Introduction to Tensor Networks

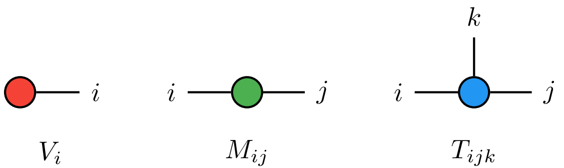
Xiangdong Zeng
Jan 19, 2026

Basic concepts

What is a tensor?

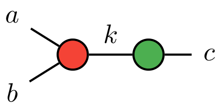
- Most general definition:
  - Tensor describes the structure that you can put two or more objects together, such that some certain **coherence conditions** can be satisfied
  - Monoidal category
- Linear algebra aspect:
  - A **multilinear map** over some linear spaces
  - A **multi-dimensional array** when we select a set of basis (defined by components)
- Terminologies:
  - **Rank** (or degree/order): dimension of the components array
  - **Dimension**: size of each component

Notations

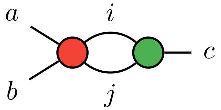
- Abstract index notation: use components to represent the tensor itself
  - Vector: $\boldsymbol{V} = \sum_{i=1}^{D} V_i \, \boldsymbol{e}_i =: V_i$
  - Matrix: $\boldsymbol{M} = \sum_{i,j=1}^{D} M_{ij} \, \boldsymbol{e}_i \otimes \boldsymbol{e}_j =: M_{ij}$
  - Rank-3 tensor: $\boldsymbol{T} = \sum_{i,j,k=1}^{D} T_{ijk} \, \boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k =: T_{ijk}$
  - $\otimes$: **tensor product**, $\boldsymbol{e}_i$: bases
- Diagram notation:
  - Use solid shapes (“balls”) for tensors, bonds (“legs”) for indices
  - Examples: 

Tensor contraction

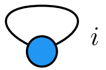
- Generalization of inner product and matrix product
- Computational cost is dependent on the contraction ordering
- Examples:
  - $C_{abc} = \sum_k A_{abk} B_{kc} =: A_{abk} B_{kc}$
  - $C_{abc} = \sum_{ij} A_{abij} B_{ijc} =: A_{abij} B_{ijc}$
- Use **Einstein notation** to simplify the expressions:
  - Omit the summation symbol
  - Sum over repeated indices
- Trace:
  - All the indices of a tensor are contracted
  - $\mathrm{tr}(\boldsymbol{M}) = \sum_i M_{ii} =: M_{ii}$
  - $\boldsymbol{AB} \neq \boldsymbol{BA}$ in general, but $\mathrm{tr}(\boldsymbol{AB}) = \mathrm{tr}(\boldsymbol{BA})$



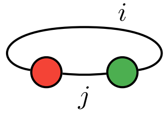
$A_{abk} B_{kc}$



$A_{abij} B_{ijc}$



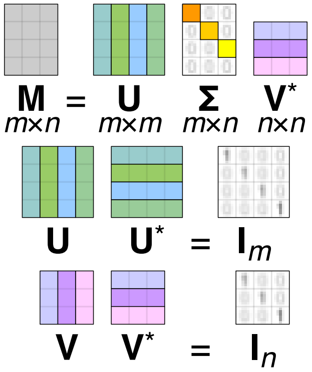
$M_{ii} = \mathrm{tr}(M)$



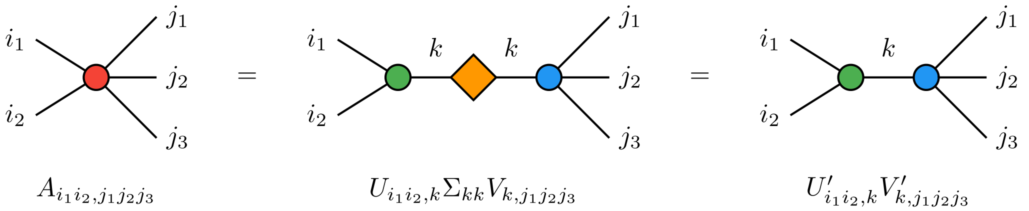
$A_{ij} B_{ji} = \mathrm{tr}(AB) = \mathrm{tr}(BA)$

Singular value decomposition (SVD)

- $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\dagger$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary, $\boldsymbol{\Sigma}$ is diagonal
- Truncated SVD: only keep first r singular values in $\boldsymbol{\Sigma}$
  - Eckart–Young–Mirsky theorem: gives best rank-r approximation of $\boldsymbol{M}$
  - Moore–Penrose (pseudo) inverse: $\boldsymbol{M}^+ = \boldsymbol{V}\boldsymbol{\Sigma}^+\boldsymbol{U}^\dagger$
- Split a single node into two nodes

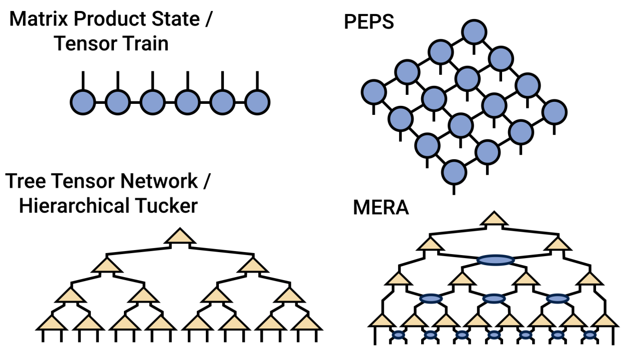


https://en.wikipedia.org/wiki/Singular_value_decomposition



Tensor network

- A network of tensor units (no magic)
- But why?
  - Time/space complexity **increases exponentially** with the size of system (i.e. Hilbert space is too large)
  - Most of the data are irrelevant and can be truncated (most interactions are **local**)
    - **Area-law**: entanglement entropy scales as the area of the space rather than the volume ($S \sim \partial A$)
    - Tensor network states can naturally reveal such properties
    - The same idea can be found in image/video compression

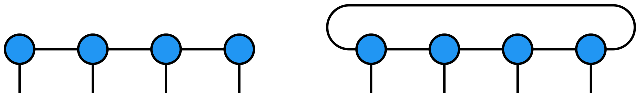


https://tensornetwork.org

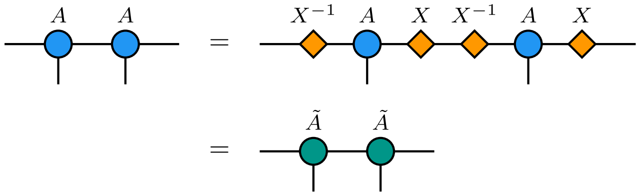
Algorithms

TN algorithms: MPS

- Matrix product state (MPS): 1D tensor chain
  - $\Psi_{i_1 i_2 \ldots i_n} = A_{i_1}^{j_1 j_2} \otimes A_{i_2}^{j_2 j_3} \otimes \cdots \otimes A_{i_n}^{j_n j_1}$
  - Physical indices: uncontracted legs (i_k)
  - Virtual indices: contracted legs (j_k)
- Entanglement entropy
  - $S_L = -\operatorname{tr}(\rho_L \log \rho_L) \sim \mathcal{O}(\log \chi)$
  - χ: bond dimension (dimension of virtual bonds)
  - Area-law for **1D gapped Hamiltonian**
- Gauge freedom
  - Canonical form
  - Schmidt decomposition for virtual bonds (SVD)



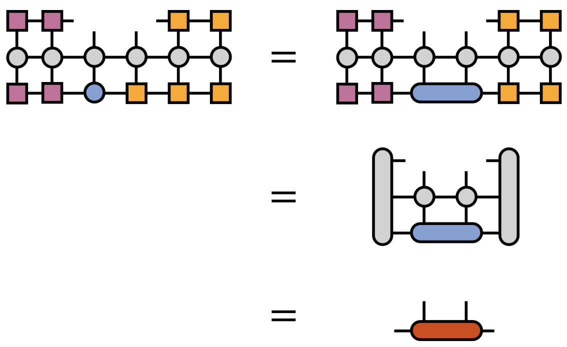
MPS with open/periodic boundary conditions



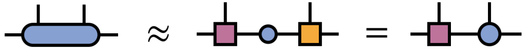
Gauge freedom in MPS

TN algorithms: DMRG

- Density matrix renormalization group (DMRG)
  - Variational optimization of MPS to find ground state
  - Ground state: $|\Psi_0\rangle = \arg\min_{|\Psi\rangle} \langle\Psi|H|\Psi\rangle \,/\, \langle\Psi|\Psi\rangle$
  - Sweep through the chain, optimize two sites at a time
- Step 1: energy functional minimization:
  - $\mathcal{L} = \langle\Psi(A)|H|\Psi(A)\rangle - \lambda\,\langle\Psi(A)|\Psi(A)\rangle$
  - λ: Lagrange multiplier
  - Minimal condition: $\partial\mathcal{L}/\partial A = 0 \implies H_{\text{eff}}A = \lambda A$
    - Partial derivative equivalent to digging a $\tilde{A}$ hole
    - Requires canonical form of MPS
- Step 2: update MPS tensors
  - Using a truncated SVD



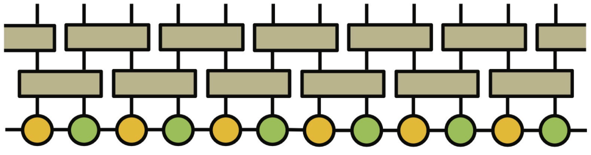
Effective Hamiltonian in DMRG



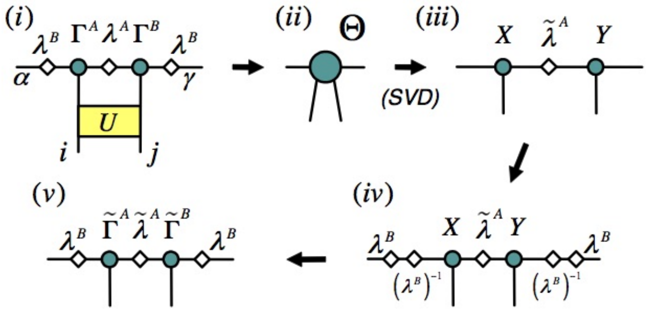
Truncated SVD

TN algorithms: TEBD

- Time-evolving block decimation (TEBD)
  - Simulate time evolution: $|\Psi_t\rangle = e^{-iHt}|\Psi_0\rangle$
  - Or imaginary time evolution: $|\Psi_\beta\rangle = e^{-\beta H}|\Psi_0\rangle$
  - Suzuki–Trotter decomposition:
    - $e^{-\tau(A+B)} = e^{-\tau A}e^{-\tau B} + \mathcal{O}(\tau^2)$
- Generalization for arbitrary operators
  - Time evolution operators are close to unitary
  - Requires extra canonicalization step
- Example: calculate partition function $Z = \mathrm{tr}(e^{-\beta H})$
  - Interpret Z as an evolution of transfer matrix
  - Contract along the imaginary time direction using iTEBD



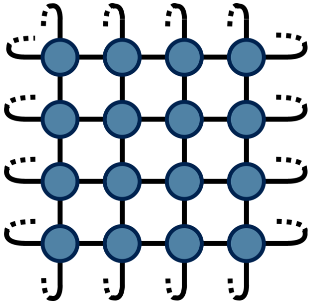
Time evolution with two-site gates



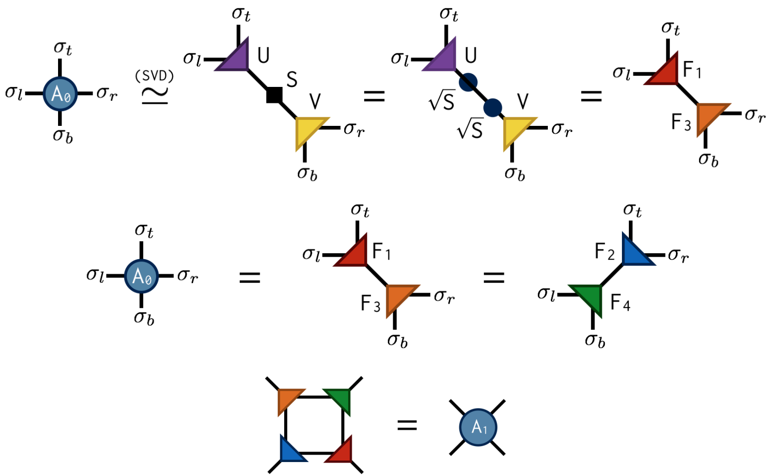
Contract and perform SVD truncation

TN algorithms: TRG & TNR (1)

- Tensor renormalization group (TRG)
  - Coarse-graining of 2D tensor network (partition function) until a single **fixed-point tensor**
  - Use SVD to split and recombine tensors



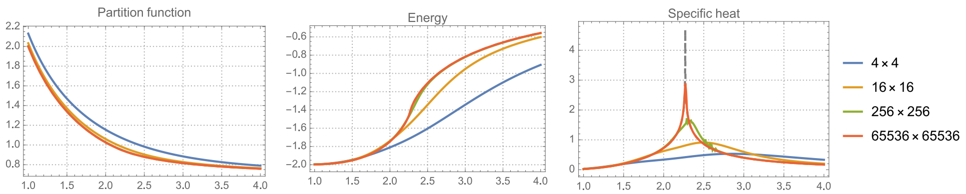
Partition function as a tensor network



Decomposition and contraction in TRG

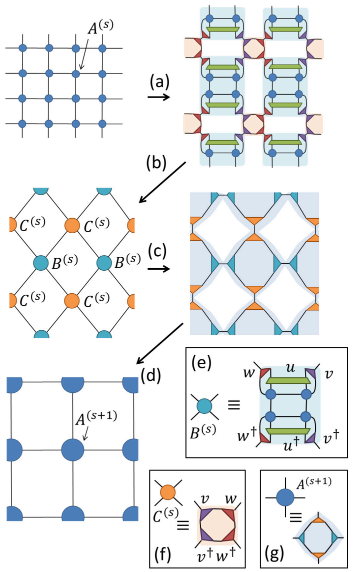
TN algorithms: TRG & TNR (2)

- TRG limitations:
  - Long-range entanglement in **critical systems**
    - Correlation length $\xi \to \infty$
    - Area-law violated with logarithmic correction: $S_L \sim \log L$
  - Fails to properly renormalize at criticality



TRG simulation of 2D Ising model

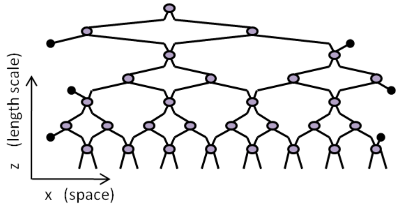
- Tensor network renormalization (TNR)
  - **Isometry**: TRG triangle tensor
  - **Disentangler**: remove short-range entanglement



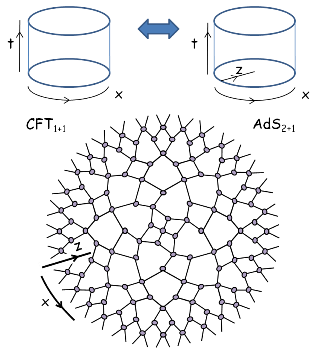
TNR steps
u: disentanglers
v: isometries

TN algorithms: MERA

- Multi-scale entanglement renormalization ansatz (MERA)
  - Variational ansatz for ground state wavefunction in 1D critical systems
  - Use isometries and disentanglers as in TNR
- Tensor network and AdS/CFT correspondence
  - Boundary: CFT (critical system)
  - Bulk: discrete AdS space (hyperbolic geometry) → gravity
  - Ryu–Takayanagi formula:
    - $S_A = \mathrm{Area}(\gamma_A)/4G_N$
    - Area law of entanglement entropy in CFT corresponds to minimal surface in AdS



MERA tensor network



MERA in hyperbolic geometry → AdS/CFT

References

- R. Orús. Tensor networks for complex quantum systems. *Nat. Rev. Phys.* **1**, 538–550 (2019)
- U. Schollwöck. The density-matrix renormalization group. *Rev. Mod. Phys.* **77**, 259 (2005)
- G. Vidal. Classical simulation of infinite-size quantum lattice systems in one spatial dimension. *Phys. Rev. Lett.* **98**, 070201 (2007)
- R. Orús and G. Vidal. Infinite time-evolving block decimation algorithm beyond unitary evolution. *Phys. Rev. B* **78**, 026117 (2008)
- M. Levin and C. P. Nave. Tensor renormalization group approach to 2D classical lattice models. *Phys. Rev. Lett.* **99**, 120601 (2007)
- G. Evenbly and G. Vidal. Tensor network renormalization. *Phys. Rev. Lett.* **115**, 180405 (2015)
- G. Vidal. Entanglement renormalization. *Phys. Rev. Lett.* **99**, 220405 (2007)
- B. Swingle. Entanglement renormalization and holography. *Phys. Rev. D* **86**, 065007 (2012)
- Tensor Network and tensors.net
- And see my PhD thesis: Aspects on tensor networks for topological orders