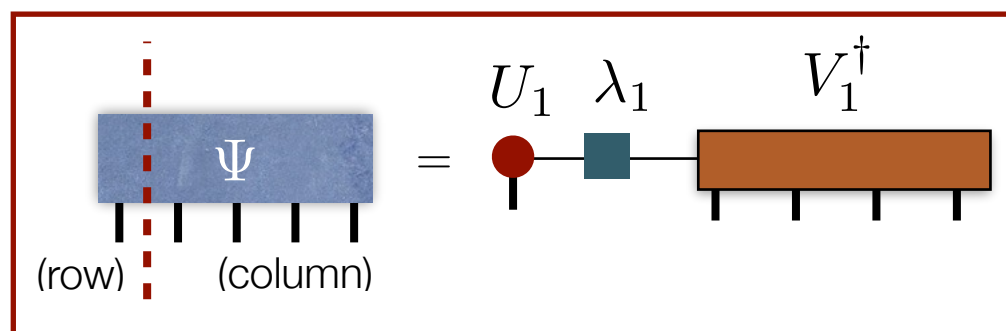
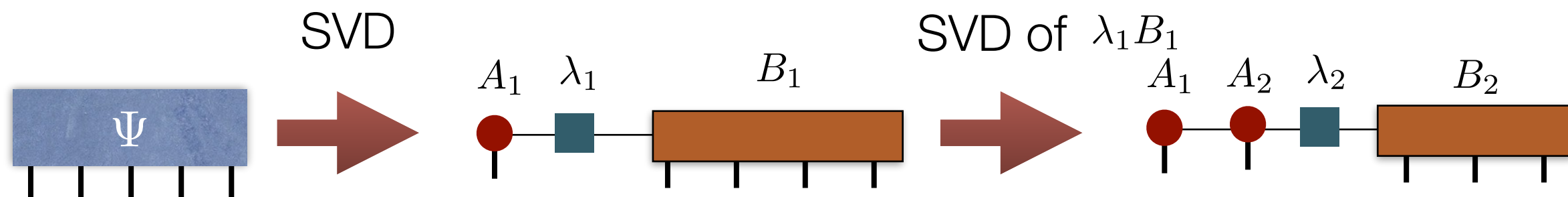
$$\Psi_{i_1 i_2 \dots i_N} \simeq A_1[i_1] A_2[i_2] \cdots A_N[i_N]$$



- MPSから外に出ている足を「物理的な足」と呼ぶ
- テンソルの間を繋ぐ足を「仮想的な足」と呼ぶ
- 仮想的な足について縮約をとると、元のテンソルと同じ形に戻る

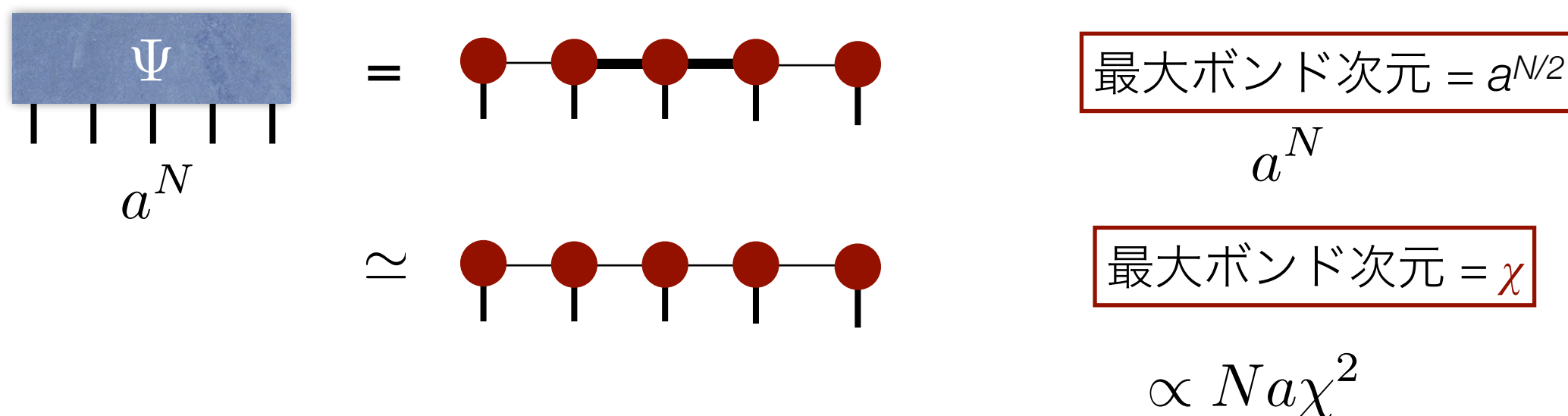
行列積状態への変換

- 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 $= O(a^{N/2})$

低ランク近似



- ゼロの特異値が現れた場合には、それに対応する行や列は捨ててよい
 - テンソルネットワークによるテンソルの圧縮表現
- 十分に小さな特異値を捨てることで、元のテンソルを精度良く近似可能
 - テンソルネットワークによるテンソルの低ランク近似
 - 指数関数的なデータ量を、多項式程度まで大幅に圧縮できる
- (どこまで効率的に近似可能かどうかは、表そうとしている「状態」の性質(=エンタングルメントの強さ)による)

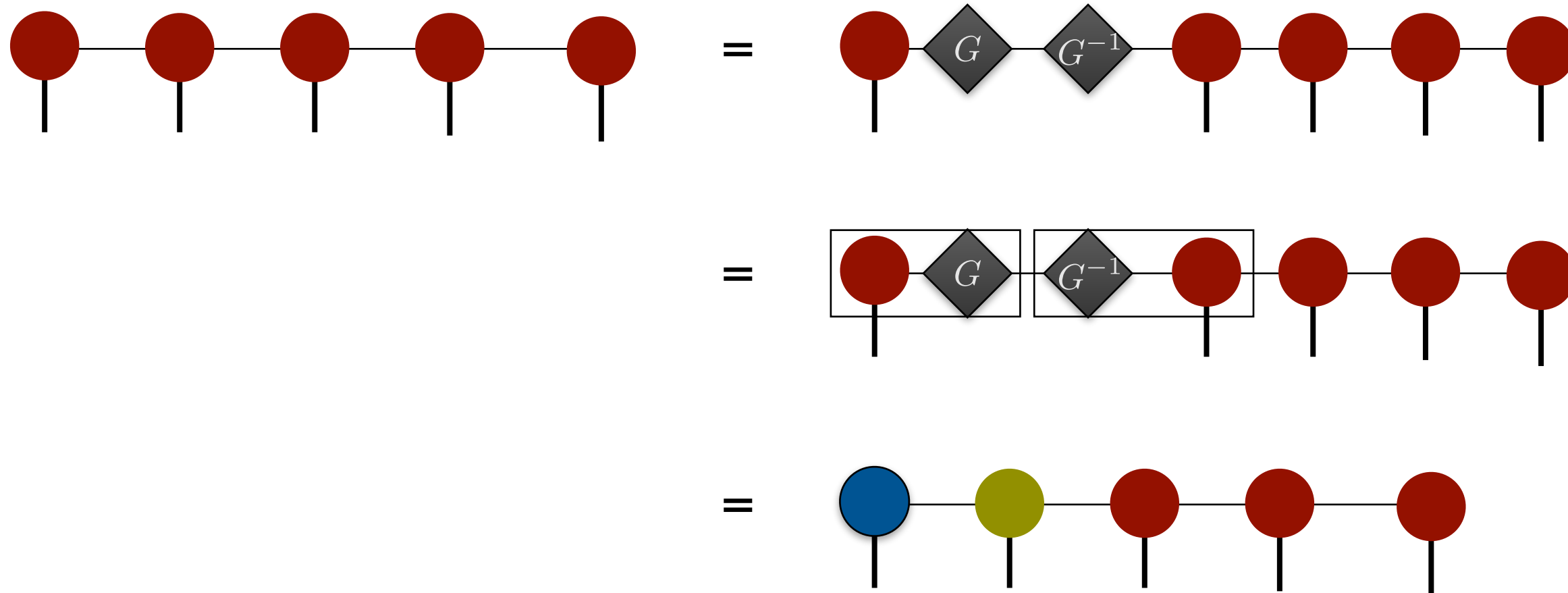
MPS のゲージ冗長性

- MPS は一意ではない

- ゲージ自由度がある

- GG^{-1} という行列を挿入しても, MPS の縮約結果は変わらない

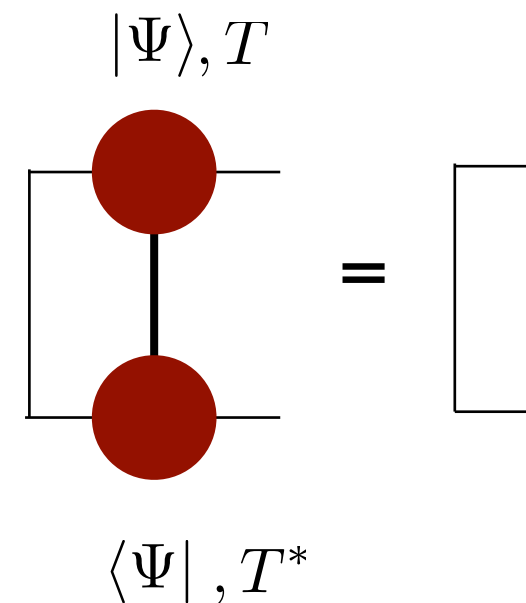
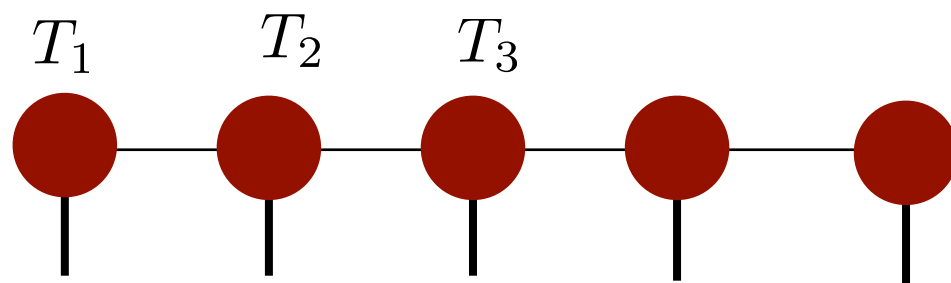
$$I = GG^{-1} \quad \text{---} = \text{---} \diamond G \text{---} \diamond G^{-1} \text{---}$$



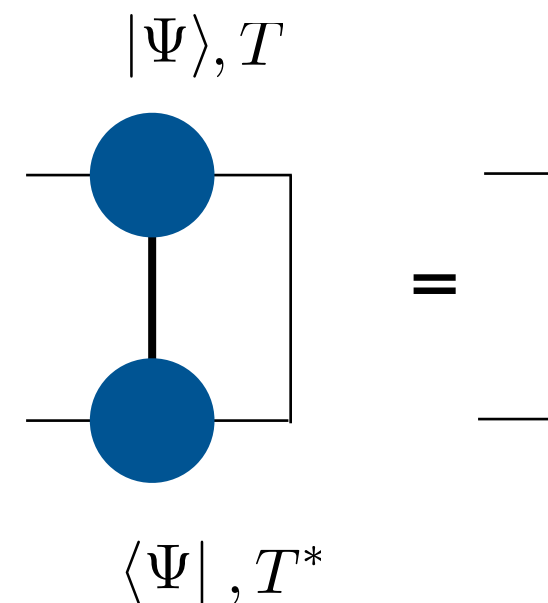
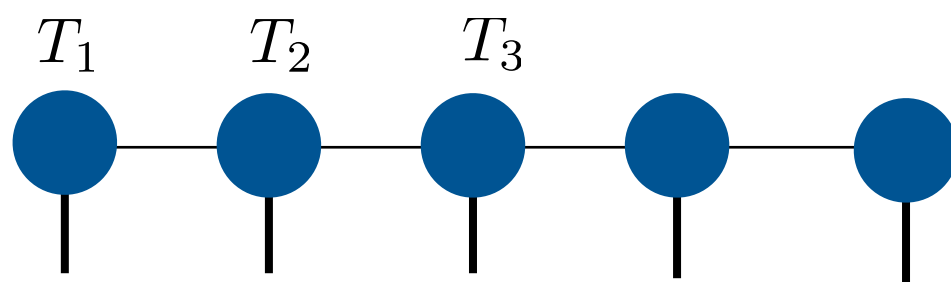
MPSのカノニカル形式(正準形)

- ゲージ自由度 G を使って MPS を「正準化」できる

- 左カノニカル形式



- 右カノニカル形式



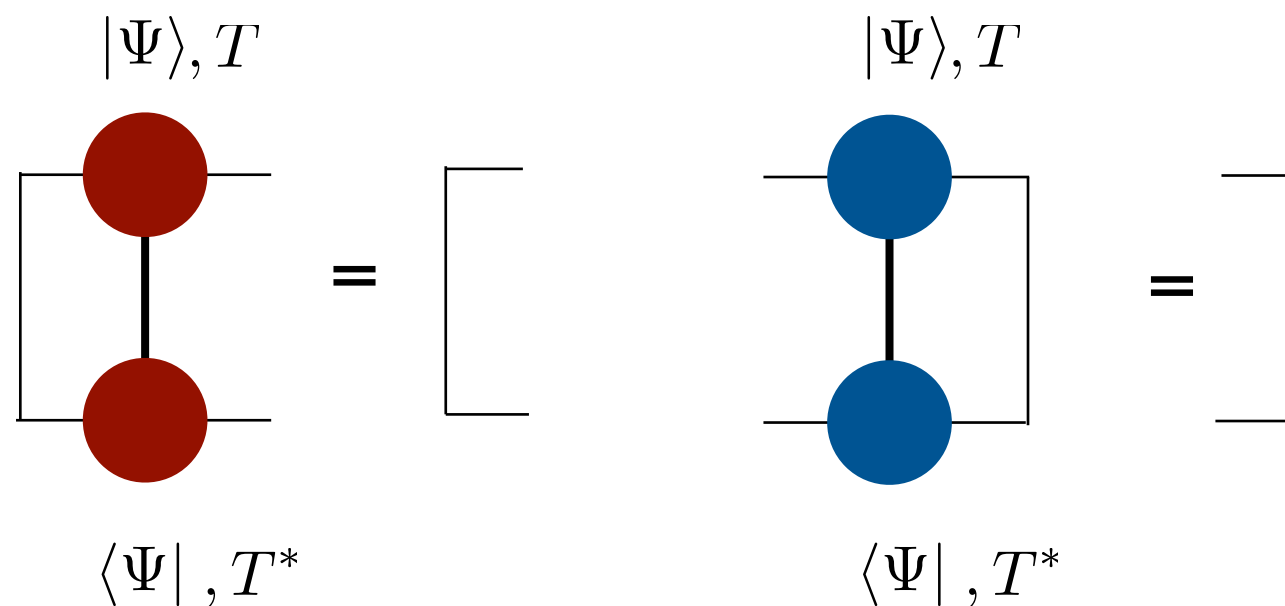
- G はSVD(または対角化)を使って求められる

混合カノニカル形式

- 混合カノニカル形式



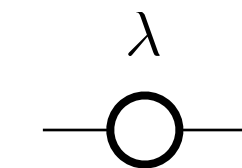
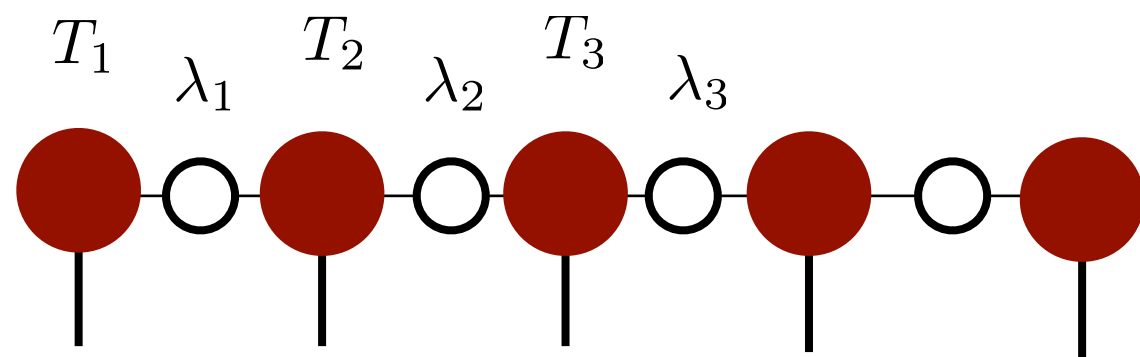
- 左・右カノニカル条件



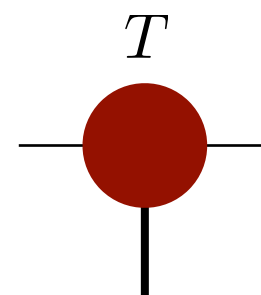
- 左側と右側の境界は SVD を使って動かせる

Vidalカノニカル形式

- MPSの別のカノニカル形式(Vidalカノニカル形式)

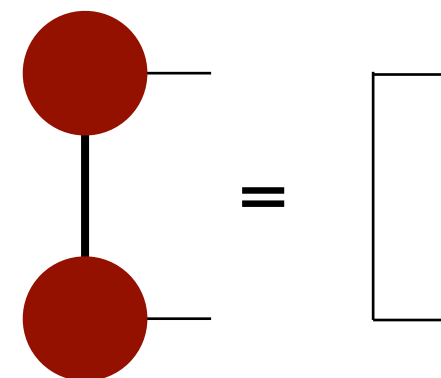
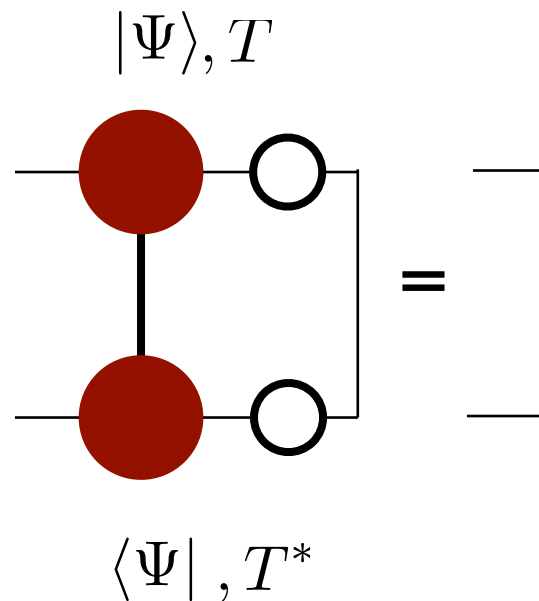
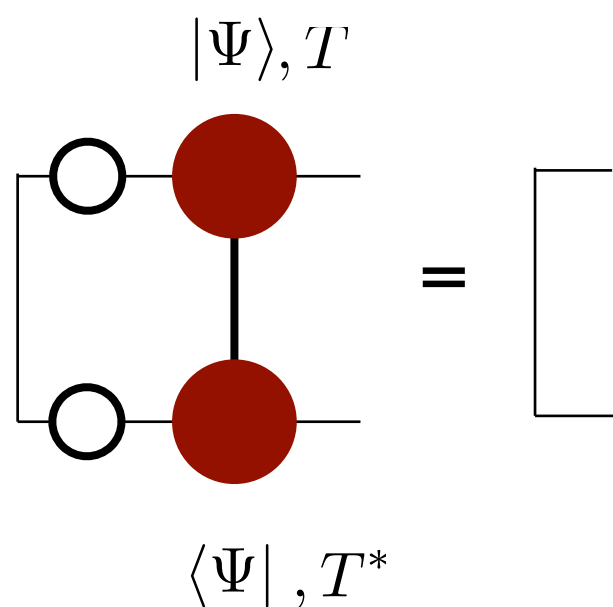


Schmidt係数に対応する対角行列



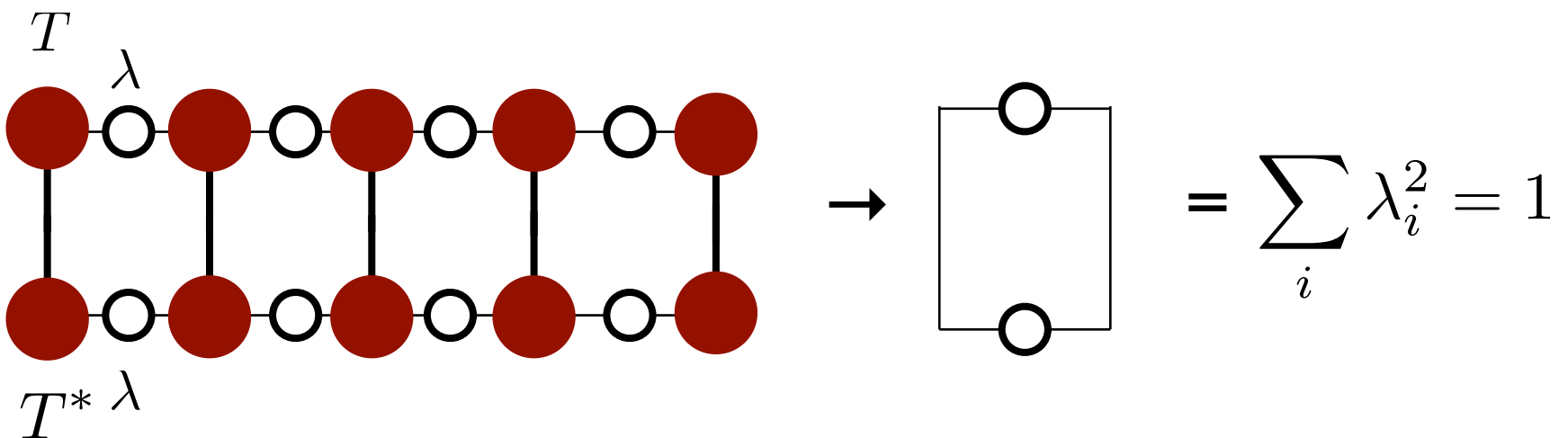
仮想インデックスは
Schmidt 基底に対応

- 右・左・境界カノニカル条件

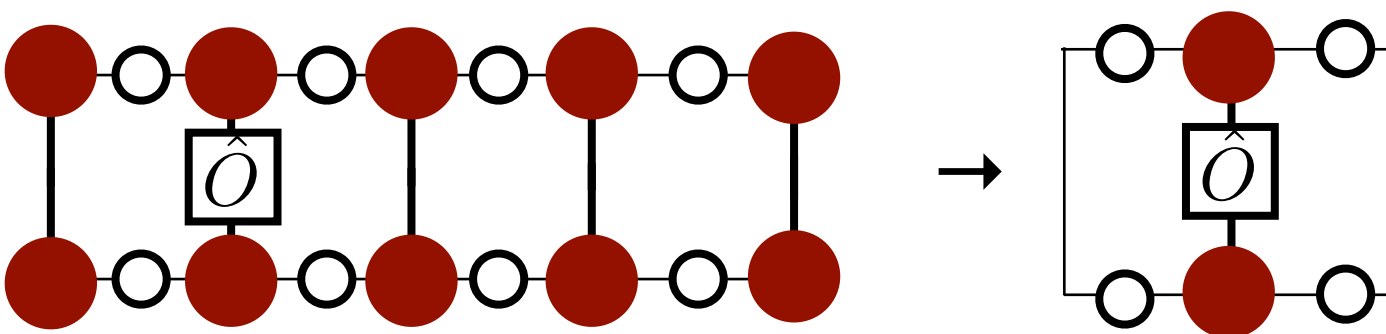


カノニカル形式のMPSにおける期待値

- カノニカル条件のもとでは期待値の評価が非常に簡単になる

$$\langle \Psi | \Psi \rangle =$$


$$= \sum_i \lambda_i^2 = 1$$

$$\langle \Psi | \hat{O} | \Psi \rangle =$$


- 混合カノニカル形式でも同様に簡単な図式が得られる

縮約と繰り込み群

テンソルネットワークの縮約

- 効率的な縮約？

- 縮約順序が性能を大きく左右する
- 悪い順序: 巨大な中間テンソルの発生
- 良い順序: 木幅を小さく維持

- 例: 行列積における縮約順序の重要性

- $ABC = (AB)C = A(BC)$?
- $A = 2 \times 1000$, $B = 1000 \times 2$, $C = 2 \times 1000$
- $(AB)C$ の場合
 - $AB: 2 \times 1,000 \times 2$, $(AB)C: 2 \times 2 \times 1,000$, 合計 8,000
- $A(BC)$ の場合
 - 途中で 1000×1000 の巨大行列が出現
 - $BC: 1,000 \times 2 \times 1,000$, $A(BC): 2 \times 1,000 \times 1,000$, 合計 4,000,000
- 最終結果は一致するにもかかわらず、順序の違いで計算コストに「500倍」の差が生じる

テンソルネットワークの「スライシング」

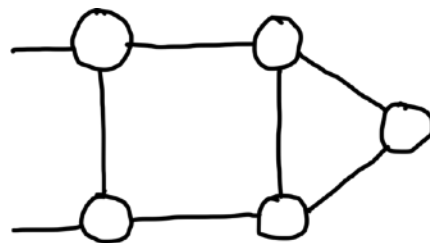
- スライシング(Slicing)

- 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がある値に固定する
 - 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

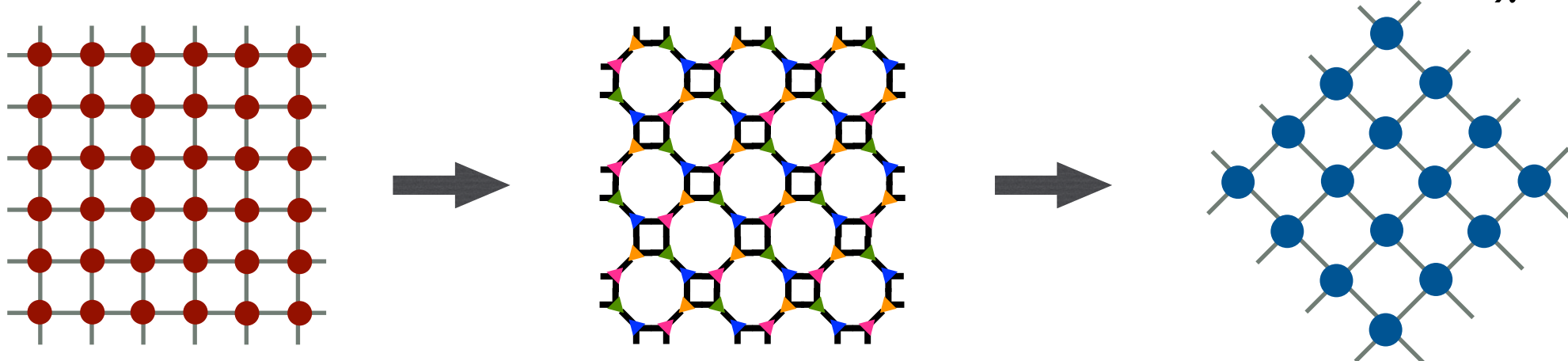
- メモリコストを減らすことが可能
- それぞれの組み合わせに関する縮約は(並列コンピュータ上で)並列に計算可能

- 例:



Tensor network methods in statistical physics

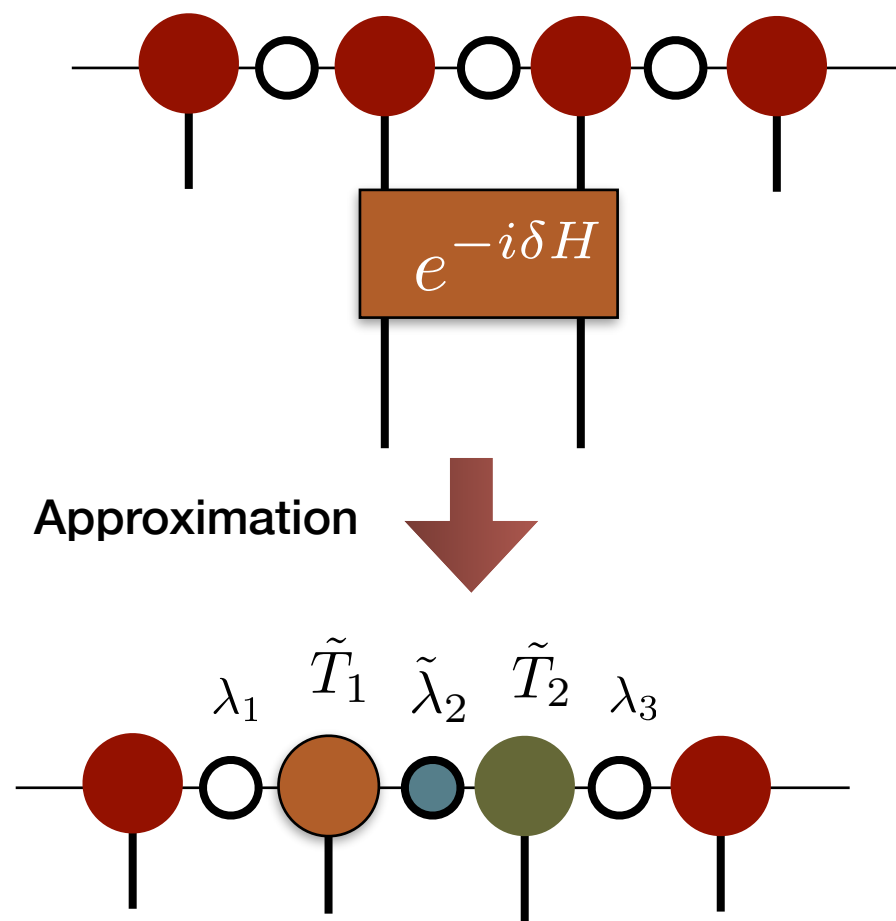
- “Renormalization” (or “Lagrangian”) approach
 - **coarse graining** of tensor network representation of the partition function
 - transfer matrix, tensor network renormalization (TRG), higher-order tensor network renormalization (HOTRG), etc
- “Variational” (or “Hamiltonian”) approach
 - **tensor-network approximation** of strongly correlated many-body quantum states
 - DMRG, PEPS, MERA, etc
- “Exact” contraction of tensor network can not be done in two and higher dimensions
 - **low-rank approximation** based on eigenvalue/singular value decomposition
 - accuracy of approximation is controlled by “**bond dimension**”: D (or χ)



TEBD アルゴリズム

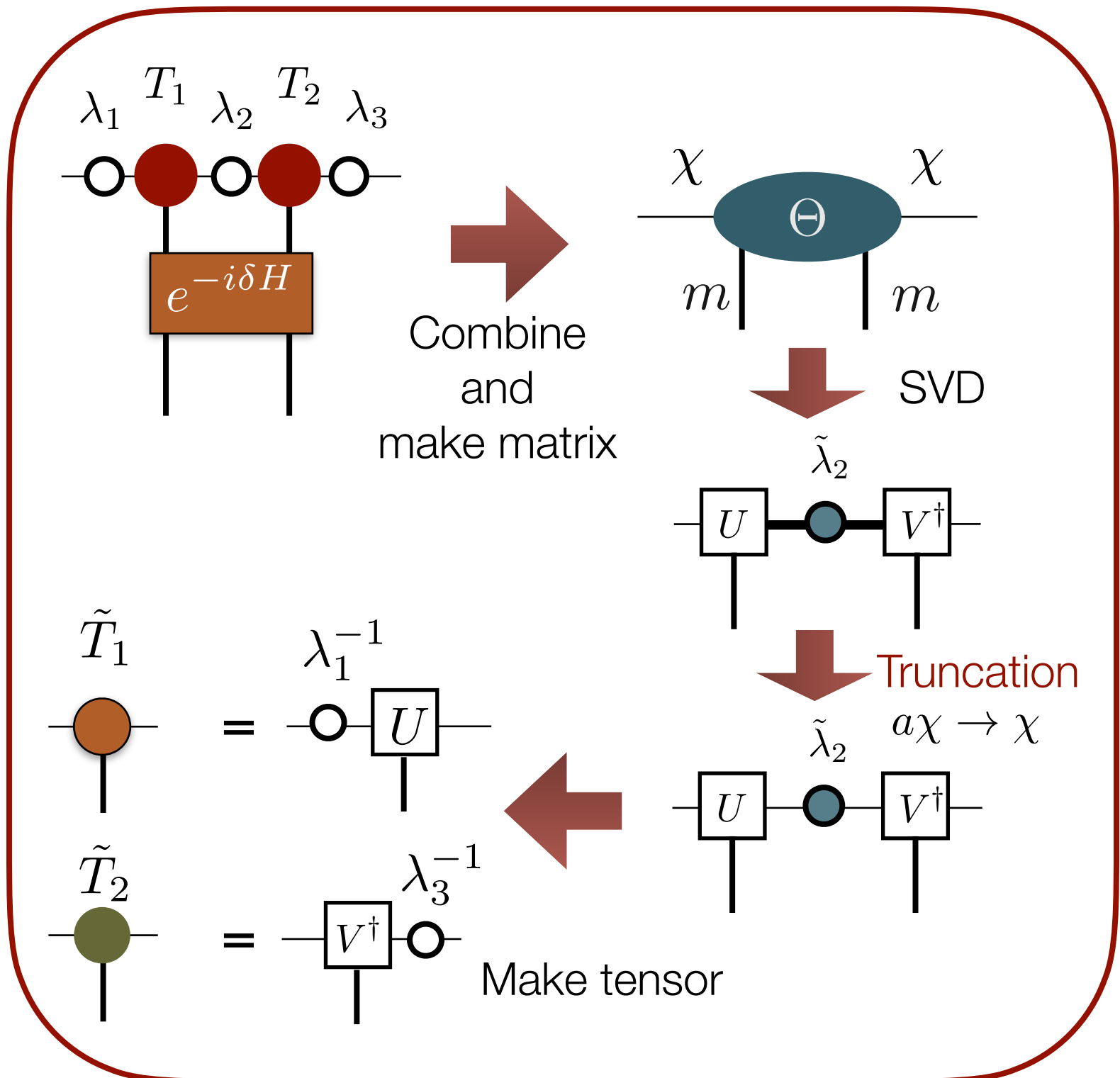
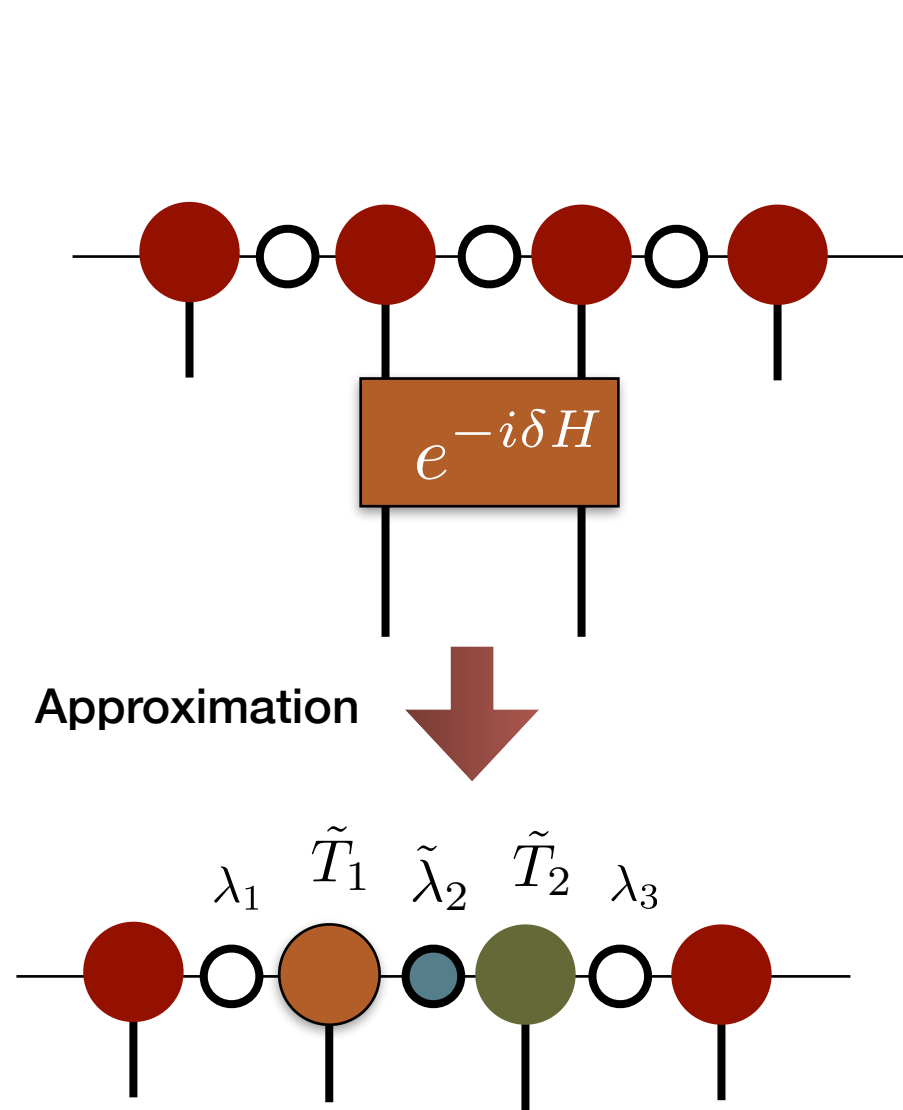
- Time Evolving Block Decimation (TEBD)

- カノニカル形式のMPSを使って, 局所的に高精度な変換を行う



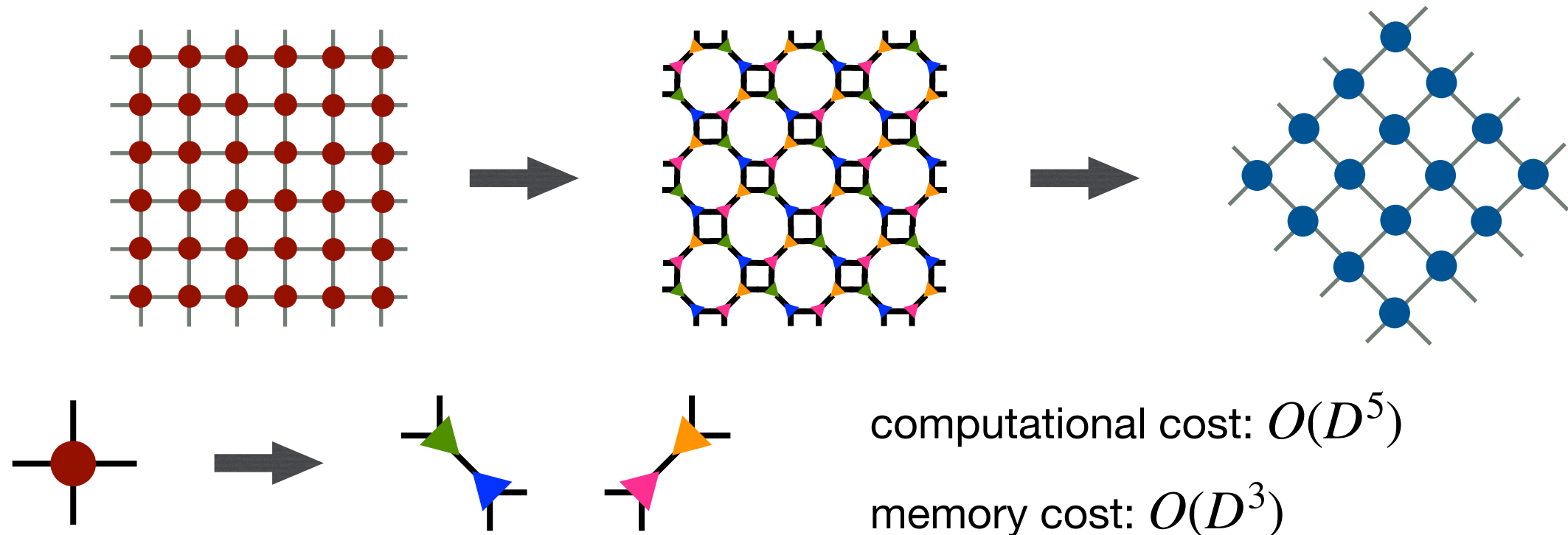
- 時間発展演算子が直接かかる2つのテンソル(+その間の対角行列)だけが変化する

TEBD algorithm



Tensor renormalization group

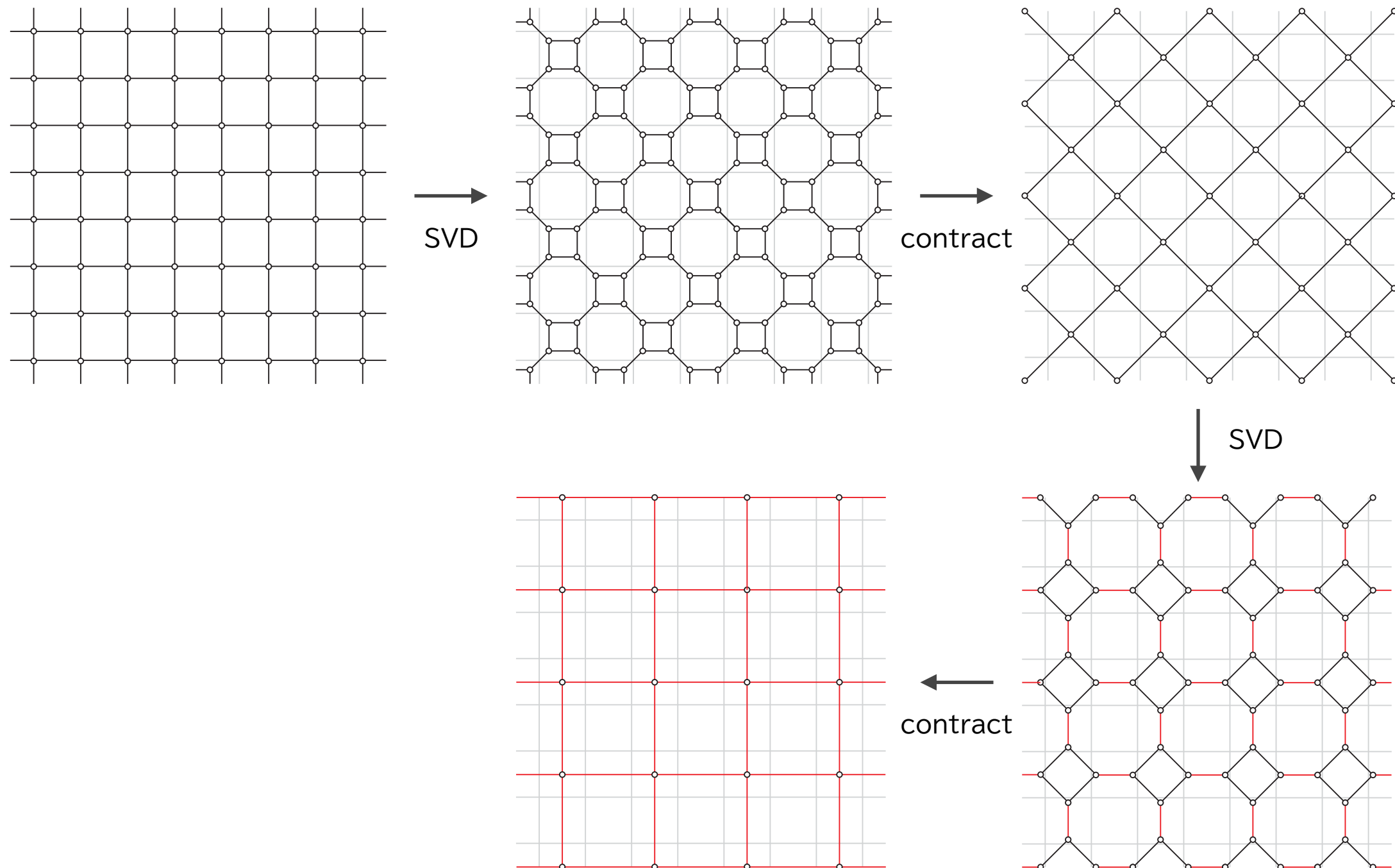
- Low-rank approximation based on SVD



- Improvement of accuracy by considering the “environment” effects or removing local correlations
 - second order renormalization (SRG), mean-field SRG, etc
 - tensor network renormalization (TNR), loop TNR, Gilt, etc
 - computational cost increases significantly

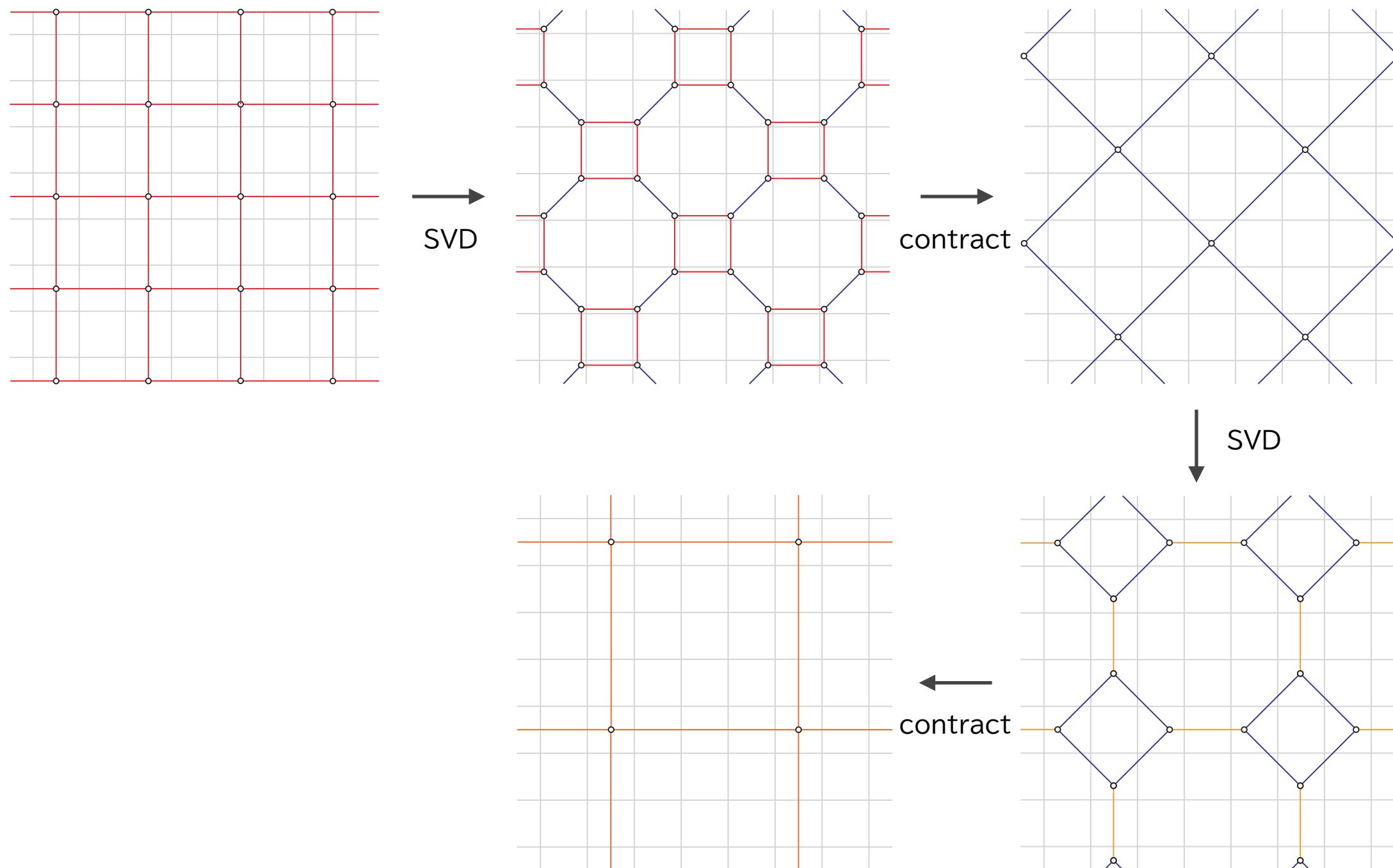
Levin-Nave tensor renormalization group

- 8×8 square lattice case ($N = 64$)



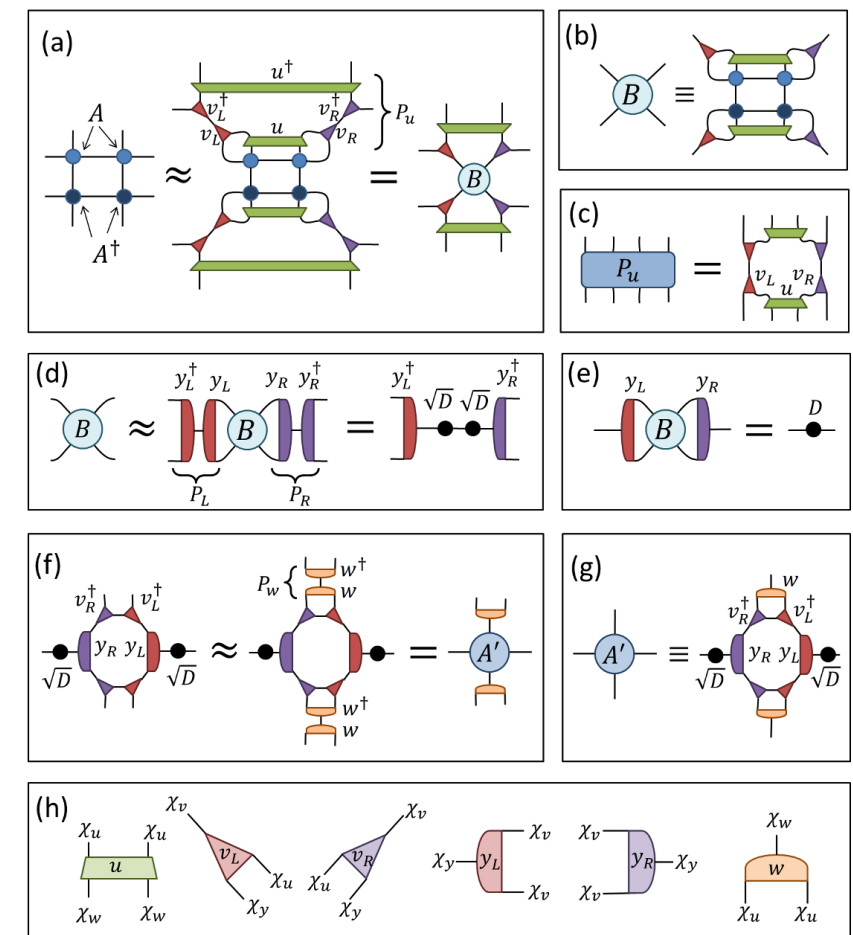
Levin-Nave tensor renormalization group

- 8×8 square lattice case ($N = 64$)



More advanced tensor-network methods

- Improving accuracy
 - including environment effect
 - SRG (2009), CTMRG (1996, 2009)
 - removing local correlations
 - TNR (2015), loop TNR (2017), Gilt (2018)
 - computational cost increases significantly
 - improve accuracy without increasing complexity
 - BTRG (2022)
- Reducing complexity
 - TRG using few-leg tensors
 - ATRG (2020), Triad RG(2019), CATN (2020)
- Generalization to higher-dimensions
 - HOTRG (2012), ATRG (2020), CATN (2020)
- Fermions
 - Grassmann tensor network (2010, 2021)



G. Evenbly (2015)

サンプリングとテンソルネットワーク

MPSからの状態のサンプリング

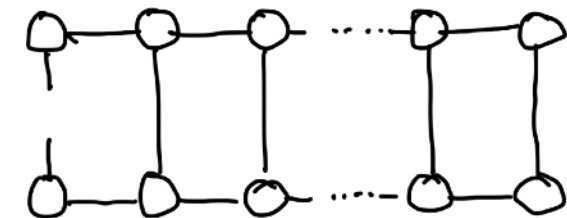
- 2^n 次元空間でのサンプリングは難しい

- $P(s_1, s_2, \dots, s_n) = |\langle s_1 s_2 \dots s_n | \Psi \rangle|^2 / \langle \Psi | \Psi \rangle$
- タワーサンプリング: メモリコスト $\sim O(2^n)$, 計算量 $\sim O(2^n)$

- いったん(近似)MPS 表現が得られれば周辺分布を使って状態を逐次にサンプルできる

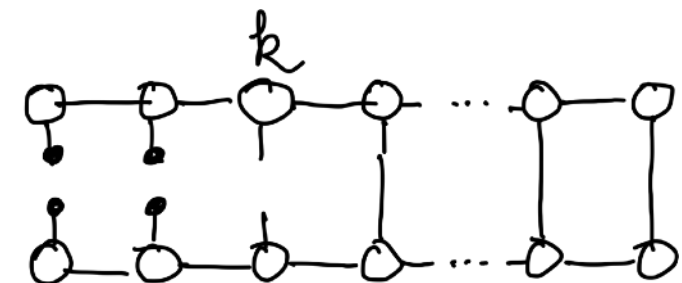
- 周辺分布:

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



- 条件付き分布:

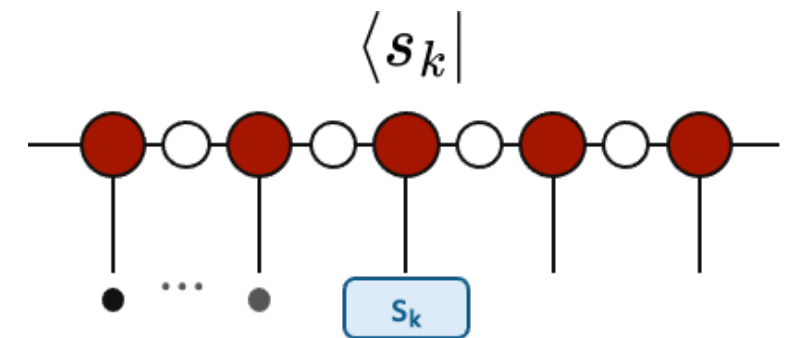
$$P(s_k | s_1, \dots, s_{k-1}) = \frac{\sum_{s_{k+1}, \dots, s_n} P(s_1, s_2, \dots, s_n)}{\sum_{s_k, \dots, s_n} P(s_1, s_2, \dots, s_n)}$$



測定後の状態

- 結果 s_k が得られたら、状態を射影して正規化する

$$|\Psi\rangle \longrightarrow \frac{P_{s_k}^{(k)} |\Psi\rangle}{\sqrt{\langle \Psi | P_{s_k}^{(k)} | \Psi \rangle}}$$

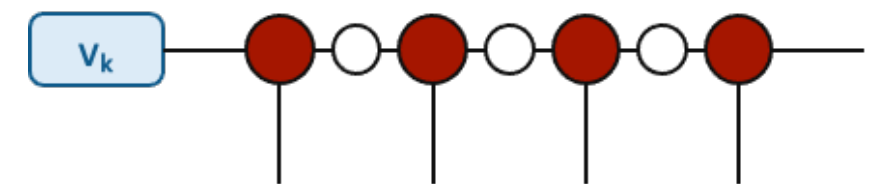


- 右カノニカル形式では、条件付き確率と更新が局所的に書ける
 - 測定後の残りの状態も再びMPSになる

$$p_k(s) = P(s_k = s \mid s_1, \dots, s_{k-1}) = \|v_{k-1} A_k[s]\|^2, \quad v_0 = 1$$

残りの状態は再び MPS

$$v_k = \frac{v_{k-1} A_k[s_k]}{\sqrt{p_k(s_k)}}$$



逐次サンプル→更新→次へ

- これを $k=1, \dots, N$ まで繰り返すと、1本のビット列が得られる

Markov-chain Monte Carlo

- Various update algorithms
 - local, cluster, worm, event-chain, hybrid (HMC), etc
 - extended ensemble methods
- Pros
 - consistent in long-time limit (if balance condition and ergodicity are satisfied)
 - computational cost increases only linearly (in many cases)
 - trivial parallelization
- Cons
 - slow convergence of statistical error ($\sim 1/\sqrt{M}$)
 - effective number of samples decreases as auto-correlation increases

$$\sigma^2 = \frac{2\sigma_0^2 \tau_{\text{int}}}{M}$$

- negative sign for frustrated quantum magnets, fermions, real-time dynamics

Tensor network renormalization group

- Many variants

- tensor renormalization group (TRG) by Levin and Nave (2007)
- higher-order tensor renormalization group (HOTRG) by Xie et al (2012)
- tensor network renormalization (TNR) by Evenbly and Vidal (2015)
- anisotropic tensor renormalization group (ATRG) by Adachi et al (2020)
- bond-weighted tensor renormalization group (BTRG) by Adachi et al (2022)

- Pros

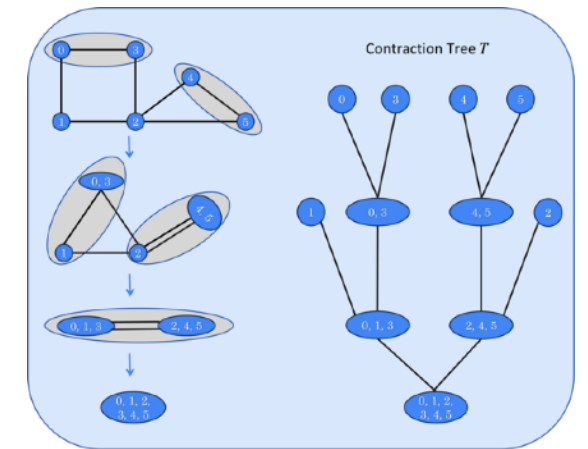
- (super-?)exponentially accurate for large bond dimension D

- Cons

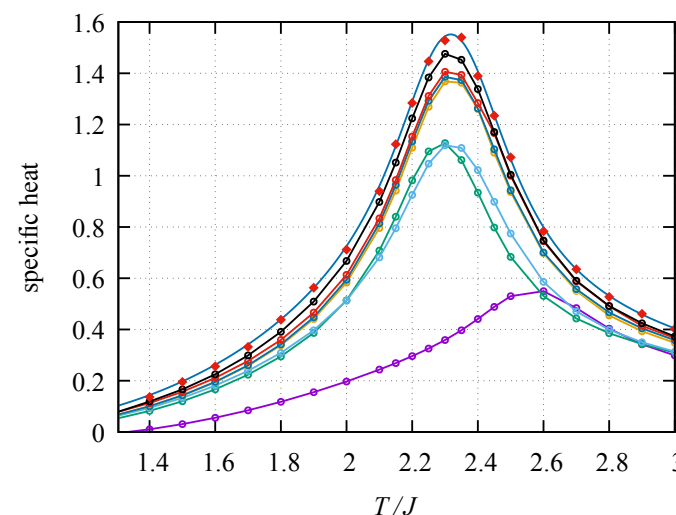
- systematic error (bias) due to finite bond dimension D
- reduction of systematic error (for larger D) is often monotonic
- computational cost increases rapidly as D^α
- few methods for higher dimensions

縮約のデザインとサンプリング

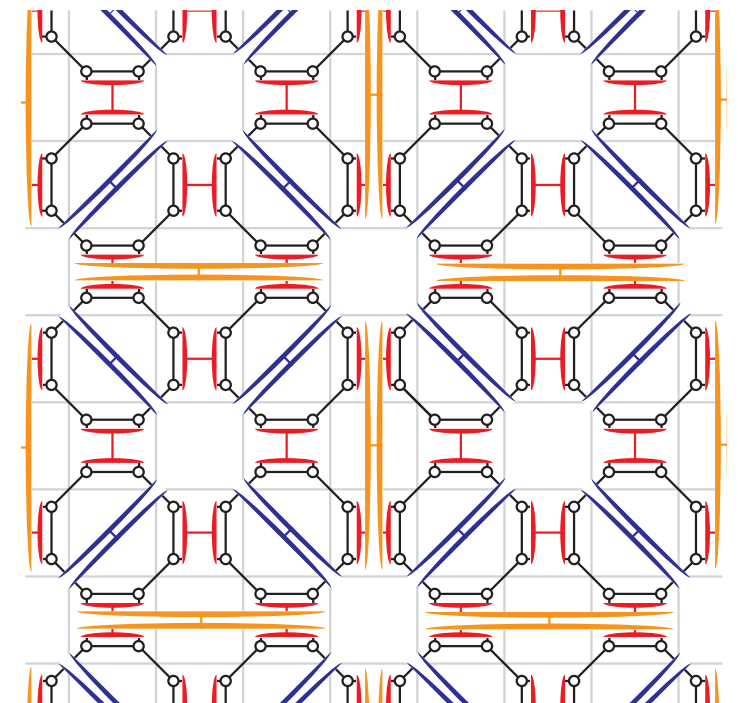
- 近似縮約のバイアスと計算量のコントロール
- 縮約順序の最適化
 - 同じTNでも順序で計算量が激変
 - グラフ理論・ヒューリスティクスで探索
 - HPC/GPU上での並列化設計
- Tensor-Network Monte Carlo (TNMC)
 - 低ランク近似 → 低次元へのプロジェクタの挿入
 - 「最適」のプロジェクタを入れる代わりに「確率的にサンプリング」
 - MCMCにより、低ランク近似による系統誤差を完全に消去



S. Mukherjee, S. Akiyama, arXiv:2510.02725

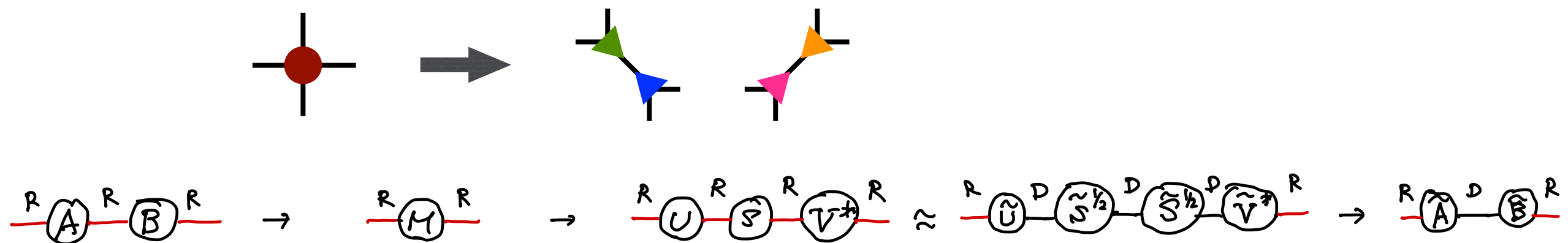


S. Todo, arXiv:2412.02974



Low-rank approximation in tensor network methods

- Low-rank approximation based on SVD
 - choose largest d singular values



• Projector formulation

- choose the best projector to d dimensions
- equivalent to low-rank approximation using SVD

$$R \text{---} (W_L)^D \equiv R \text{---} (B) \text{---} (\tilde{V})^D \text{---} (\tilde{S}^{-1/2})^D$$

$$D \text{---} (W_R^*)^R \equiv D \text{---} (\tilde{S}^{-1/2})^D \text{---} (\tilde{U}^*)^R \text{---} (A)^R$$

$$R \text{---} (A) \text{---} (W_L)^D \text{---} (W_R^*)^R \text{---} (B)^R = R \text{---} (\tilde{A})^D \text{---} (\tilde{B})^R$$

Core idea of tensor-network MCMC

- Random projectors

- instead of choosing the “best” projector, all possible projectors are sampled according to some weights, such that on average (Ferris 2015)

$$\langle W_R(\theta)W_L^*(\theta) \rangle_\theta = \sum_{\theta=1}^{n_c} W_R(\theta)W_L^*(\theta)p(\theta) = I_r$$

- Use Markov chain Monte Carlo to control variance

- as, in random sampling and importance sampling, statistical error diverges exponentially as system size increases

- Use different projector configurations in different positions

- to remove systematic errors from correlation

Markov chain Monte Carlo approach

- Determine projector candidates from SVD during the conventional (deterministic) TRG

- projectors becomes independent with each other and can be sampled **independently**

$$p(\theta_1, \theta_2, \dots, \theta_n) = p(\theta_1)p(\theta_2)\dots p(\theta_n)$$

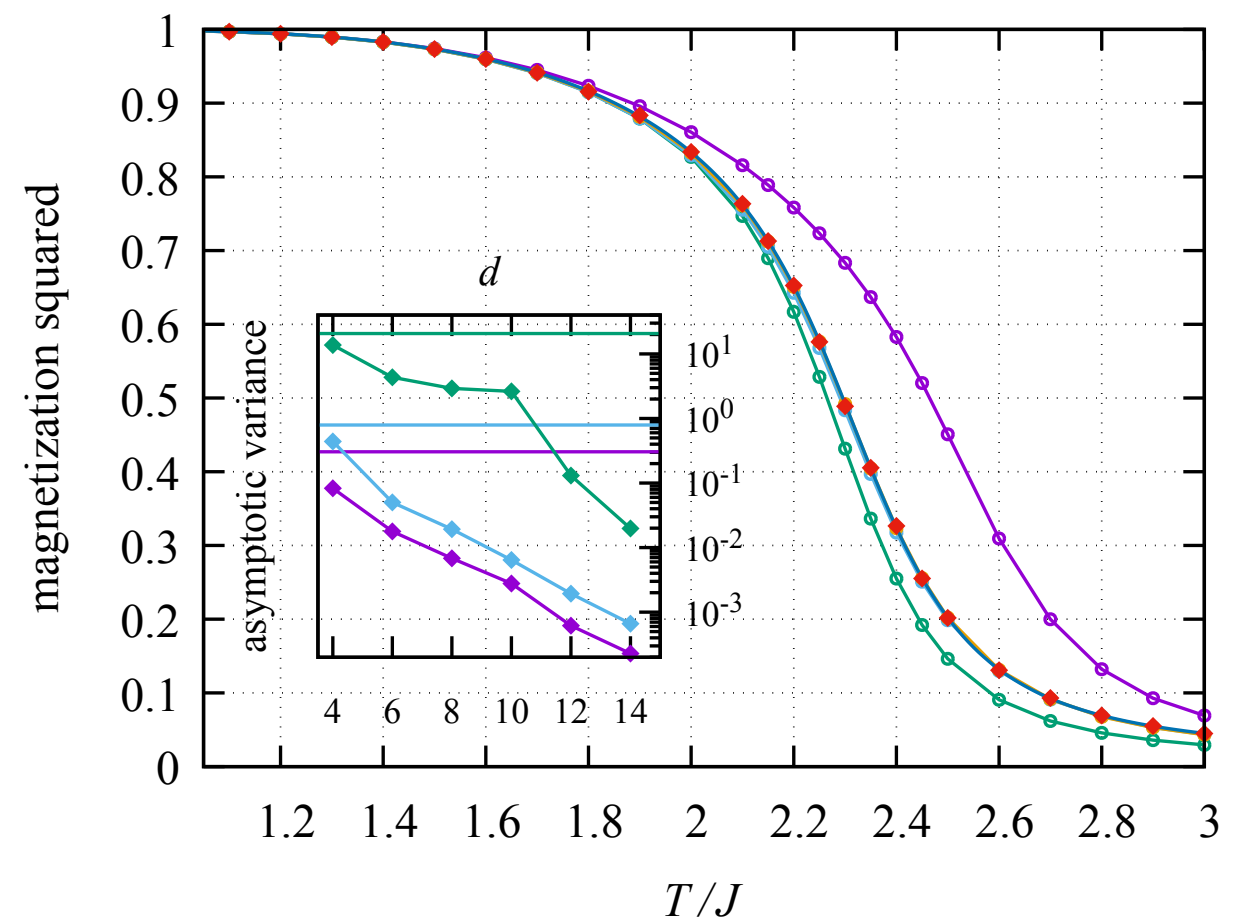
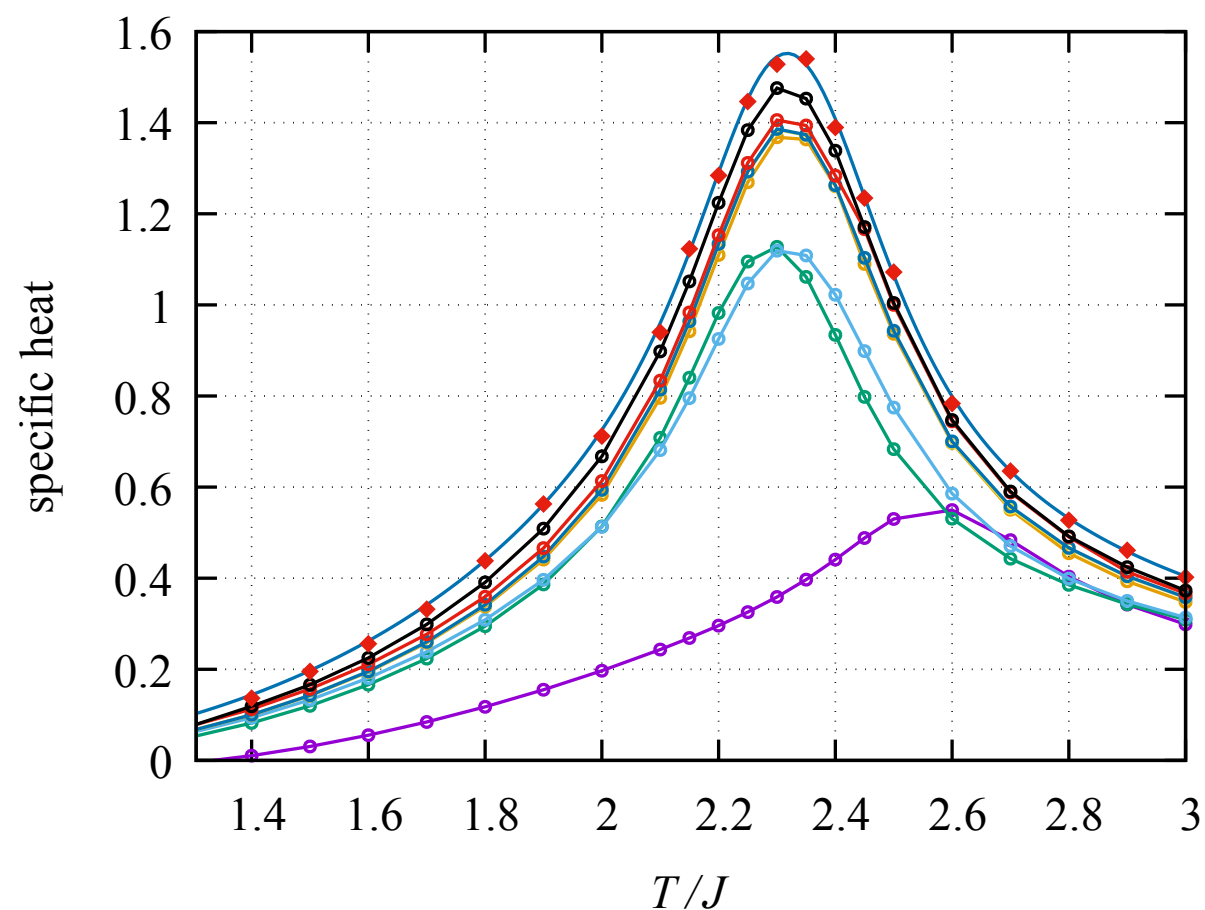
- (exact) tensor network representation of partition function

$$Z = \sum_{\{\theta_i\}} g(\theta_1, \theta_2, \dots, \theta_n) p(\theta_1, \theta_2, \dots, \theta_n) = \sum_{\{\theta_i\}} g(\theta_1, \theta_2, \dots, \theta_n) p(\theta_1) p(\theta_2) \dots p(\theta_n)$$

- Sample projectors $\{\theta_i\}$ using **Markov-chain Monte Carlo**
 - propose new θ_i according to $p(\theta_i)$
 - Metropolis update with $P = \min(1, g(\theta_1, \theta_2, \dots, \theta'_i, \dots, \theta_n) / g(\theta_1, \theta_2, \dots, \theta_i, \dots, \theta_n))$
 - update of weights is $O(\log N)$ and includes matmul only
 - SVDs are required during **initialization stage** only
- Physical quantities

Comparison with Levin-Nave TRG + impurity tensor

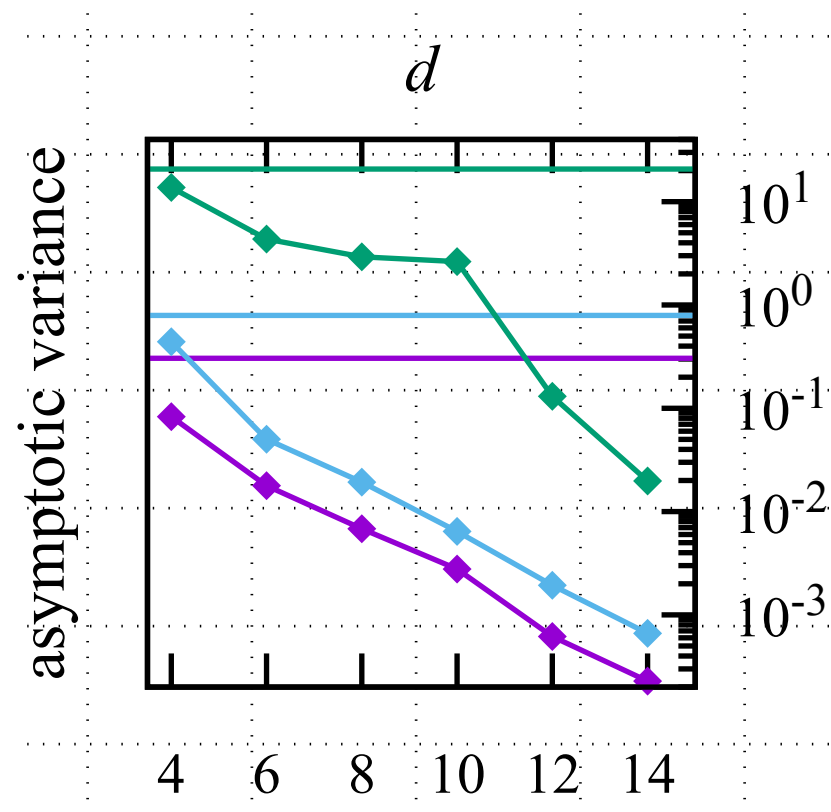
- Square-lattice Ising model ($L = 16$)
 - Exact results (transfer matrix): dark blue
 - TensorMC: $d=6$ (red symbols)
 - Levin-Nave TRG: $d=2$ (purple), 4 (green), 6 (cyan), ..., 16 (black)



Comparison with Metropolis algorithm

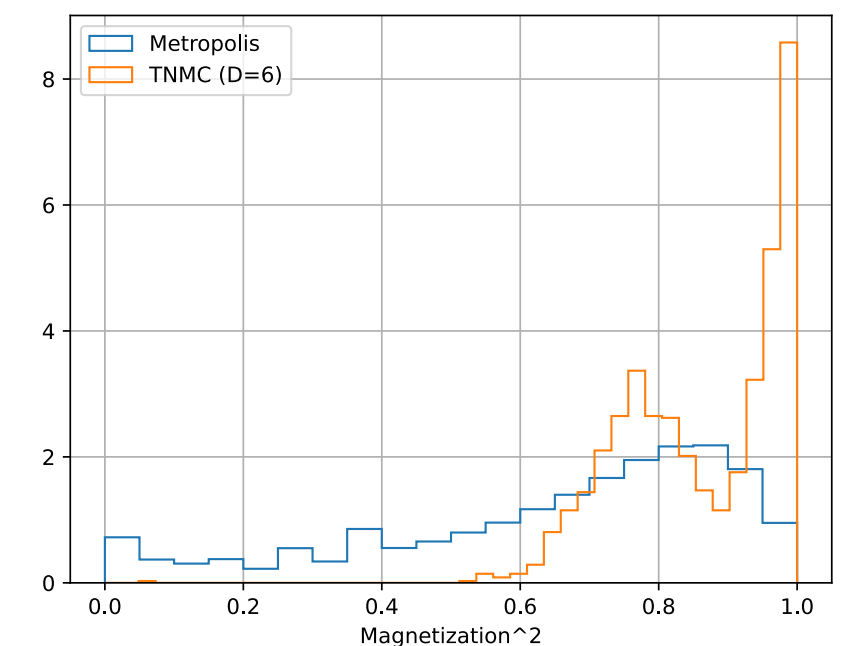
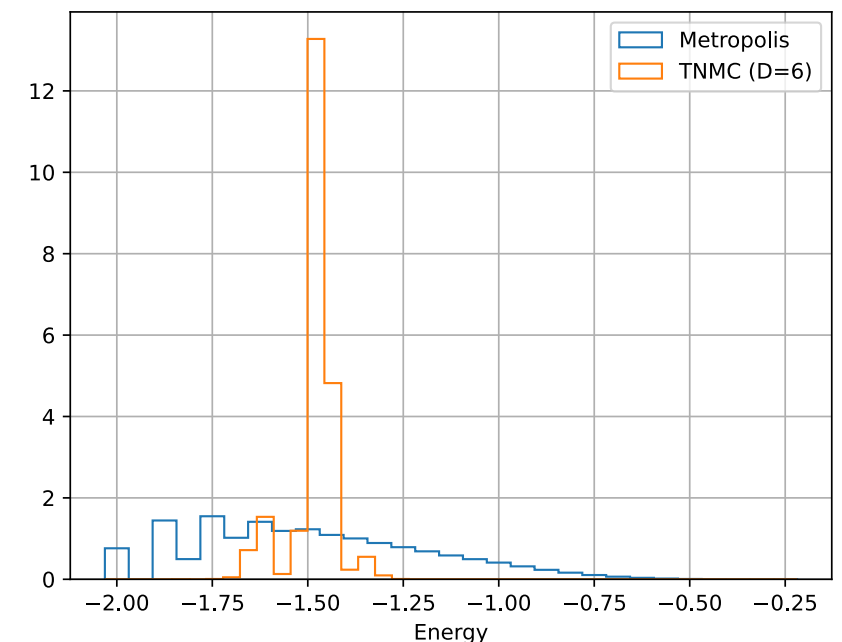
- Square-lattice Ising model

- asymptotic variance is smaller by **orders of magnitude**
- asymptotic variance decreases **exponentially** as d increases



$$\sigma^2 = \frac{2\sigma_0^2 \tau_{\text{int}}}{M}$$

- energy (purple), specific heat (green),



Ising model in imaginary external field

- Square lattice Ising model

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

- pure imaginary external field

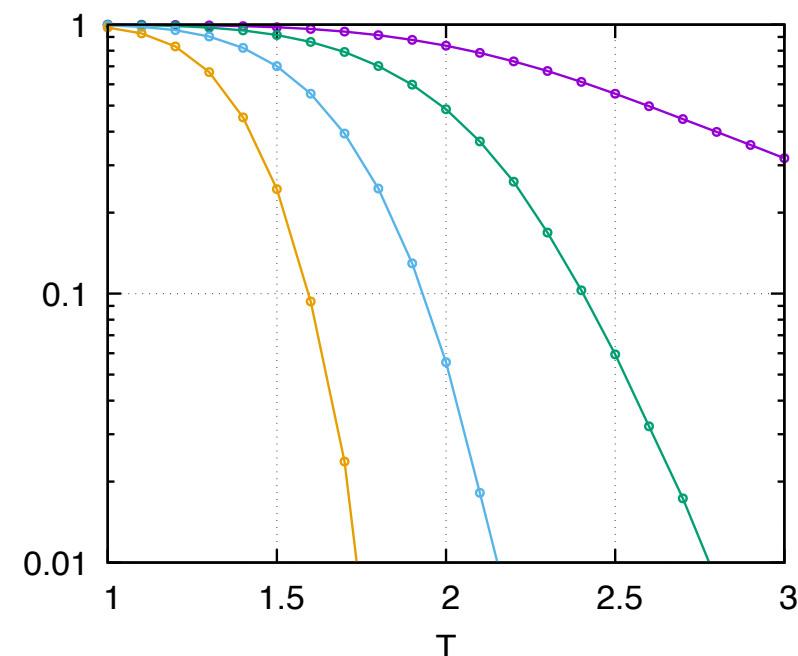
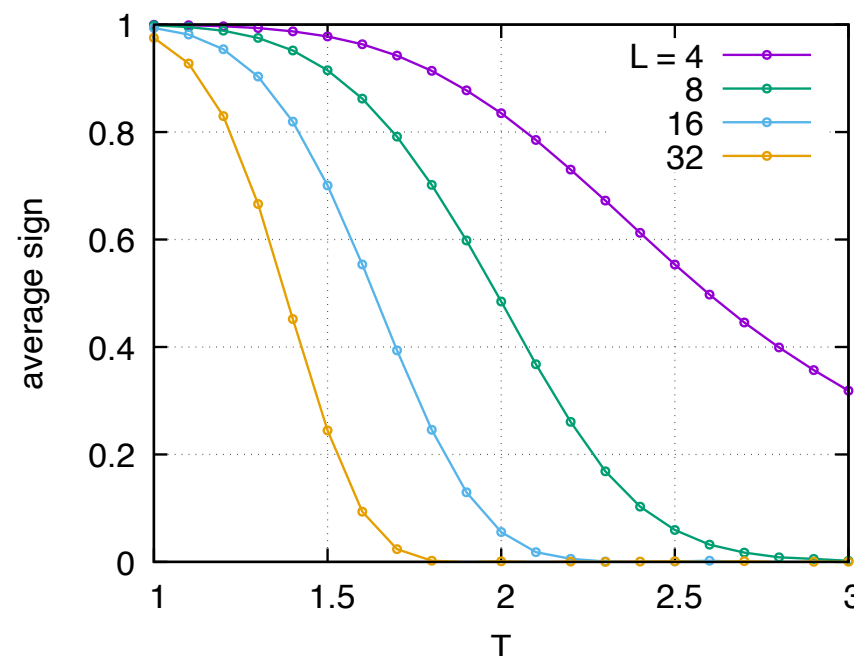
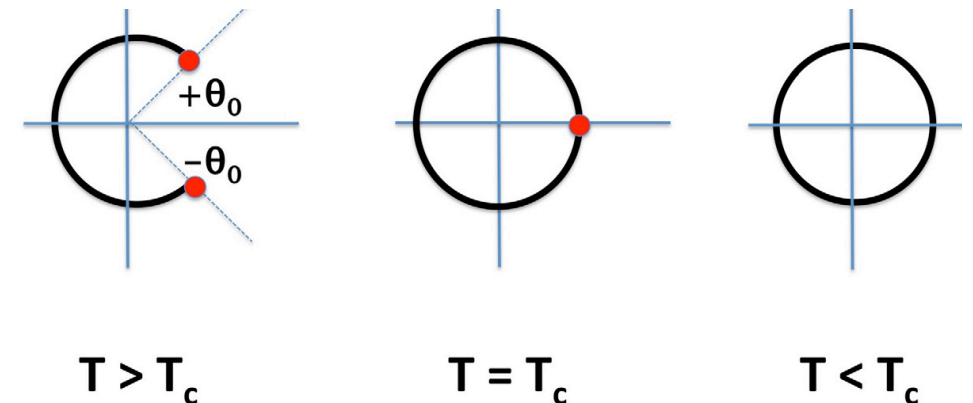
$$h = i\pi/2\beta \quad \Rightarrow \quad z = e^{-2\beta h} = -1$$

- non-positive Boltzmann weight (m : total magnetization)

$$W = e^{\beta \sum \sigma_i \sigma_j} \times (-1)^{m/2} \quad (m: \text{total magnetization})$$

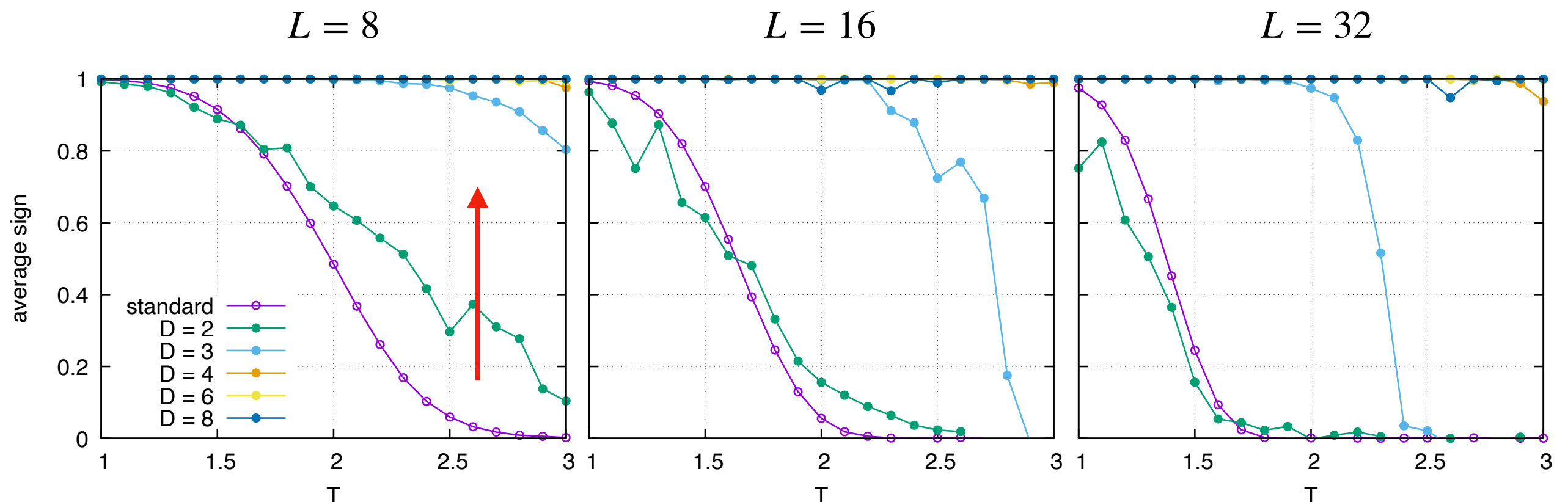
- Standard Markov chain Monte Carlo suffers from severe negative sign problem

Yang-Lee zeros on complex plane of fugacity z



Ising model in imaginary external field

- Our proposed method also **has negative signs** for small d
 - $d = 2$ results are almost similar to the standard method
 - NB: negative signs can appear for small d even if the original model is free from negative sign
- However, **average sign is improved drastically** as we increase d



Summary of Tensor MCMC

- Markov chain Monte Carlo in tensor network representation

- reducing statistical error using approximate tensor network contraction
- removing systematic bias by sampling singular vectors (projectors) using MCMC
 - avoid divergence of statistical error, negative signs, and systematic bias

$$\sigma^2 = \frac{2\sigma_0^2 \tau_{\text{int}}}{M}$$

- Computational complexity of one Monte Carlo update

- $O(d^\alpha N \log N)$
- matmul only (no SVD) during MCMC sampling → ideal for modern GPGPU or HPC

- Combination with various advanced sampling techniques (to reduce τ_{int})

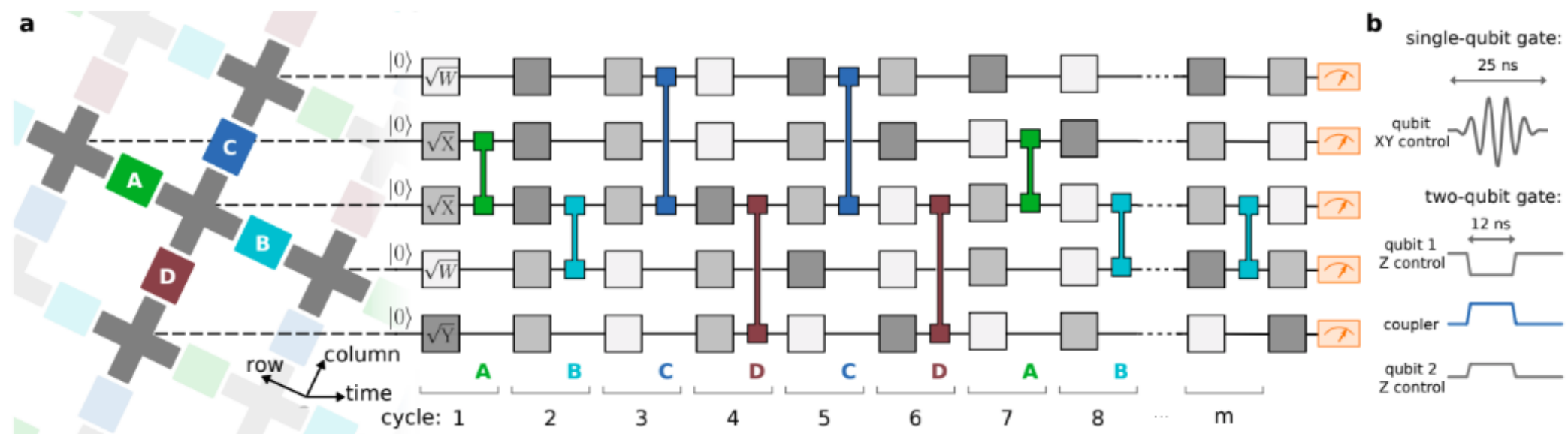
- Applications:

- higher dimensions: HOTRG (2012), ATRG (2020)
- quantum spin models, fermions, quantum circuits, etc

テンソルネットワークから見た量子計算

量子回路

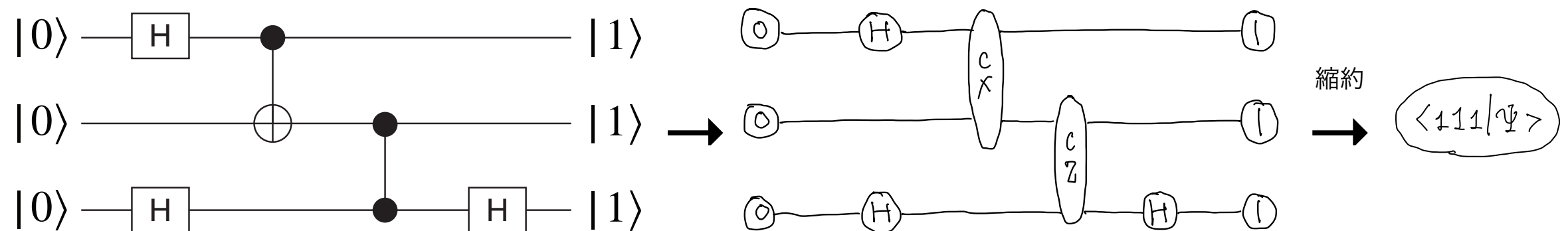
- 量子回路 = 量子ビットに演算するゲート操作の回路図
 - 量子ビット(qubit)
 - $|0\rangle$ と $|1\rangle$ の重ね合わせの状態をとる
 - N -qubitの量子回路 $\rightarrow N$ 個の $S = 1/2$ スピン
 - ヒルベルト空間の次元: 2^N
- 量子計算 = 重ね合わせ + 並列性 + 干渉 + 測定による収縮



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

量子回路シミュレータ

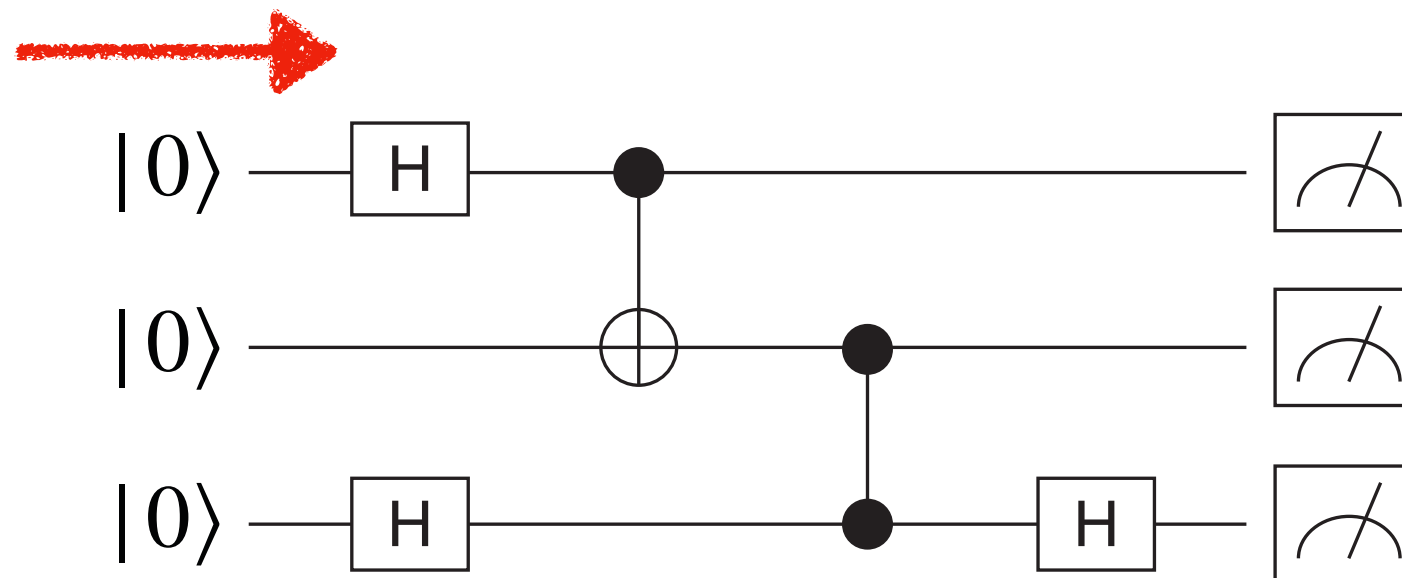
- 任意の量子回路はテンソルネットワークとして表現可能



- 全てのボンド次元は2 (qubitの場合)
- 1量子ビットゲート $\rightarrow$ 2脚テンソル、2量子ビットゲート $\rightarrow$ 4脚テンソル
- 多制御量子ゲート(CCCXなど) $\rightarrow$ ボンド次元2のMPO (行列積演算子)
- 出力ビット列の確率・振幅 = TN縮約の結果
- 利点
 - 縮約の順序を変えても、最終結果は変わらない
 - 縮約順序には自由度がある $\rightarrow$ 工夫次第で計算コストを削減できる可能性がある
- 課題
 - 「縮約順序の最適化」と「中間テンソルのサイズ」がボトルネック

Schrödinger シミュレーション

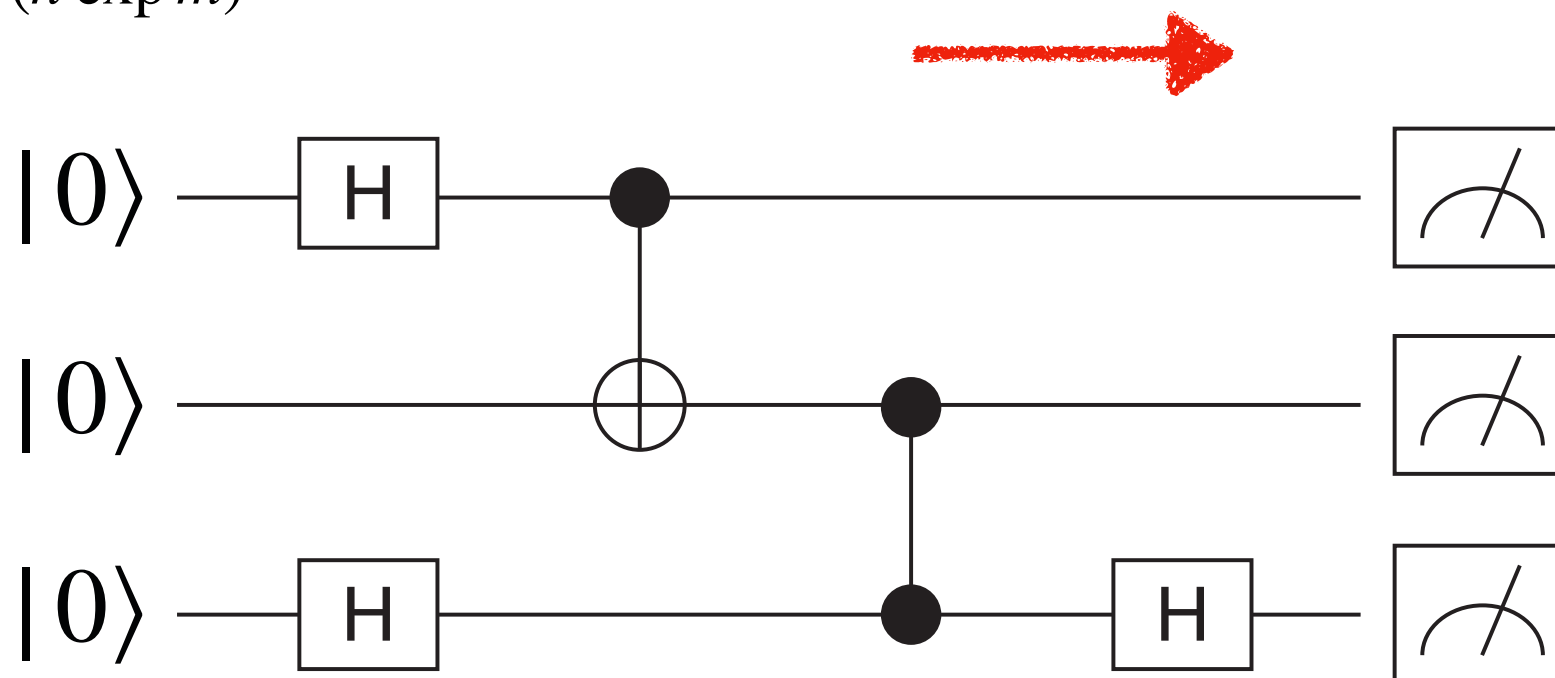
- 波動関数の係数(状態の重ね合わせ係数)をベクトルとして保持する
 - n 量子ビット → サイズ 2^n の複素ベクトル
 - 量子ゲート → $2^n \times 2^n$ ユニタリ行列 (ただし通常は非常に疎)
 - ゲート操作 → 行列ベクトル積
- シミュレーションコスト
 - n 量子ビット, m ゲート
 - メモリコスト: $O(\exp n)$ $n = 50 \rightarrow 9$ PB (単精度複素数)
 - 計算量: $O(m \exp n)$ $n = 50, m = 1000 \rightarrow 1$ EFLOP



- テンソル表現で疎行列を効率的に扱える

Feynman シミュレーション

- すべての「経路」からの寄与を足し合わせる
 - 量子力学の経路積分に対応
 - Hadamard のような分岐ゲートを通るたびに、異なる経路へ分岐する
 - 各経路の寄与(重み)は行列要素の積で与えられる
- シミュレーションコスト
 - n qubits, m 個の分岐ゲート
 - 経路数: $O(\exp m)$
 - メモリコスト: $O(n)$ (何を測るかに依存)
 - 計算量: $O(n \exp m)$



Feynman サンプリング

- 別名 Quantum Monte Carlo

- 各分岐において, 行列要素に比例した確率で次状態を確率的に選ぶ

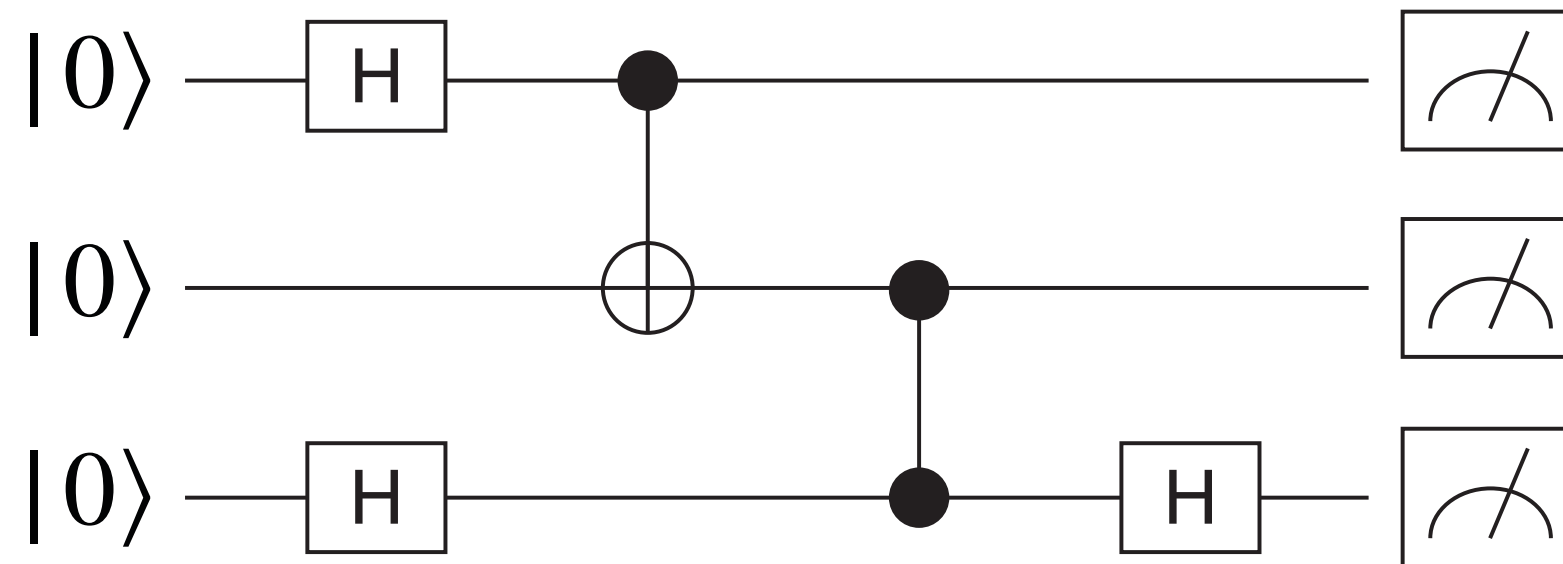
- シミュレーションコスト

- n qubits, m 個の分岐ゲート
 - 1サンプルあたりの計算量: $O(m)$
 - 総計算量: $O(m) \times (\text{サンプル数})$

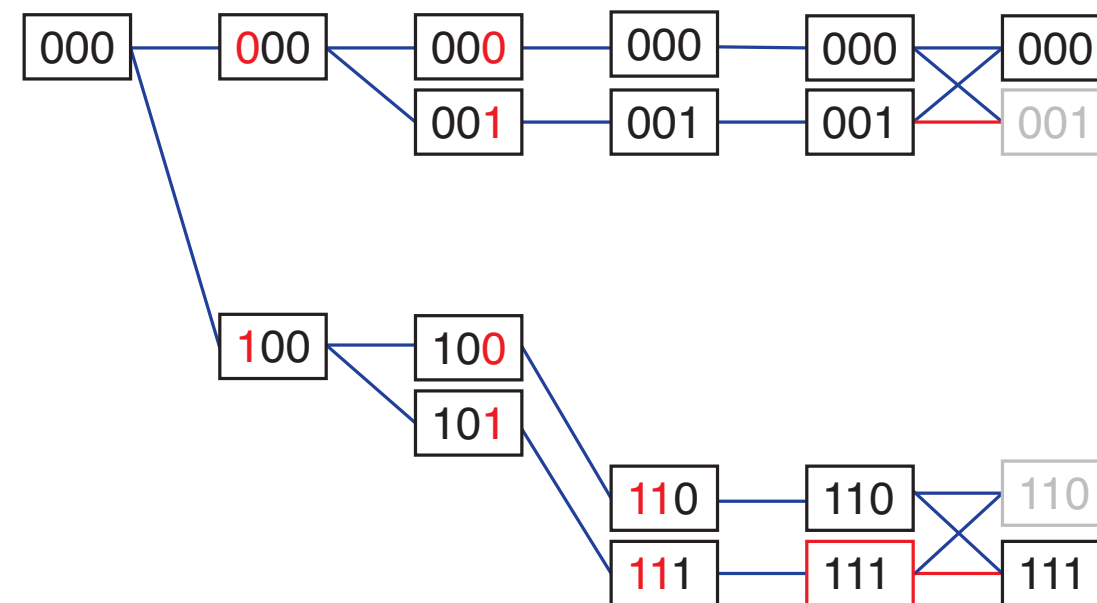
- 符号問題

- 行列要素が負(あるいは複素数)のときどうするか?
 - 行列要素の絶対値を「重み」として使う
 - 符号(または位相)は別に保持する
 - 符号が混ざると強い相殺が起きる
 - 回路が深くなると, ほとんどランダムサンプリングになってしまう
 - 必要サンプル数: $O(\exp m)$

量子回路の経路サンプリング



• 状態遷移図



- 干渉によって多くの状態が消える → 負符号(複素位相)問題

よりモダンな手法

- テンソルネットワークに基づく方法

- MPS (TEBD)

- 一次元的な回路・エンタングルメントが小さい場合に強い
 - 長距離相互作用がある場合、回路が深い場合は破綻する

- (狭義の)テンソルネットワーク法

- 量子回路全体をテンソルネットワークと見立て、縮約を行う
 - 縮約順序が性能を大きく左右する
 - 幅が狭い回路・浅い回路で有効

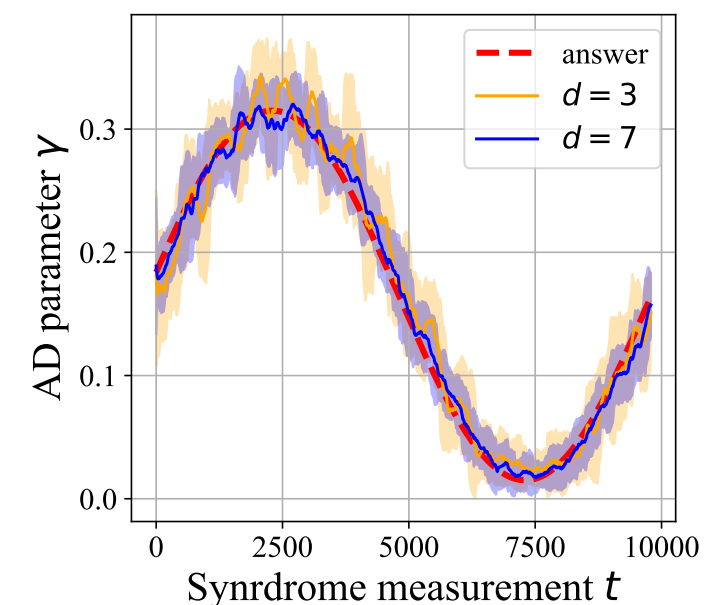
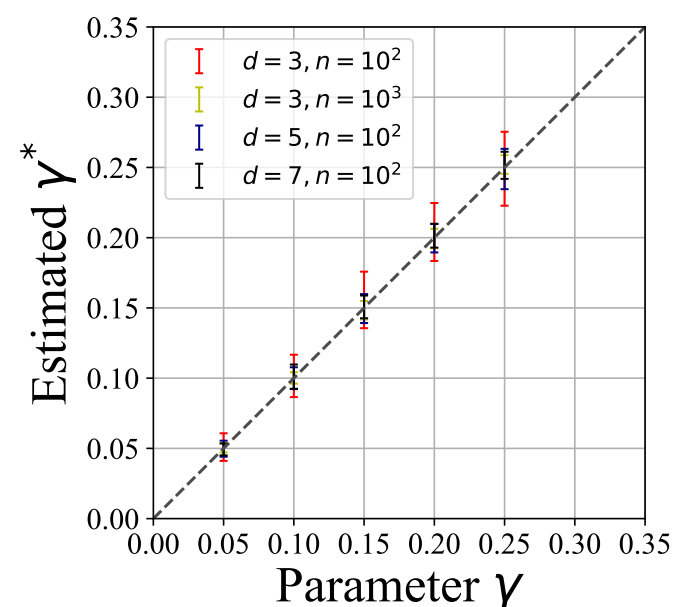
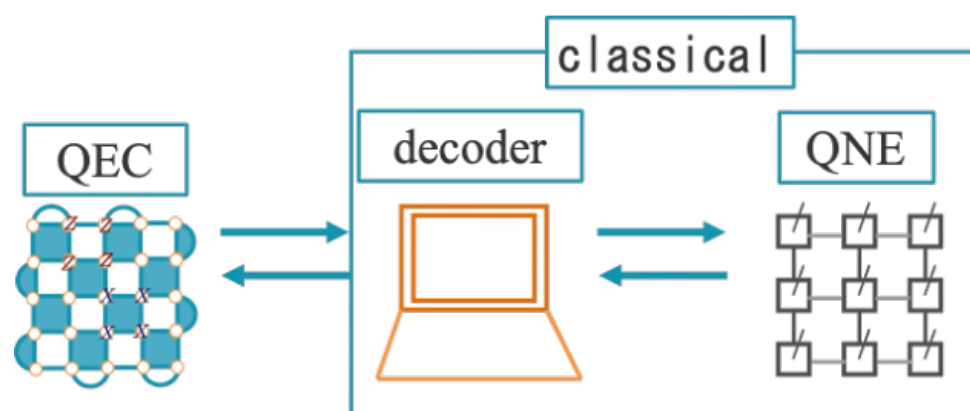
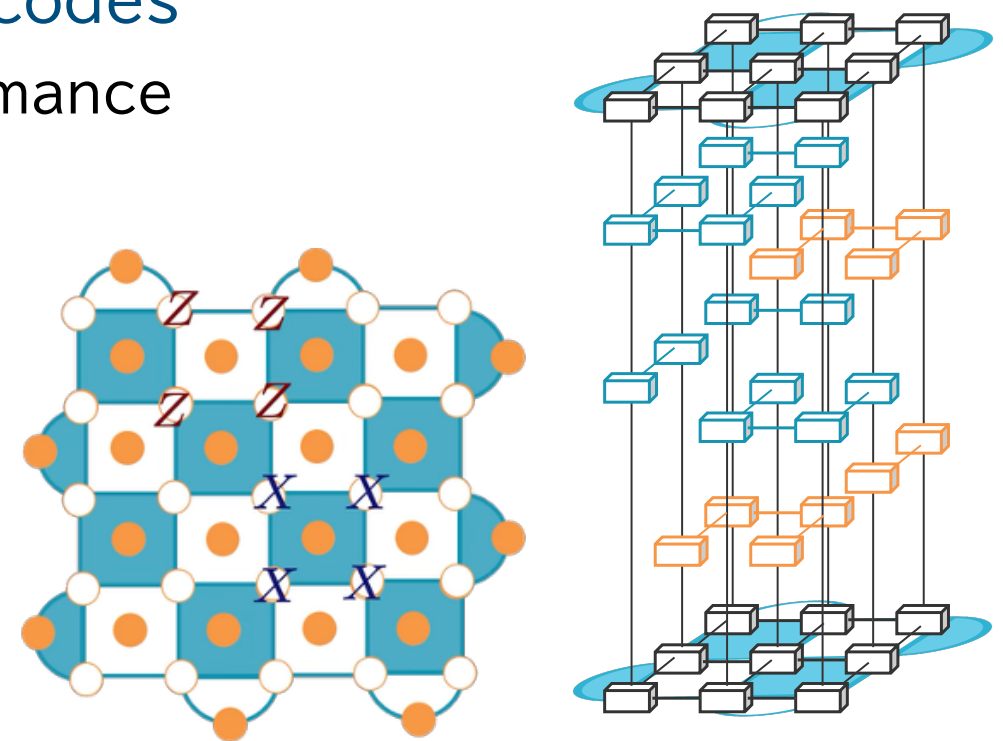
- スタビライザー表現に基づく方法

- Cliffordシミュレータ

- 非常に高速
 - 限られた種類の回路(=Clifford回路)しか扱えない → Clifford+Tシミュレータ

Tensor Network Decoder for Error Correction

- Decoding stabilizer code, such as surface codes
 - Noise model information can improve performance
- Bayesian inference of noise model
 - Using syndrome measurement results only
 - syndrome measurement + TN likelihood calculation + Bayesian inference + Monte Carlo sampling
- TNs can improve decoding and noise characterization for fault-tolerant quantum computing (FTQC)



Kobori and Todo (2025)

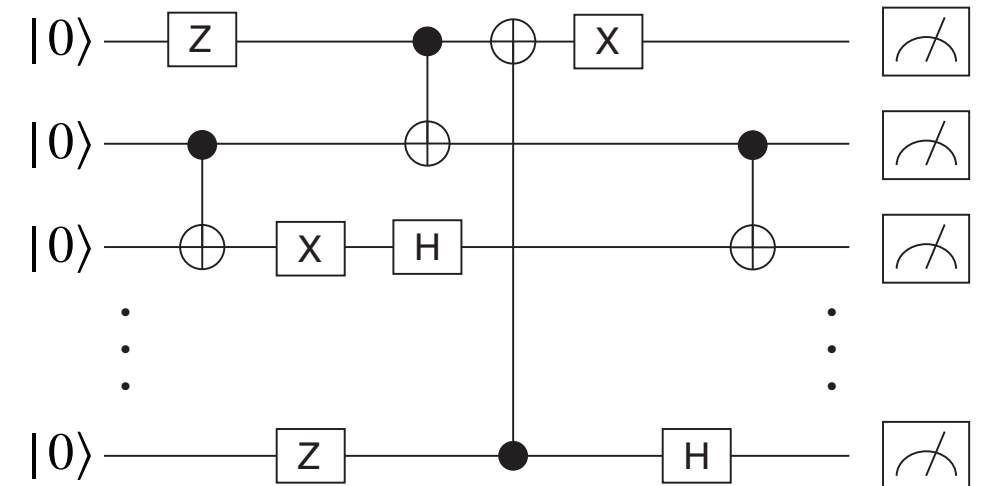
量子回路への埋め込み: 利点と制約

• 発想

- テンソルネットワークが表す量子状態や線形写像を、「量子回路+測定」として実装する

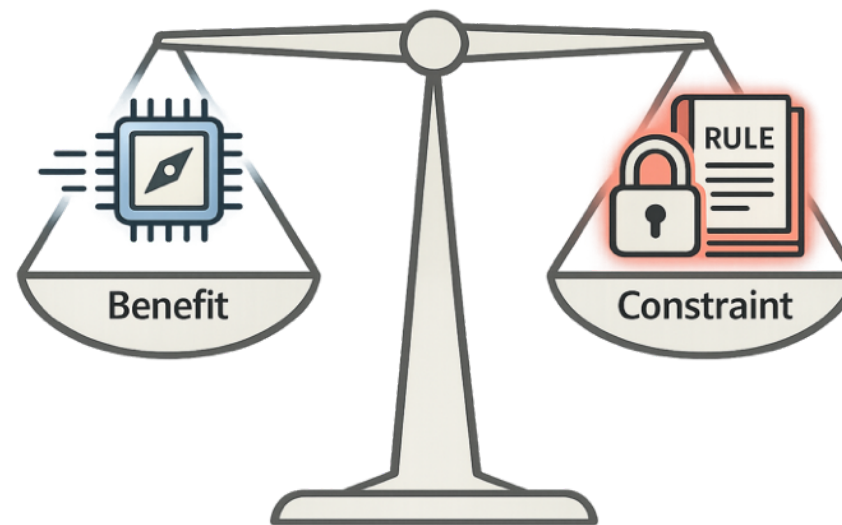
• 量子回路による量子計算

- 状態準備→測定→サンプル生成(ビット列)
- 量子回路は、確率的にビット列を吐き出す「サンプリングマシン」と見なせる



• 「ユニタリ制約」

サンプル生成を量子回路+測定で
高速に実行可能
(古典計算のボトルネック回避)

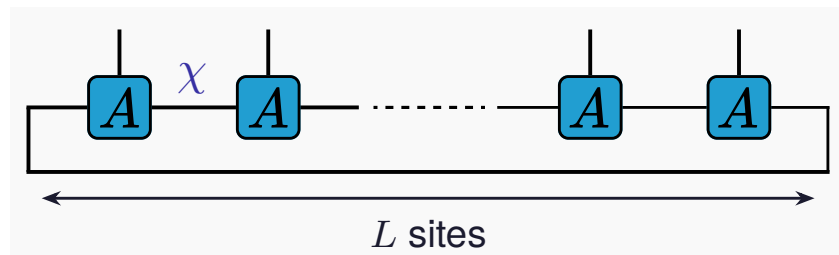
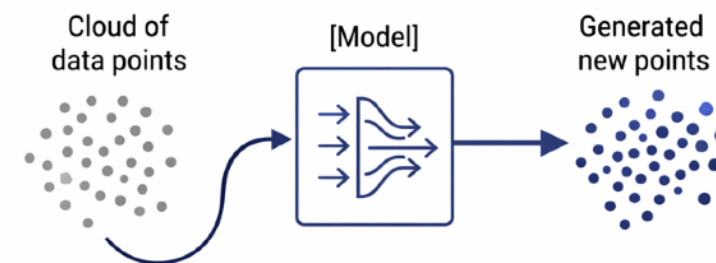


量子ゲートは「ユニタリ」
一方、一般のテンソルはユニ
タリとは限らない

• 一般のテンソルネットワーク

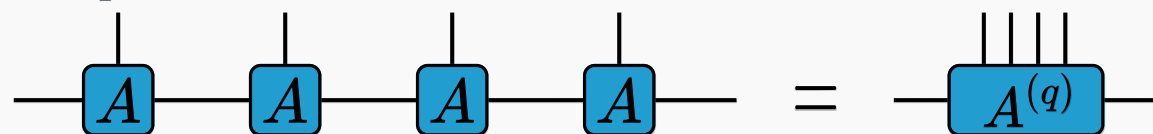
- 非ユニタリな行列を含むため、単純な変換では量子回路に埋め込めない

MPSの厳密な対数深さ状態準備

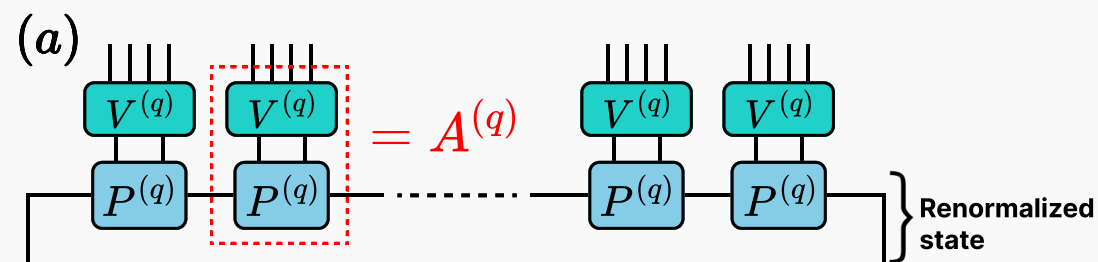
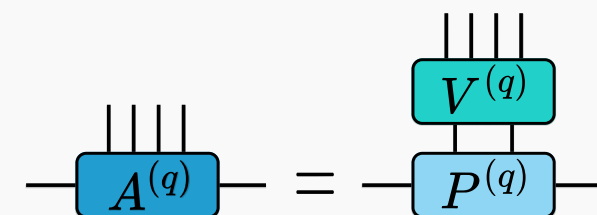


- 長さ L 、ボンド次元 χ のMPSを、量子コンピュータ上にできるだけ浅い回路で準備する

Block q sites into one tensor $A^{(q)}$



Polar-decompose it

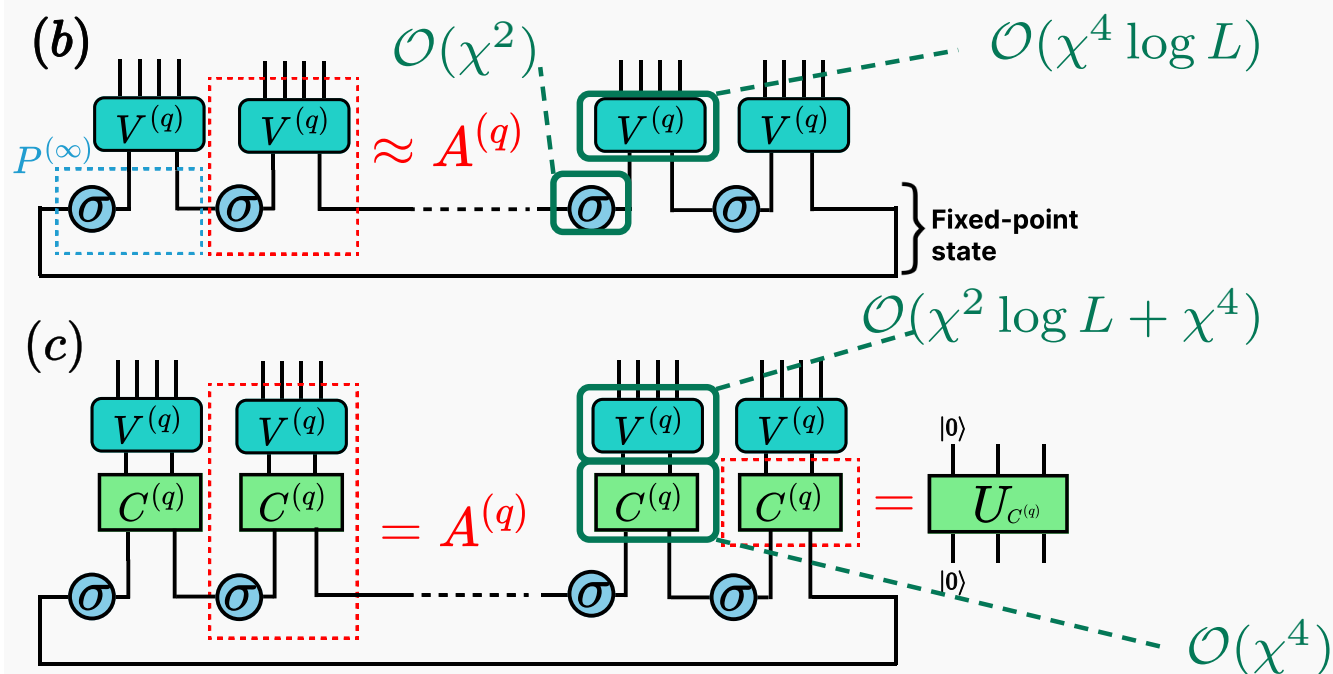


(b): 先行研究

- 極分解であらわれる繰り込み部分 P をブロック長 q が無限の場合で近似
- 深さ $\sim O(\chi^4 \log(L/\epsilon))$

(c): 新手法

- 有限 q による繰り込み部分 P の誤差を修正する写像 C を導入
- C をブロックエンコーディングにより確率的に埋め込む。ポストセレクションの成功率は L 無限の極限でも有限にとどまる
- 深さ $\sim O(\chi^2 \log L + \chi^4)$



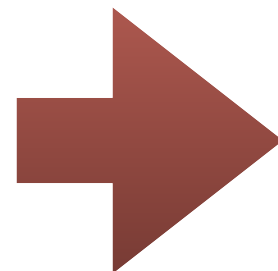
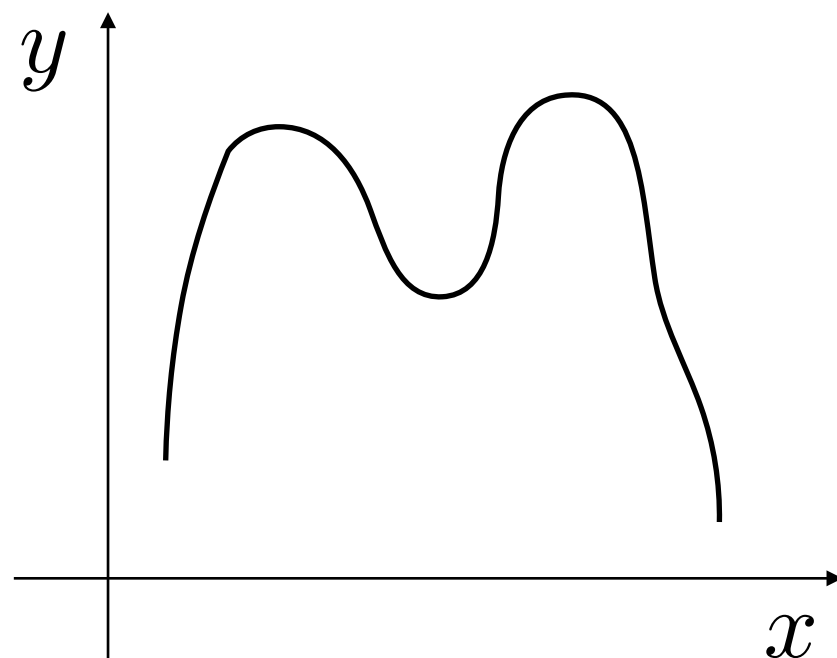
テンソルネットワークによる関数表現と機械学習

滑らかな関数の離散化

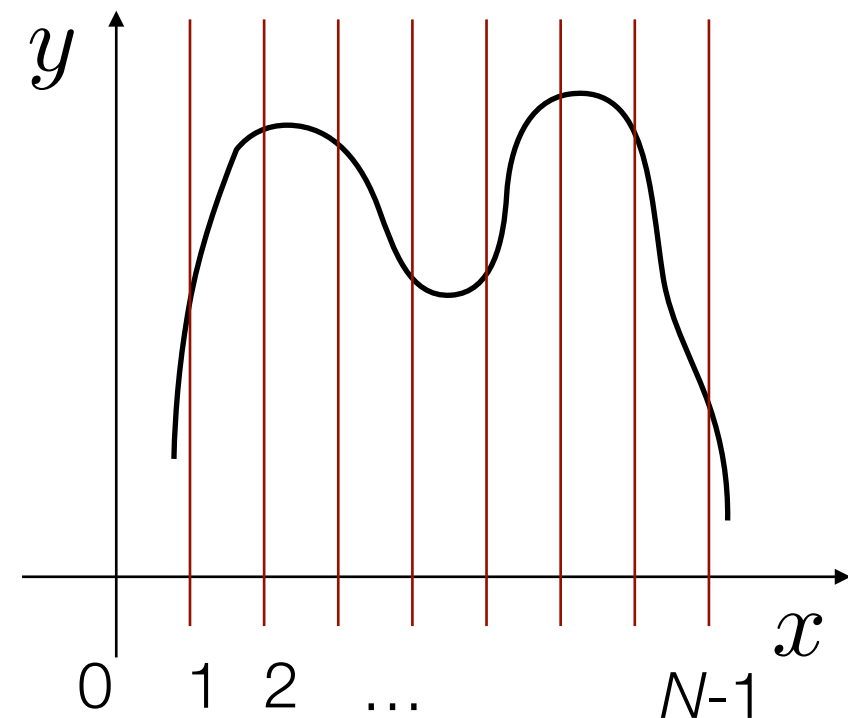
$$y = f(x) \quad x \in [0, 1)$$

$$x_i = \frac{i}{N}, i = 0, 1, \dots, N - 1$$

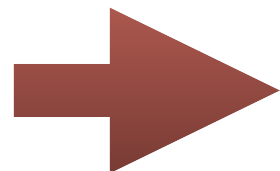
$$y_i = f(x_i)$$



離散化



離散化された関数値のテーブル y_i は N 成分ベクトル

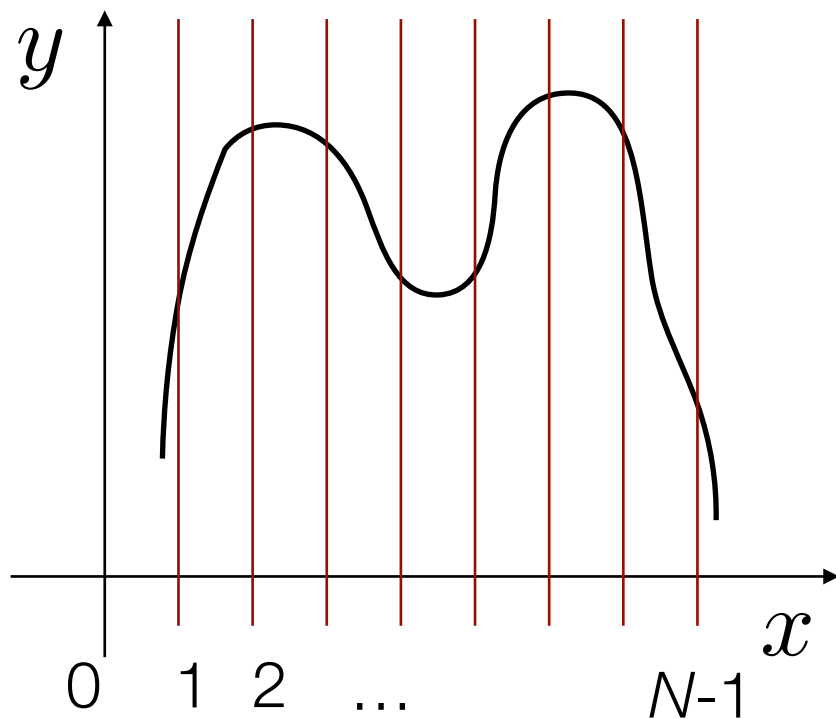


関数の情報圧縮？

関数のテンソル化

$$x_i = \frac{i}{N}, i = 0, 1, \dots, N - 1$$

$$y_i = f(x_i)$$

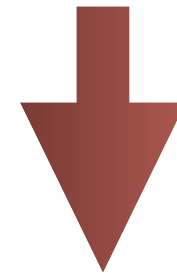


インデックス i の二進数表示

$$N = 2^M$$

$$i = i_0 2^{M-1} + i_1 2^{M-2} + \dots + i_{M-2} 2^1 + i_{M-1}$$

$$i_m = \{0, 1\}$$



$$x_i = x_{i_0, i_1, \dots, i_{M-1}} = \sum_{m=0}^{M-1} \frac{1}{2^{m+1}} i_m$$

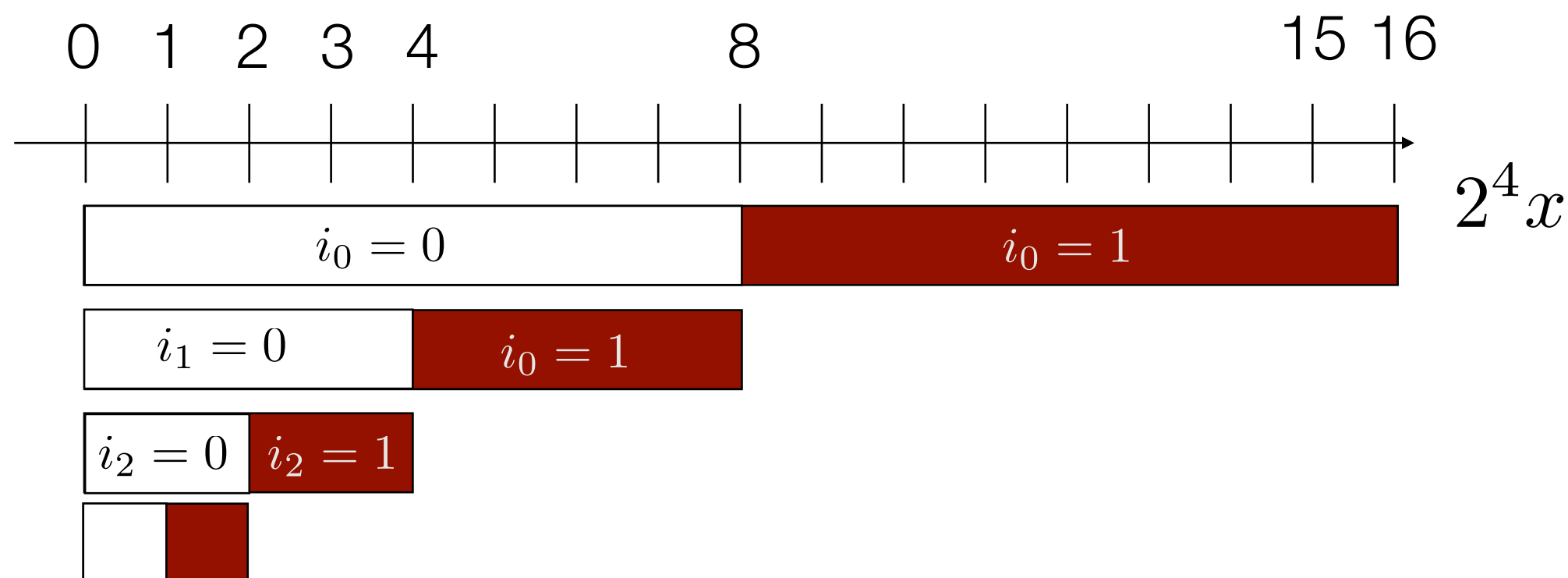
$$y_i = f(x_i) = y_{i_0, i_1, \dots, i_{M-1}}$$

2^M グリッド上の関数 = M 本脚のテンソル

二進数表記と長さスケール

$$x \in [0,1) \text{ の } M \text{ ビット二進数表記 : } x_i = x_{i_0, i_1, \dots, i_{M-1}} = \sum_{m=0}^{M-1} \frac{1}{2^{m+1}} i_m$$

各ビットは“長さスケール”に対応している



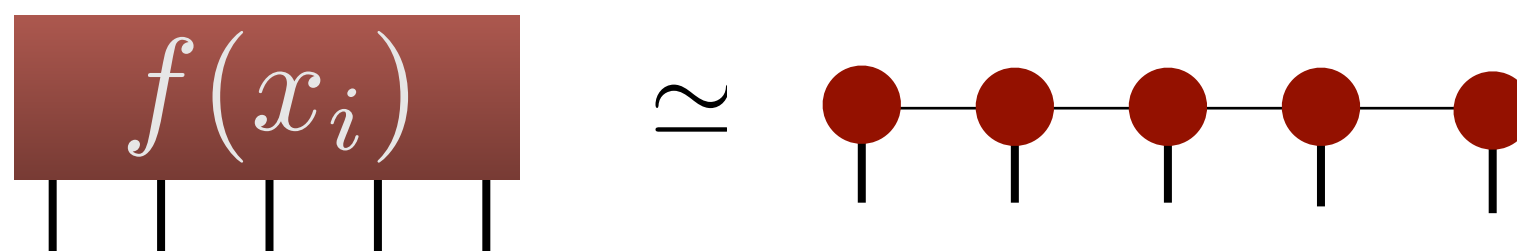
m 番目のビットの変化 = 長さ $\frac{1}{2^{m+1}}$ の変化

関数のテンソルネットワーク近似

$$x_i = x_{i_0, i_1, \dots, i_{M-1}} = \sum_{m=0}^{M-1} \frac{1}{2^{m+1}} i_m$$

$$y_i = f(x_i) = y_{i_0, i_1, \dots, i_{M-1}}$$

長さスケールを意識したテンソルネットワーク = MPS



Quantized/Quantics Tensor Train (QTT)

$N = 2^M$ の情報を $O(M = \log_2 N)$ に圧縮

指数関数的な圧縮の可能性

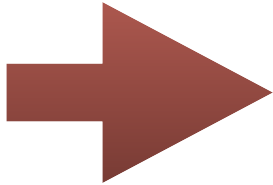
I. V. Oseledets, Dokl. Math. **80**, 653 (2009).

B. N. Khoromskij, Constr. Approx. **34**, 257 (2011).

I. V. Oseledets, Constr. Approx. **37**, 1 (2013)

QTT表現の例 1: 指数関数

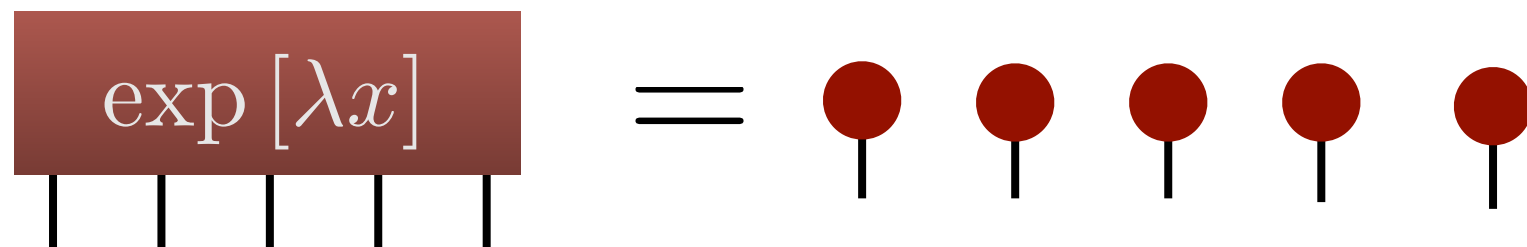
指数関数 : $f(x) = \exp[\lambda x]$


 $f(x_i) = \exp \left[\lambda \sum_m \frac{1}{2^{m+1}} i_m \right] = \prod_{m=0}^{M-1} \exp \left[\lambda \frac{1}{2^{m+1}} i_m \right]$

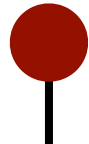
二進数表記

インデックス毎
の指数関数の積

= 量子系の積状態



指数関数 = ボンド次元 1 の QTT


 $= \exp \left[\lambda \frac{1}{2^{m+1}} i_m \right]$

QTT表現での微分

$$y = f(x) \rightarrow \text{微分} : \frac{dy}{dx} = \frac{d}{dx} f(x)$$

$$\text{微分の差分近似 : } \frac{dy}{dx_i} \simeq \frac{y_{i+1} - y_{i-1}}{2\Delta x} \quad \Delta x = \frac{1}{N}$$

(中心差分)

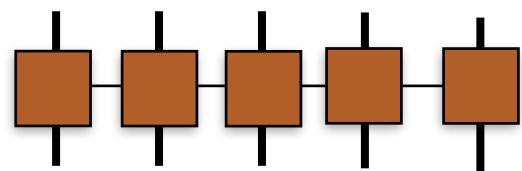
$$= \frac{1}{2\Delta x} \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & -1 \\ -1 & 0 & 1 & 0 & & \vdots \\ 0 & -1 & 0 & 1 & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 1 & \dots & 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ y_{N-1} \end{pmatrix}$$

QTTでの効率的な微分計算？ *周期境界条件

QTT表現での微分: 行列積演算子

$$\frac{dy}{dx_i} \simeq \frac{1}{2\Delta x} \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & -1 \\ -1 & 0 & 1 & 0 & & \vdots \\ 0 & -1 & 0 & 1 & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 1 & \dots & 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{pmatrix}$$

行列積演算子
(MPO)

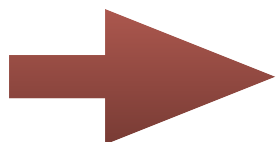
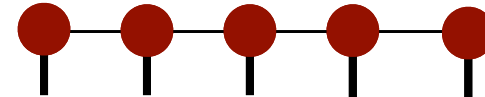


*ボンド次元=3

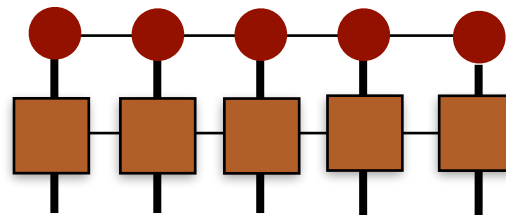
QTT



$\simeq$



$$\frac{dy}{dx_i} \simeq$$

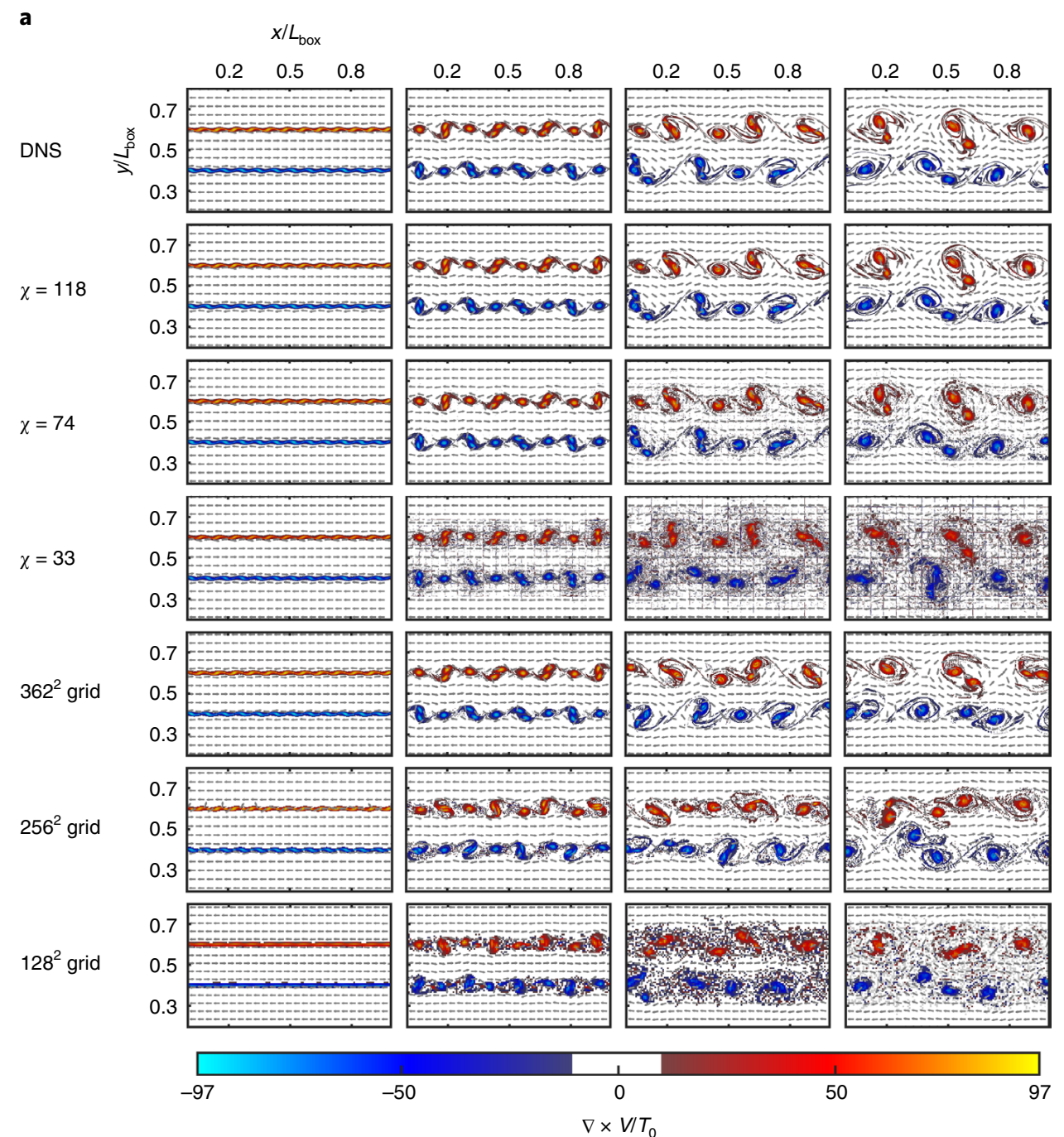


$O(M = \log_2 N)$
で微分が計算可能

偏微分方程式のシミュレーション

場のテンソルネットワーク表現

- 流れの場、密度の場
 - 同次多項式表現 (quantics tensor train (QTT) representation)
 - 長い空間スケールと短い空間スケールの分離 (低ランク近似)
 - 乱流のように無数の長さスケールが混合場合に有効
-
- テンソルネットワークで圧縮したままでシミュレーションを実行
-
- テンソルネットワークを使うことで、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



Quantum-inspired TN Option Pricing

- Accelerating option pricing with tensor networks

- Traditional Monte Carlo: $V = \mathbb{E}[f_{\text{pay}}(S)]$
- Challenges:
 - Millions of simulation paths
 - Expensive Greek calculations
 - Repeated evaluations during risk analysis

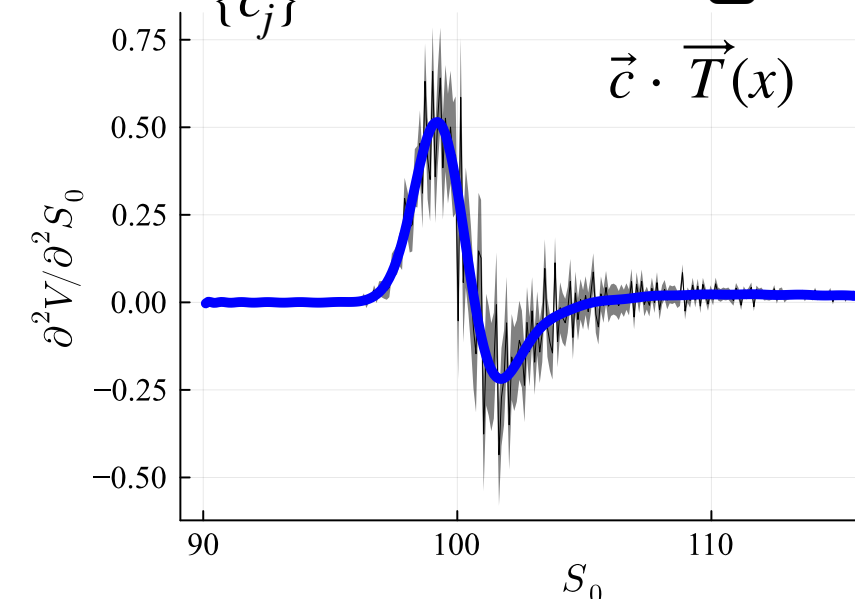
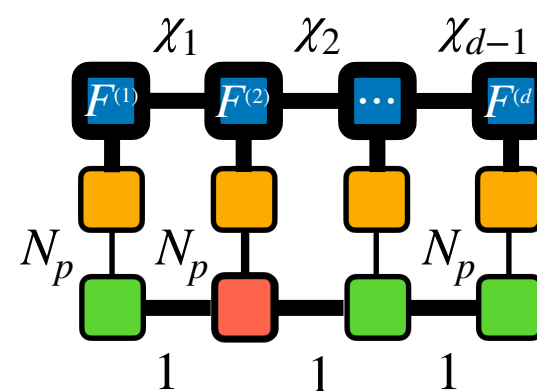
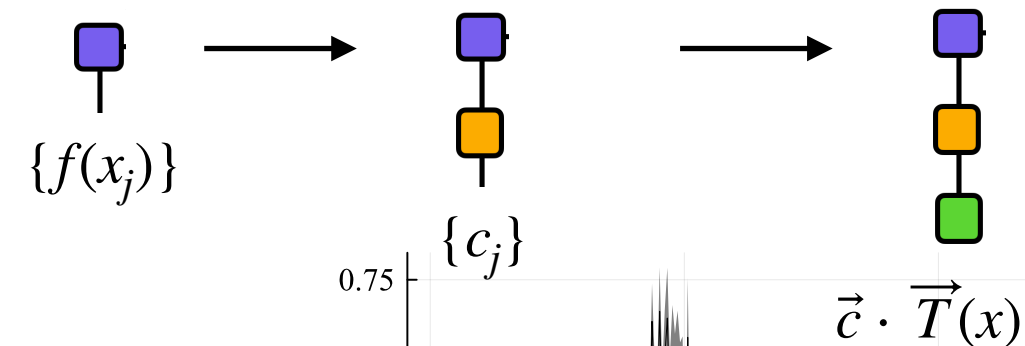
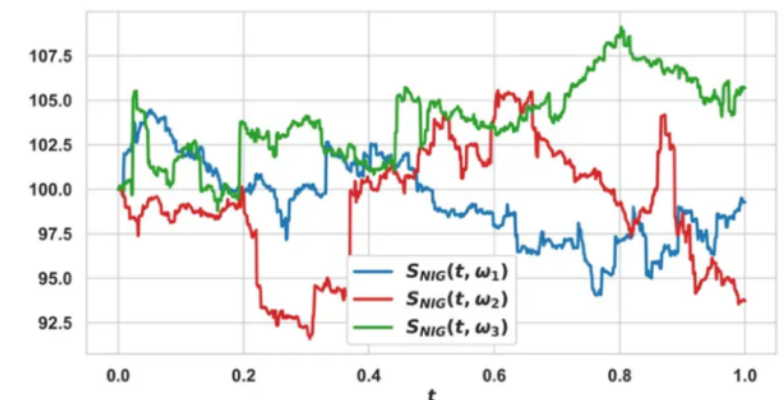
- Our tensor-network based approach

- Chebyshev interpolation
- Tensor cross interpolation
- Analytical differentiation

- Quantum-Inspired Risk Analysis

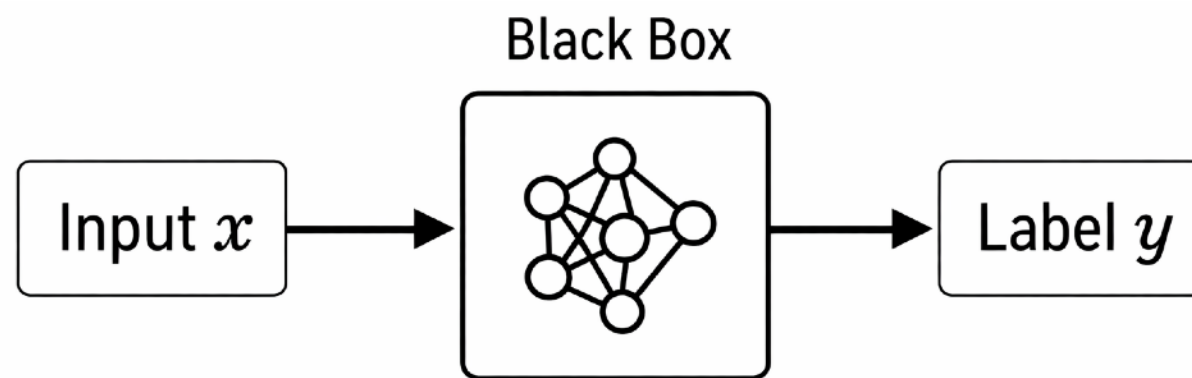
- Asian barrier option
- Multiple market parameters
- Greeks and sensitivities

- TNs provides reusable surrogate model, enabling fast evaluation of prices and Greeks across many market scenarios



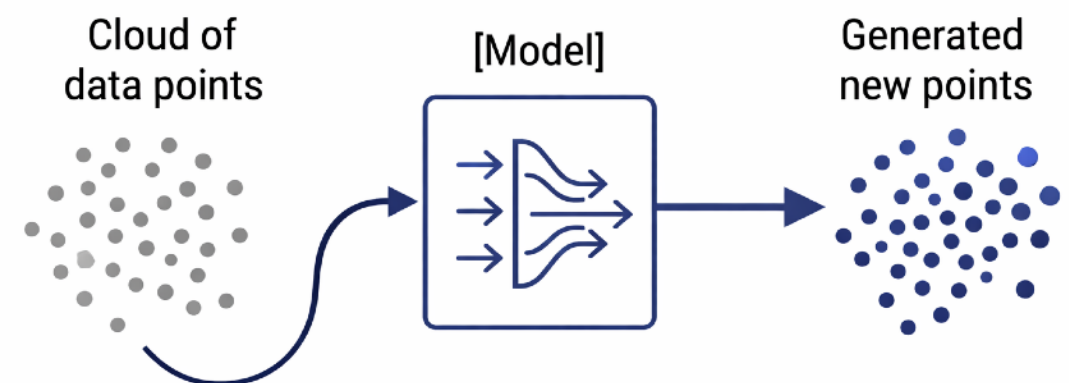
教師あり学習と教師なし学習

教師あり(supervised)



入力からラベルを予測(識別)

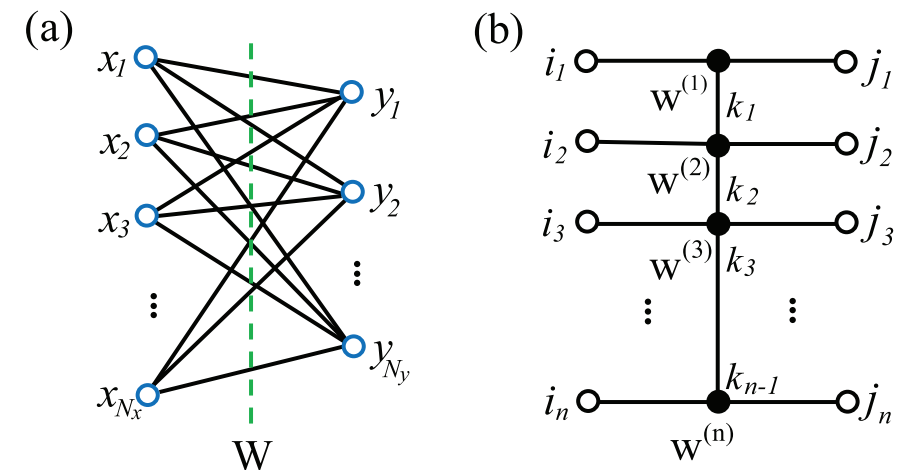
教師なし(unsupervised)



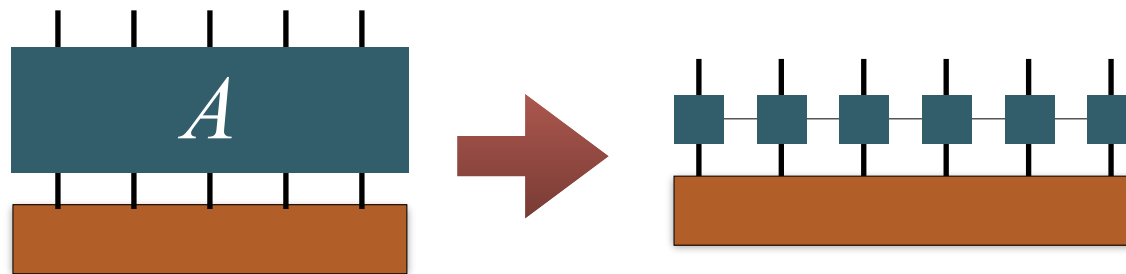
データ分布そのものを学ぶ

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ることでコストが低下



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



- 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
 - 過学習の抑制、汎化性能の向上も期待できる

教師なし学習 - 生成モデル

• 手書き数字画像



$$\vec{v} \in \mathbb{V} = \{0, 1\}^{\otimes N}$$

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

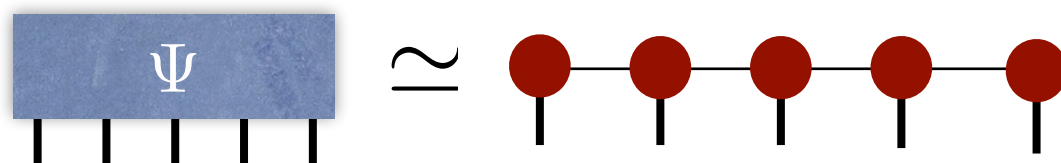
- 数字として意味のある画像の確率分布: $P(\vec{v})$
- Born機械(Born-machine)表現

- 確率分布を量子状態と対応づける

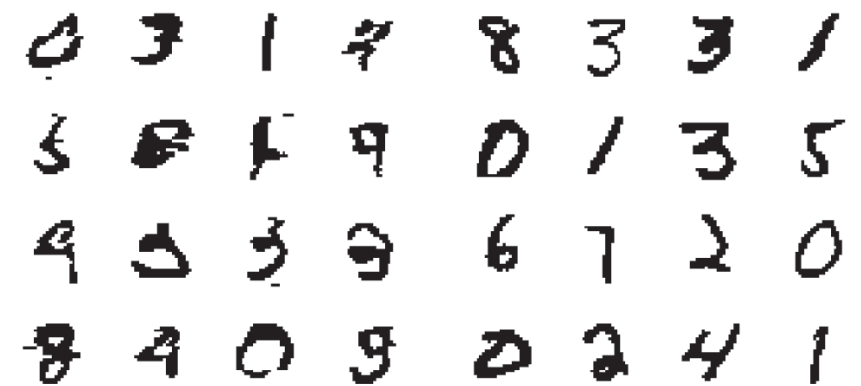
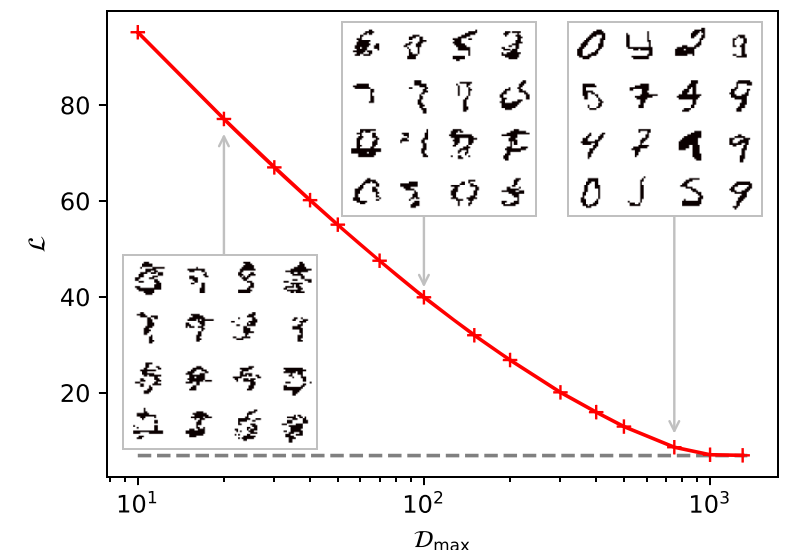
$$|\Psi\rangle = \sum_i \Psi(\vec{v}_i) |\vec{v}_i\rangle \quad P(\vec{v}) = \frac{|\Psi(\vec{v})|^2}{Z}$$

- この量子状態に対して測定を行うと $P(\vec{v})$ の確率で $\vec{v}$ が得られる

• 量子状態のテンソルネットワーク表現



- 負の対数尤度を最小化するようにテンソルを最適化



(a) Generated

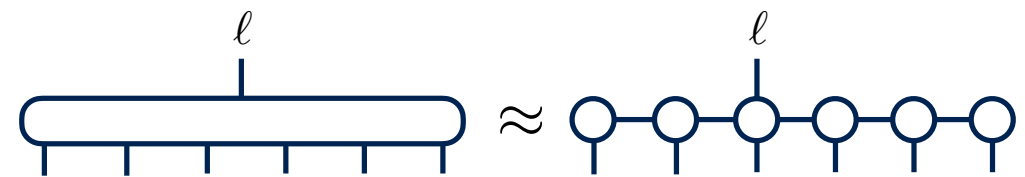
(b) Original

教師あり学習の埋め込み

• テンソルネットワークによる教師あり学習

- 教師あり学習もテンソルネットワークで実現可能
- 入力データ: 量子回路の 初期状態 として符号化
- ラベル: 量子状態の 一部を測定 して取得
- 分類問題を「状態準備 + 測定」として定式化

$$\begin{array}{c} \ell \\ \hline \text{---} \\ \text{---} \end{array} \begin{array}{c} W^\ell \\ \Phi(\mathbf{x}) \end{array} = \begin{array}{c} \ell \\ \text{---} \\ \text{---} \end{array} f^\ell(\mathbf{x})$$

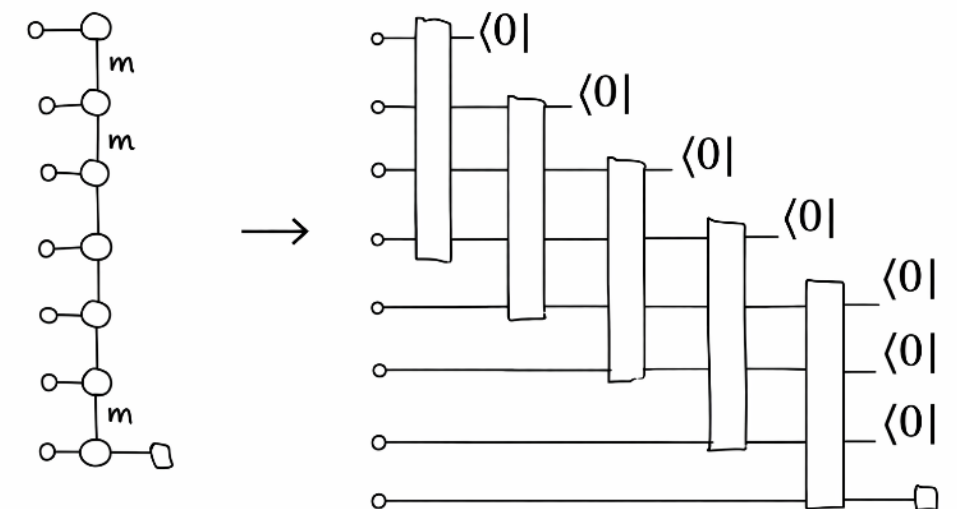
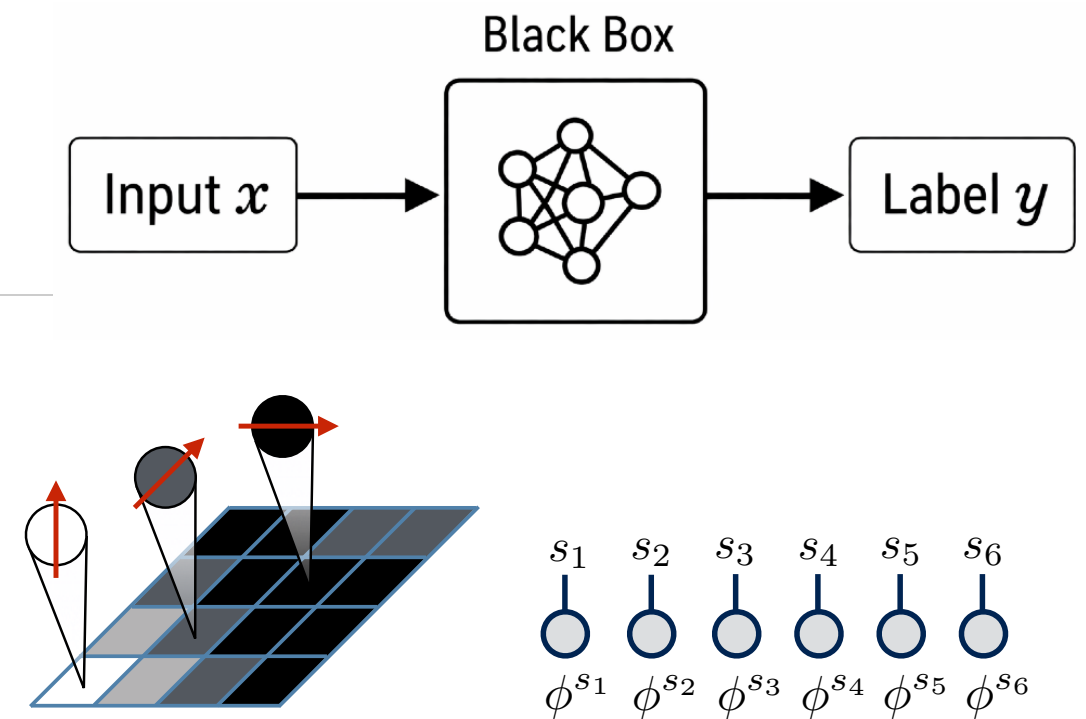


Stoudenmire, E. M., & Schwab, D. J. (2016)

• ループのないテンソルネットワーク

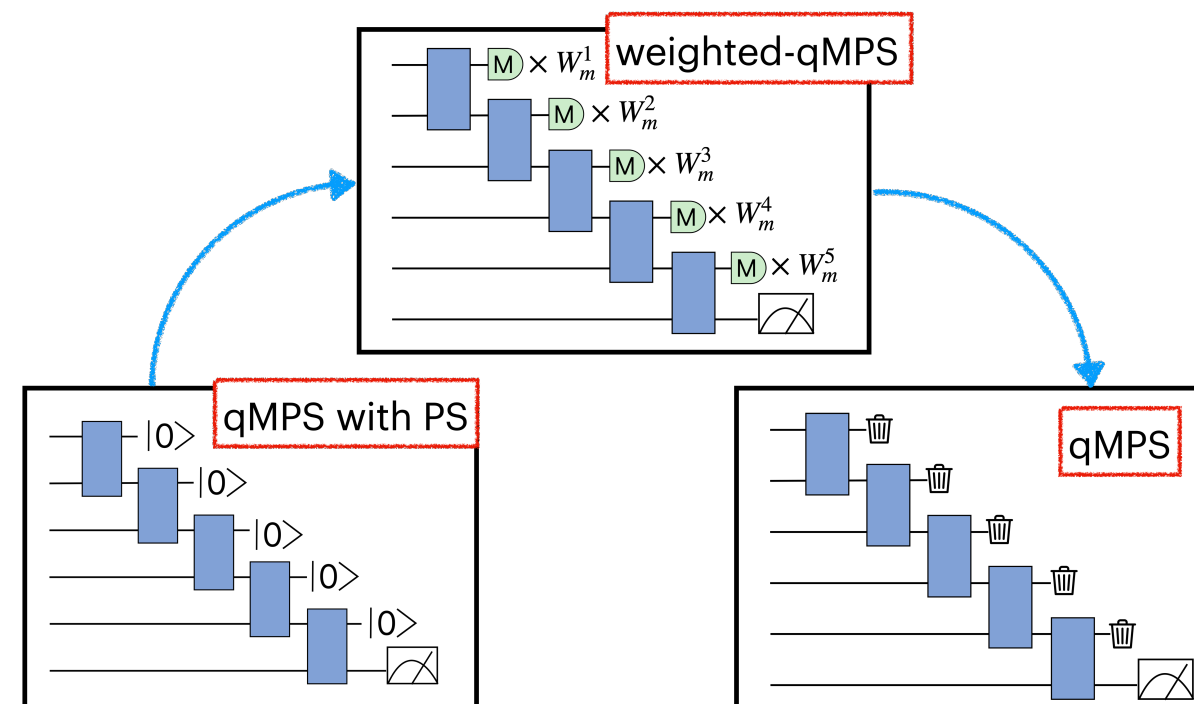
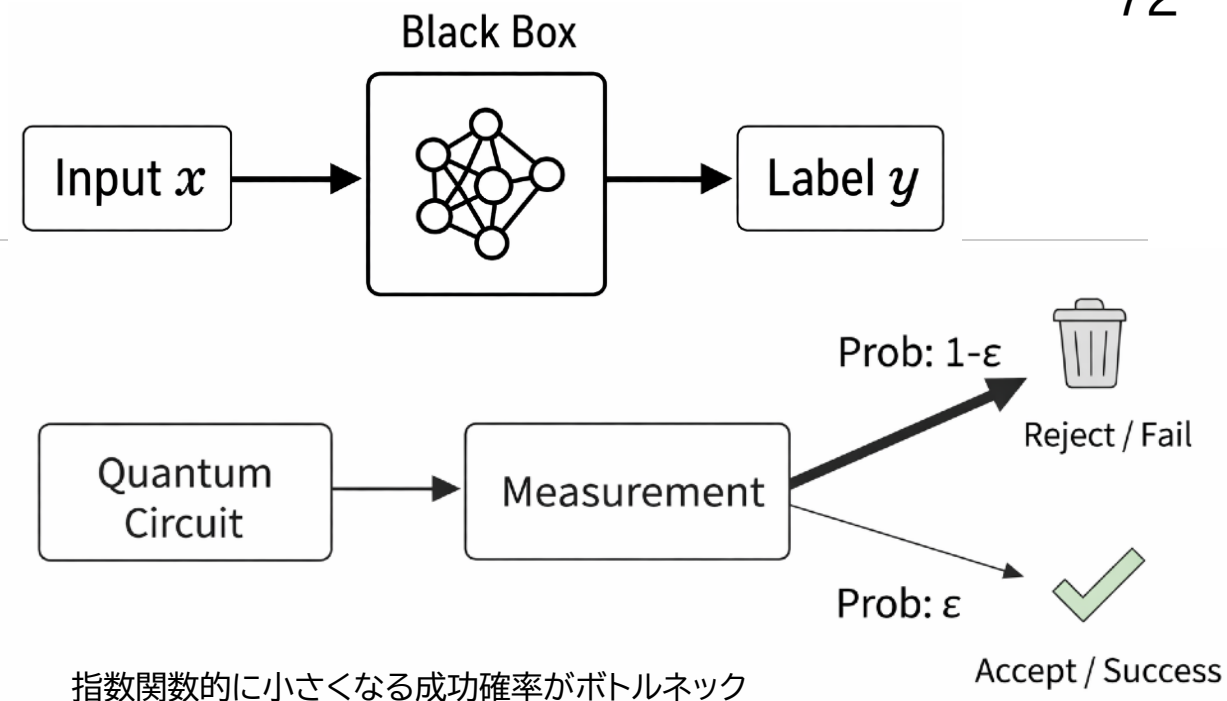
- 行列積状態 MPS、ツリーテンソルネットワーク TTN
- 葉から順次「アイソメトリ(isometry)」に変換
- 変換後のテンソルを「ユニタリ」の一部として量子回路に埋め込み

Sugawara, Inomata, Okubo, Todo (2025), Inomata (2026)



ポストセレクション問題

- MPSによる教師あり学習の埋め込み
- ポストセレクション
 - 一般のテンソル(線形写像)は ユニタリではない
 - 回路化の典型的な方法
 - ユニタリの一部に埋め込み
 - 測定により写像を実装
 - 成功=補助量子ビットの測定結果が全て0
 - 成功の場合のみ採用(post-selection)
- 問題点
 - 成功確率が小さいとサンプル効率が著しく低下
- 断熱エンコーディング
 - MPS を断熱的に変形
 - 補助量子ビットの状態に依らず
分類結果がほぼ同一になるよう設計
 - 「MPSで(古典)学習 → ポストセレクションあり量子回路 → 断熱エンコーディング → ポストセレクションなし量子回路」により、「バレンプラトー問題」も回避できると期待



まとめ: テンソルネットワークを通して見る量子多体系計算

- 表現

- 量子状態・分配関数・確率分布・離散化した関数を、共通のテンソルネットワークとして扱える
- 必要なボンド次元は対象の相関構造に依存する

- 計算

- 縮約は和を直接評価し、サンプリングはその一部を確率的に評価する
- 両者は対立する方法ではなく、組み合わせて使える

- TNMC

- 射影子のMCMCにより、低ランク切断に由来する系統誤差を取り除き、TNの構造を統計誤差の低減と符号問題の緩和に利用する

- 展開

- 同じ表現と計算技術が、量子回路シミュレーション、状態準備、関数計算、機械学習へとつながる

- 古典サンプルへ変換せず、準備した量子状態を次の量子処理へ直接渡す量子-量子の連携も、今後の重要な方向