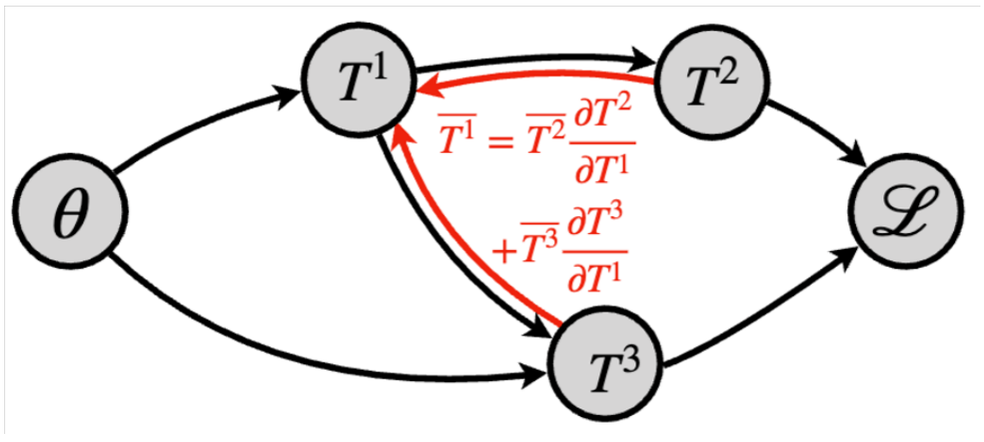
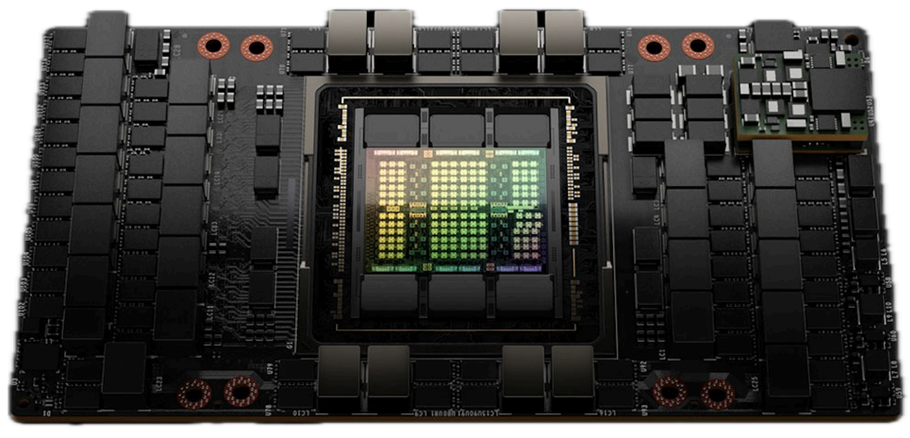
# Scalable iPEPS Simulations in the Era of AD and GPUs

Presenter: Qi Yang @ UNIVERSITY OF AMSTERDAM

**We want larger Models !!**
**We want better PyTorch !!**




Automatic Differentiation



Modern GPUs

EuroHPC Joint Undertaking

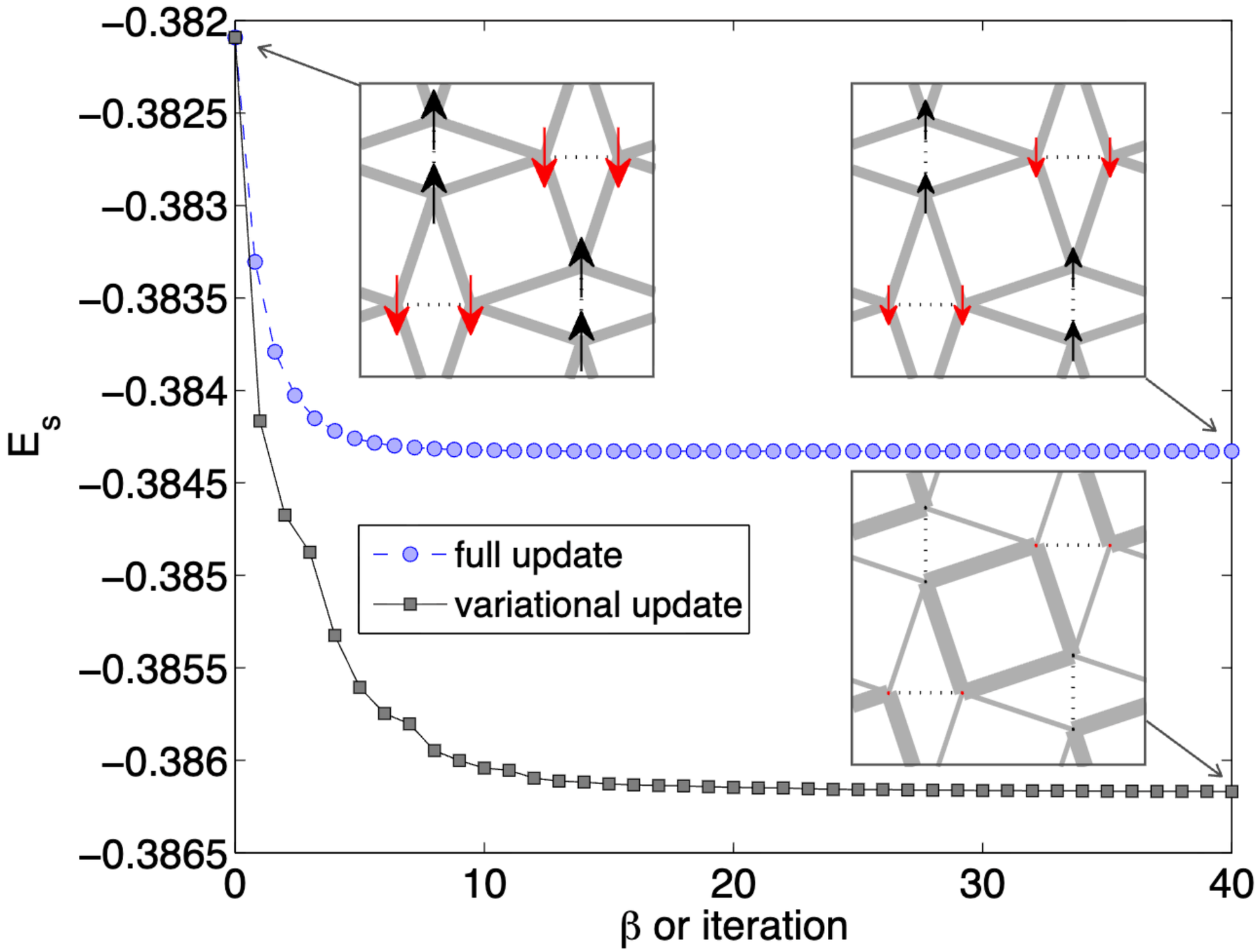
erc

**Can we take a ride?**

# Table of contents

- A brief introduction of AD and iPEPS
- QRCTM:  A powerful method hiding in plain sight for 29 years.
- Application: High Spin Kitaev model
- Riemannian optimization for iPEPS
- Technical Tricks

# A brief introduction of AD: Why AD?




Corboz, PhysRevB.94.035133 (2016)

# A brief introduction of AD: Backward

Scalar

$$E = E(\mathbf{x}_N) \qquad \mathbf{x}_{n+1} = f_n(\mathbf{x}_n)$$

Jacobian

$$\frac{\partial E}{\partial \mathbf{x}_1} = \frac{\partial E}{\partial \mathbf{x}_2}\frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1} = \frac{\partial E}{\partial \mathbf{x}_3}\frac{\partial \mathbf{x}_3}{\partial \mathbf{x}_2}\frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1} = \ldots\ldots = \partial_N E \times \prod_{i=1}^{N-1} \partial_i f_i(x_i)$$

# A brief introduction of AD: Backward

Scalar

$$E = E(\mathbf{x}_N) \qquad \mathbf{x}_{n+1} = f_n(\mathbf{x}_n)$$

Jacobian

$$\frac{\partial E}{\partial \mathbf{x}_1} = \frac{\partial E}{\partial \mathbf{x}_2}\frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1} = \frac{\partial E}{\partial \mathbf{x}_3}\frac{\partial \mathbf{x}_3}{\partial \mathbf{x}_2}\frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1} = \ldots\ldots = \partial_N E \times \prod_{i=1}^{N-1} \partial_i f_i(x_i)$$

Step 1: $G_N = \partial_N E, G_i = \partial_i E$

Step i: $G_{i-1} = G_i \times \partial_{i-1} f_{i-1}(x_{i-1})$

How to know it?

Case 1: Reversible functions, e.g. $f(x) = x^3, \partial_x f = 3x^2 = 3(f(x))^{2/3}$

# A brief introduction of AD: Backward

Scalar

$$E = E(\mathbf{x}_N) \qquad \mathbf{x}_{n+1} = f_n(\mathbf{x}_n)$$

Jacobian

$$\frac{\partial E}{\partial \mathbf{x}_1} = \frac{\partial E}{\partial \mathbf{x}_2}\frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1} = \frac{\partial E}{\partial \mathbf{x}_3}\frac{\partial \mathbf{x}_3}{\partial \mathbf{x}_2}\frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1} = \ldots\ldots = \partial_N E \times \prod_{i=1}^{N-1} \partial_i f_i(x_i)$$

Step 1: $G_N = \partial_N E, G_i = \partial_i E$

Step i: $G_{i-1} = G_i \times \partial_{i-1} f_{i-1}(x_{i-1})$

How to know it?

Case 1: Reversible functions, e.g. $f(x) = x^3, \partial_x f = 3x^2 = 3(f(x))^{2/3}$

Case 2: other functions, e.g. $f(x) = \cos(x) \longrightarrow$ Cache x! **Bottleneck**!

D=12, chi=1536, each tensor: 2.7Gb

# Variational optimization with AD

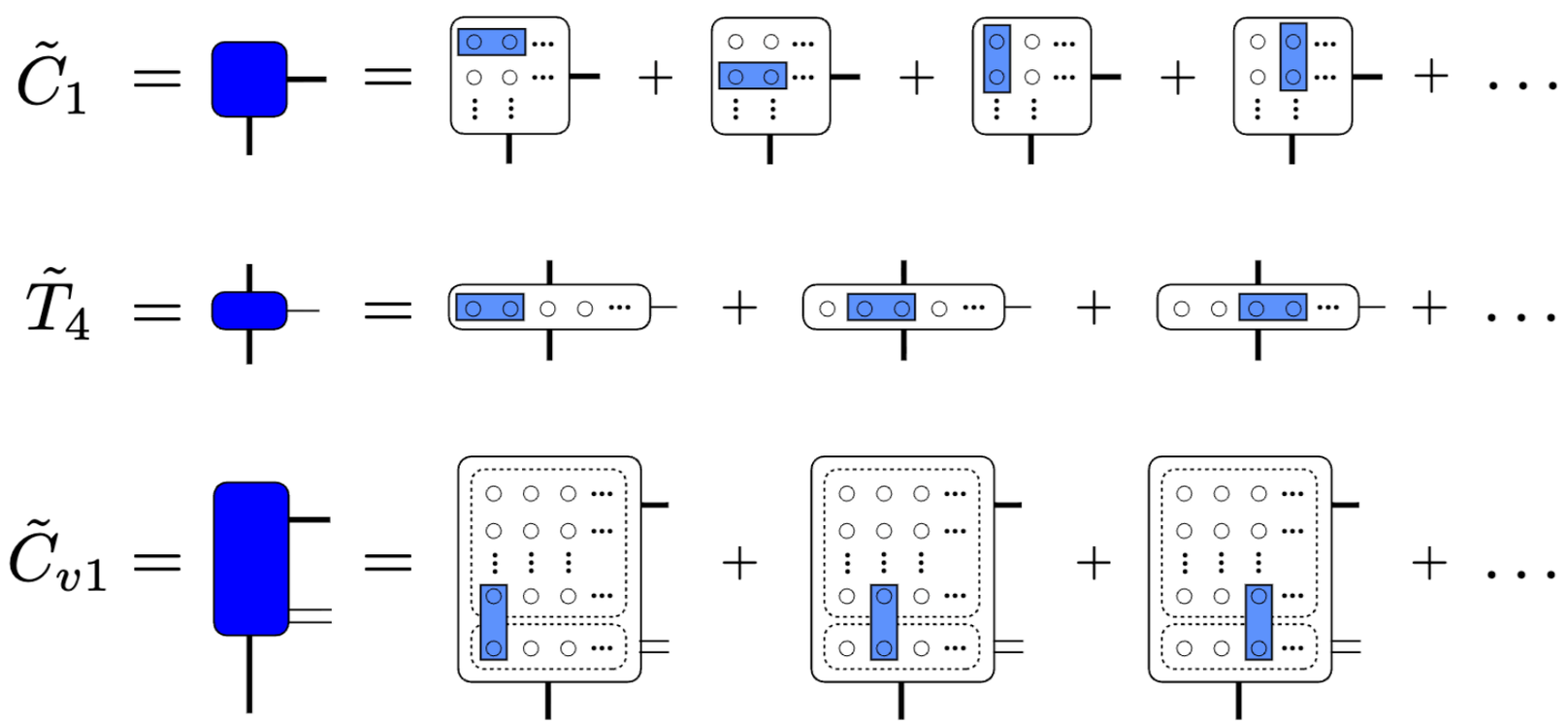
**One sentence introduction of its application for iPEPS**

Once you can calculate: $\langle \psi(A) | H | \psi(A) \rangle$

You automatically get: $\partial_A \langle \psi(A) | H | \psi(A) \rangle$

Liao, Liu, Wang, Xiang, PRX (2019)

Instead of summing over manually:



Corboz, PRB.94.035133 (2016)

# A brief introduction of AD: Forward

Vector

$$\mathbf{y} = \mathbf{x}_N \qquad \mathbf{x}_{n+1} = f_n(\mathbf{x}_n)$$

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}_1} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}_2}\frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}_3}\frac{\partial \mathbf{x}_3}{\partial \mathbf{x}_2}\frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1} = \ldots\ldots = \partial_N \mathbf{y} \times \prod_{i=1}^{N-1} \partial_i f_i(x_i)$$

But you only want to know $\Delta \mathbf{y} = \partial_1 \mathbf{y} \times \Delta x_1$

# A brief introduction of AD: Forward

Vector

$$\mathbf{y} = \mathbf{x}_N \qquad \mathbf{x}_{n+1} = f_n(\mathbf{x}_n)$$

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}_1} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}_2}\frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}_3}\frac{\partial \mathbf{x}_3}{\partial \mathbf{x}_2}\frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1} = \ldots\ldots = \partial_N \mathbf{y} \times \prod_{i=1}^{N-1} \partial_i f_i(x_i)$$

But you only want to know $\Delta \mathbf{y} = \partial_1 \mathbf{y} \times \Delta x_1$

Step 1: $x_1, \Delta x_1$

Step i: $x_{i+1} = f_i(x_i) \qquad \Delta x_{i+1} = \partial_i f_i \times \Delta x_i$

**Faster and Cache Free!**

# A brief introduction of AD: Unified framework

$$\xrightarrow{\text{Bwd}} \quad \text{🍎} \times \prod_i \frac{\partial x_{i+1}}{\partial x_i} \times \text{🍊} \quad \xleftarrow{\text{Fwd}}$$

Backward mode:

| | | |
|---|---|---|
| 🍎 | $\dim\ 1$ | Energy |
| 🍊 | $\dim\ N$ | Gradients |

Forward mode:

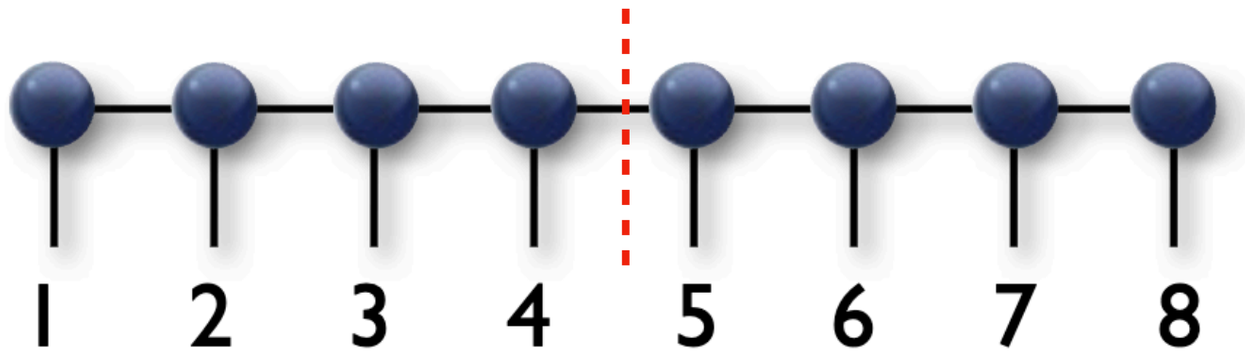
| | | |
|---|---|---|
| 🍎 | $\dim\ N$ | Tangents |
| 🍊 | $\dim\ 1$ | Step size |

# From MPS to PEPS: entanglement entropy

Gapped 2D system Area Law(argued): $S \sim L$

1D MPS $|\psi\rangle =$

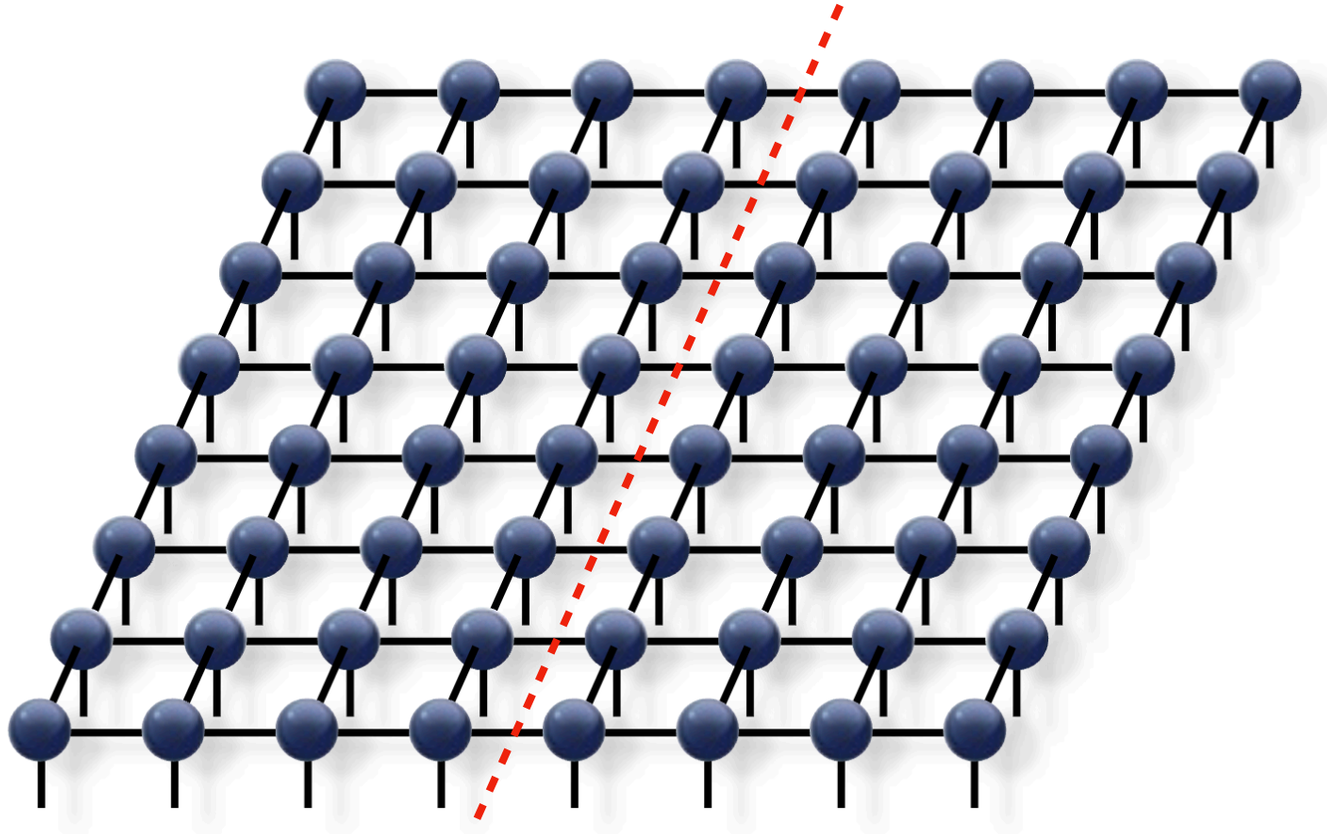
$S < \ln(D)$




**Matrix Product states**

2D PEPS $|\psi\rangle =$

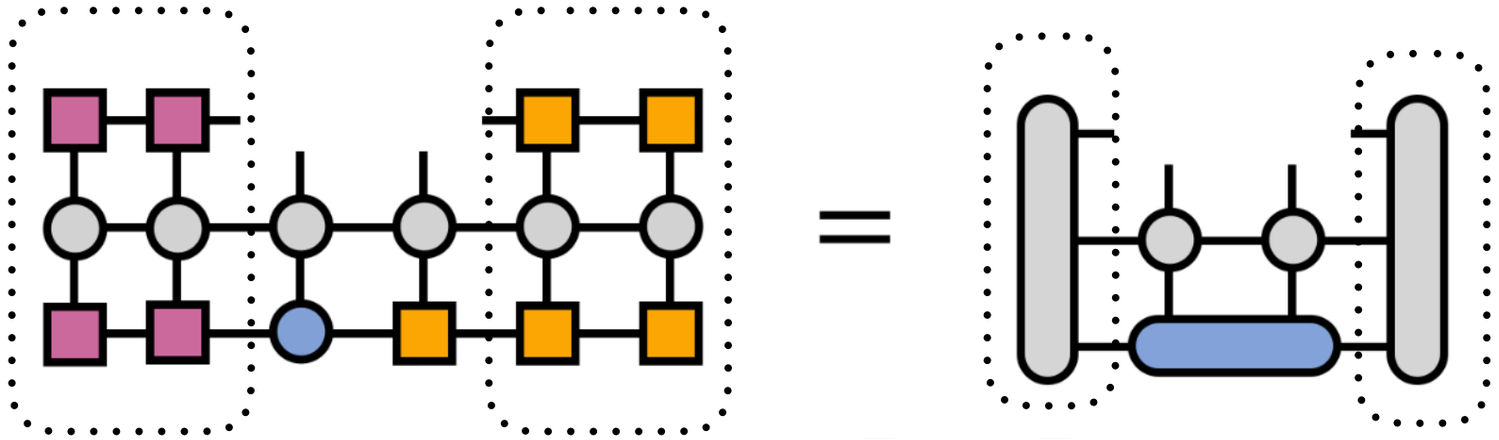
$S < L \ln(D)$



**Projected entangled pair states**

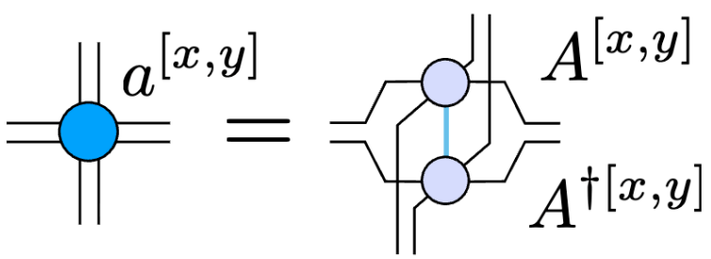
# Environments: Exact vs Approximated.

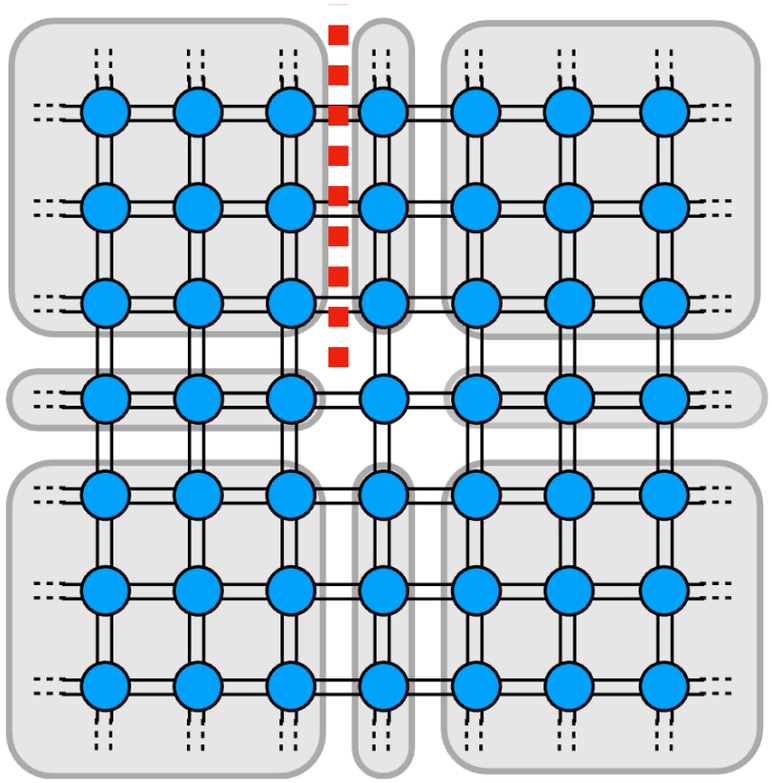
1D MPS

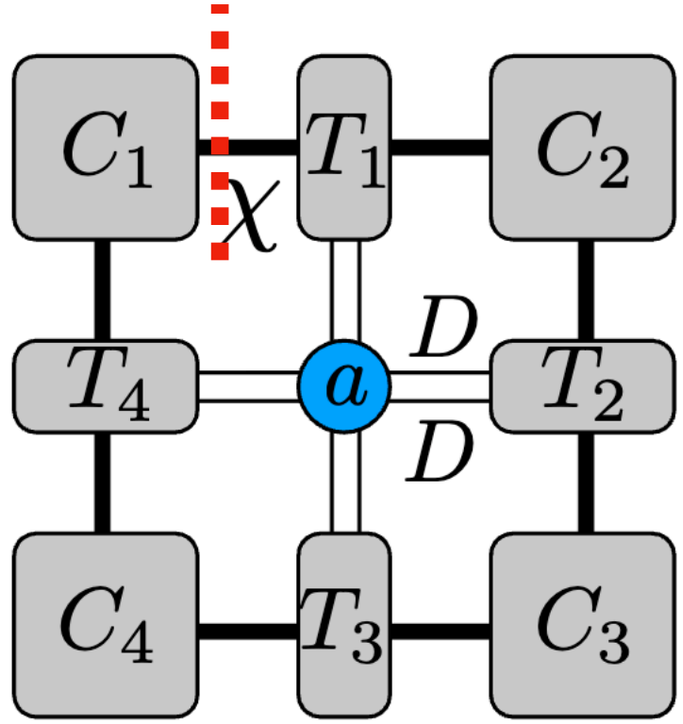


Exact Env

2D PEPS

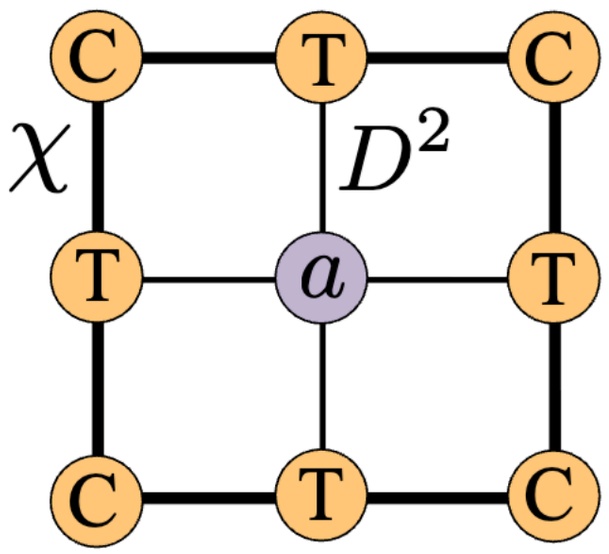


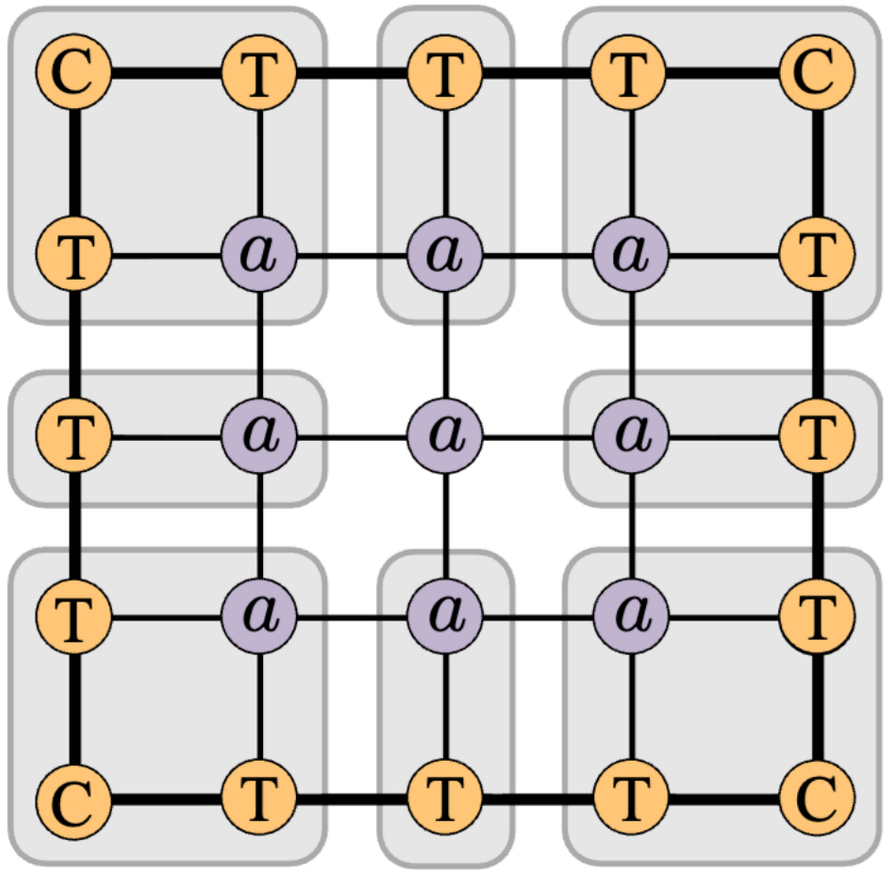



Approximated env

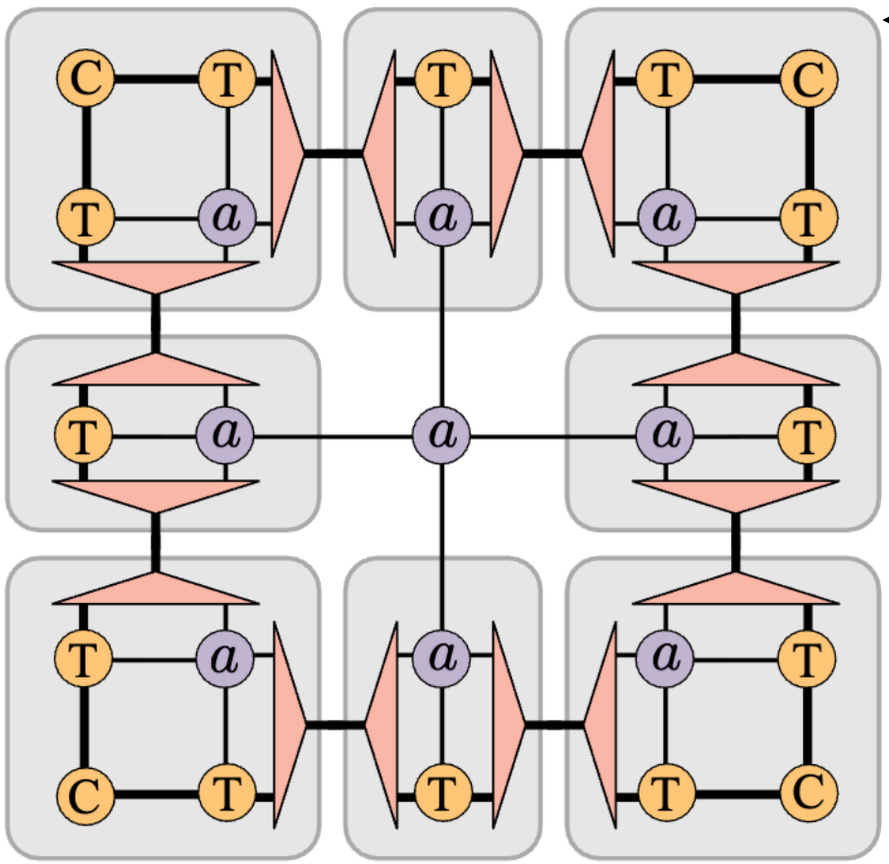
# Textbook C4v CTMRG iterations



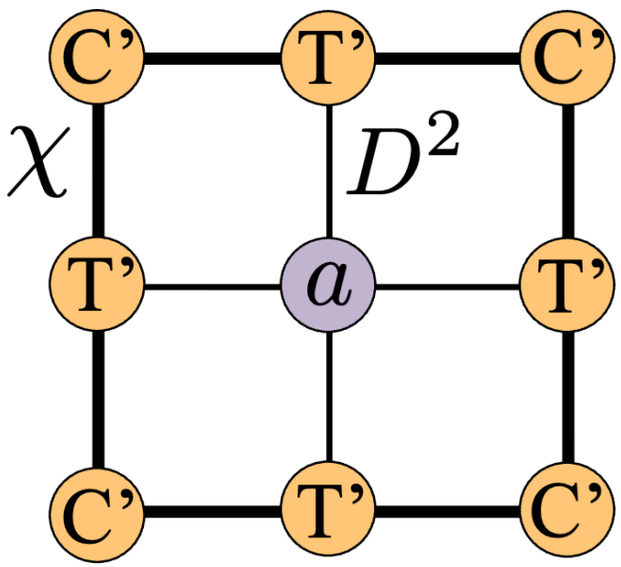

Expand



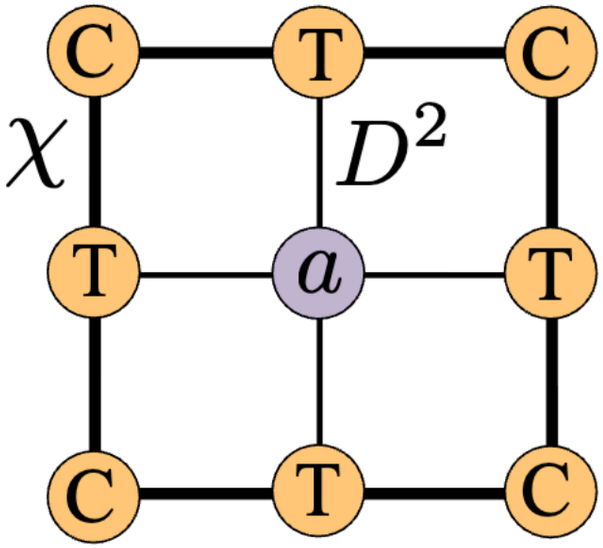

Truncate



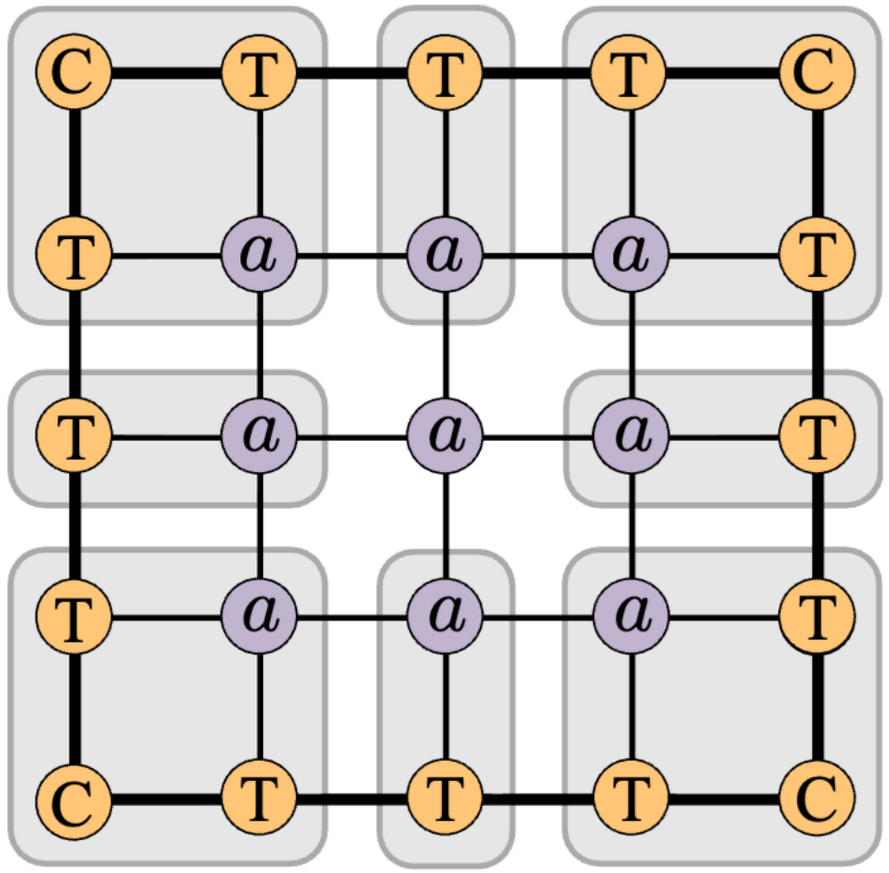

Update



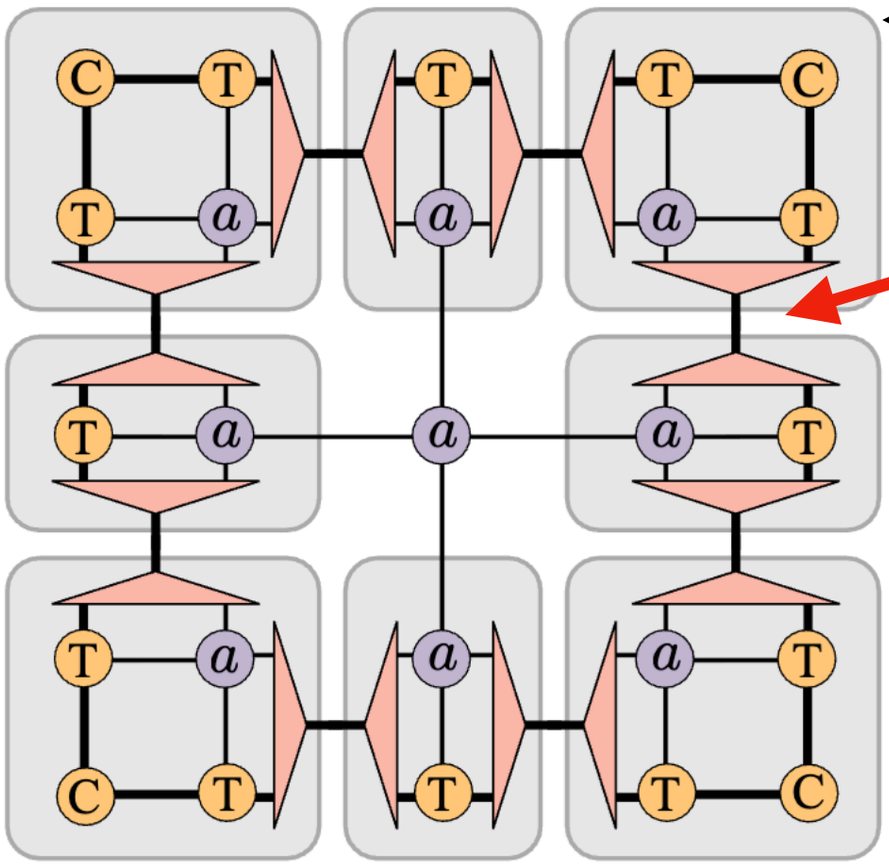
# Projectors?




Expand

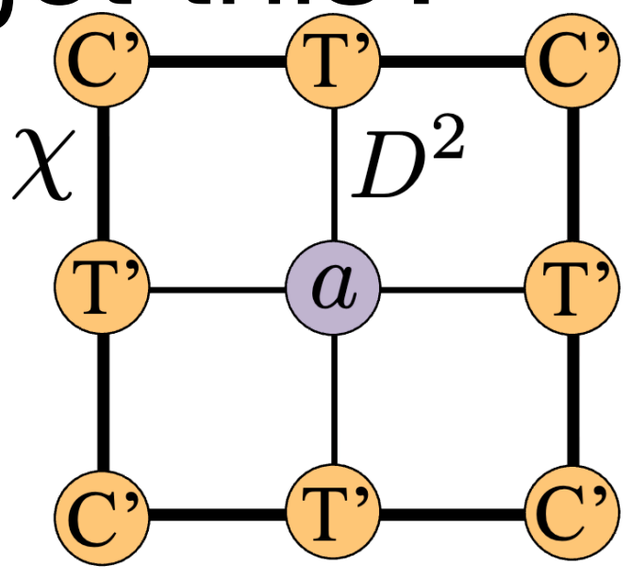



Truncate

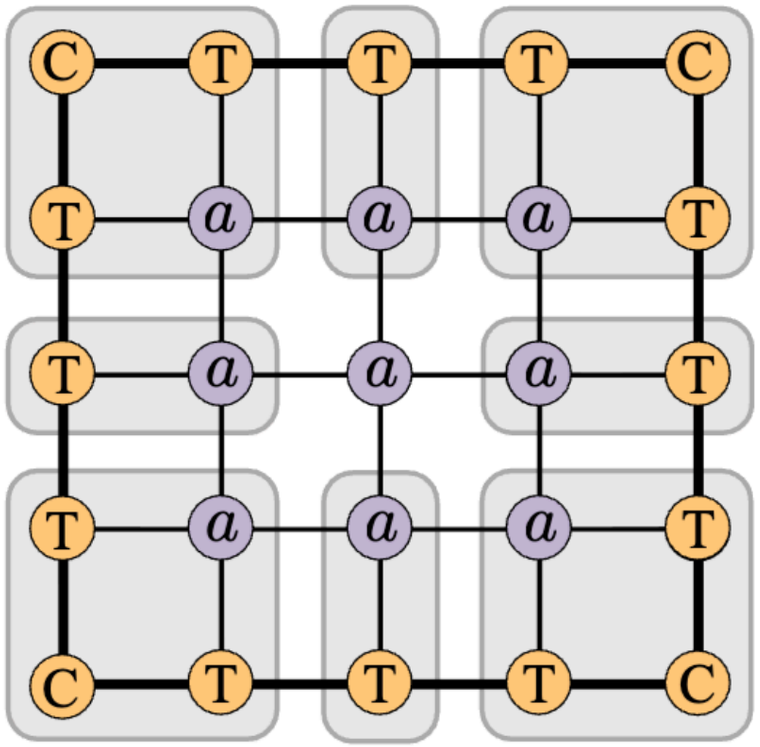


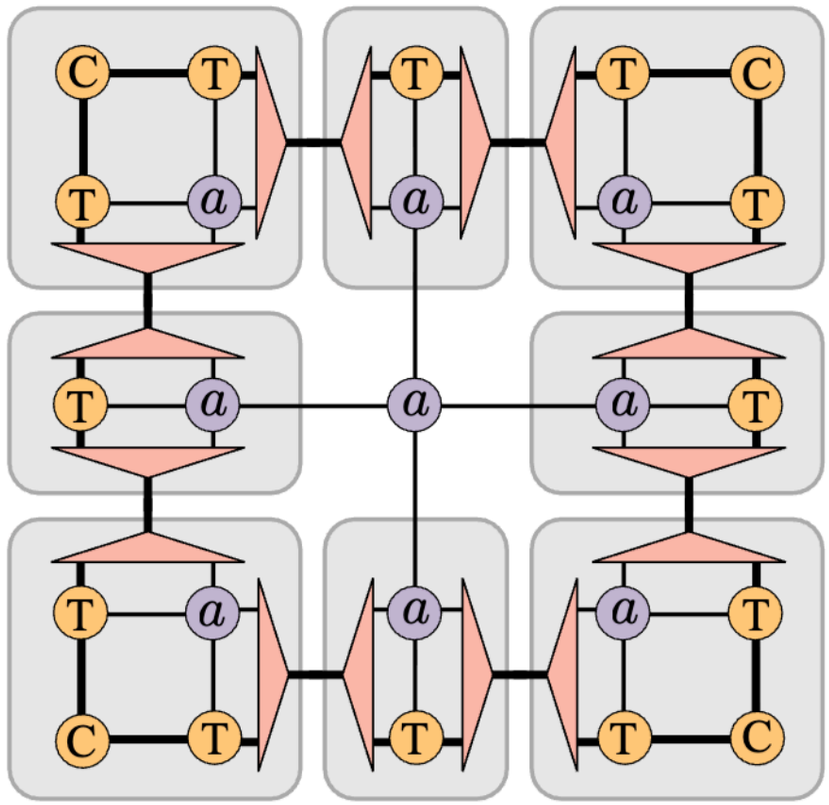

? How to get this?

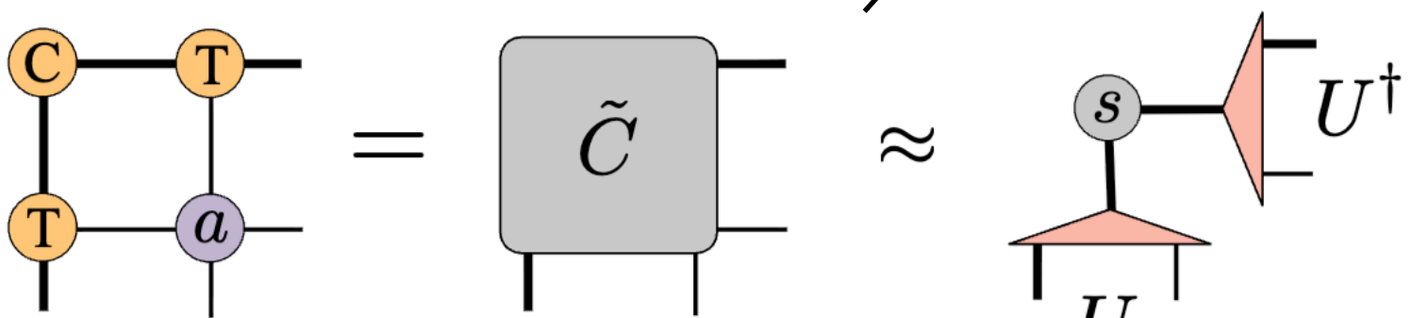
Update



# By diagonalizing enlarged corner



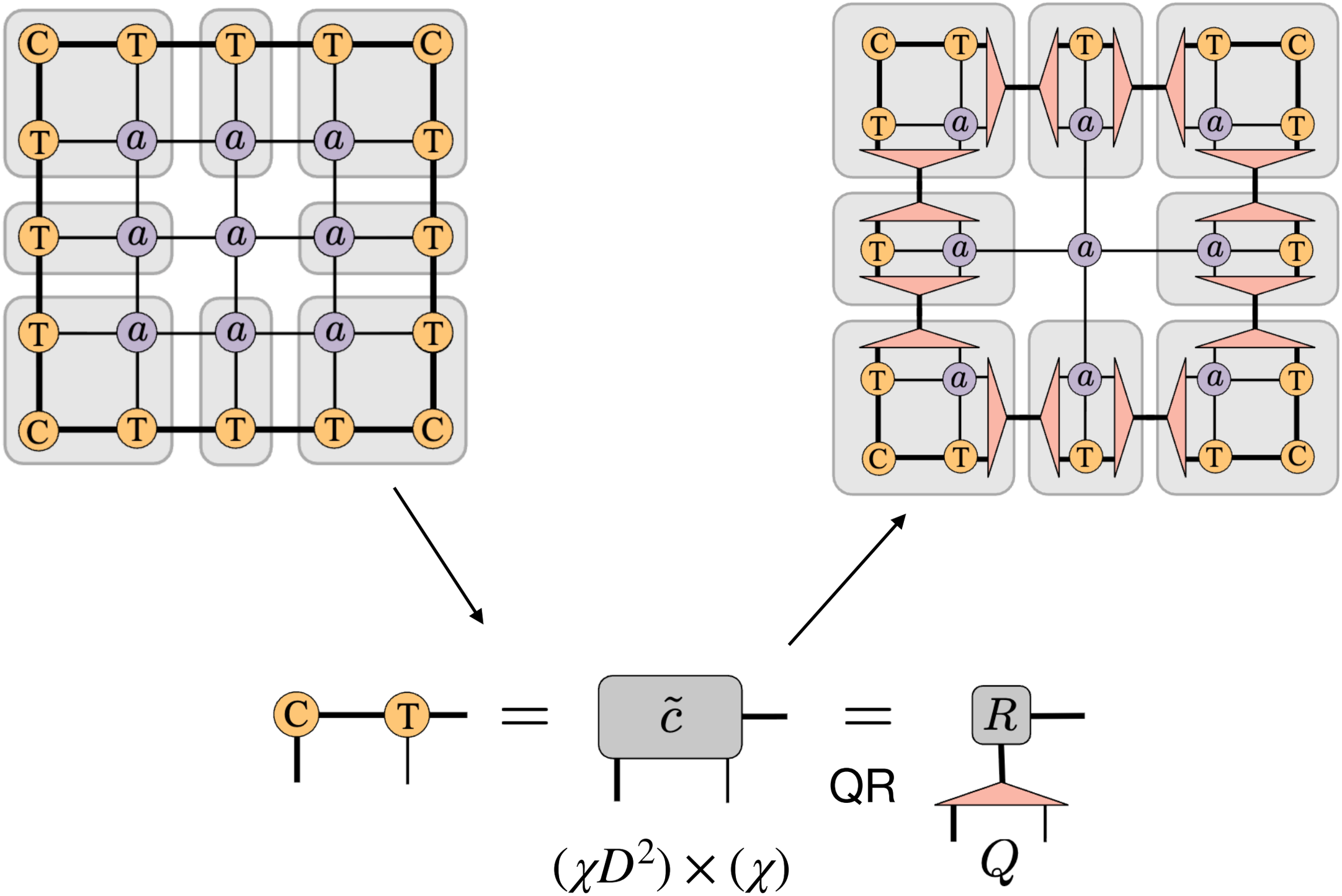




$(\chi D^2) \times (\chi D^2)$ SVD/eigh

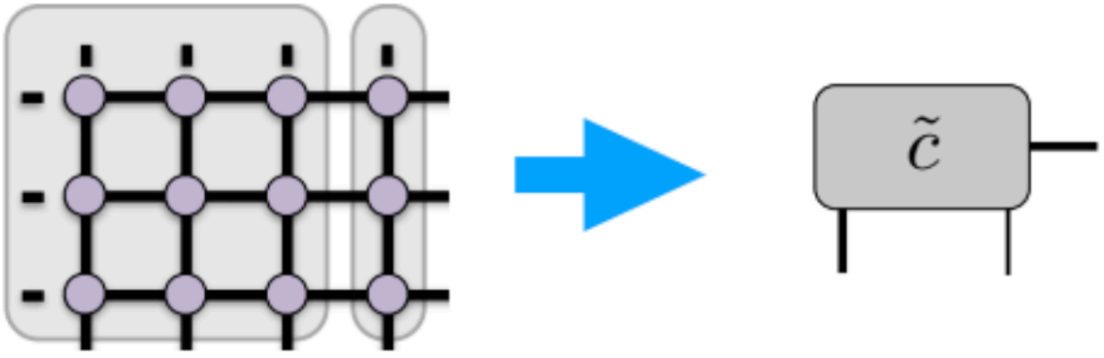
Nishino & Okunishi, **1996** , JPSJ 65, 891

# A method hiding in plain sight for 29 years?



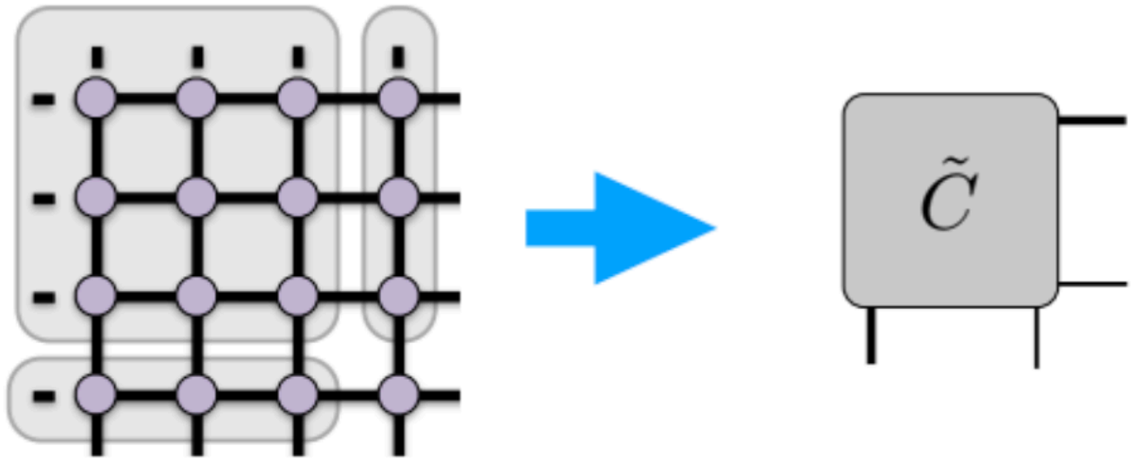

Y. Zhang* & Q. Yang* & P. R. Corboz (**2025**) arXiv:2505.00494

# Why this can work?




Reduced corner

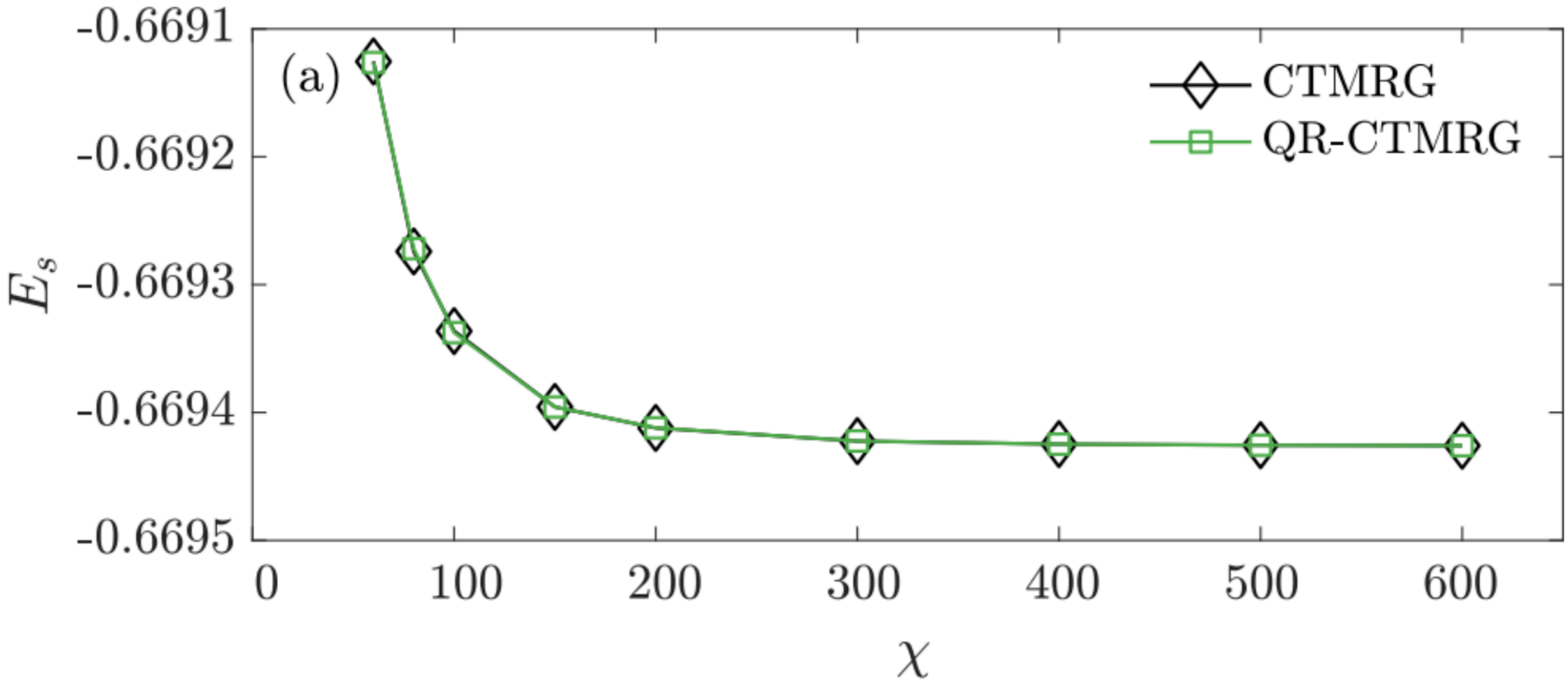
QR: $(\chi D^2) \times (\chi)$




Enlarged corner

SVD: $(\chi D^2) \times (\chi D^2)$

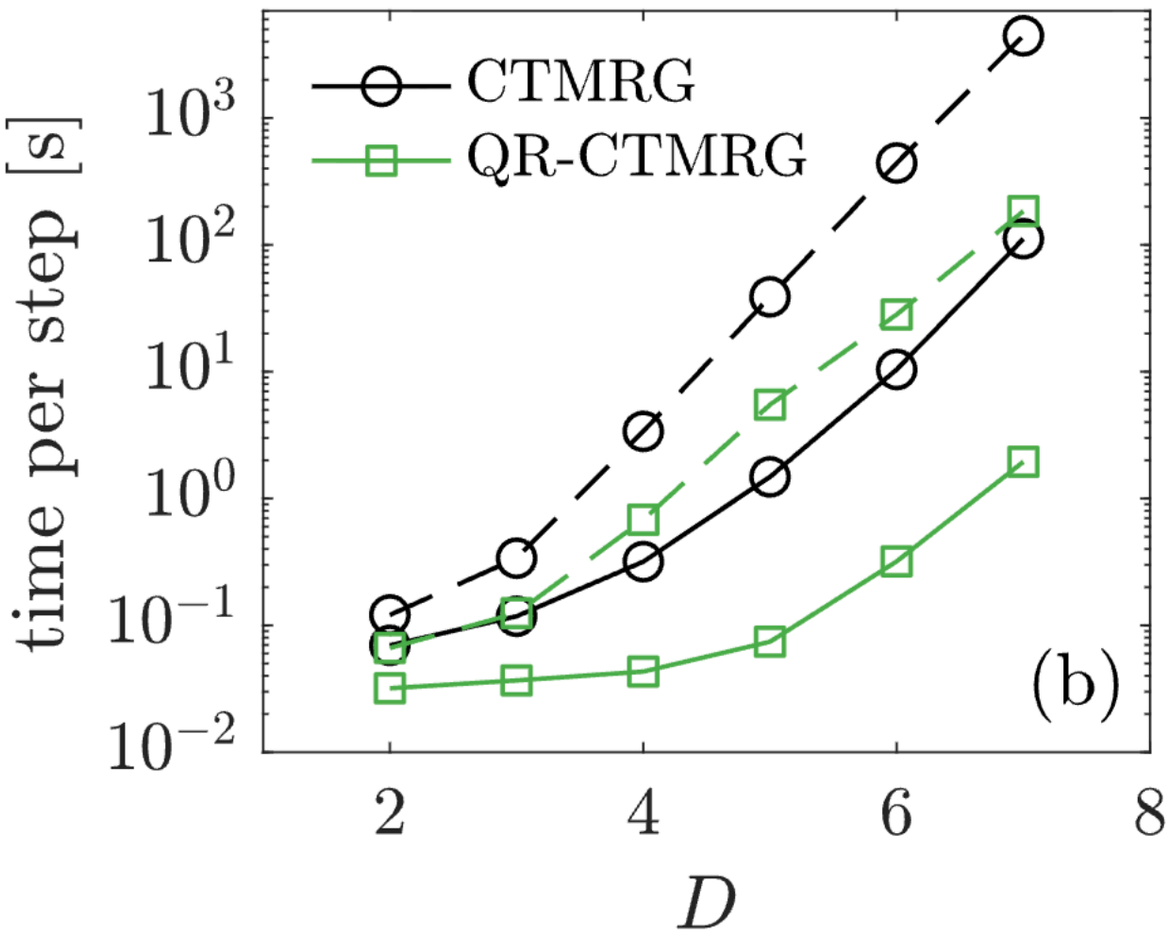
# Benchmark with SVD-CTMRG

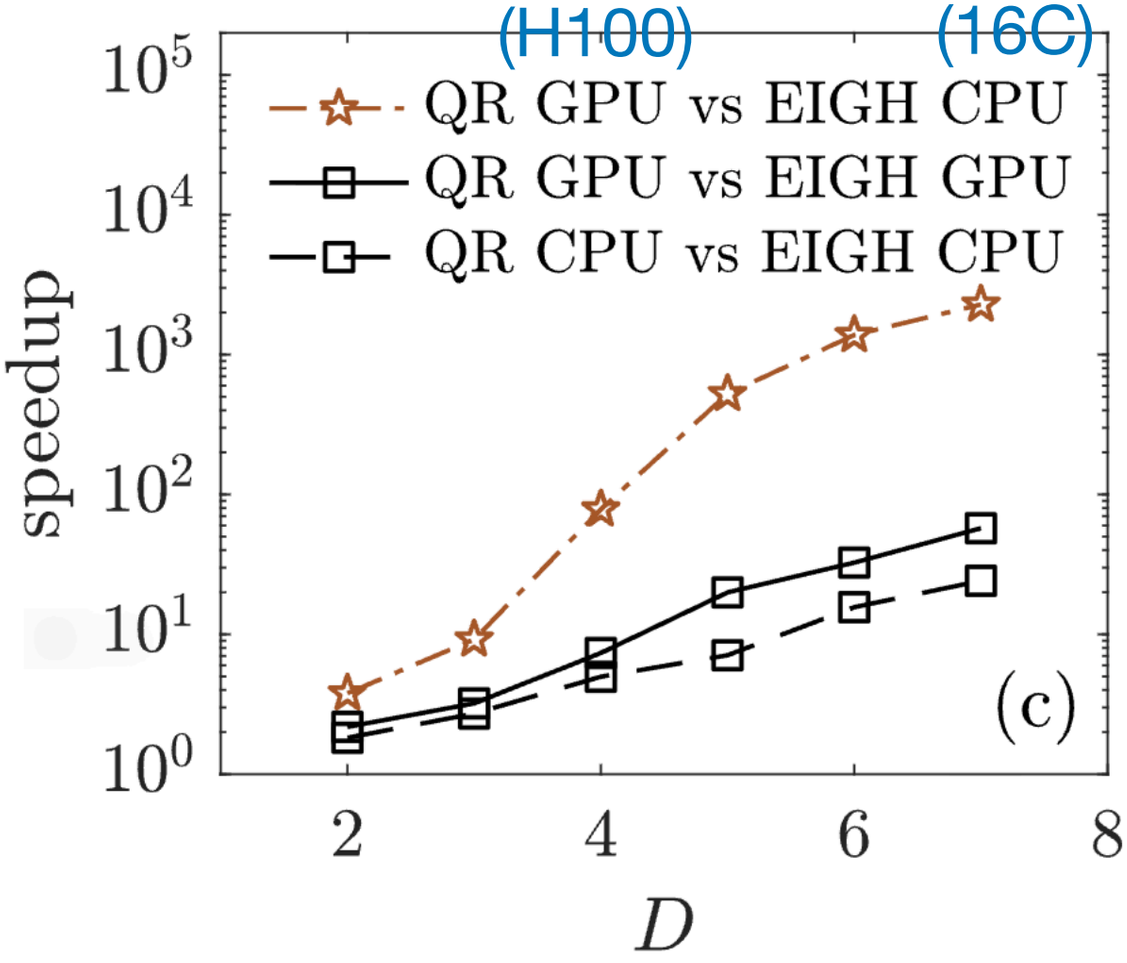



2D Heisenberg model, D=6

# Order of magnitude optimization speed up

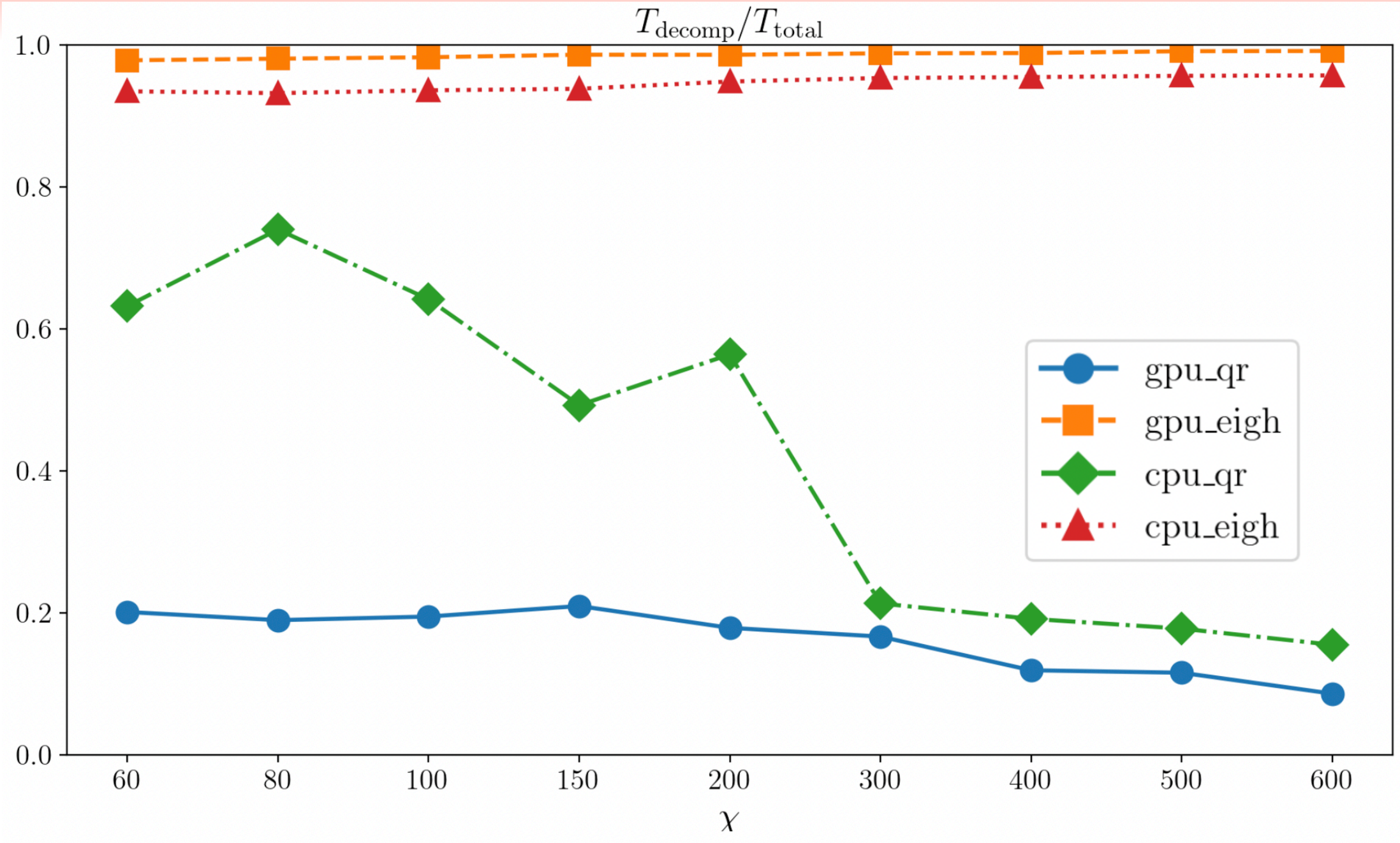
2D Heisenberg model, $\chi = 5D^2$




Even complexity are the same, pre-factor of SVD is very large

# Reason: Contraction become dominant



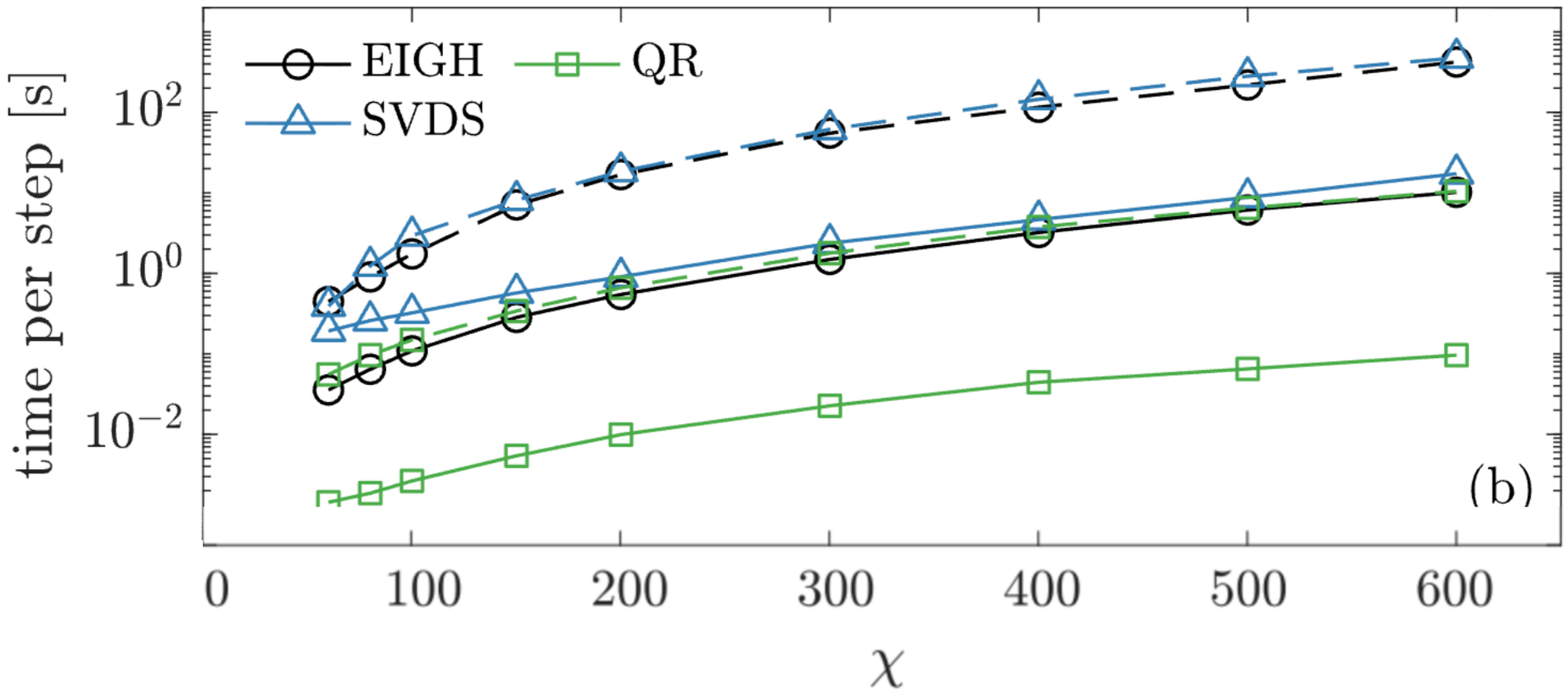

🚀 GEMM

🚗 Reshape, QR

🐌 SVD, Eigh

D=6, real, H100

# How about SVDs?



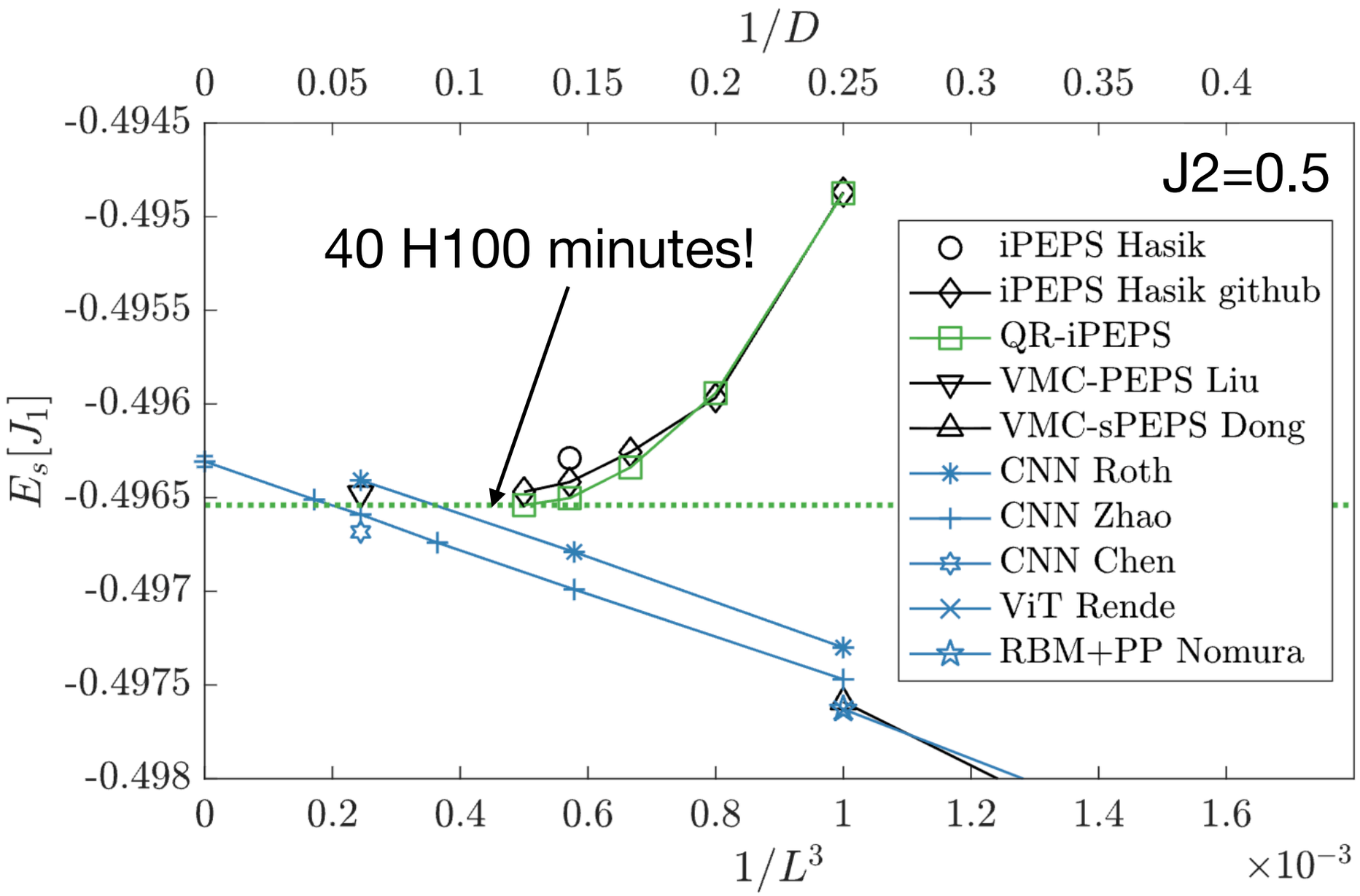

QR : Householder reflection

SVD : Householder reflection + implicit QR iterations to converge

SVDs: A very large Krylov space construction

D=6, real, H100

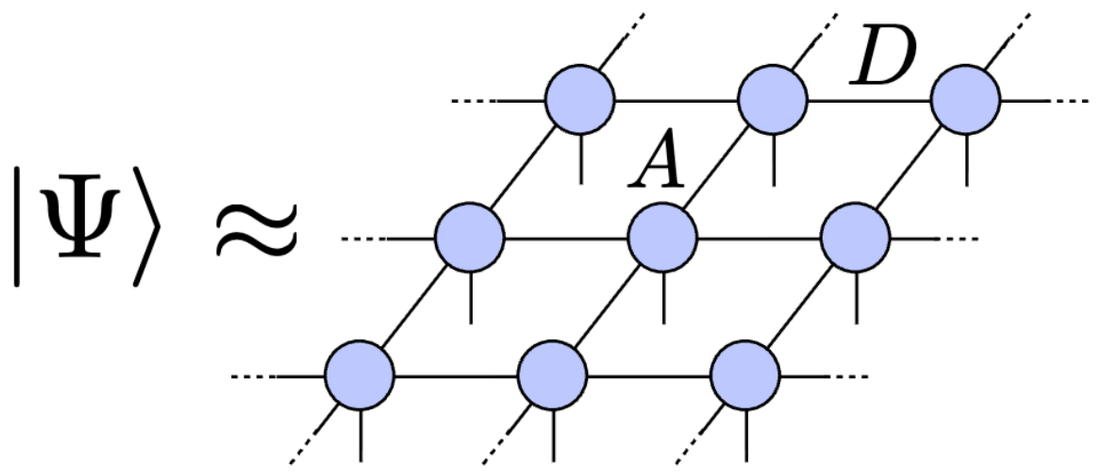
# J1-J2 on Square lattice: SOTA



# Why wasn't QRCTM invented earlier?

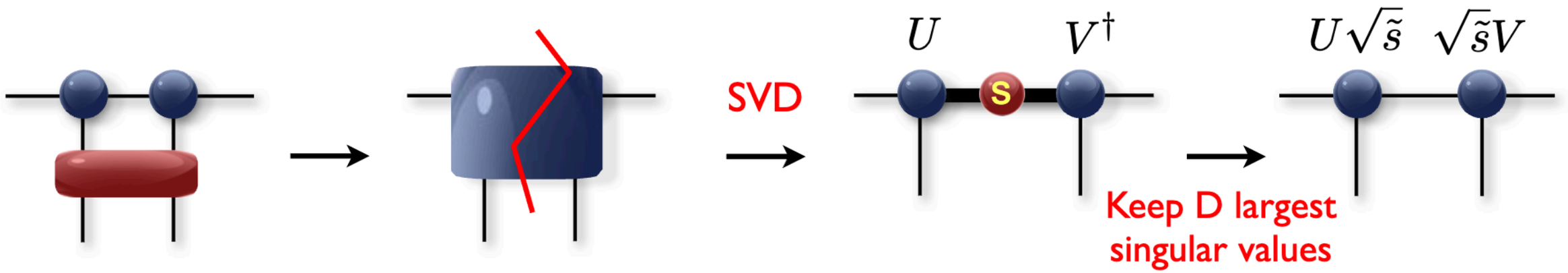
- Classical: Critical.
  Preferred: TRG, Fix point approach.

- Quantum:
  This ansatz was also overlooked!

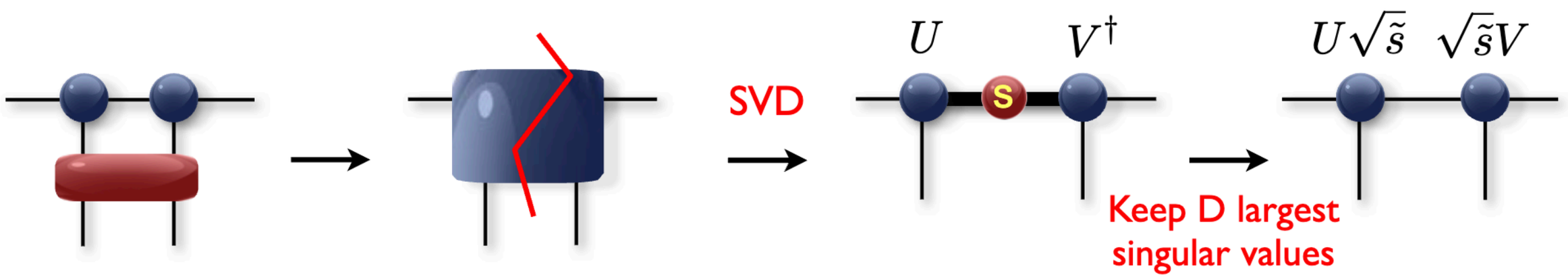


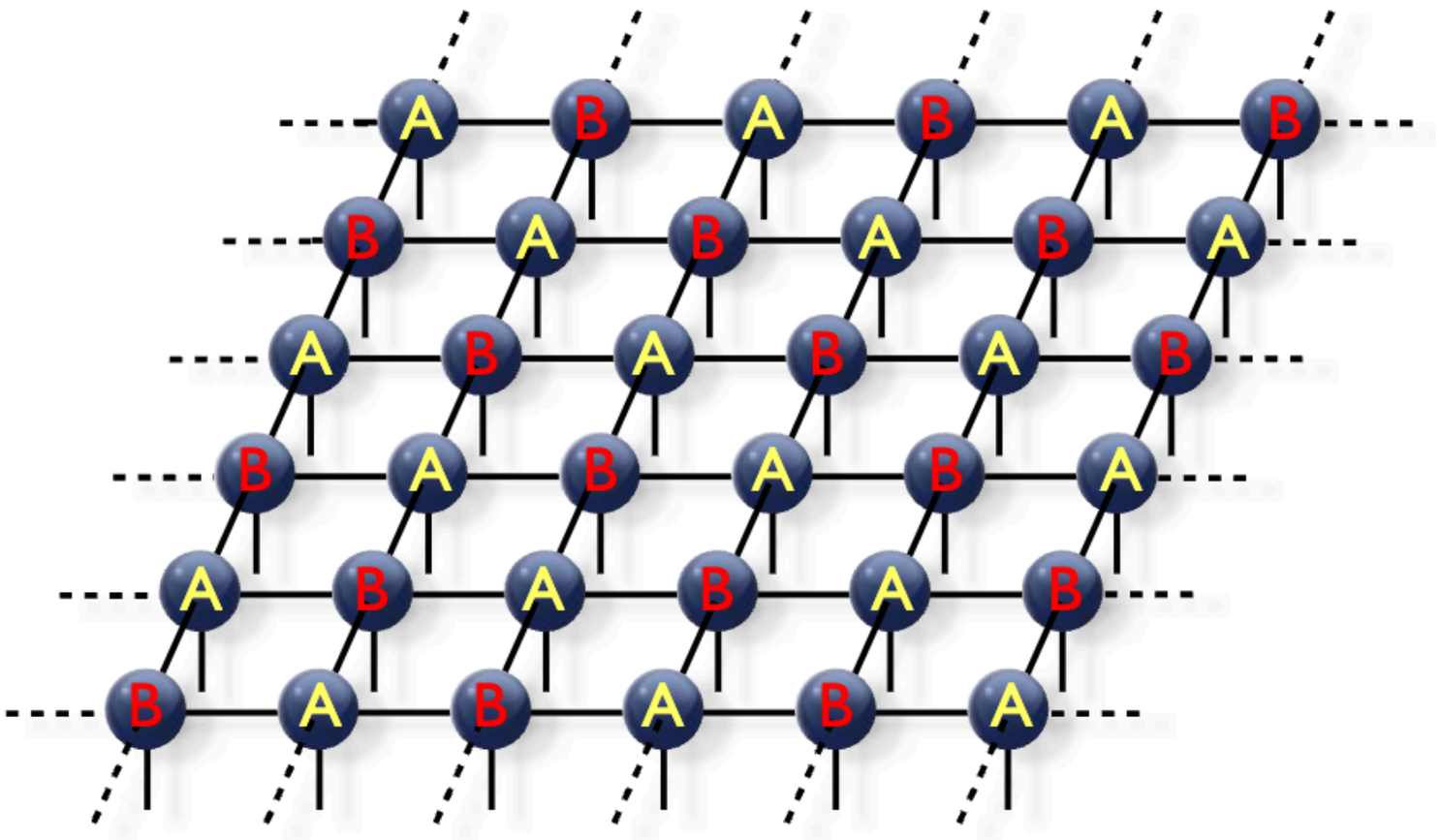
# ITE in iPEPS breaks lattice symmetry!

Imaginary time evolution



# ITE in iPEPS breaks lattice symmetry!




ITE will not give you a C4v iPEPS.

**But AD enable it!**

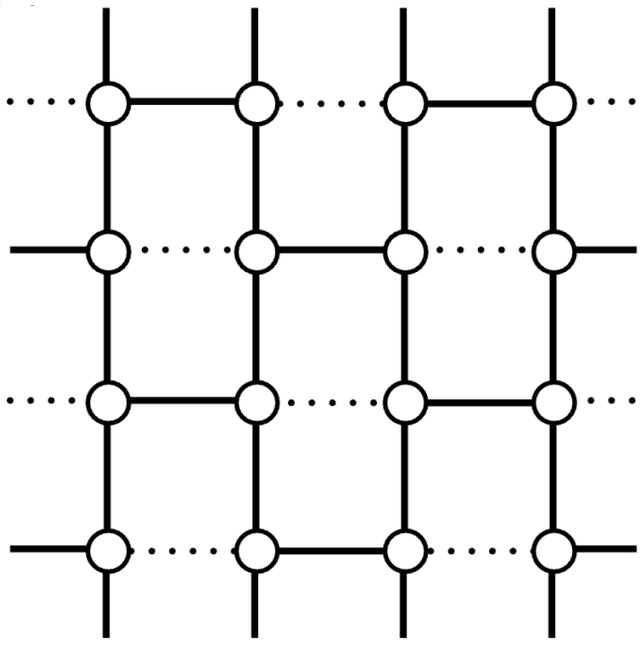
# A halfway summary

- **QRCTM** and **GPUs** accelerate the contraction of C4v symmetric iPEPS
- **AD** provides practical algorithms to optimize this iPEPS.
- **SVD Free**: Free from "$S^{-1/2}$" instability.
- Avoid "**large unit-cell**" and "**bi-orthogonalization**", the origins of instability in iPEPS.
- Everything is fine, but,

  **where is its killing application**?

# iPEPS on the Honeycomb Lattice: Brick-Wall Approach

## Brick Wall



PhysRevX.2.041013

## Large Unit Cell VUMPS

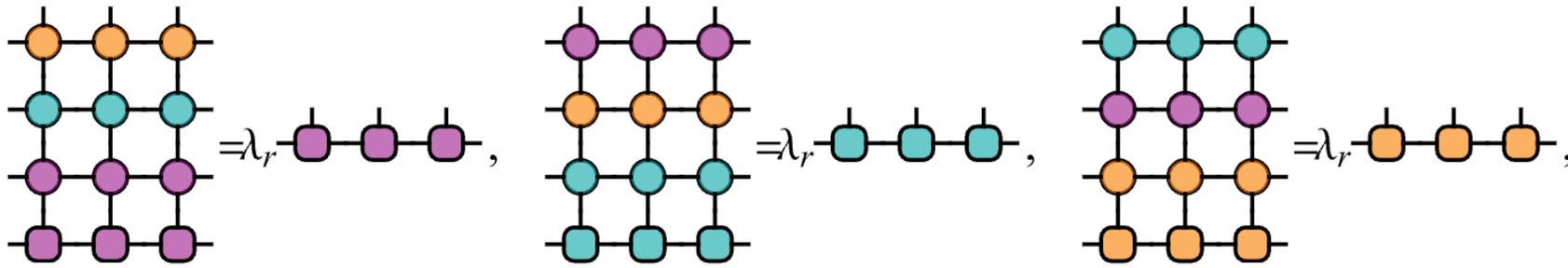


Fig: PhysRevB.108.085103

## Large Unit Cell CTMRG

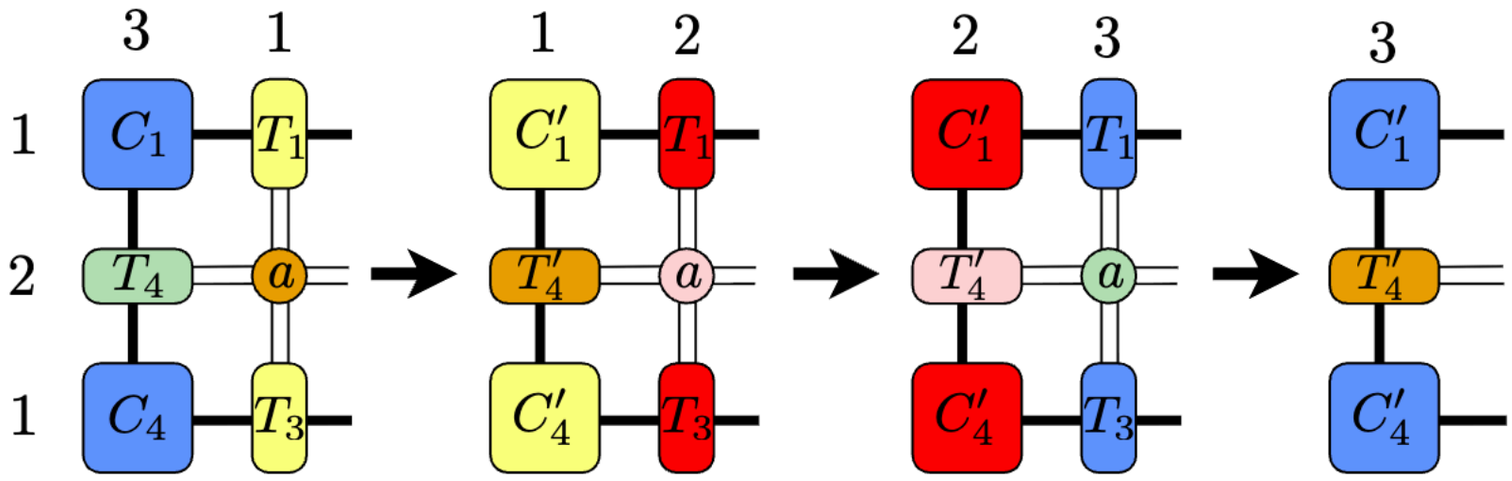
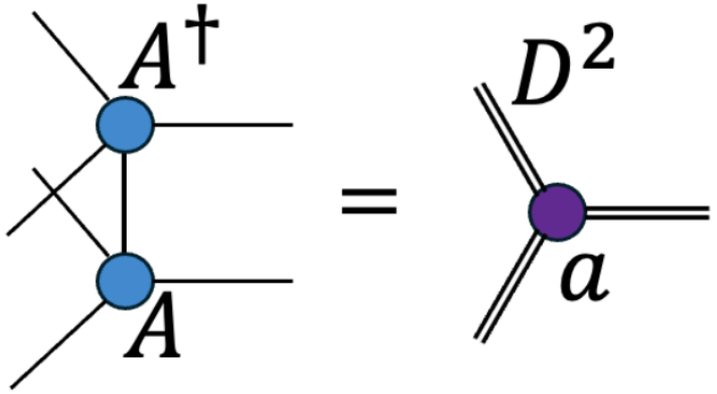



Fig: Philippe Corboz's slides@tnqmp2016

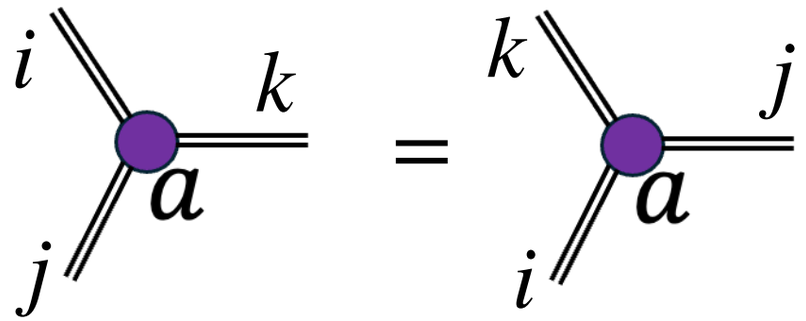
**Complex Coding, QR magic is lost.**

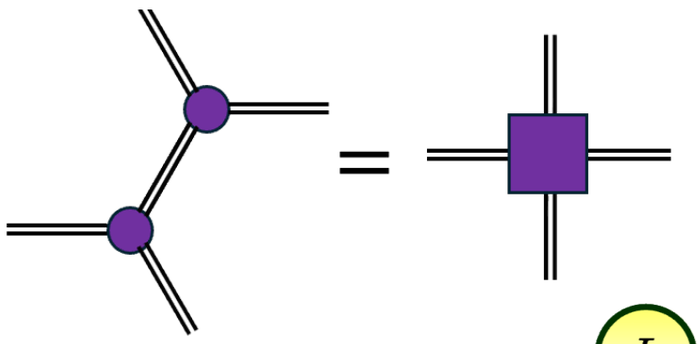
Figures shown for illustration only; credit for large–unit–cell algorithms goes to the original authors.

# Uniform $C_{3v}$ CTMRG on Honeycomb Lattice

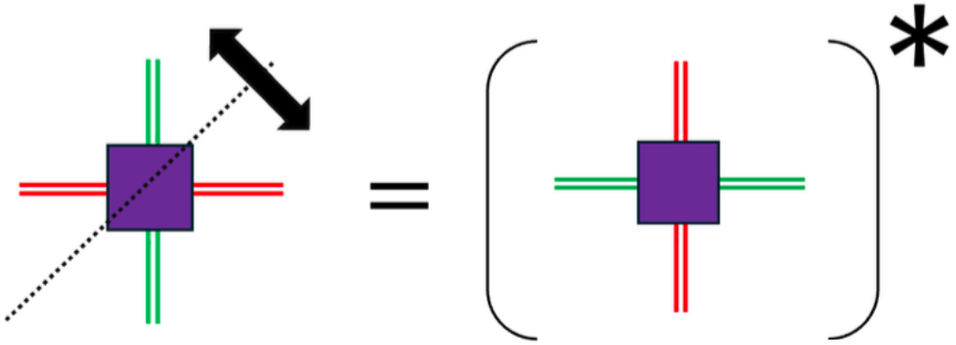


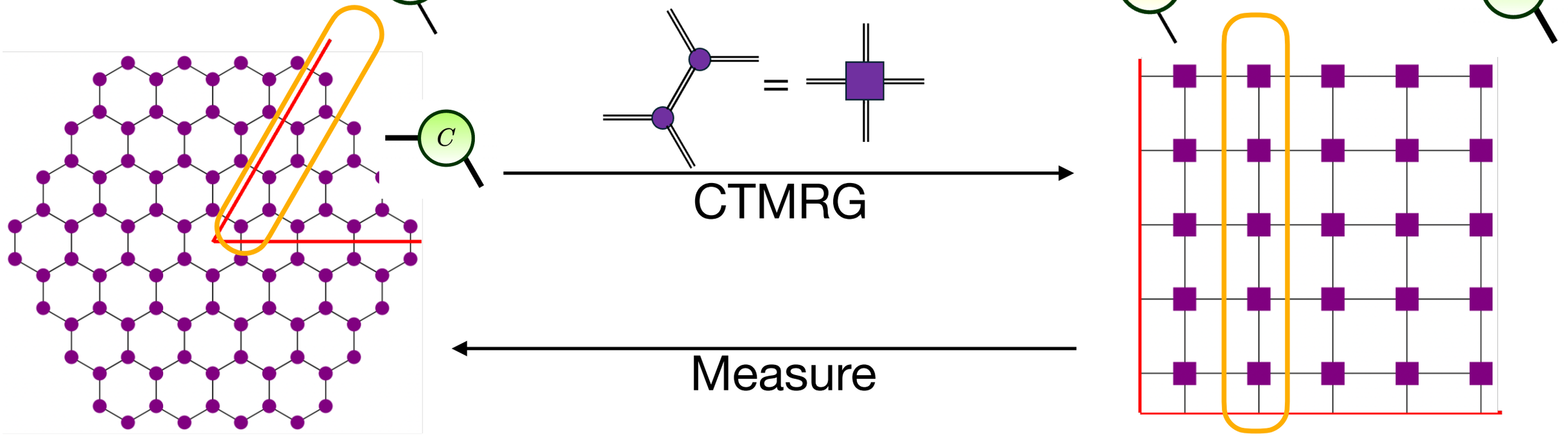

Rot. Symm. :



Refl. Symm. :






Thanks to our Ukrainian colleagues—their work paved the way

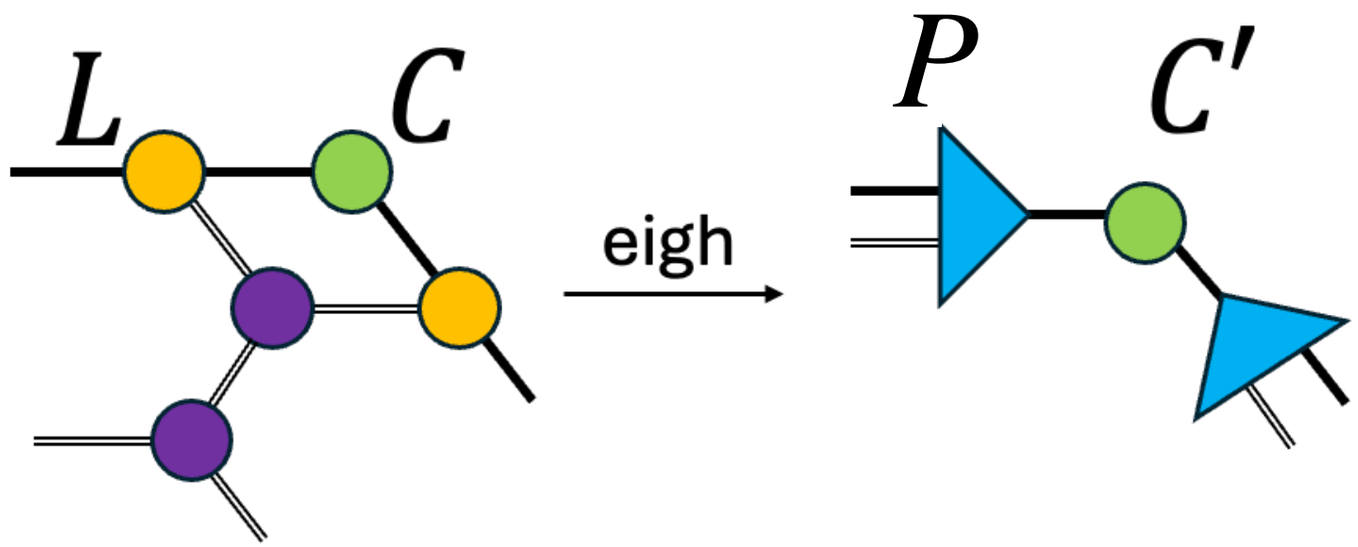
Lukin et al., PhysRevB.107.054424(2023)

# CTMRG for uniform $C_{3v}$ iPEPS

Lukin et al., PhysRevB.107.054424(2023)

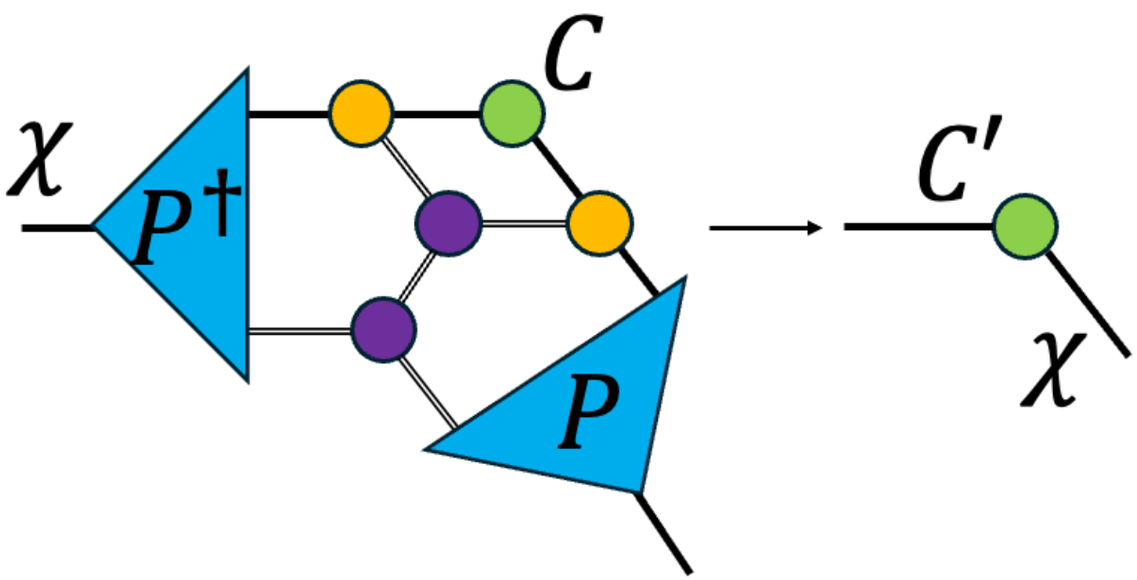
## Step 1: Build projector. (expensive EVD)

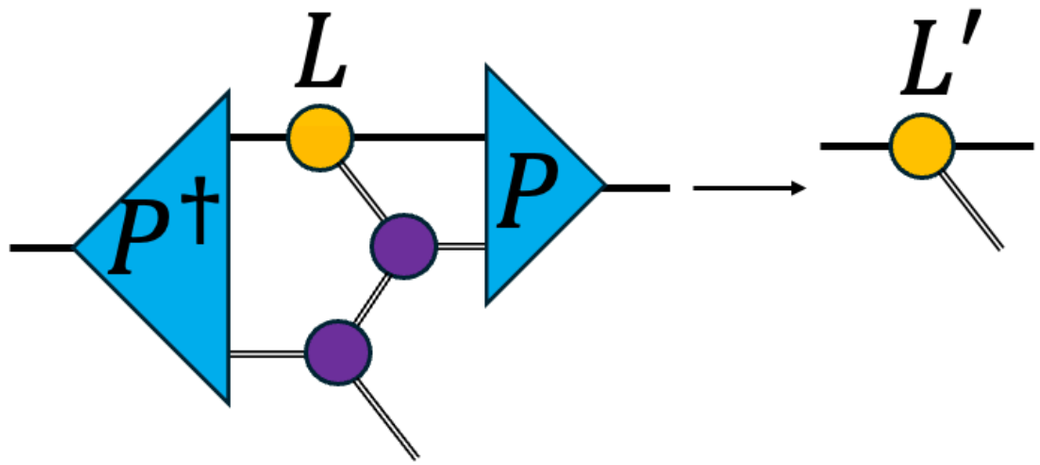
~95% time




Step 2: Update edge and corner tensors.

~ 5% time



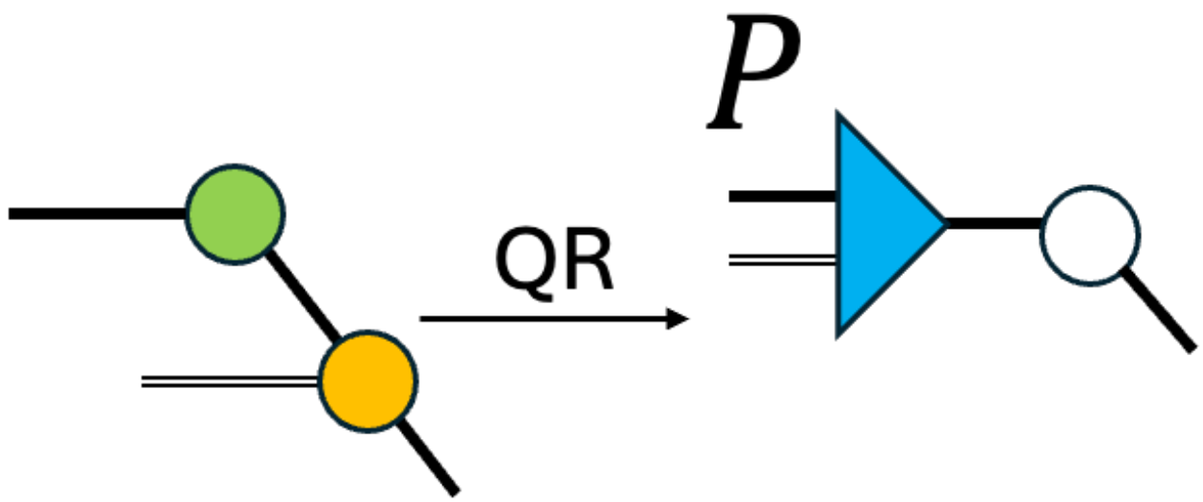




H100, double, very large D>6 and chi>600

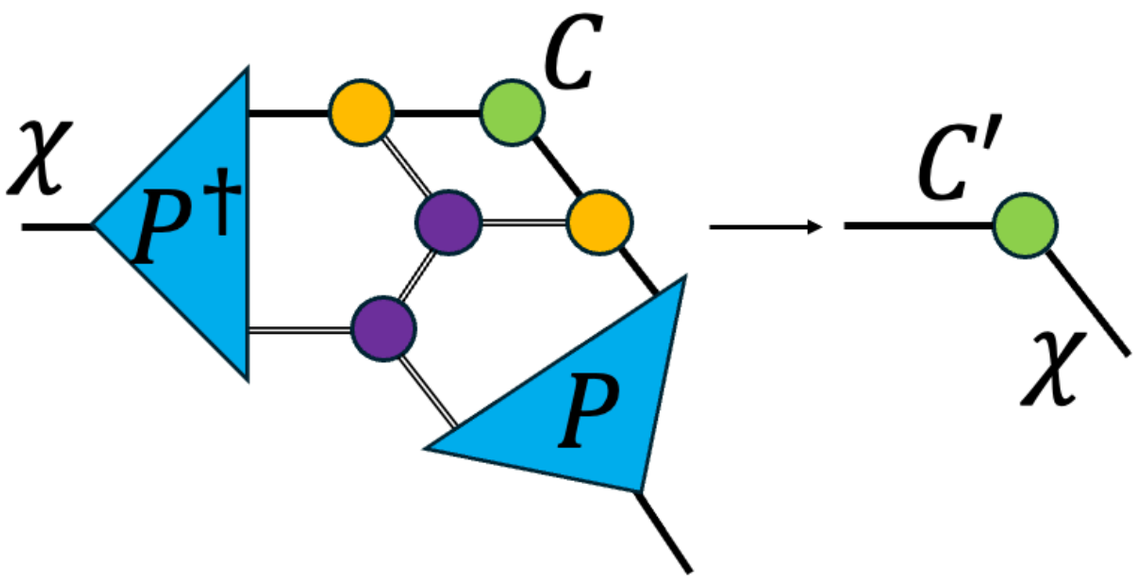
# QRCTM for uniform $C_{3v}$ iPEPS

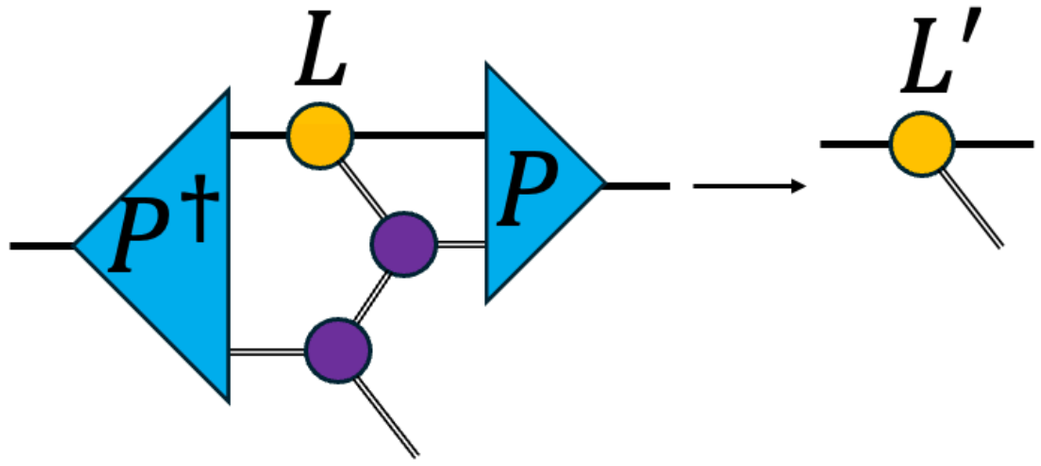
arXiv:2509.05090

Step 1: Build projector via QR (cheap) ~10% time




**Step 2: Update edge and corner Tensors.** ~90% time



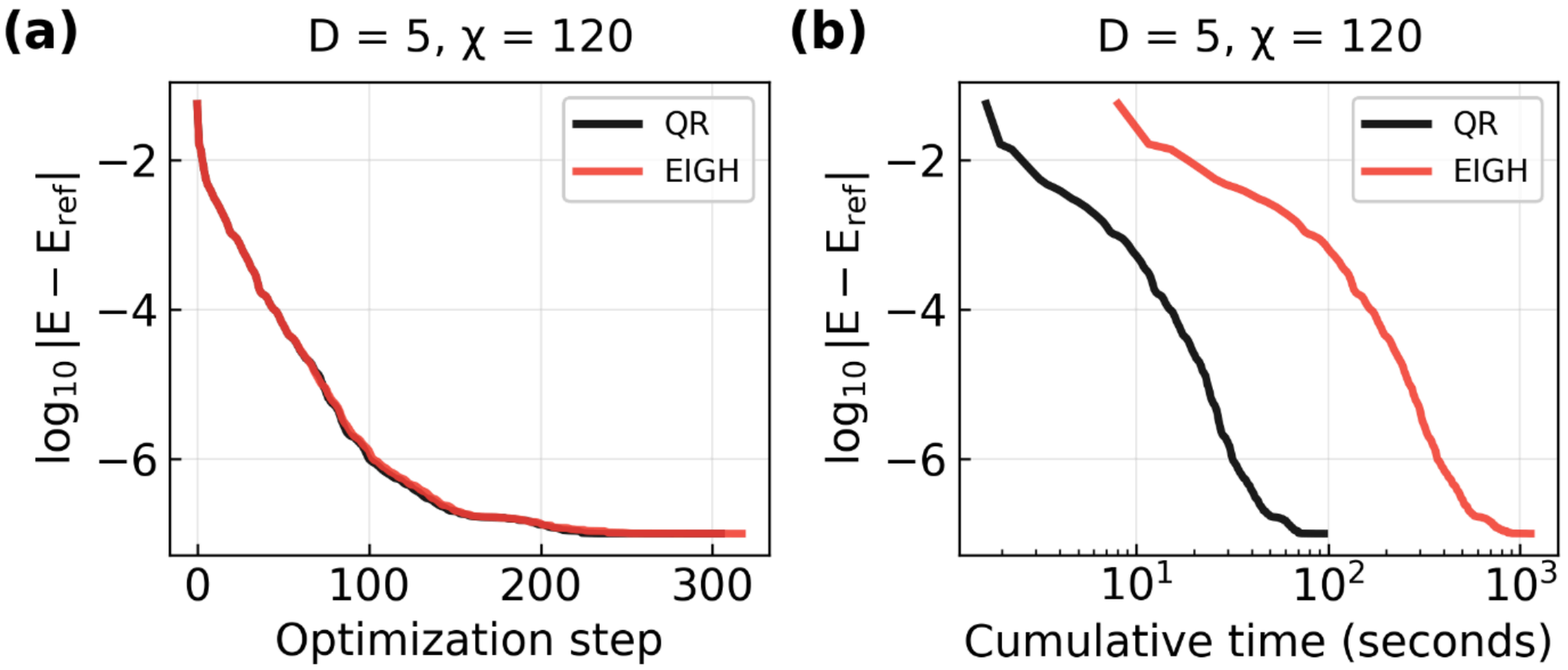

H100, double, D=6 and chi=600

# AFM Heisenberg on the Honeycomb Lattice

$$H = \sum_{\langle ij \rangle} \vec{\mathbf{S}}_i \cdot \vec{\mathbf{S}}_j$$

Training Dynamics

Speed up

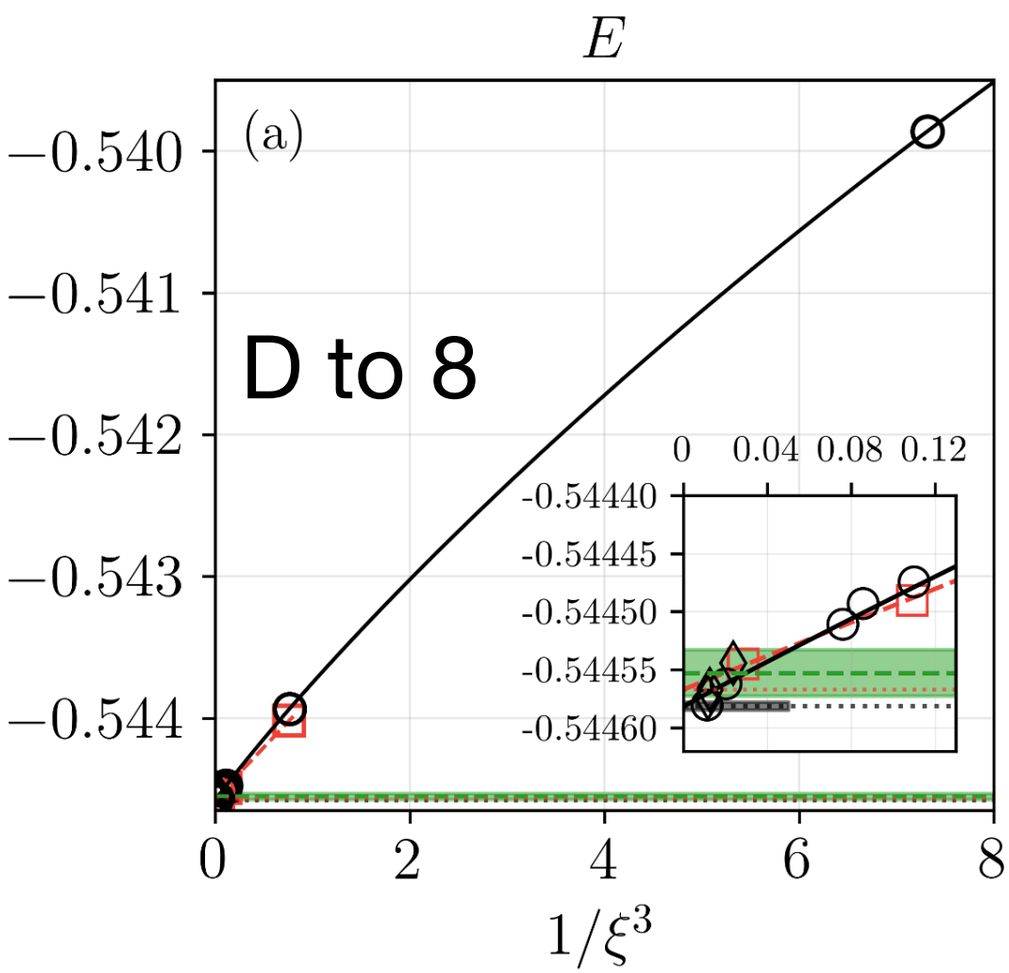


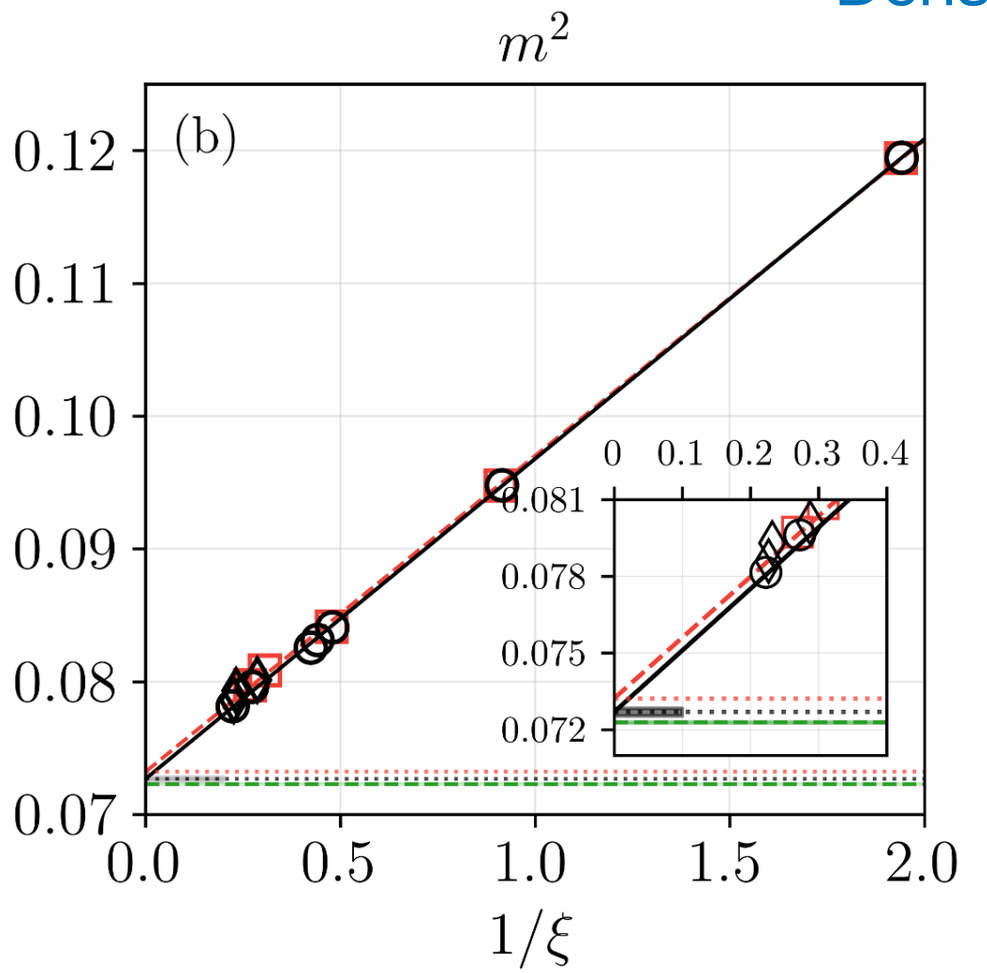

**Similar training dynamics, SUPER FAST!**

H100, double, real, same initialization, Eref=Emin+1E-7

# AFM Heisenberg on the Honeycomb Lattice

"Dense U1"




"Lower" $\longrightarrow$ $E_0 = -0.544582(3)$ $\qquad$ $m_0 = 0.2696(3)$

$E_{QMC} = -0.544553(20)$ $\qquad$ $m_{QMC} = 0.26885(2)$

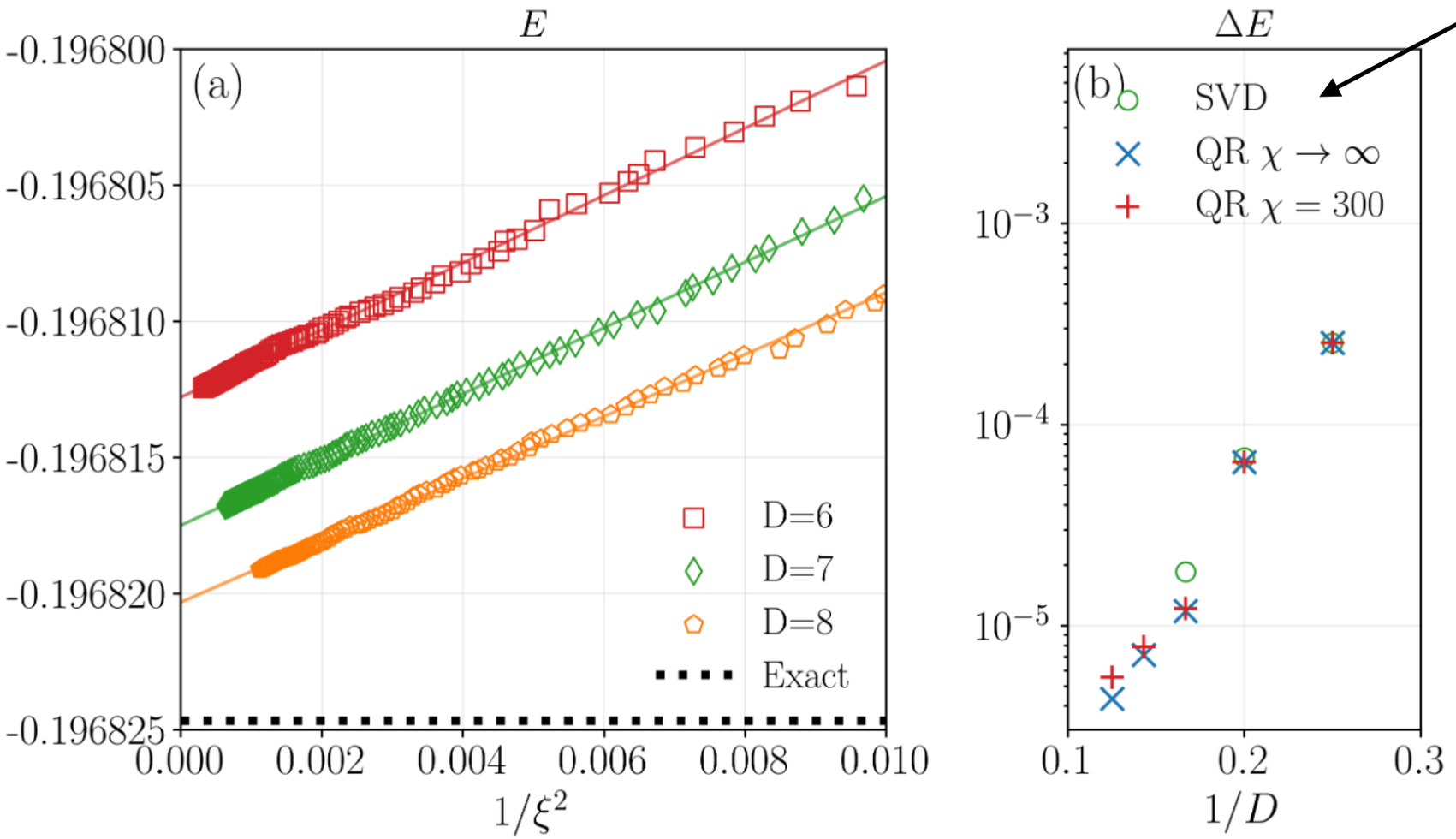
**D=8, accurate results—consistent with QMC**

F. Jiang, Eur. Phys. J. B 85, 1(2012)
U. Low, Condens. Matter Phys. (2009)
Both QMC and our iPEPS results are extrapolated (FSS, FCLS)

# Isotropic Kitaev model (S=1/2) energy

j.aop.2005.10.005

Lukin et al., PhysRevB.107.054424(2023)







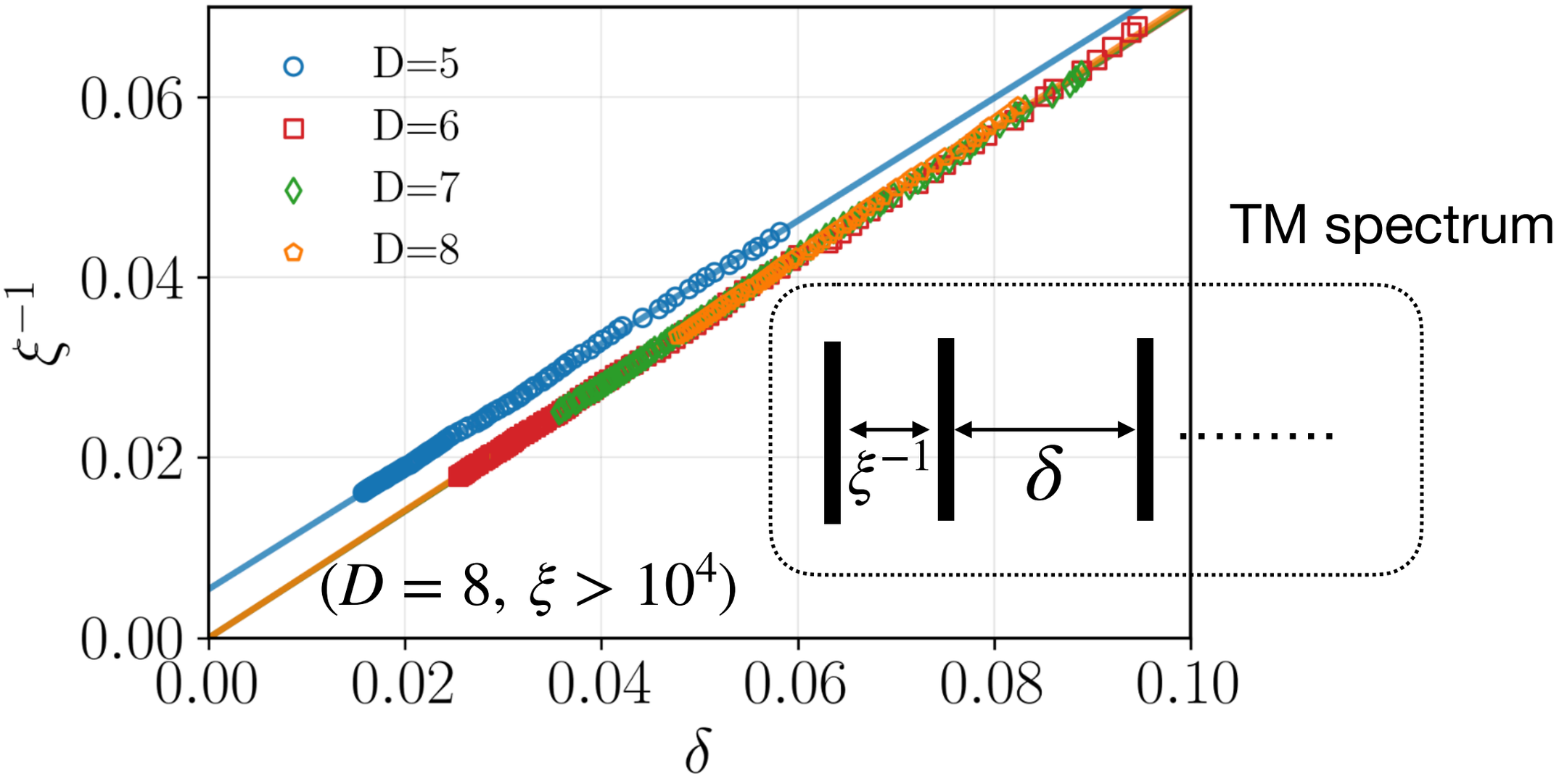
| | | |
|---|---|---|
| Critical SG | $D = 8,\ \|\Delta E/E\| = 7 \times 10^{-5}$ | PhysRevLett.123.087203(2019) |
| AD+CTMRG | $D = 6,\ \|\Delta E/E\| = 9 \times 10^{-5}$ | PhysRevB.107.054424(2023) |
| AD+QRCTM | $D = 8,\ \|\Delta E/E\| = 2 \times 10^{-5}$ | |

**SOTA reported energy with AD+iPEPS**

Not extrapolated, energy per site.

# Kitaev model: Finite Correlation Length Scaling

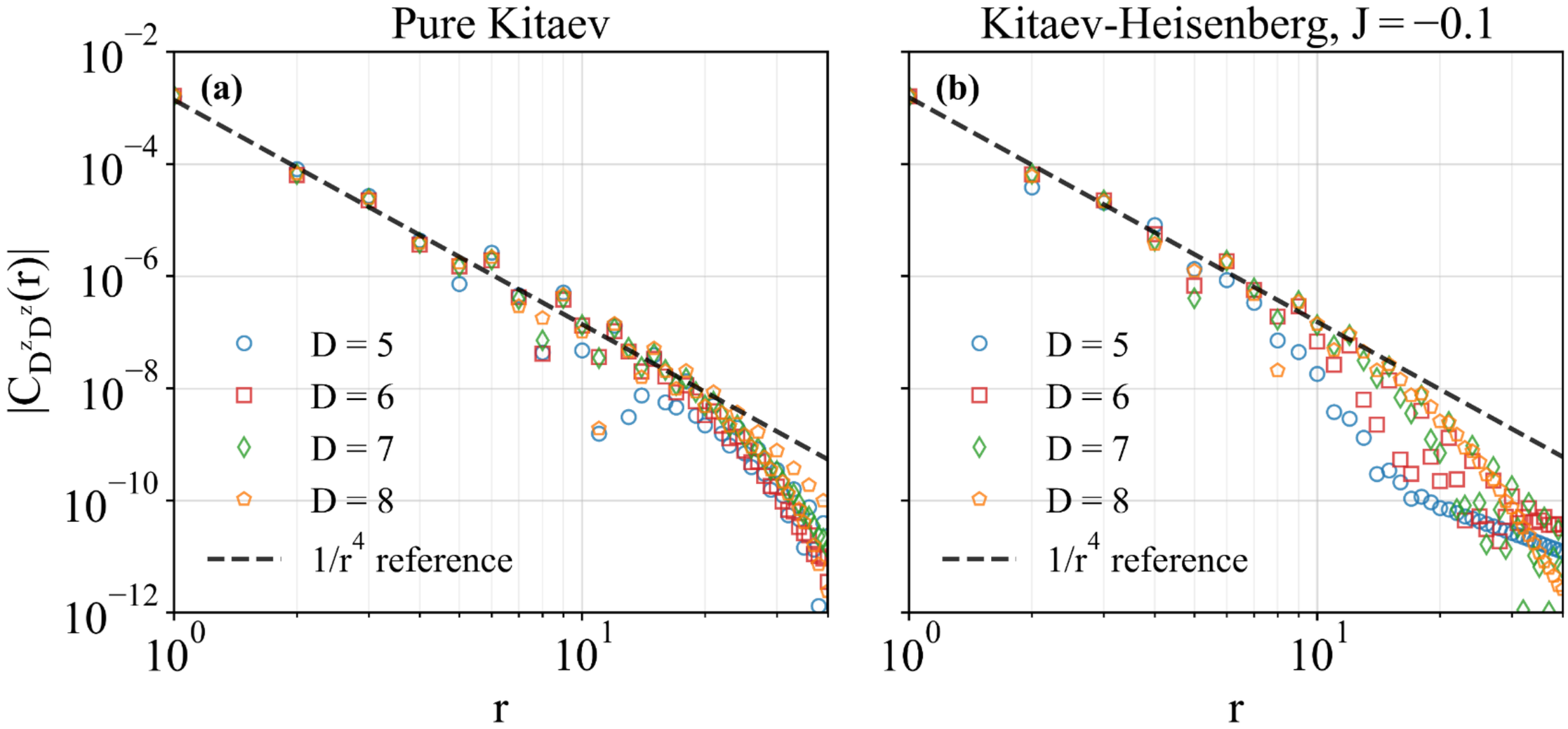
Phys. Rev. X 8, 031030 (2018)
Phys. Rev. X 8, 031031 (2018)




**Finite D, infinite $\xi$ !**

Finite-correlation-length-scaling reveals gapless nature.

# iPEPS evidence for Dimer-Dimer correlation decay




**Dimer–Dimer $r^{-4}$ tail persists in Kitaev-Heisenberg model**

Phys. Rev. A 78, 012304 (2008)
Phys. Rev. Res. 2, 013005 (2020)

As predicted by field theory; to our knowledge, first confirmed with iPEPS

# Higher Spin Kitaev model

$$H = \sum_{\langle i,j \rangle_\gamma} K_\gamma S_i^\gamma S_j^\gamma$$

$$S > 1/2,\ K_\gamma = -1$$

Is the ground state a **QSL**?

Any **symmetry breaking**?

Semi-classical analysis

PRB 78, 115116 (2008)
Nat. Commun. 9, 1575 (2018)
…….
Results: **Bond order** for Large: S ≧ 2
S=3/2 (unclear), S=1(no SSB)

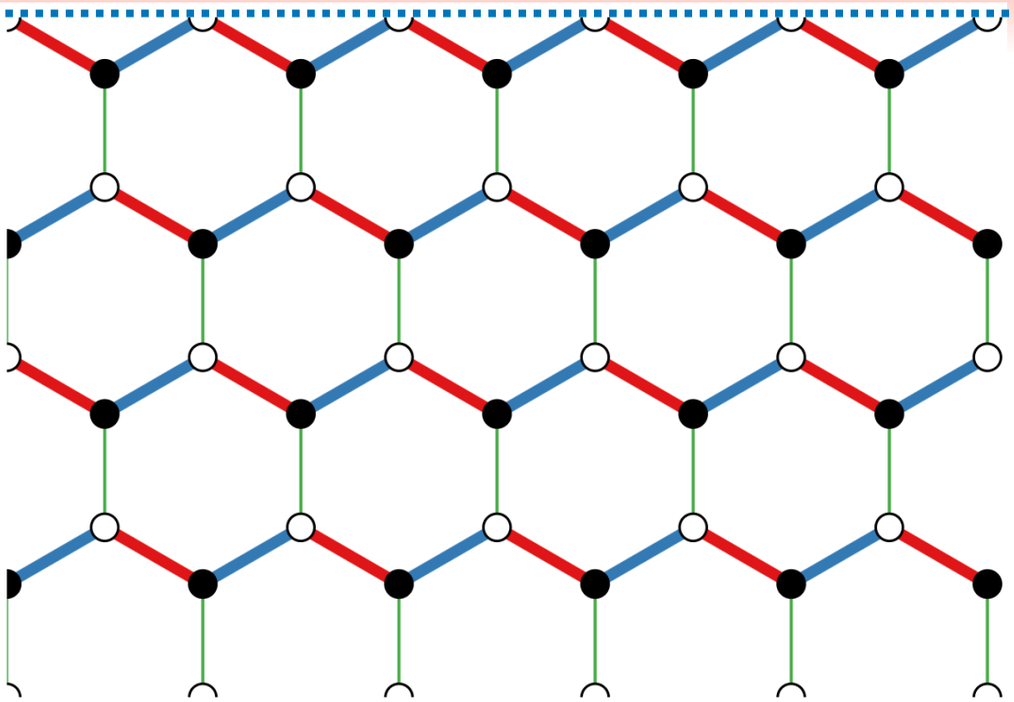
Quantum many-body methods

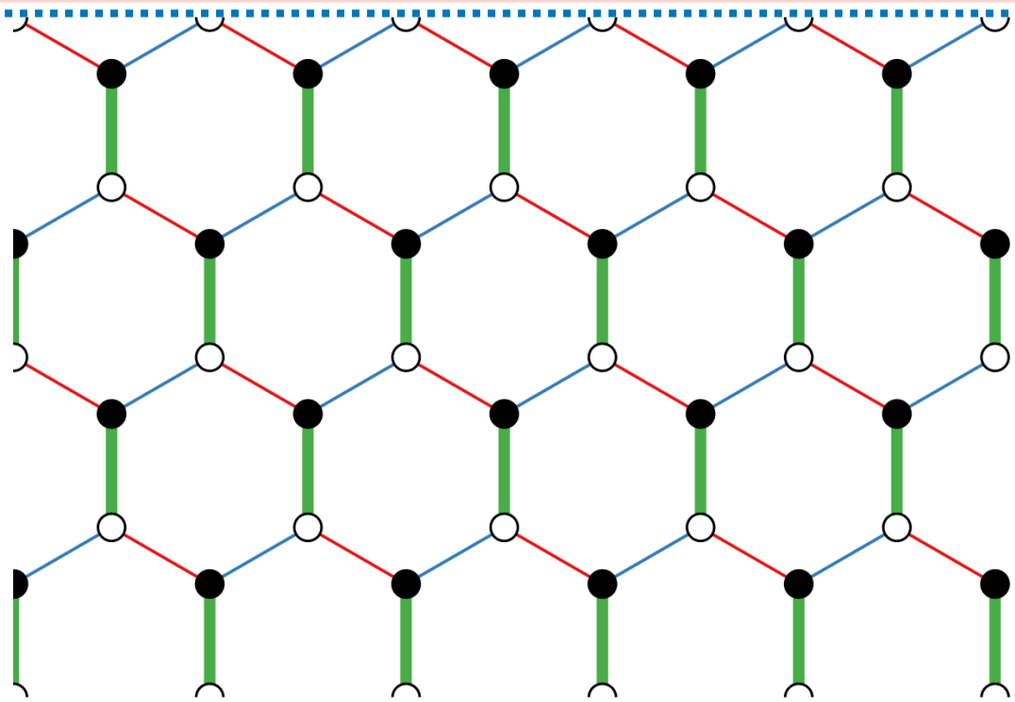
JPSJ 87, 063703 (2018)
PRR 2, 033318 (2020)
PRR 3, 013160 (2021)
PRB 105, L060403 (2022)
Nat. Commun. 13, 3813 (2022)
PRB 106, 174416 (2022)
SciPost Phys. 16, 100 (2024)
PRR 6, 033077 (2024)
PRB 110, 134402 (2024)
PRR 6, 033168 (2024)
arXiv:2503.24246 (2025)
…….
Results: **QSL, no symmetry breaking**

Commun. Phys. 7, 319 (2024)
Result: **(2x2 unit-cell SU), anisotropic**

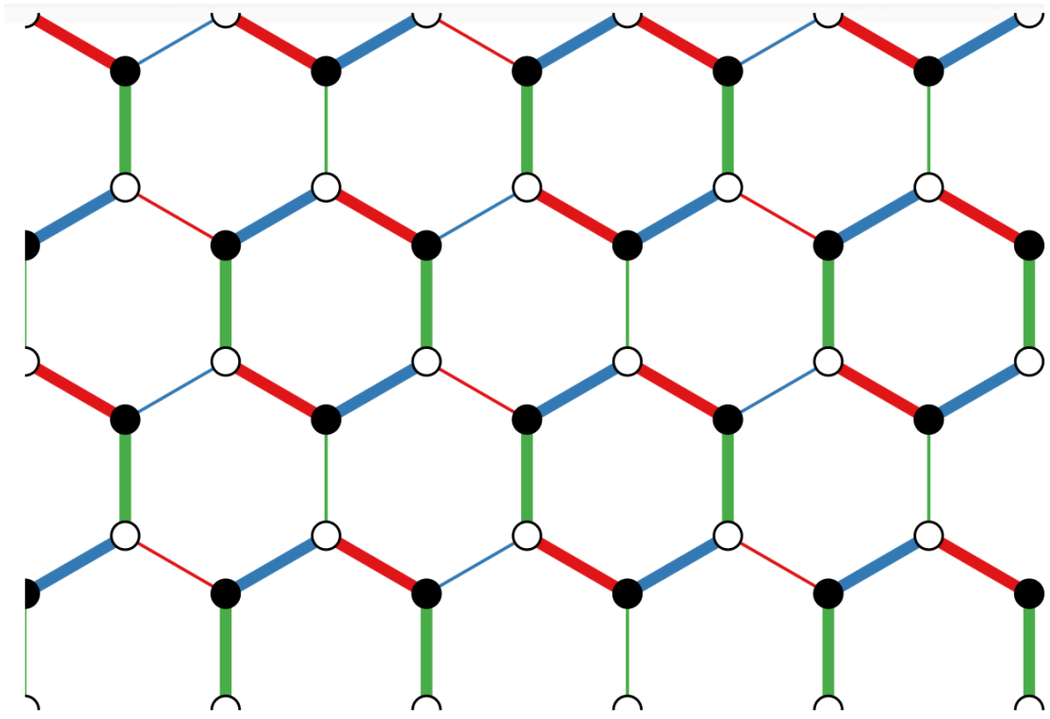
arXiv.2511.07415. Collaboration with Xingyu Zhang*, Jutho Heageman, Yuchi He @Ghent

# Ground states candidates from SU

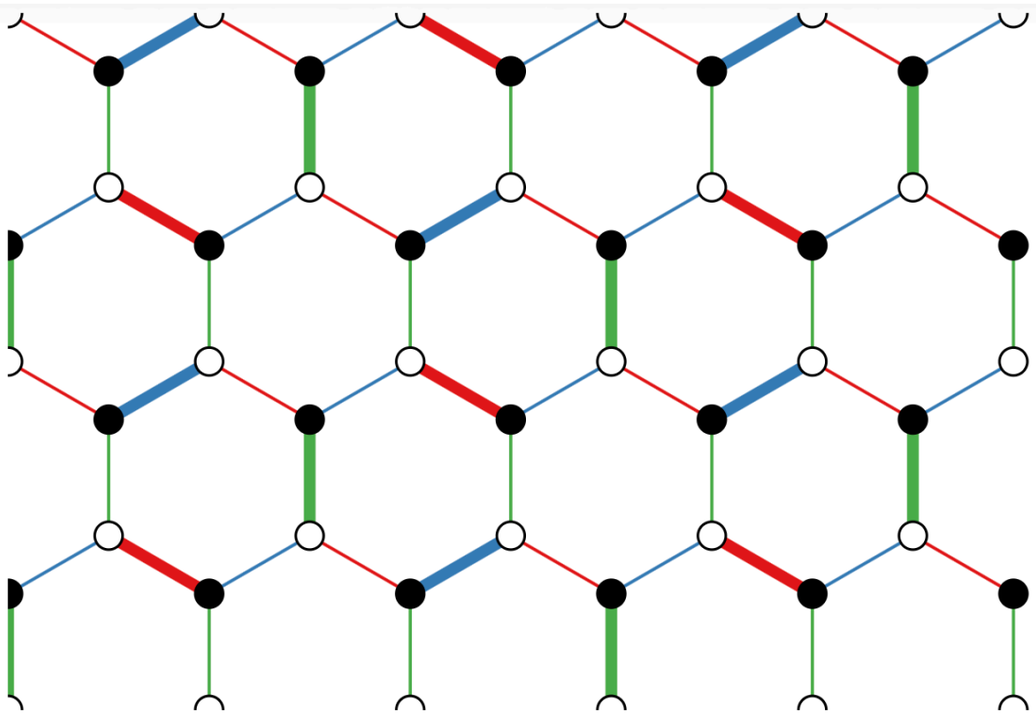




Uniform anisotropic, high E, 2x1 unit-cell



Plaquette

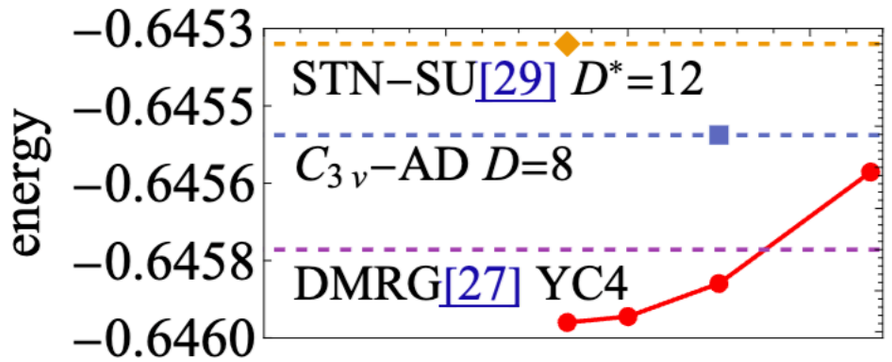


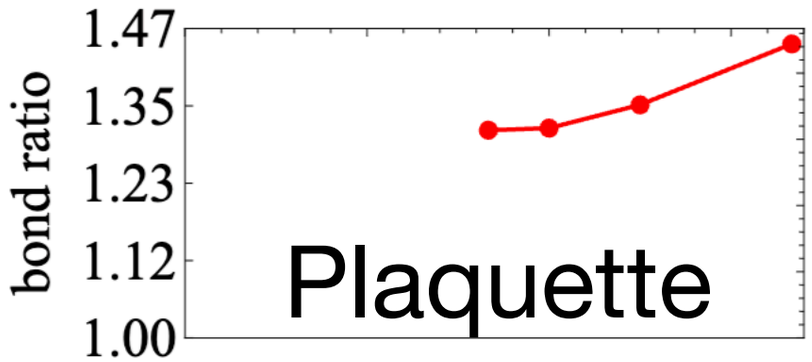
Dimer

# Detect SSB: Energy comparison

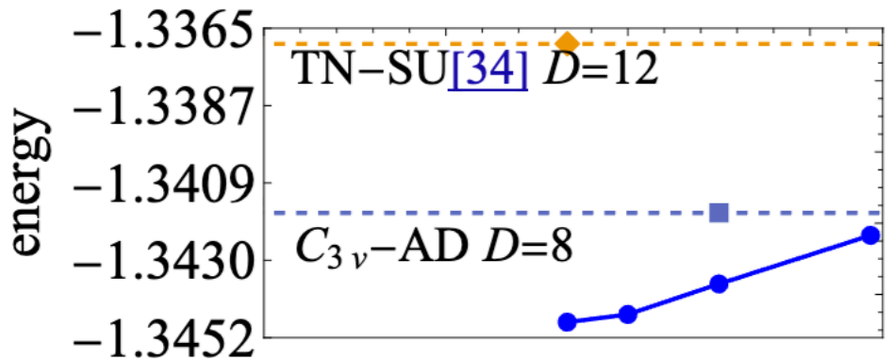
Comparing with AD+VUMPS, Precondition + multiple GPUs

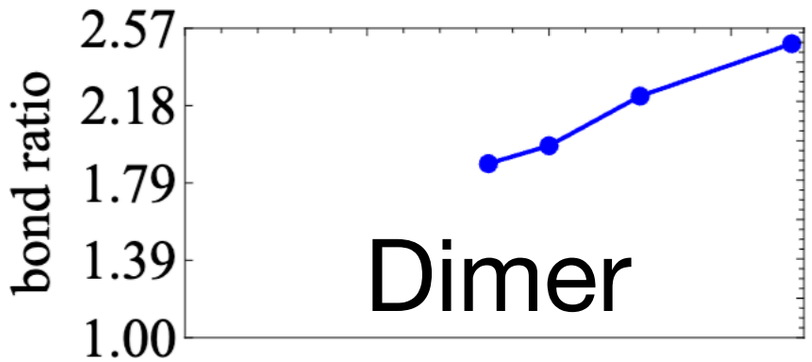
(a) S=1



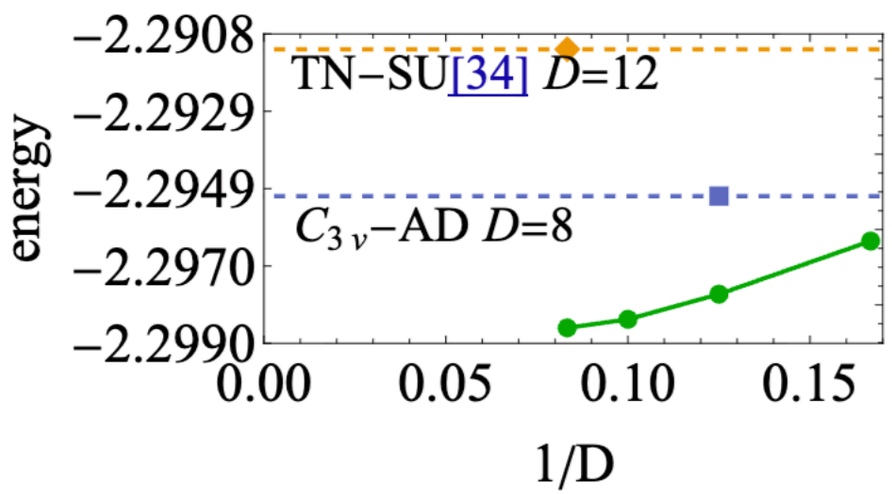



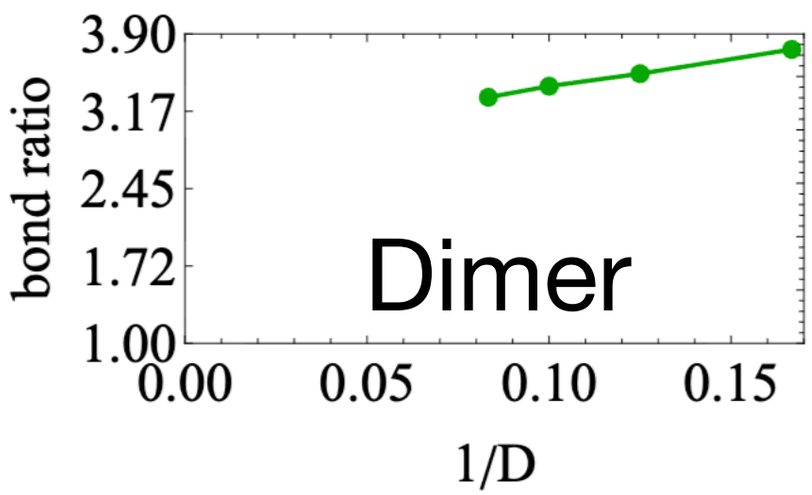

(b) S=$\frac{3}{2}$




(c) S=2

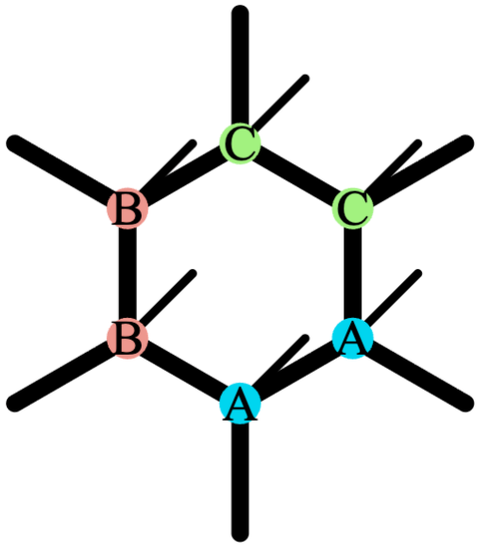


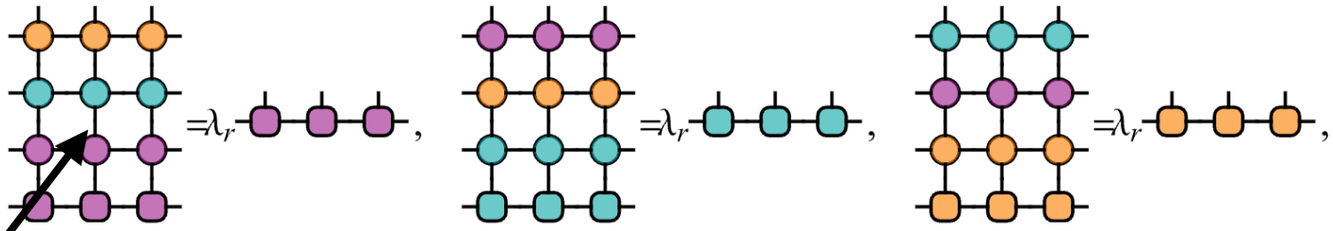




VUMPS: Xingyu Zhang@Ghent

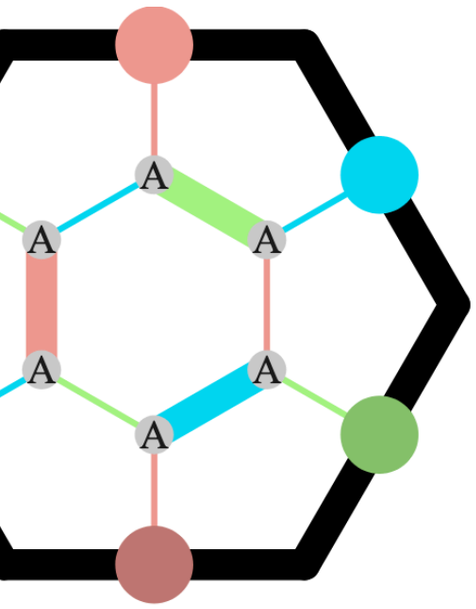
# Detect SSB: A New Diagnostic

VUMPS

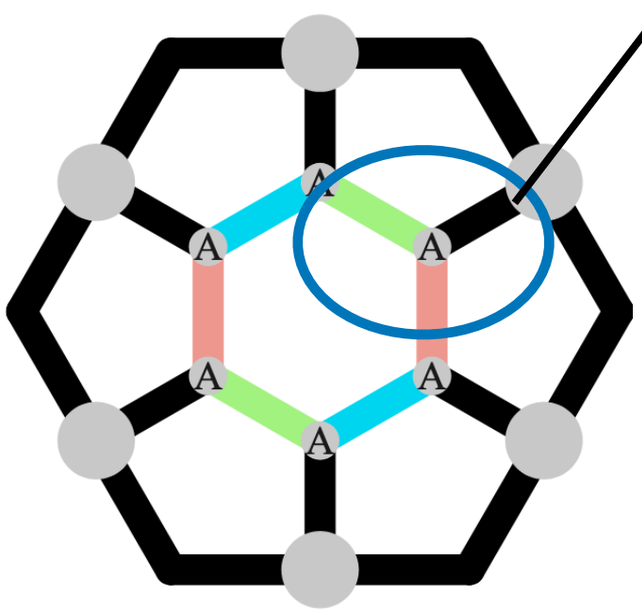



Env. SSB

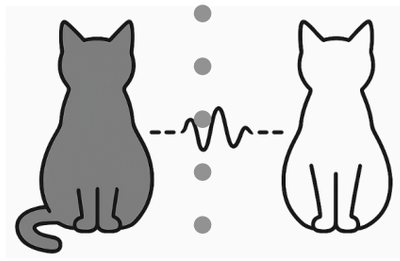


QRCTM





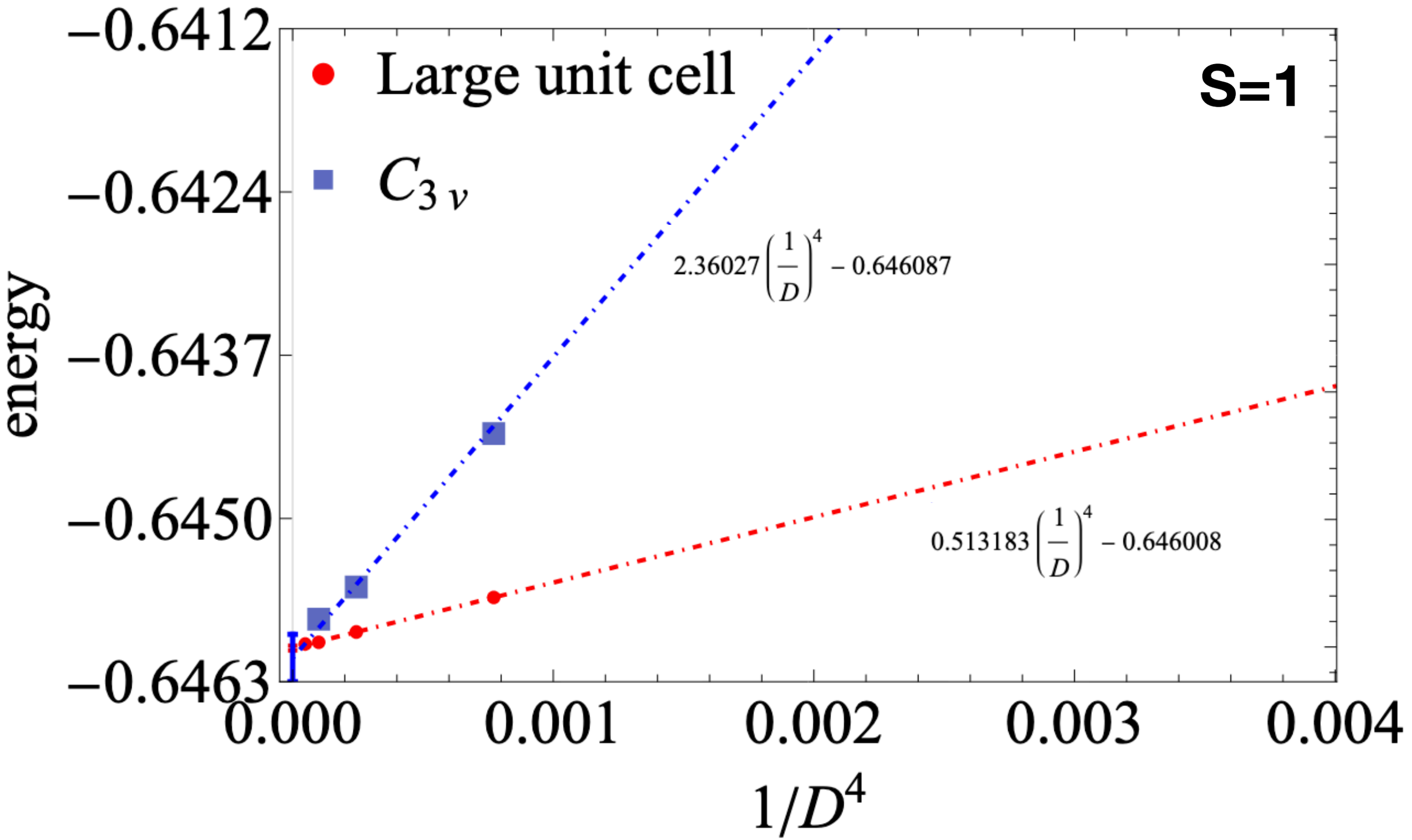




observables uniform ← cat states → observables non-uniform!

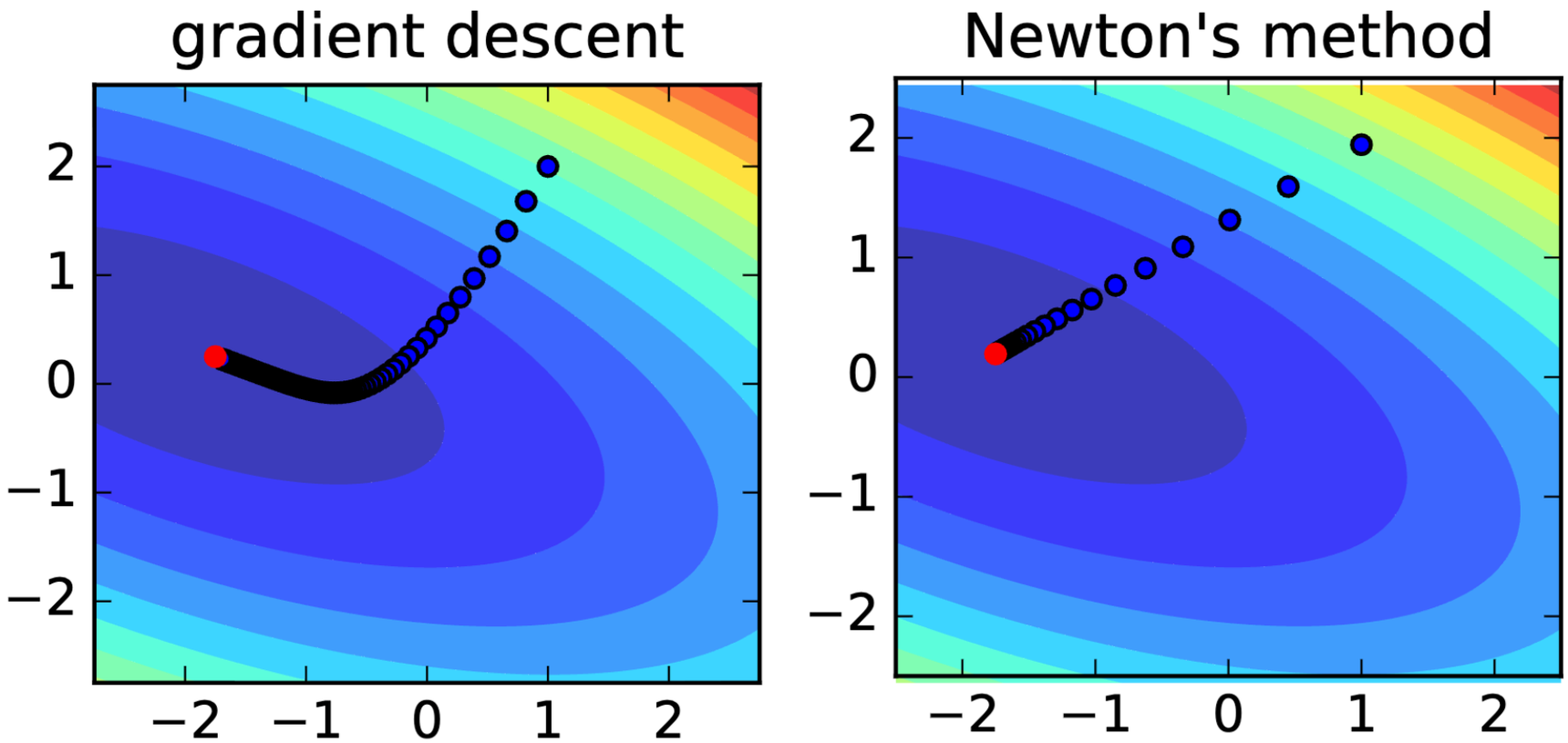
**Another peace of evidence of SSB for S=1,3/2,2**

# Quantifying SSB: Energy and Scaling?



# Riemannian optimization

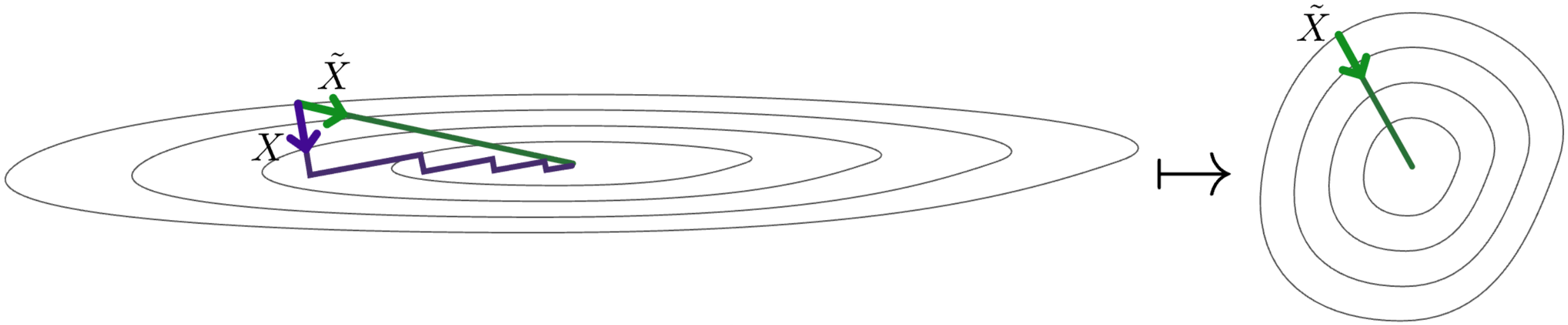
$$f(x_1, x_2) = x_1^2 + a x_2^2 = y_1^2 + y_2^2 \qquad g = \begin{pmatrix} 1 & 0 \\ 0 & a \end{pmatrix}$$




$$\partial_i f = 2x_1 + 2a x_2 \qquad g^{-1}\partial_i f = 2x_1 + 2x_2$$

Fig: arxiv:1612.01789

# Riemannian optimization for TNS




$$g_{ij}(A) = \frac{\partial^2 \langle \mathrm{TNS}(A_i) \,|\, \mathrm{TNS}(A_j') \rangle}{\partial A_i \partial A_j'}$$

Gradients G: $G = \partial_A E \to g^{-1} \partial_A E$

Equivalent: $gF = \partial_A E$

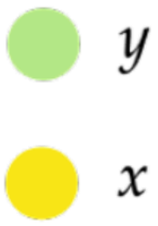
Markus Huaru, et al, SciPostPhys.10.2.040

# Towards TDVP for 2D?

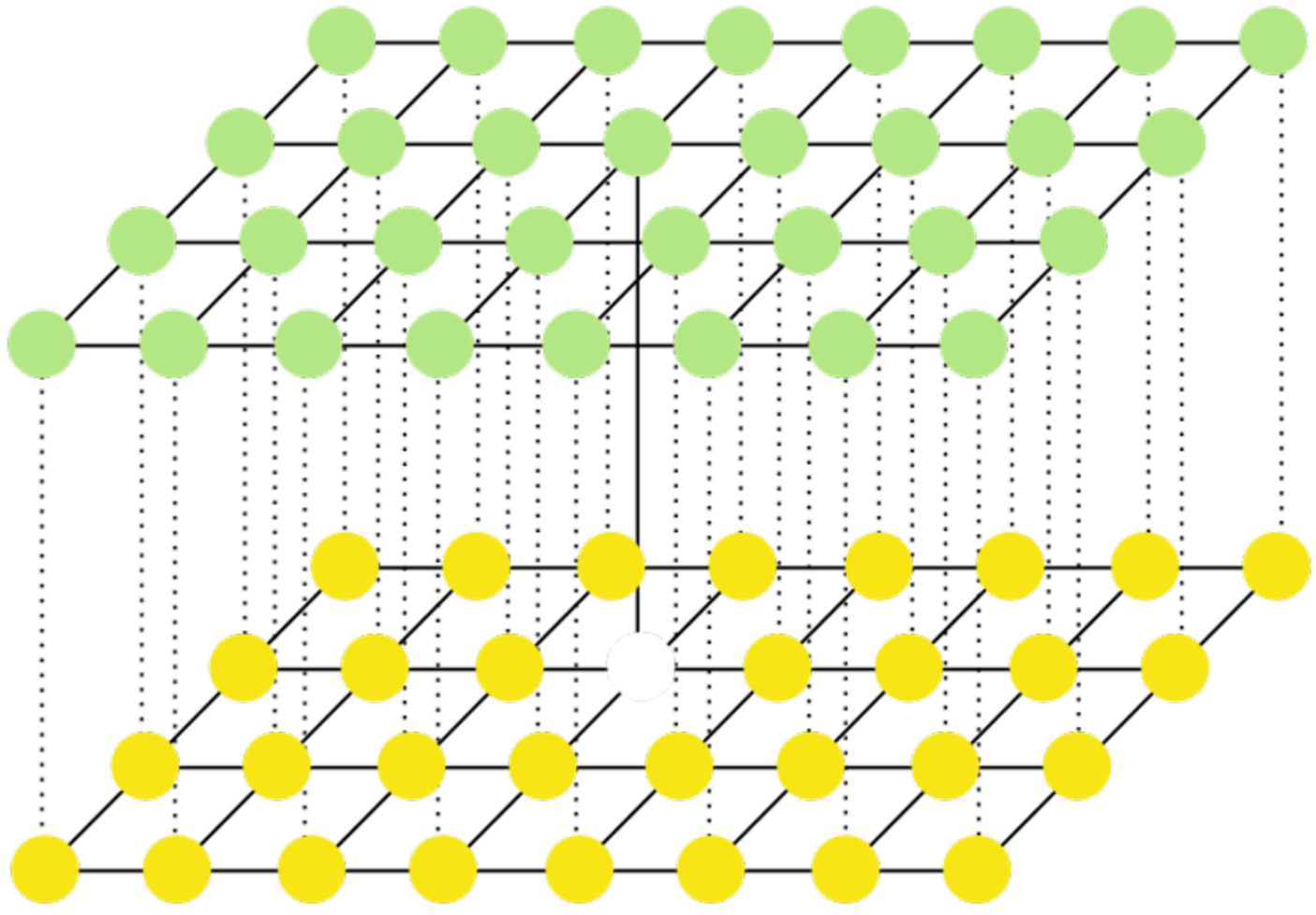
$$G_{xy} = \frac{\partial \langle \psi |}{\partial x} \frac{\partial |\psi \rangle}{\partial y}$$

# Towards TDVP for 2D?

$$G_{xy} = \frac{\partial \langle \psi |}{\partial x} \frac{\partial |\psi \rangle}{\partial y}$$




$$\partial_x \langle \psi(y) | \psi(x) \rangle =$$



$$G_{xy} = \partial_y [\partial_x \langle \psi(y) | \psi(x) \rangle]$$

# Towards TDVP for 2D?

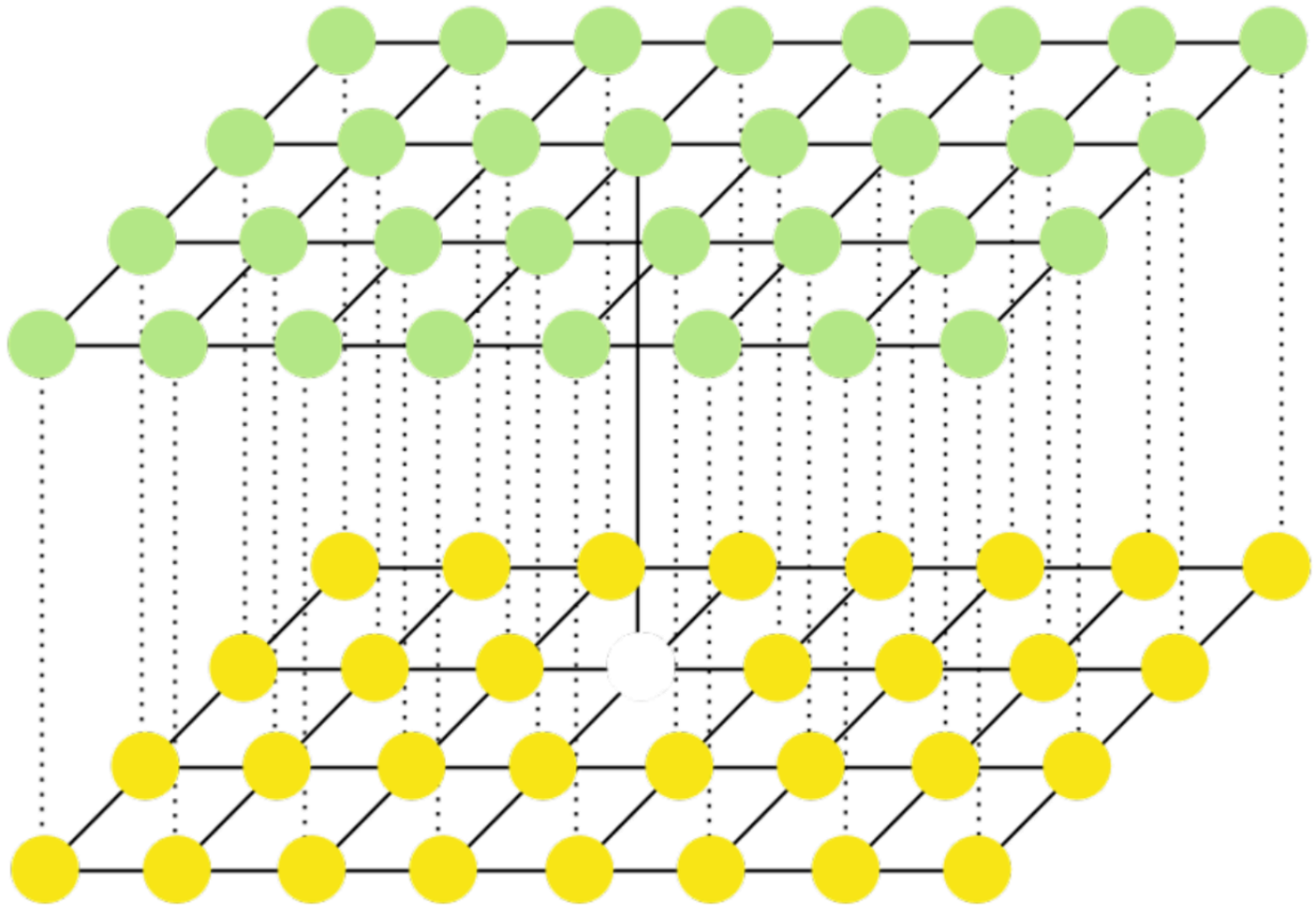
$$G_{xy} = \frac{\partial \langle \psi |}{\partial x} \frac{\partial |\psi \rangle}{\partial y}$$

$y$

$x$

$$\partial_x \langle \psi(y) | \psi(x) \rangle =$$

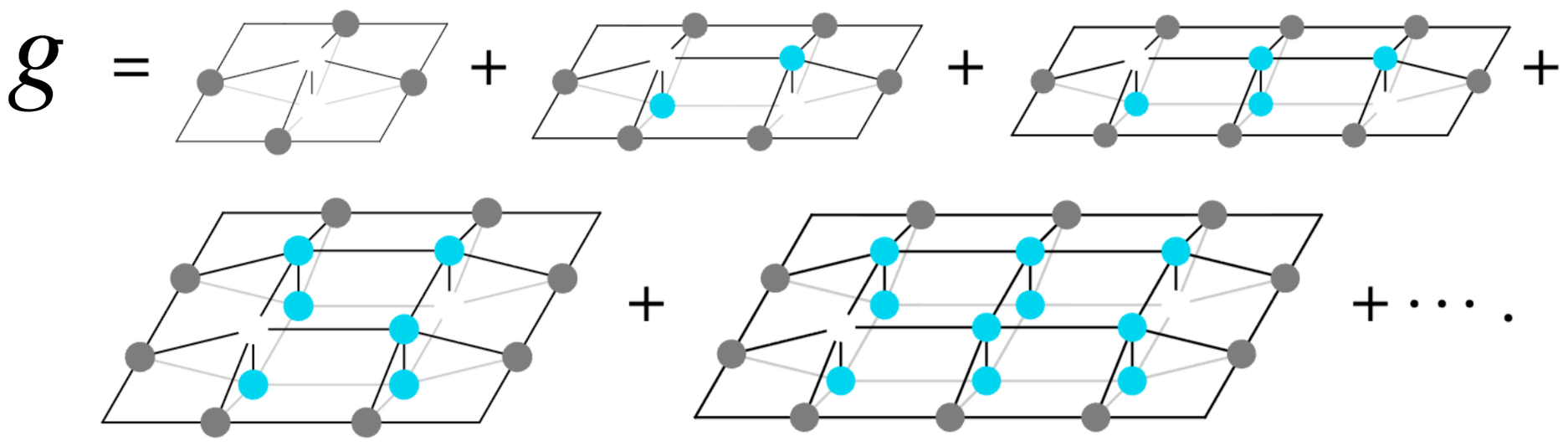
Analytic backward



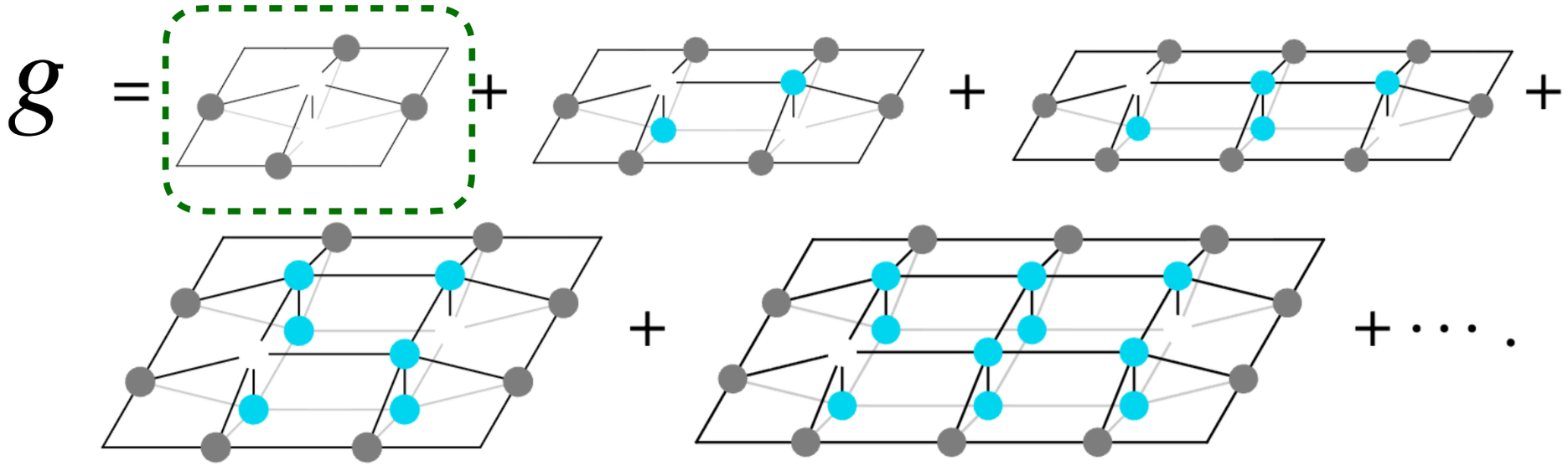
$$G_{xy} = \partial_y [\partial_x \langle \psi(y) | \psi(x) \rangle]$$

Only forward mode AD! Cheaper and Scalable!

# Metric for uniform iPEPS

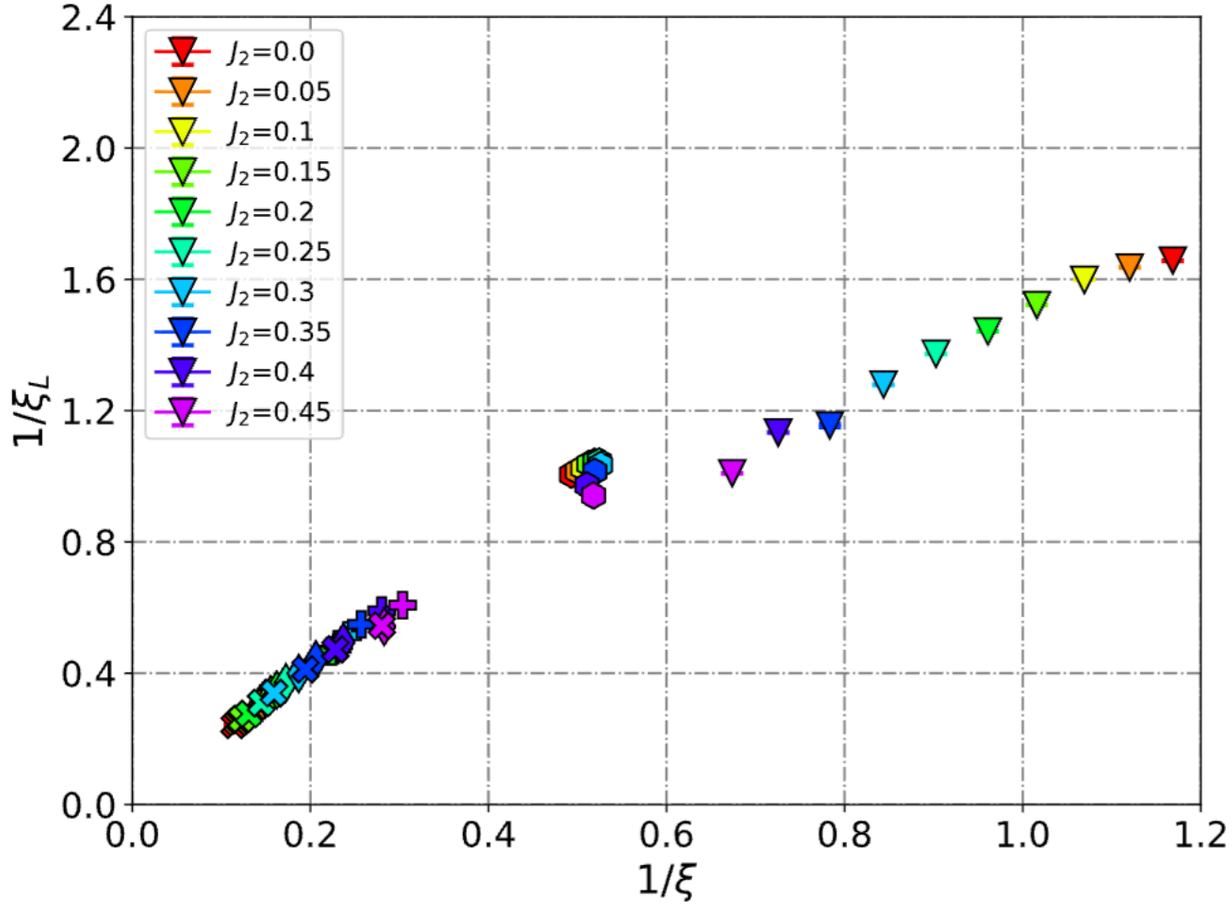


# Metric for uniform iPEPS



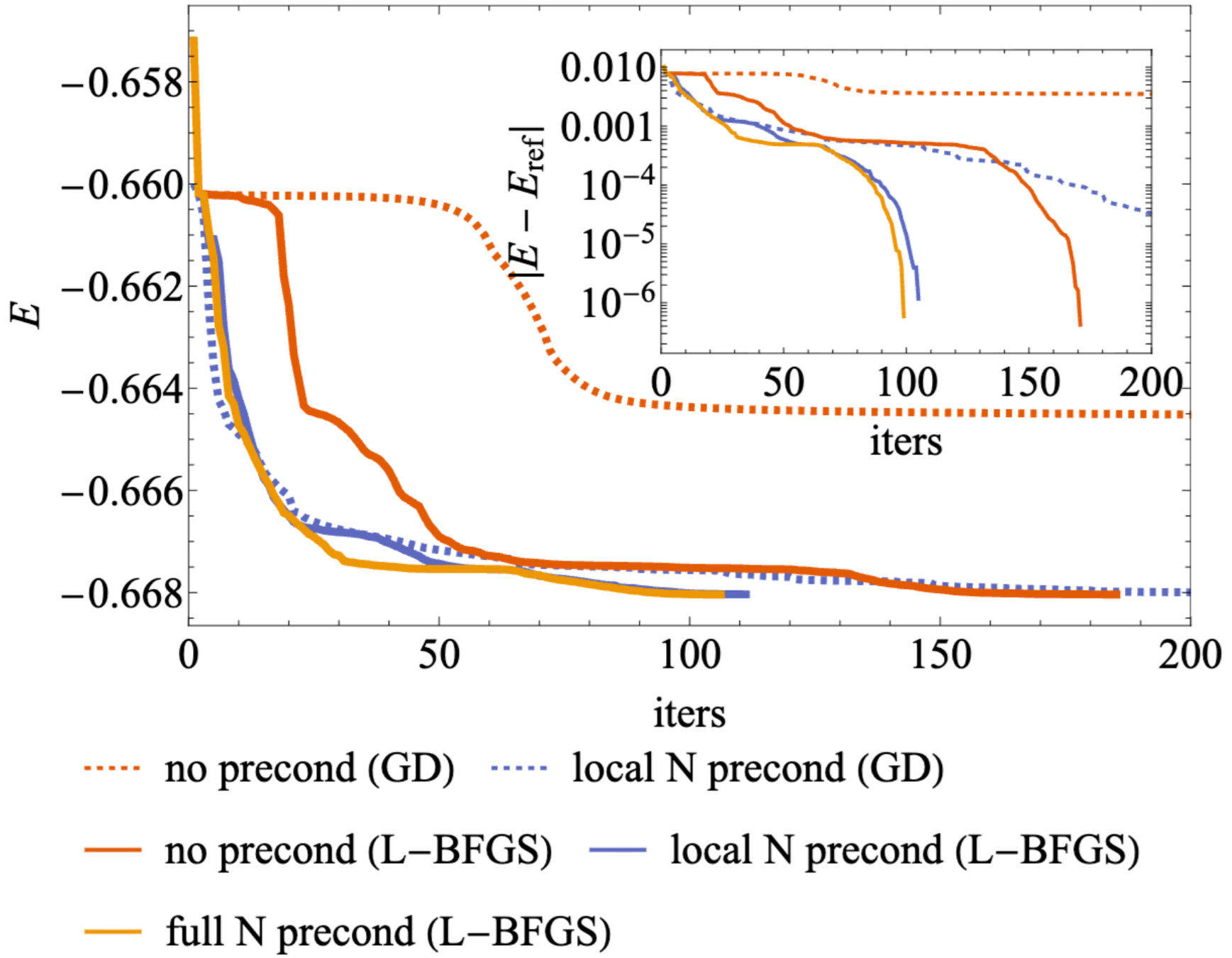

**Infinite summation!**

**O(1) “observed” correlation length**



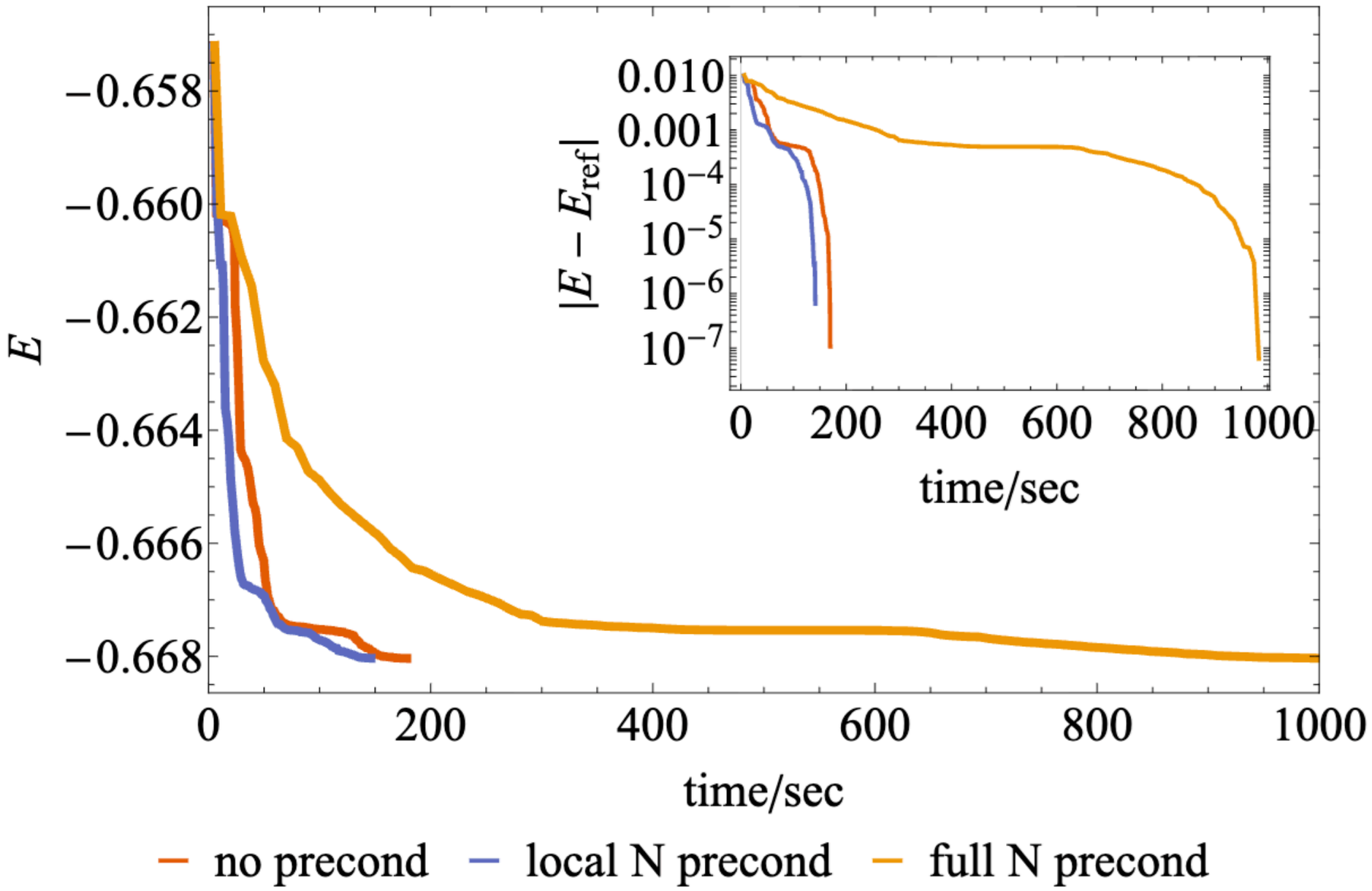

Juraj Hasik, et al, SciPostPhys.10.1.012

# Benchmark with Heisenberg model (D=3)



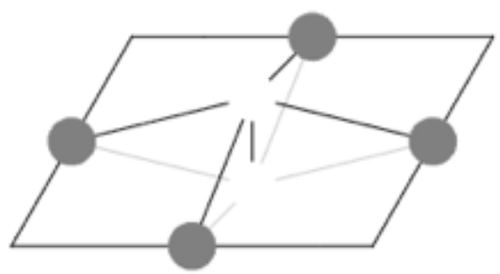

arXiv.2511.09546

# Benchmark with Heisenberg model (D=3)



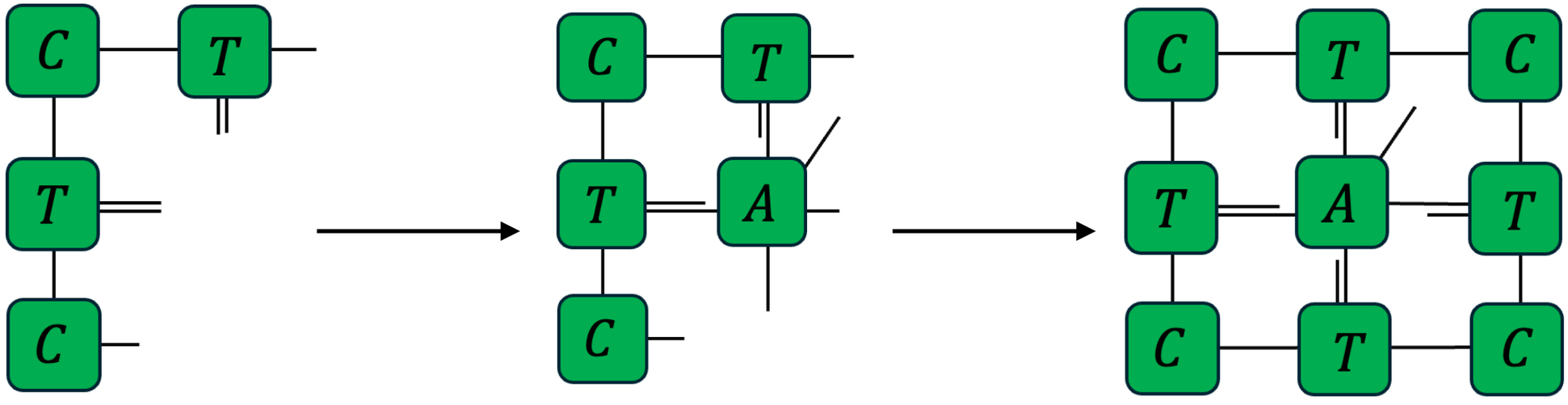
# “Matrix-free” Trick

Path A: explicitly construct:



$$O(\chi^3 D^4 + \chi^2 D^8)$$

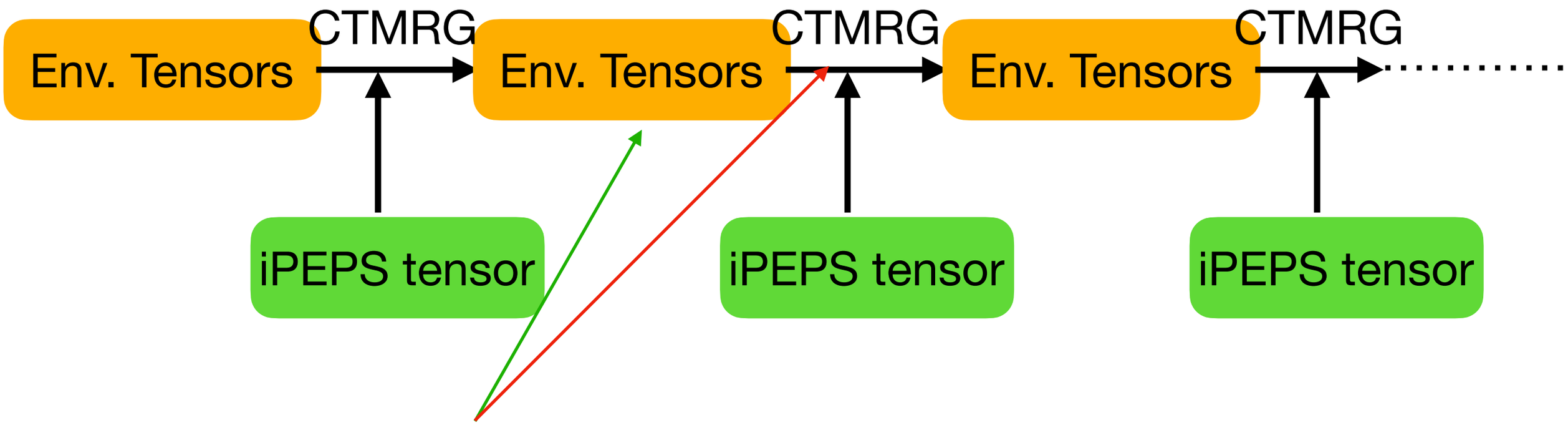
Path B: implicit construct:




$O(\chi^3 D^4)$ $O(\chi^2 D^6 d)$ $O(\chi^2 D^6 d)$

Then use Krylov solver

# Scalability Trick 1: CPU-offload + Slicing



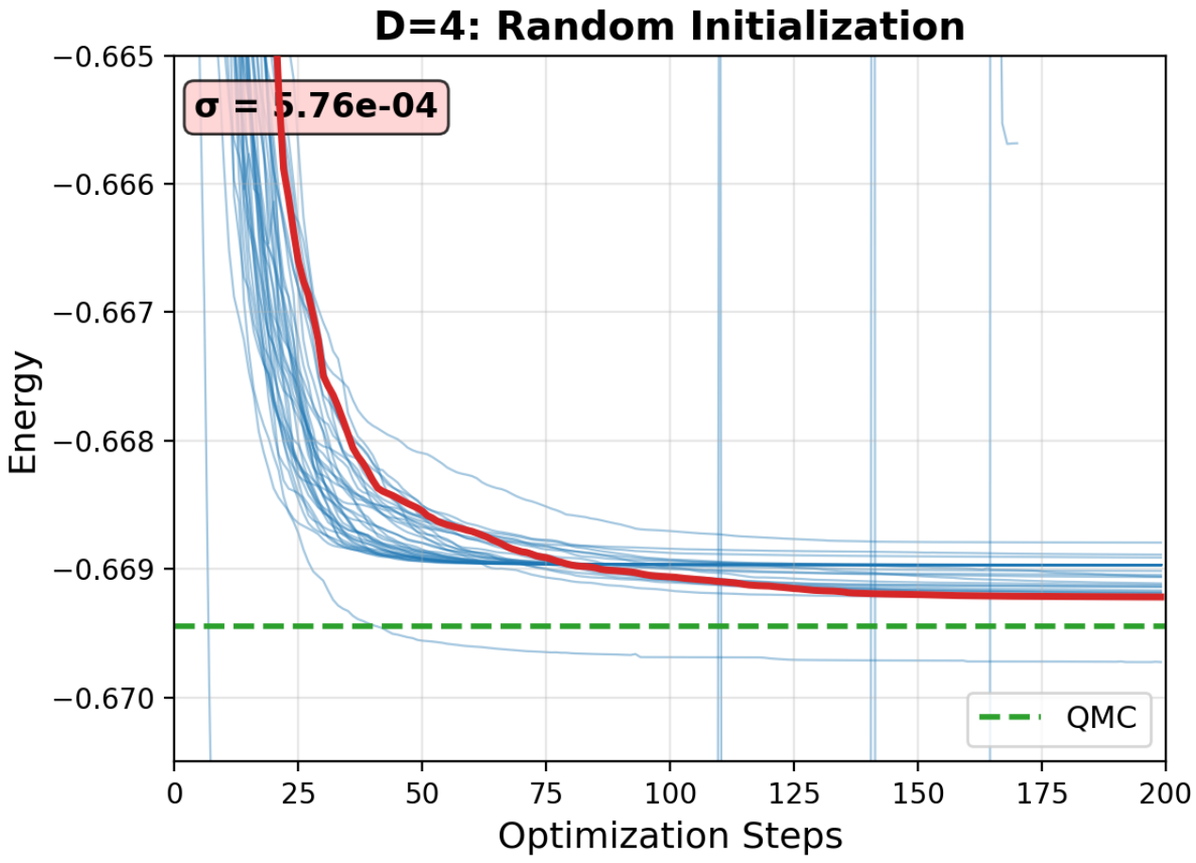

**1. Using loop to reduce peak**
**2. Multiple GPU dispatch**
**3. Cache this on CPU memory!**

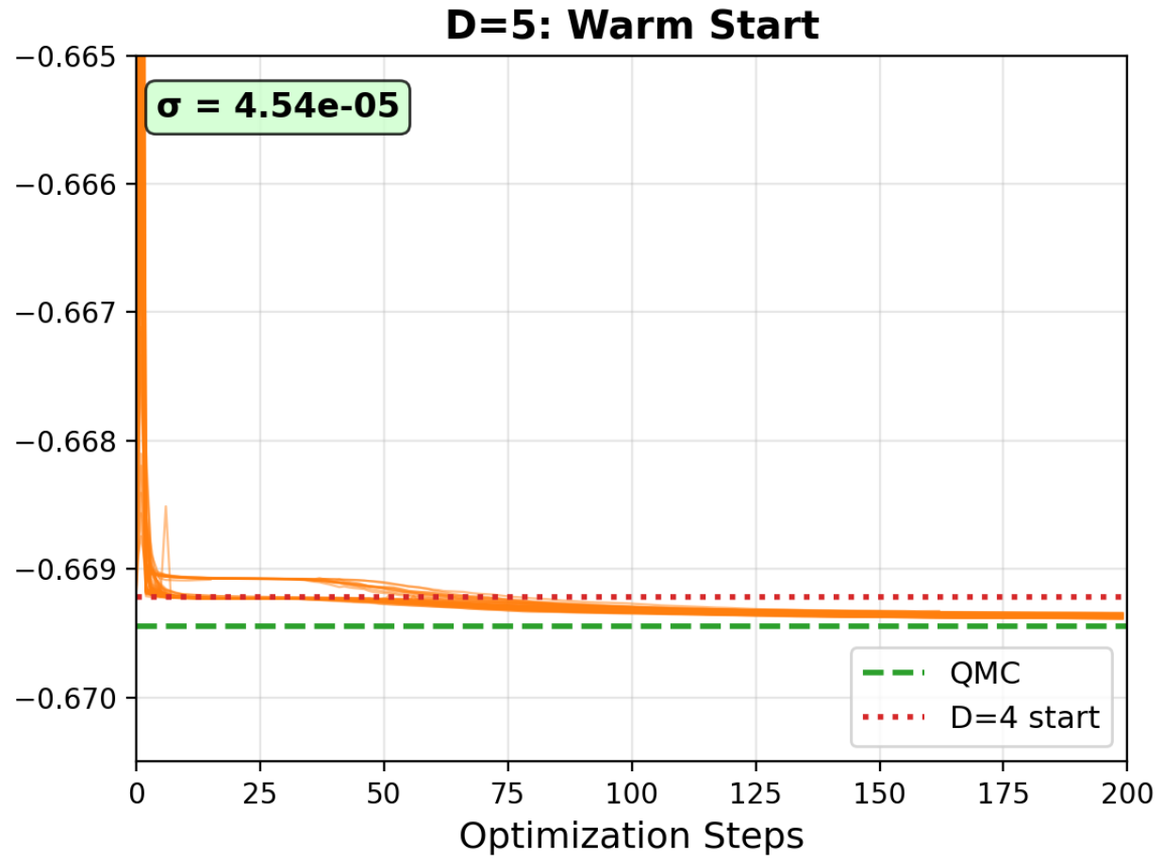
*https://github.com/qiyang-ustc/omeinsum*

One H100 can run AD+iPEPS for Heisenberg model with D=12, χ=1536!

# Stability Trick 2: Annealing + Ramping

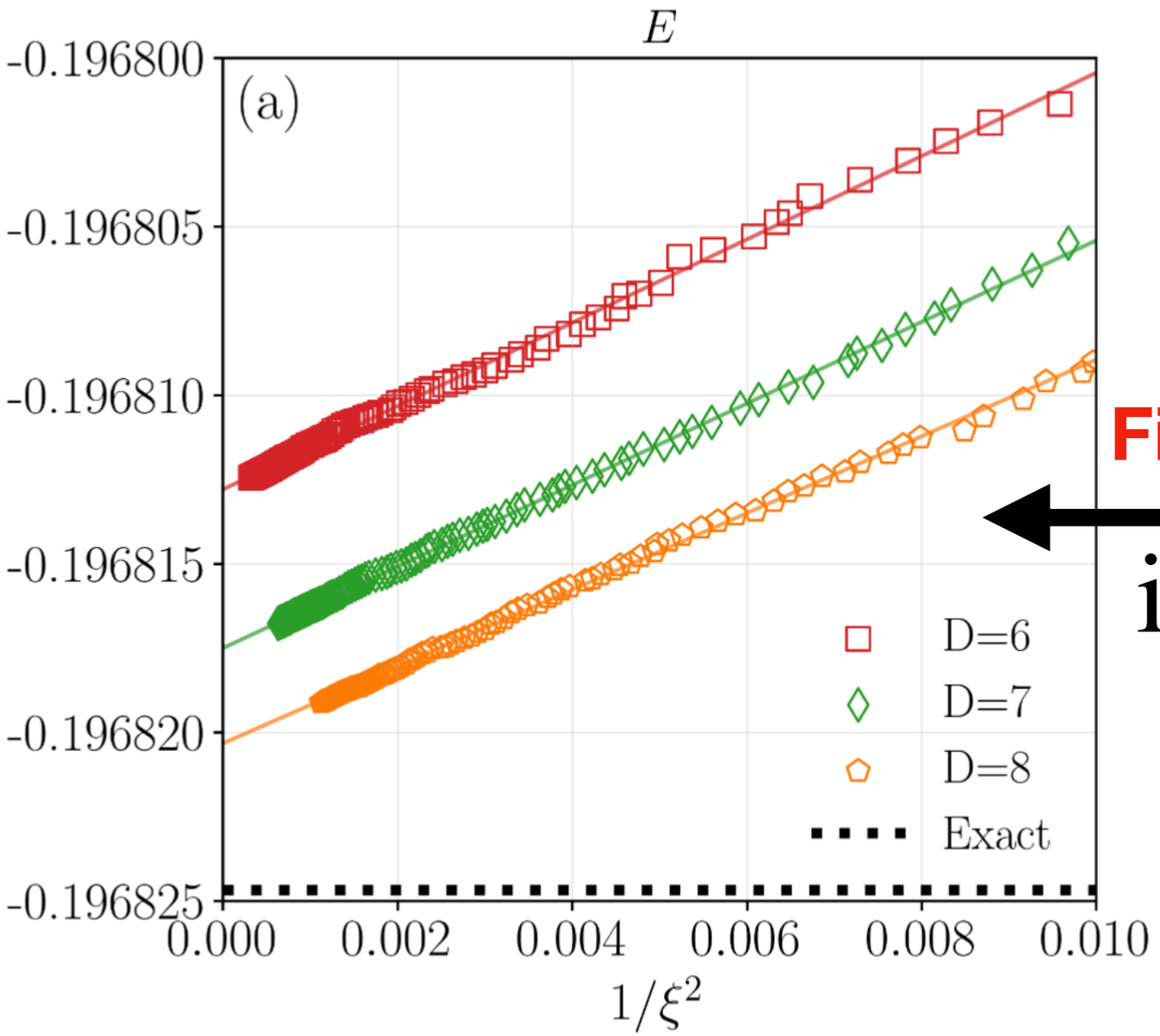
$D$ ramping



# Sanity check: Variational Principle?

$\chi$ ramping




**Fix iPEPS tensor**

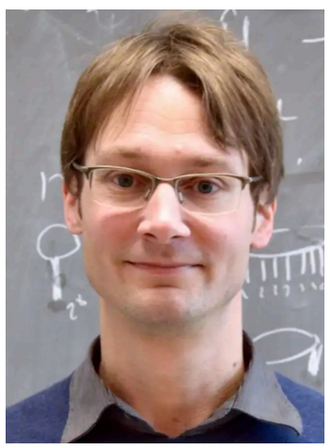
increase $\chi$

# Outlook

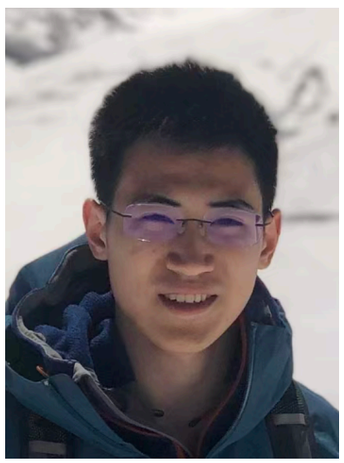
- Beyond Spatial symmetry and Fermionic system?

- 2D TDVP?

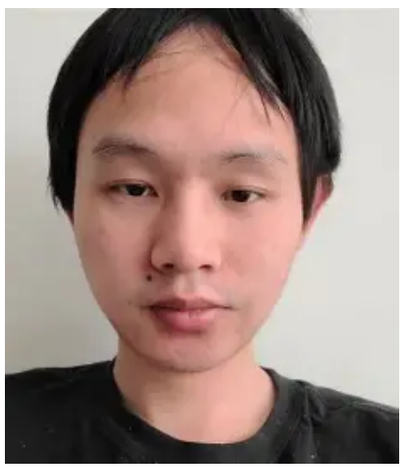
# Thanks to my collaborators



P. R. Corboz

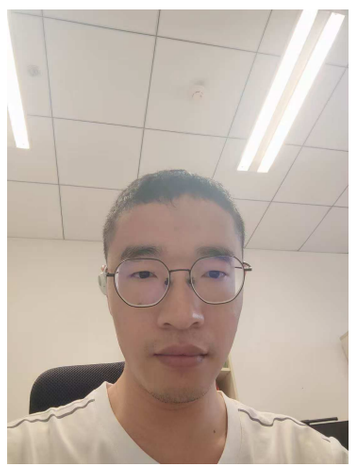


Yining Zhang



Yuchi He

Wei Tang



X.Y. Zhang

Wei Tang

Philippe Corboz
Qi Yang †*

Jutho Haegeman
Xingyu Zhang*

Yuchi He

Yining Zhang†

Jutho Haegeman

UNIVERSITY OF AMSTERDAM

GHENT UNIVERSITY

1. QRCTM C4v †
2. QRCTM C3v
3. Higher Spin Kitaev *
4. Precondition *

†*: equal first