# Corner Transfer Matrix Renormalization Group

Efficient Contraction of 2D Classical Lattice Models

Your Name

Your Institution

January 5, 2026

# Outline

**1D Ising Model:**

$$H = -J \sum_i \sigma_i \sigma_{i+1}$$
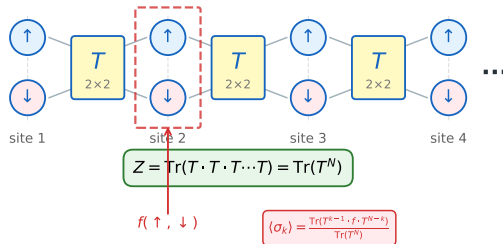
**Partition function as contraction:**

$$Z = \text{Tr}(T \cdot T \cdot T \cdots T) = \text{Tr}(T^N)$$

where $T \in \mathbb{R}^{2 \times 2}$ encodes local interactions.

**Local observable:**

$$\langle \sigma_k \rangle = \frac{\text{Tr}(T^k \cdot f(\uparrow, \downarrow) \cdot T^{N-k})}{\text{Tr}(T^N)}$$

All $T$ identical ⇒ commute ⇒ diagonalize once!



$$Z = \text{Tr}(T \cdot T \cdot T \cdots T) = \text{Tr}(T^N)$$

$f(\uparrow, \downarrow)$

$$\langle \sigma_k \rangle = \frac{\text{Tr}(T^{k-1} \cdot f \cdot T^{N-k})}{\text{Tr}(T^N)}$$
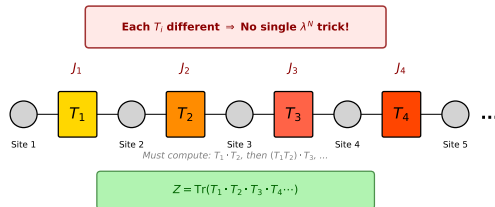
**Key Point**

# When Even 1D Classical Needs Renormalization Group

**Recall:** For uniform 1D Ising, $Z = \text{Tr}(T^N)$ has no contraction issues.

**But what if...**

- Couplings $J_i$ are **site-dependent**?
- Random disorder: $J_i \sim \mathcal{N}(\bar{J}, \sigma)$?
- Open boundary conditions (no translation symmetry)?

**Then:** Cannot diagonalize $T$ once!
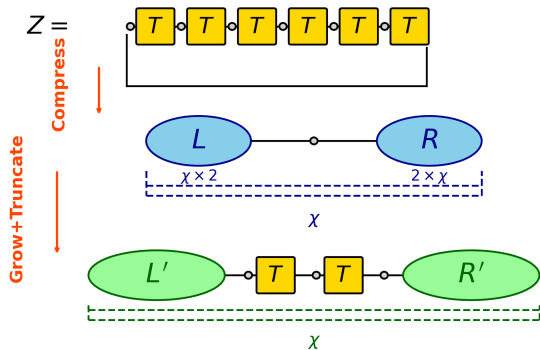
$$Z = \text{Tr}(T_1 \cdot T_2 \cdot T_3 \cdots T_N)$$

Each $T_i$ is different $\Rightarrow$ no simple $\lambda^N$ formula.

Each $T_i$ different $\Rightarrow$ No single $\lambda^N$ trick!

$J_1 \qquad J_2 \qquad J_3 \qquad J_4$

$T_1 \qquad T_2 \qquad T_3 \qquad T_4$

Site 1  Site 2  Site 3  Site 4  Site 5  ...

*Must compute: $T_1 \cdot T_2$, then $(T_1 T_2) \cdot T_3$, ...*

$Z = \text{Tr}(T_1 \cdot T_2 \cdot T_3 \cdot T_4 \cdots)$

## The Problem

For $N$ sites: need $O(N)$ matrix multiplications. Not exponential, but **no closed-form solution**.

# Solution: Transfer Matrix Renormalization Group (TMRG)

**Insight:** Not all configurations contribute equally to $Z \Rightarrow$ Keep only the **most relevant** ones!



$Z =$

Compress

Grow+Truncate

$\chi = $ *bond dimension (most relevant configurations in compressed basis)*

**TMRG Algorithm:**

1. **Grow:** $L' = L \cdot T$, $R' = T \cdot R$
   (add sites)

2. **Truncate:** SVD $\rightarrow$ keep $\chi$ largest
   (coarse-grain)

### RG Fixed Point

Iterate until $L^*, R^*$ converge $\Rightarrow$
Thermodynamic limit!
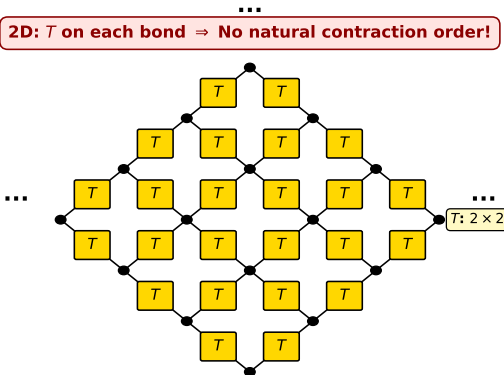
*Kind-of like infinite-DMRG by adding new of sites*

# 2D: The Transfer Matrix Becomes a **Network**

**1D:** Ordered product of matrices

$$Z = \text{Tr}(T_1 \cdot T_2 \cdots T_N)$$

Contract left-to-right: $O(N)$.

**2D:** $T$ still $2 \times 2$ on each **bond**,
but bonds form a **2D network**!
$\Rightarrow$ No natural ordering.



2D: $T$ on each bond $\Rightarrow$ No natural contraction order!

$T: 2 \times 2$

## The Fundamental Problem

**No natural contraction order!** The 2D tensor network cannot be reduced to a simple trace of matrix products. **Exact contraction is #P-hard!**

## One Approach: Row-to-Row Transfer (DMRG-style)

**Idea:** Group one row of $L$ spins $\rightarrow$ treat as a "super-spin" with $2^L$ states.

**Apply DMRG/TMRG ideas:**

- Row config space $\rightarrow$ MPS
- Row-to-row transfer $\rightarrow$ MPO
- Truncate via SVD

*[Figure: Row $\rightarrow$ super-spin, $T_{row} \in \mathbb{R}^{2^L \times 2^L}$]*

### Pros & Cons

+ Systematic, well-understood
− Breaks 2D symmetry
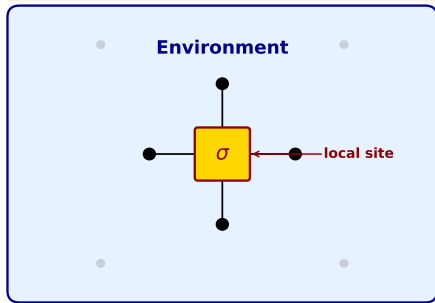− Hard to generalize to other lattices

Then $Z = \text{Tr}(T_{\text{row}}^M)$ looks like 1D!

*(Details in Appendix)*

# A More Natural Approach: Think in 2D!

**Question:** If we only want **local observables** $\langle \sigma_{i,j} \rangle$,
do we really need to contract the *entire* infinite lattice?

**Environment**

$\sigma$ ←— **local site**

$$\langle \sigma \rangle = \frac{\text{Tr}(\text{Environment} \cdot \sigma)}{\text{Tr}(\text{Environment})}$$

**Key Insight:**
Decompose the infinite 2D environment into
**geometrically natural** pieces:

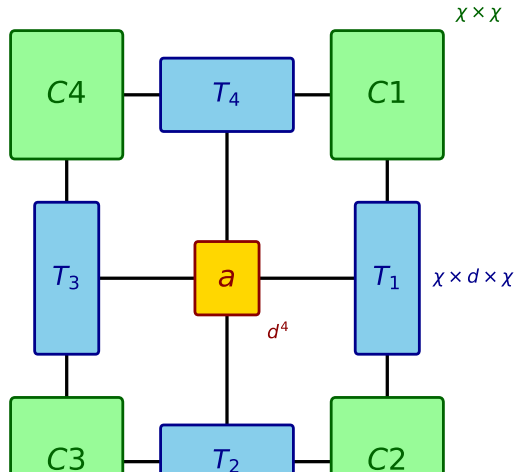- **4 Corners** (quarter-planes)
- **4 Edges** (half-infinite strips)

## Baxter's Insight (1968)

The **Corner Transfer Matrix** encodes a
quarter of the infinite plane!

**CTMRG: Decompose 2D environment into 4 Corners + 4 Edges**

# The Plaquette Tensor: From Bonds to Faces

**Key transformation:**

Bond-based $T$ (rank-2) $\rightarrow$ Plaquette-based $W$ (rank-4)

[Figure: 4 bonds around a plaquette $\rightarrow$ W]

**For 2D Ising:**
Combine 4 bond tensors $T$ around a face:

$$W_{u,d,l,r} = \sum_\sigma T_{\sigma,u} T_{\sigma,d} T_{\sigma,l} T_{\sigma,r}$$

where $T_{\sigma,\sigma'} = e^{\frac{\beta J}{2}\sigma\sigma'}$

## Why Plaquette?

$W$ is the **rank-4 local tensor** used in CTMRG. Each $W$ encodes one face of the tilted lattice.

# CTMRG Iteration: Step 1 — Grow (Absorb Row/Column)

**Absorption:** Add one row **and** one column to expand the environment.

*[Figure: Absorption step — corner grows by absorbing a and edges]*

# CTMRG Iteration: Step 2 — Truncate (The Key Difference!)

**Truncation in 2D:** Use the **full environment** to determine projectors.

*[Figure: Building density matrix from environment for truncation]*

**Compute projector $P$:**                    **Apply truncation:**

# CTMRG: Complete Algorithm

*[Figure: CTMRG algorithm flowchart]*

local ……

# Computing Observables with CTMRG

**Partition function:**

$$Z \propto \text{Tr}(C_1 T_1 C_2 T_2 C_3 T_3 C_4 T_4)$$

*[Figure: Computing $\langle \sigma \rangle$ by inserting operator]*

*[Results to be added: magnetization, critical behavior, etc.]*

# Summary: The Core of CTMRG

## CTMRG in Three Steps

1. **Decompose:** Infinite 2D lattice $\rightarrow$ 4 corners + 4 edges
2. **Grow:** Absorb local tensors (add row/column)
3. **Truncate:** SVD-based RG to keep $\chi$ most relevant states

*[Figure: Visual summary — Decompose $\rightarrow$ Grow $\rightarrow$ Truncate]*

## Comparison: iDMRG vs CTMRG

| Aspect | iDMRG (1D) | CTMRG (2D) |
|---|---|---|
| Dimension | 1D chain | 2D square lattice |
| Environment | Left + Right | 4 Corners + 4 Edges |
| Grow step | Add site pair | Add row + column |
| Truncation | SVD on center bond | SVD on corner boundaries |
| Bond dimension | $\chi$ (MPS) | $\chi$ (environment) |
| Fixed point | $L^*, R^*$ | $C_i^*, T_i^*$ |
| Computational cost | $O(\chi^3)$ | $O(\chi^6)$ or $O(\chi^5)$ |

# Thank You!

Questions?

## Appendix: TMRG Setup in Detail

**Goal:** Compute $Z = \text{Tr}(T_1 \cdot T_2 \cdots T_N)$ for large $N$.

*[Figure: Left environment L, local site T, right environment R]*

**Left Environment $L$:**

$$L_\sigma = \sum T_1 \cdot T_2 \cdots T_{k-1}$$

**Right Environment $R$:**

$$R_\sigma = \sum T_{k+1} \cdots T_N$$

# Appendix: TMRG Step 1 — Grow (Absorption)

**Absorption:** Add one more site to the environment.

*[Figure: $L' = L \cdot T_k$, growing the left environment]*

**Mathematically:**

$$L'_{\sigma_k} = \sum_{\sigma_{k-1}} L_{\sigma_{k-1}} \cdot (T_k)_{\sigma_{k-1}, \sigma_k}$$

## Appendix: TMRG Step 2 — Truncate (SVD)

**Idea:** Compress $L'$ back to dimension $\chi$ using SVD.

**Form the "density matrix":**

$$\rho = L' \cdot R'^T$$

**SVD:**

$$\rho = U \Sigma V^\dagger$$

**Truncate:**

$$P = U_{:,1:\chi}$$

**New environment:**

$$L_{\text{new}} = P^\dagger L'$$

[Figure: SVD spectrum, keep $\chi$ largest]

### Eckart-Young Theorem

SVD gives the **optimal** rank-$\chi$ approximation

## Appendix: TMRG Convergence and Observables

**Iterate:** Grow $\rightarrow$ Truncate $\rightarrow$ Grow $\rightarrow$ Truncate $\rightarrow \cdots$

**Convergence criterion:**
Fixed point: $L^*, R^*$ such that

$$L^* \xrightarrow{\text{grow+truncate}} L^*$$

**Free energy:**

$$f = -k_B T \lim_{N \to \infty} \frac{1}{N} \ln Z$$

$$= -k_B T \ln \sigma_1^*$$

**Local observables:**

$$\langle \sigma_k \rangle = \frac{L^* \cdot \sigma_k \cdot R^*}{L^* \cdot R^*}$$

**Correlation functions:**

$$\langle \sigma_i \sigma_j \rangle = \frac{L^* \cdot \sigma_i \cdot T^{|i-j|} \cdot \sigma_j \cdot R^*}{Z}$$

### Physical Meaning

$L^*, R^*$ encode the **thermodynamic limit**.

**Idea:** Treat one row of $L$ spins as a single "super-spin".

[Figure: Row $\rightarrow$ super-spin with $2^L$ states]

**Row configuration:**

$$\vec{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_L)$$

Total states: $2^L$.

**Row-to-row transfer:**

$$(T_{\text{row}})_{\vec{\sigma}, \vec{\sigma}'} = \prod_{i=1}^{L} e^{\beta J \sigma_i \sigma_i'} \prod_{i=1}^{L-1} e^{\beta J \sigma_i \sigma_{i+1}}$$

$T_{\text{row}} \in \mathbb{R}^{2^L \times 2^L}$.

### Key Point

Now, $Z = \text{Tr}(T_{\text{row}}^M)$ looks like 1D problem with $2^L$-dimensional transfer matrix!

**Key insight:** $T_{\text{row}}$ has a **tensor network** structure!

**Decompose** $T_{\text{row}}$ into local tensors:

$$T_{\vec{\sigma},\vec{\sigma}'} = \sum_{\alpha_1,\dots} W_{\alpha_0\alpha_1}^{\sigma_1\sigma_1'} W_{\alpha_1\alpha_2}^{\sigma_2\sigma_2'} \cdots$$

**Local tensor** $W$:

$$W_{\alpha\beta}^{\sigma\sigma'} = \text{local Boltzmann weight}$$

- $\sigma, \sigma'$: spins in rows $n$, $n+1$
- $\alpha, \beta$: auxiliary (horizontal bonds)

*[Figure: MPO structure of $T_{row}$]*

### No Quantum!
"MPO" is just a factorization of the classical transfer matrix.

## Appendix: Boundary = MPS

**MPS:** Efficient representation of row configuration space.

**Full vector:**

$$|R\rangle = \sum_{\vec{\sigma}} R_{\vec{\sigma}} |\vec{\sigma}\rangle$$

has $2^L$ components.

**MPS compression:**

$$R_{\vec{\sigma}} = A^{\sigma_1} A^{\sigma_2} \cdots A^{\sigma_L}$$

Only $L \cdot \chi^2 \cdot 2$ parameters!

*[Figure: MPS tensor network]*

### Physical Meaning

MPS compresses $2^L$ row configs into $\chi$ effective states with **limited entanglement**.

## Appendix: Row-to-Row DMRG Algorithm

**Combine:** MPS (boundary) + MPO (transfer) + SVD (truncation).

1. Initialize MPS $|L\rangle$, $|R\rangle$ for boundaries
2. **Grow:** Apply MPO $T_{\text{row}}$ to boundaries

$$|L'\rangle = T_{\text{row}}|L\rangle$$

3. **Truncate:** SVD to compress bond dimension back to $\chi$
4. Iterate until convergence to fixed point $|L^*\rangle$, $|R^*\rangle$

| Pros | Cons |
|---|---|
| • Systematic, well-understood | • Breaks 2D rotational symmetry |
| • Controlled approximation | • Hard to generalize to other lattices |

# Appendix: No Monte Carlo Sign Problem

**Classical models:** All Boltzmann weights are **positive**!

$$W = e^{-\beta H} > 0 \quad \text{always}$$

*[Figure: Classical (positive) vs Quantum (sign problem)]*

## Appendix: Why is 3D Difficult?

**In 2D:** Environment tensors are **1D objects** (edges).
**In 3D:** Environment would be **2D surfaces** — back to exponential!

*[Figure: 2D boundary problem in 3D systems]*

# Appendix: Extensions and Generalizations

**Beyond square lattice Ising:**

**Different lattices:**

- Honeycomb
- Triangular
- Kagome

**Different models:**

- Potts model
- Clock model
- Vertex models

**Quantum systems (via iPEPS):**

- 2D Heisenberg model
- Frustrated magnets
- Topological phases

**Improvements:**

- Directional CTMRG
- Full-update vs simple-update
- Gradient optimization

# Appendix: Computational Complexity

**CTMRG scaling:**

| Operation | Cost | Bottleneck? |
|---|---|---|
| Corner absorption | $O(\chi^4 d^2)$ | |
| Edge absorption | $O(\chi^3 d^2)$ | |
| Build density matrix | $O(\chi^4)$ | |
| SVD for projector | $O(\chi^3 d^3)$ | ✓ |
| Apply truncation | $O(\chi^3 d)$ | |
| **Total per iteration** | $O(\chi^3 d^3)$ to $O(\chi^6)$ | |

## Comparison

- iDMRG (1D): $O(\chi^3)$ per sweep
- CTMRG (2D): $O(\chi^5)$ – $O(\chi^6)$ per iteration
- More expensive, but still **polynomial** in $\chi$!

## Appendix: Key References

**Original works:**

- R. J. Baxter, *J. Math. Phys.* **9**, 650 (1968) — Corner transfer matrices
- R. J. Baxter, *J. Stat. Phys.* **19**, 461 (1978) — CTM method

**Modern CTMRG:**

- T. Nishino & K. Okunishi, *J. Phys. Soc. Jpn.* **65**, 891 (1996)
- T. Nishino & K. Okunishi, *J. Phys. Soc. Jpn.* **66**, 3040 (1997)

**CTMRG for iPEPS:**

- R. Orús & G. Vidal, *Phys. Rev. B* **80**, 094403 (2009)
- P. Corboz et al., *Phys. Rev. B* **84**, 041108(R) (2011)

**Reviews:**

- R. Orús, *Ann. Phys.* **349**, 117 (2014) — Tensor networks review