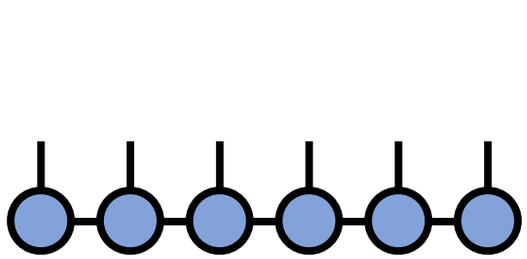
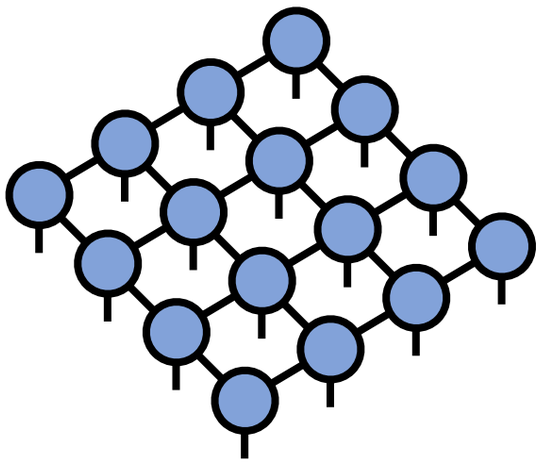
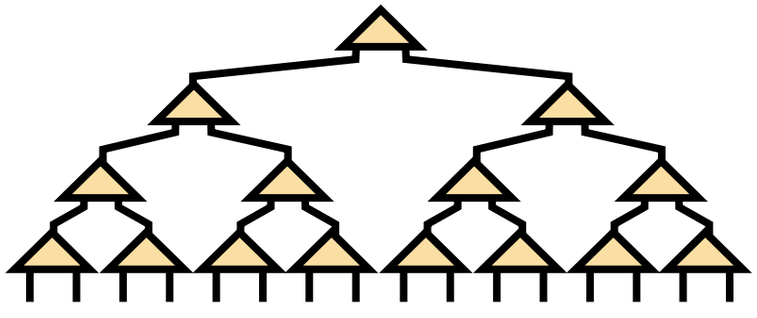
Tensor Networks

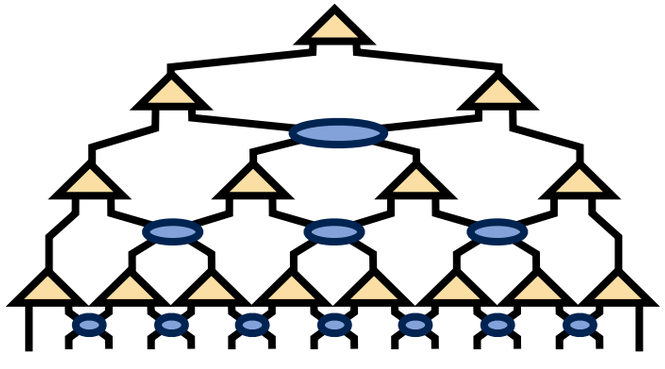
And The

ITensor Library









E.M. Stoudenmire

Jan 2019 - MIT

FLATIRON INSTITUTE

SIMONS FOUNDATION

Flatiron Institute

FLATIRON INSTITUTE

SIMONS FOUNDATION

The mission of the Flatiron Institute is to advance scientific research through computational methods, including data analysis, modeling and simulation.

CCA: Center for Computational Astrophysics

CCB: Center for Computational Biology

CCQ: Center for Computational Quantum Physics

CCM: Center for Computational Mathematics

Why tensors?

Probability distribution of N classical bits:

p_{000000} = probability to observe 0,0,0,0,0,0

p_{000001} = probability to observe 0,0,0,0,0,1

p_{000010} = probability to observe 0,0,0,0,1,0

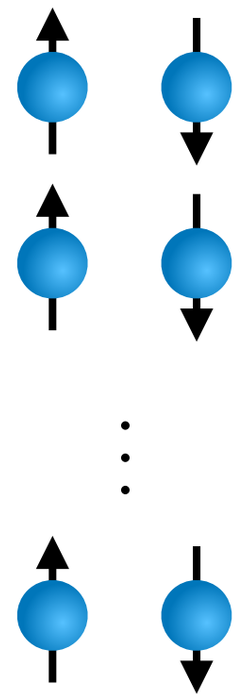
p_{000011} = probability to observe 0,0,0,0,1,1

p_{111111} = probability to observe 1,1,1,1,1,1

It is an order-N tensor

Similarly, quantum amplitudes of N
"quantum bits" form an order-N tensor

$$\Psi^{s_1 s_2 s_3 \cdots s_N}$$

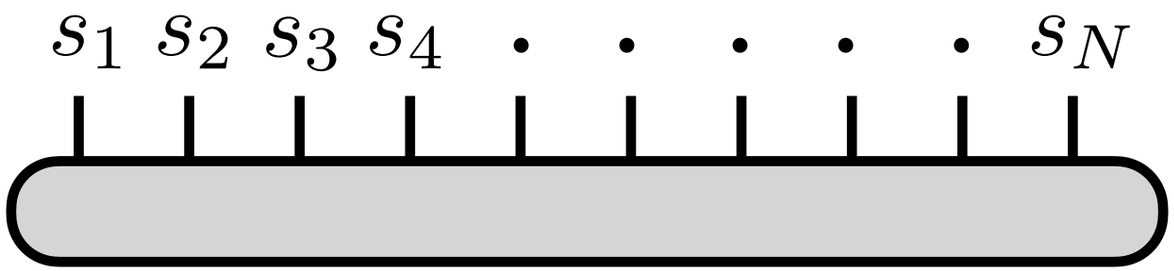


Called the "many-body quantum wavefunction"

Components are probability *amplitudes*
(square roots of probabilities)

Of course, an N-index tensor is a forbidding, intractable object

$$\Psi^{s_1 s_2 s_3 \cdots s_N} =$$

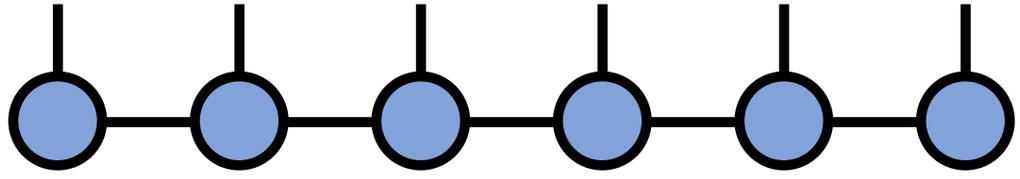


Number of components is 2^N

Tensor networks provide interesting way around this problem

$$\Psi^{s_1 s_2 s_3 s_4 s_5 s_6} = \sum_{\mathbf{i}} M^{s_1}_{i_1} M^{s_2}_{i_1 i_2} M^{s_3}_{i_2 i_3} M^{s_4}_{i_3 i_4} M^{s_5}_{i_4 i_5} M^{s_6}_{i_5}$$

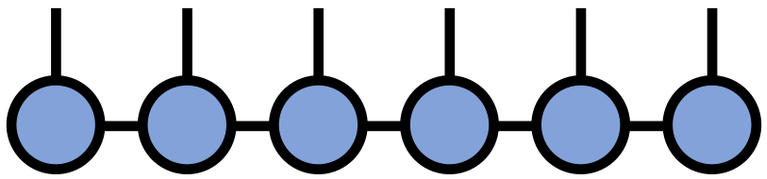
=



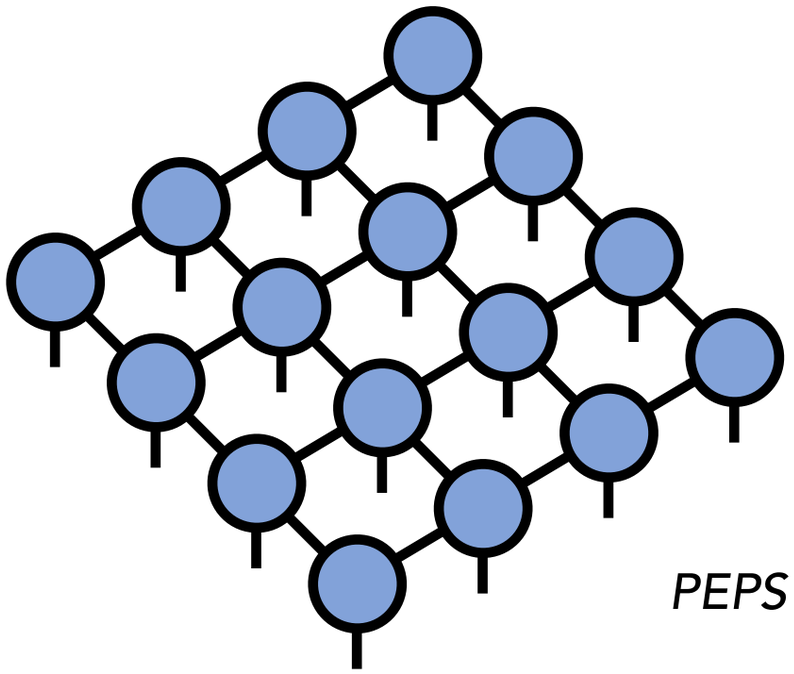
Generalization of singular value decomposition for tensors

Like a tensor version of hierarchical matrix approximations

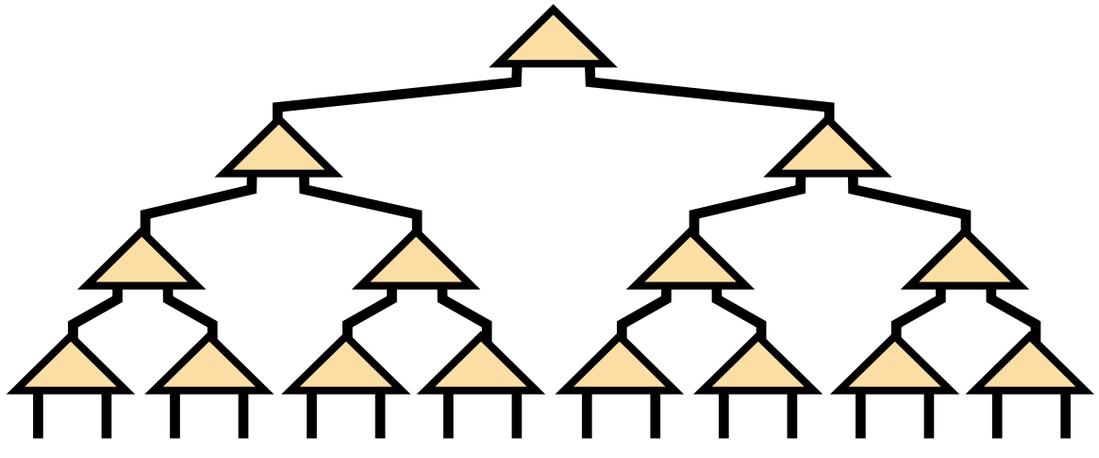
Many interesting tensor network formats, suited for different types of distributions



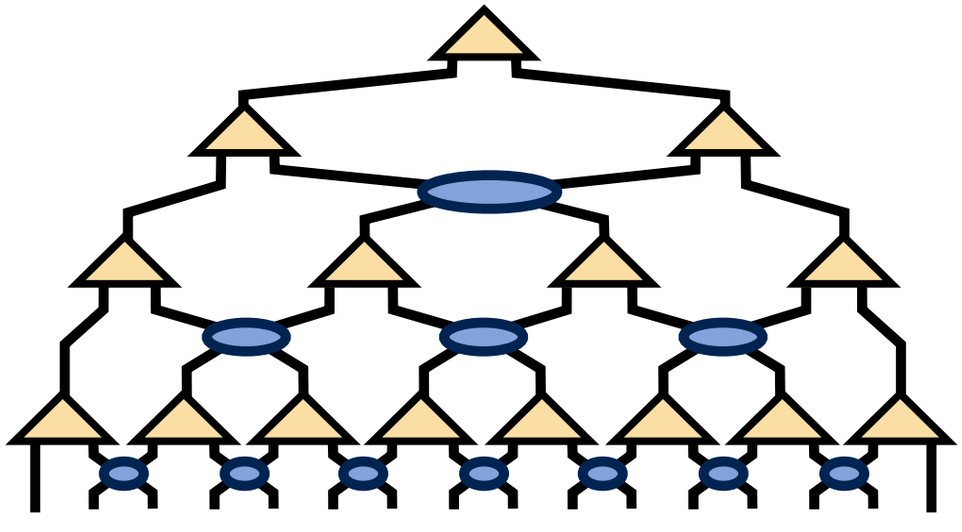
matrix product state / tensor train



PEPS network

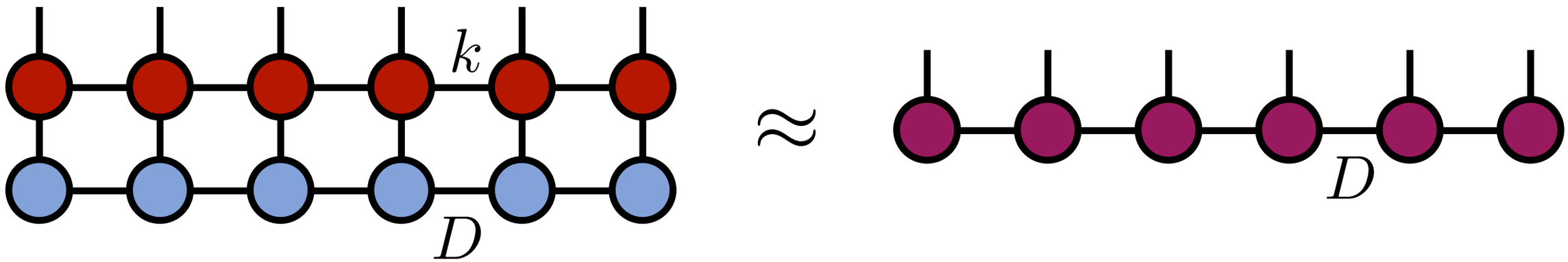


tree tensor network / hierarchical Tucker



MERA network

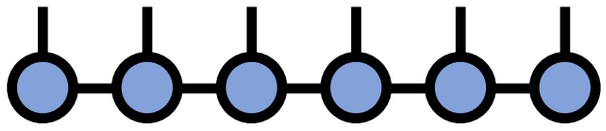
Controlled, efficient algorithms for
non-trivial operations:



Tensor networks are a set of techniques for
approximate linear algebra
in high-dimensional
tensor product spaces

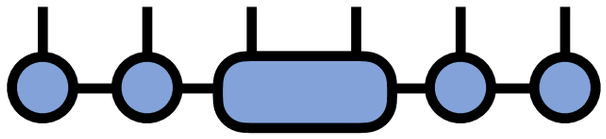
These algorithms are often adaptive (in index/mode size)

1) merge (contract) a pair of tensors



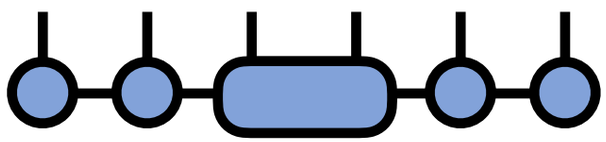
These algorithms are often adaptive (in index/mode size)

1) merge (contract) a pair of tensors

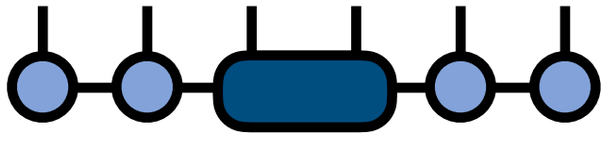


These algorithms are often adaptive (in index/mode size)

1) merge (contract) a pair of tensors

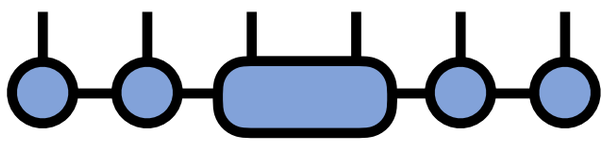


2) optimize parameters

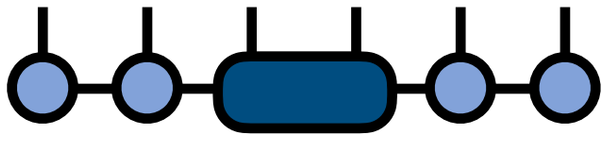


These algorithms are often adaptive (in index/mode size)

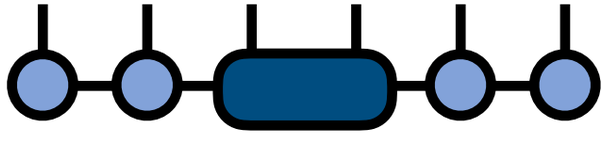
1) merge (contract) a pair of tensors



2) optimize parameters

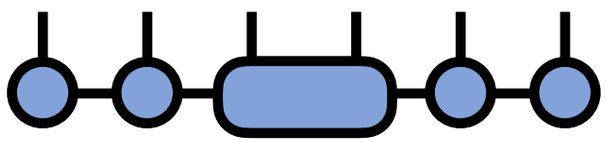


3) SVD factorization to adapt size of bond index

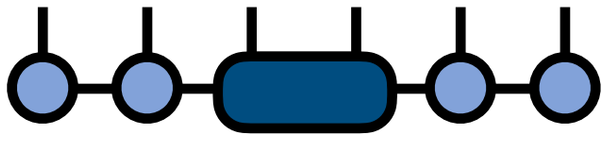


These algorithms are often adaptive (in index/mode size)

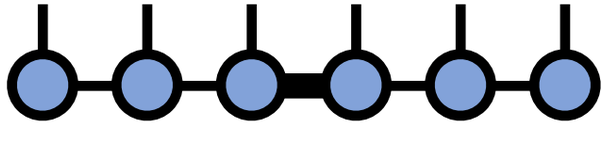
1) merge (contract) a pair of tensors



2) optimize parameters

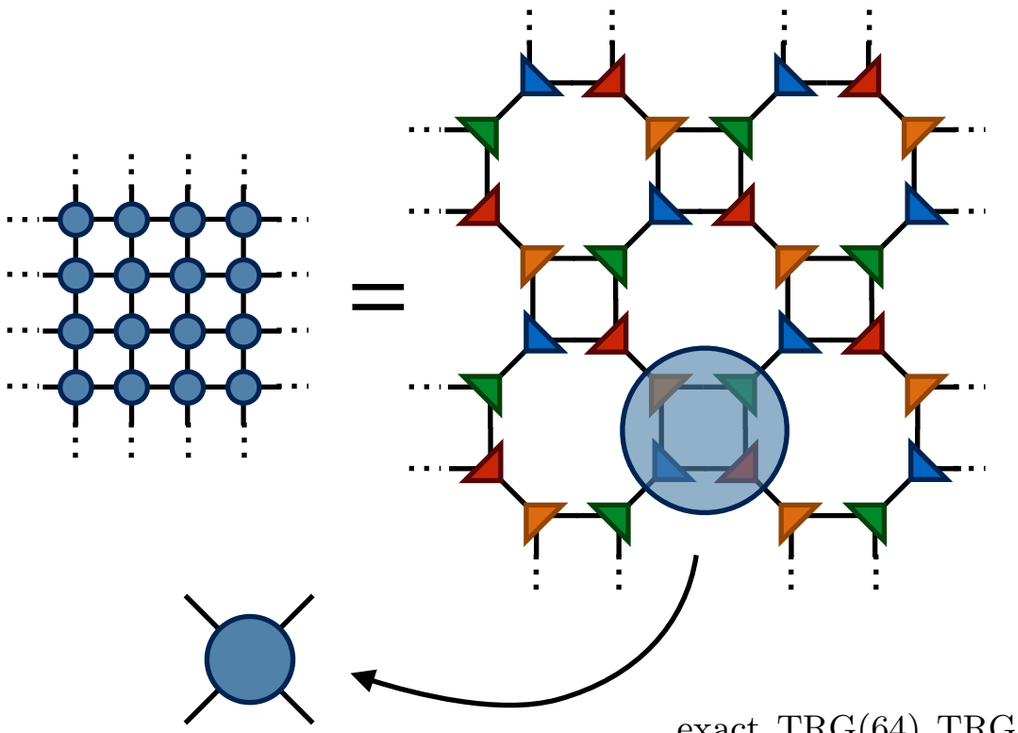


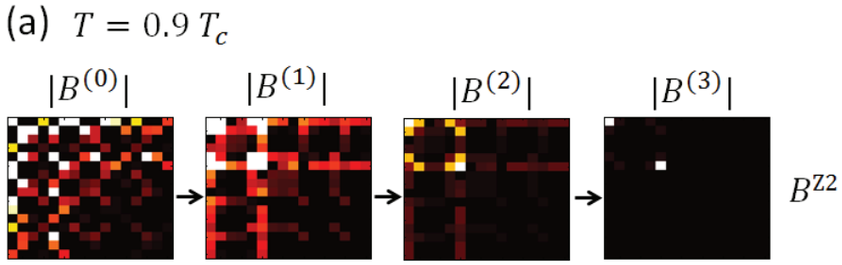
3) SVD factorization to adapt size of bond index

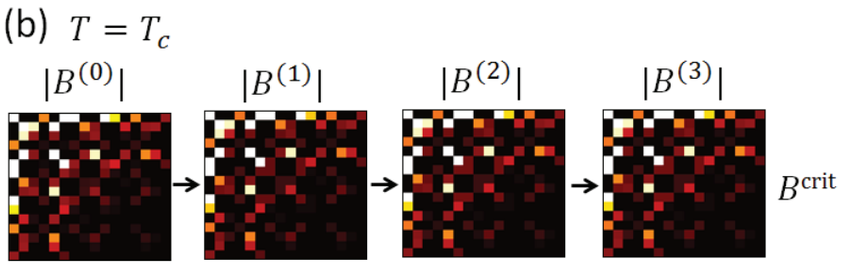


Resulting methods powerful for physics applications

coarse-graining of statistical models – compute partition function







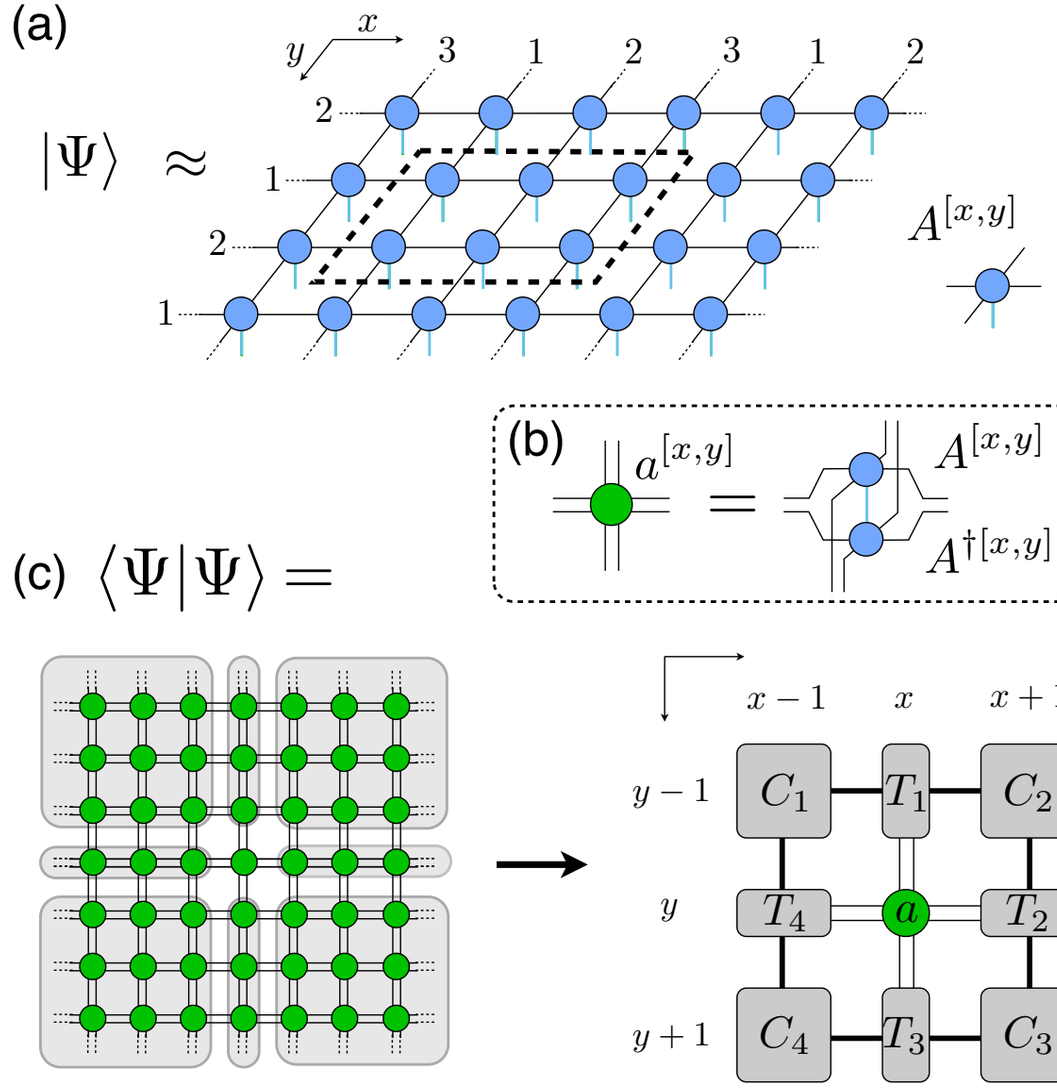
	exact	TRG(64)	TRG+env(64)	TEFR(64)	TNR(24)
c	0.5	0.49982	0.49988	0.49942	0.50001
σ	0.125	0.12498	0.12498	0.12504	0.1250004
ϵ	1	1.00055	1.00040	0.99996	1.00009
	1.125	1.12615	1.12659	1.12256	1.12492
	1.125	1.12635	1.12659	1.12403	1.12510
	2	2.00243	2.00549	-	1.99922
	2	2.00579	2.00557	-	1.99986
	2	2.00750	2.00566	-	2.00006
	2	2.01061	2.00567	-	2.00168

*Levin, Nave, PRL **99**, 120601 (2007)*

*Evenbly, Vidal, PRL **115**, 200401 (2015)*

Resulting methods powerful for physics applications

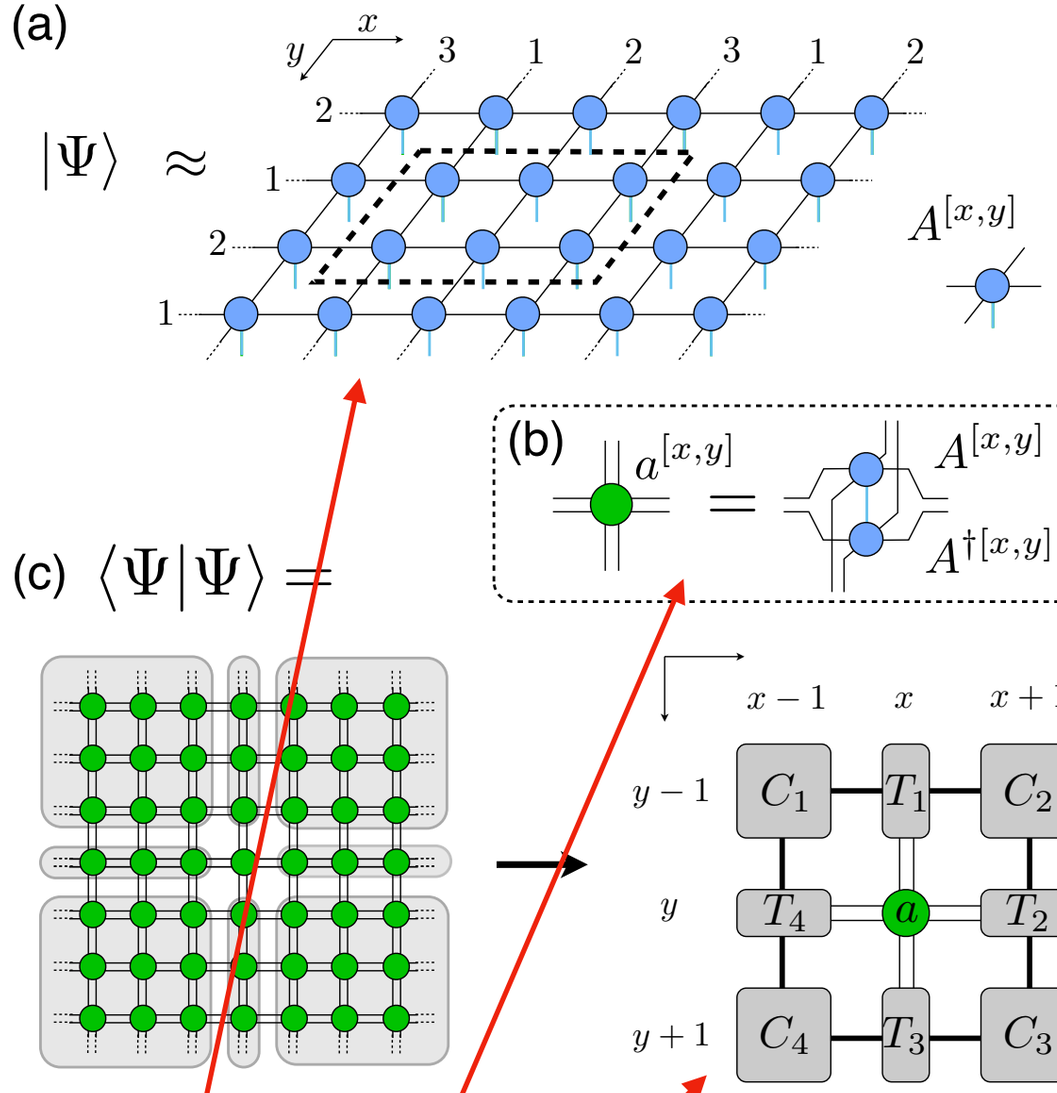
dominant eigenvector of exponentially large matrices (Hamiltonian)



*Corboz, Phys. Rev. B **94**, 035133 (2016)*

Resulting methods powerful for physics applications

dominant eigenvector of exponentially large matrices (Hamiltonian)

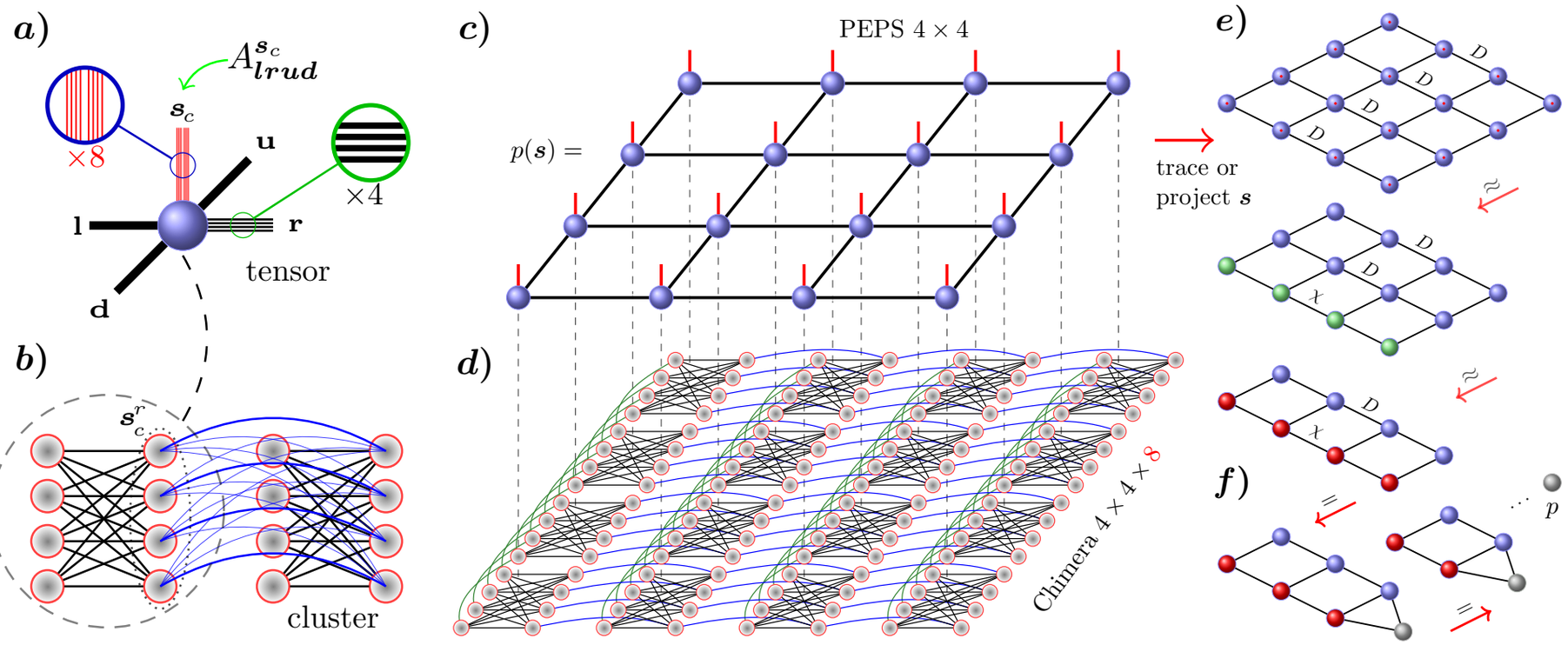


Highly mode non-generic, dynamic

*Corboz, Phys. Rev. B **94**, 035133 (2016)*

Starting to be used much more widely for applied math, classical systems

Can write distribution of spin glass as a 2D tensor network (= graphical model)



Use non-trivial tensor network algorithms to contract, sometimes provably bound the lowest energy

Rams, Mohseni, Gardas, arxiv:1811.06518

tensornetwork.org website:

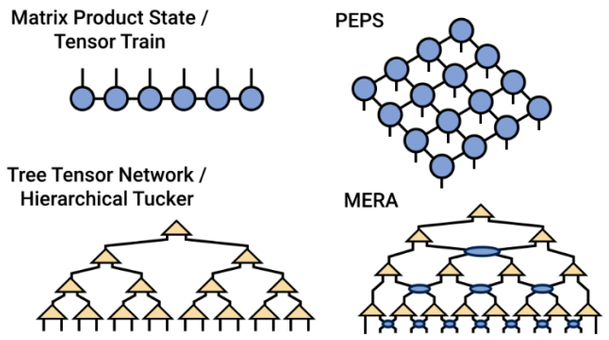
Welcome to the Tensor Network

This site is a resource for tensor network algorithms, theory, and software.

Tensor networks are factorizations of very large tensors into networks of smaller tensors, with applications in applied mathematics, chemistry, physics, machine learning, and many other fields.

Fundamentals:

- Tensor Diagram Notation



MPO-MPS Multiplication: Density Matrix Algorithm

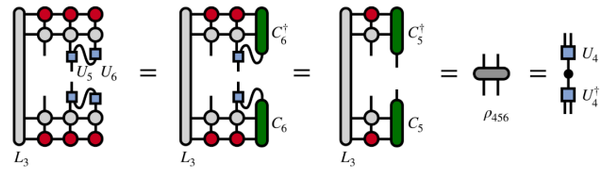
One way to multiply a matrix product state (MPS) (or *tensor train*) tensor network by a matrix product operator (MPO) is to use the product of these two networks as input to the density matrix MPS compression algorithm.

The page will describe the algorithm in detail, but it may be helpful to first review the simpler case of the algorithm that only involves compressing an MPS to a smaller MPS.

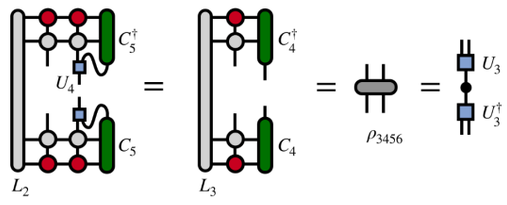
Statement of the Problem

Given an MPS of bond dimension M and an MPO of bond dimension k, we wish to represent their product as another MPS of bond dimension m. That is, we wish to find the MPS on the right-hand side of the following equation:

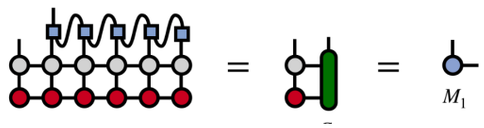
Having obtained U_5 above, one applies it to transform the basis and uncovers another external index of the MPO-MPS product, reusing the saved L_3 tensor to obtain the density matrix ρ_{456}. Diagonalizing this density matrix (with truncation) gives the next MPS tensor U_4:



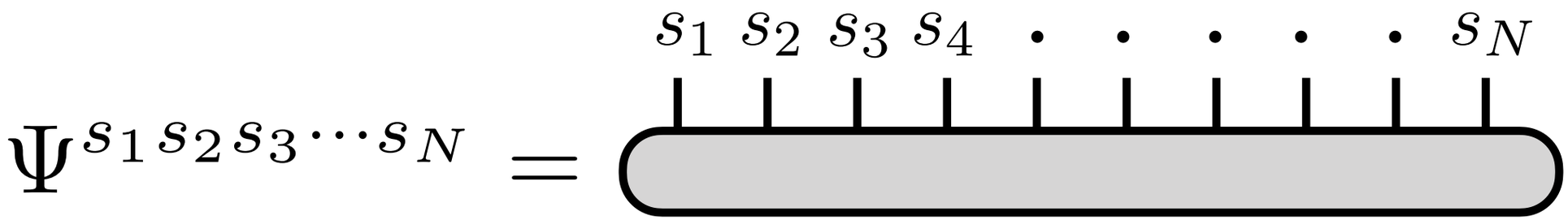
Continuing with steps similar to the ones above, one can continue to obtain the tensors U_3, U_2, etc. which diagonalize the reduced density matrices obtained by exposing each previous external index. For example, the steps to obtain U_3 are:



Finally, once all of the tensors down to U_2 have been obtained, the MPS tensor M_1 carrying the first external index can be computed as:



Draw N-index tensor as blob with N lines



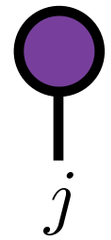
Diagrams for simple tensors

Diagrams for simple tensors

vector v_j

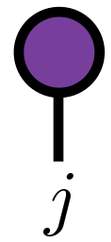
Diagrams for simple tensors

vector v_j



Diagrams for simple tensors

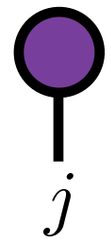
vector v_j



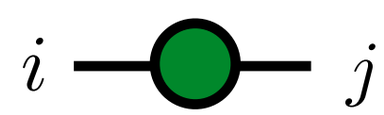
matrix M_{ij}

Diagrams for simple tensors

vector v_j

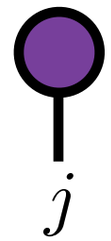


matrix M_{ij}

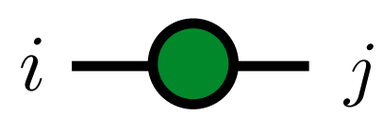


Diagrams for simple tensors

vector v_j



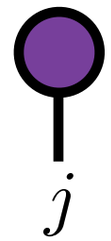
matrix M_{ij}



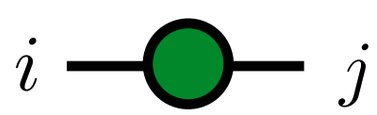
3-index tensor T_{ijk}

Diagrams for simple tensors

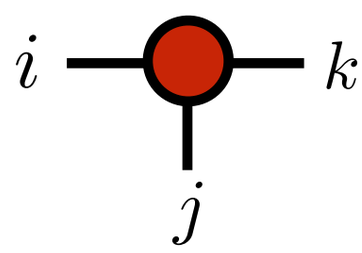
vector v_j



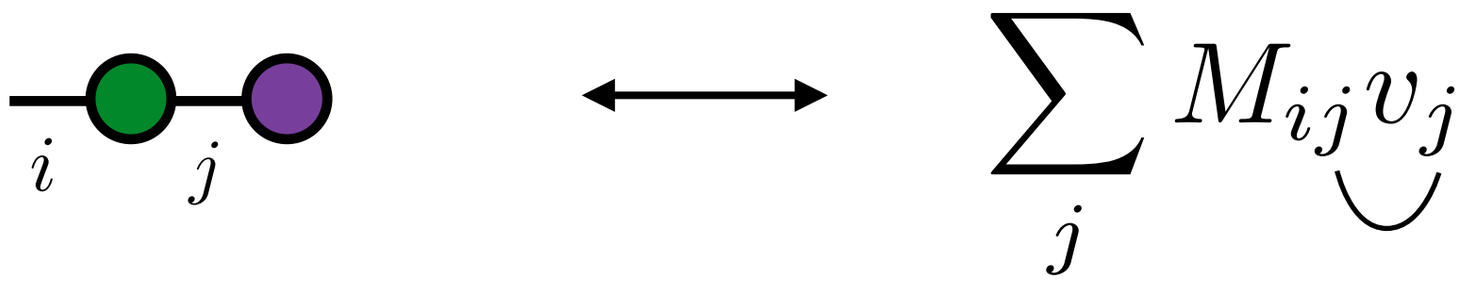
matrix M_{ij}



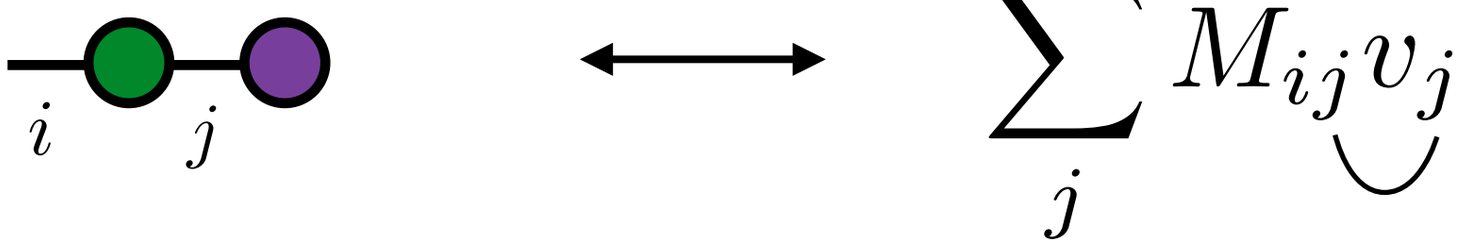
3-index tensor T_{ijk}

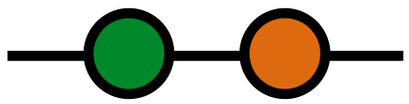


Joining lines means contraction, omit names

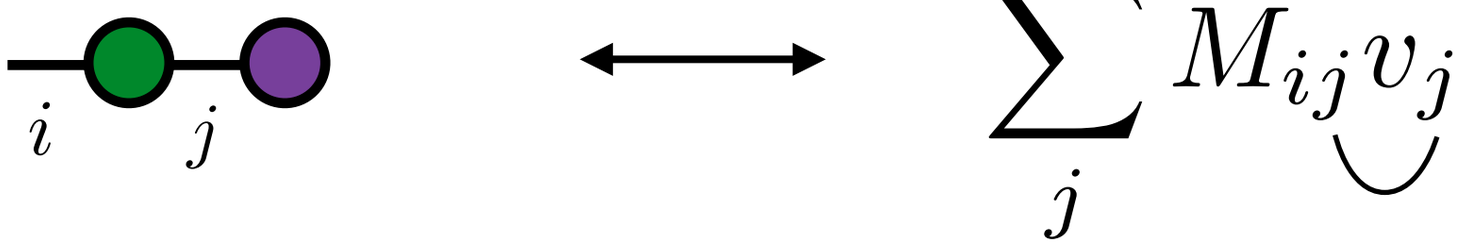


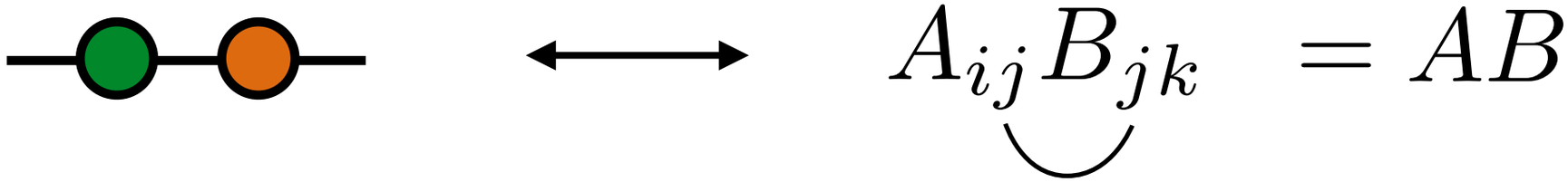
Joining lines means contraction, omit names



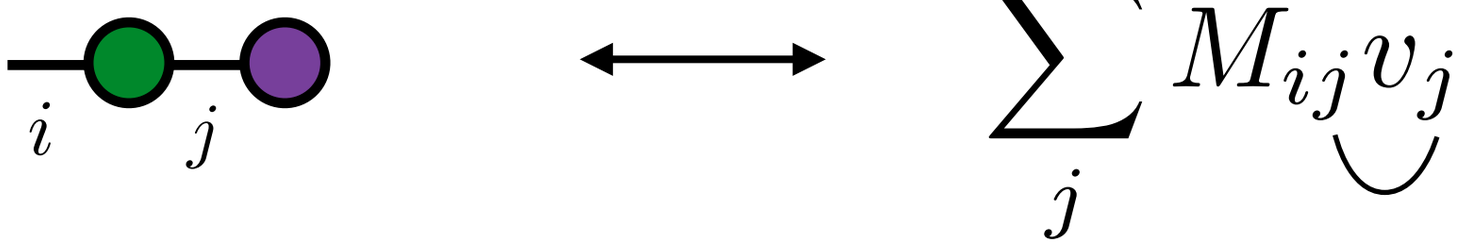


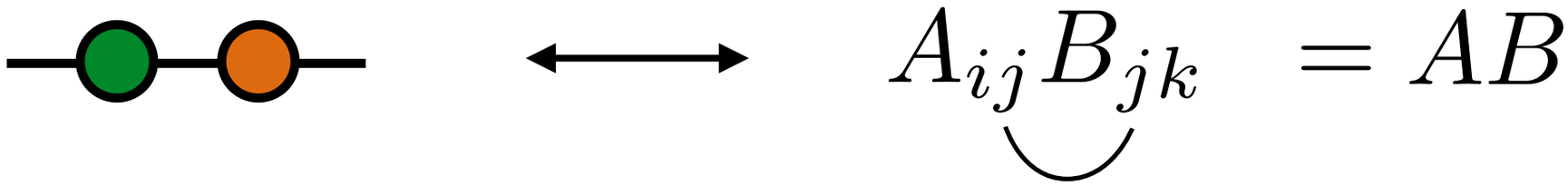
Joining lines means contraction, omit names

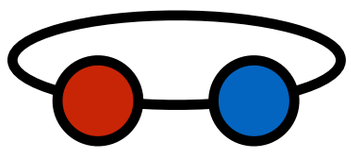




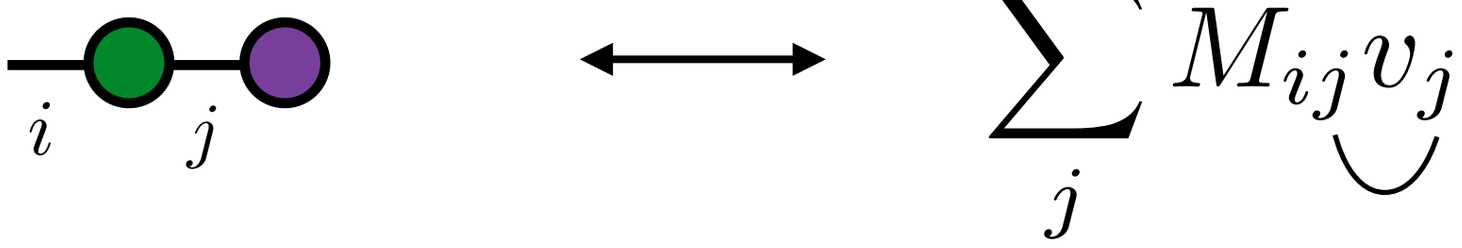
Joining lines means contraction, omit names

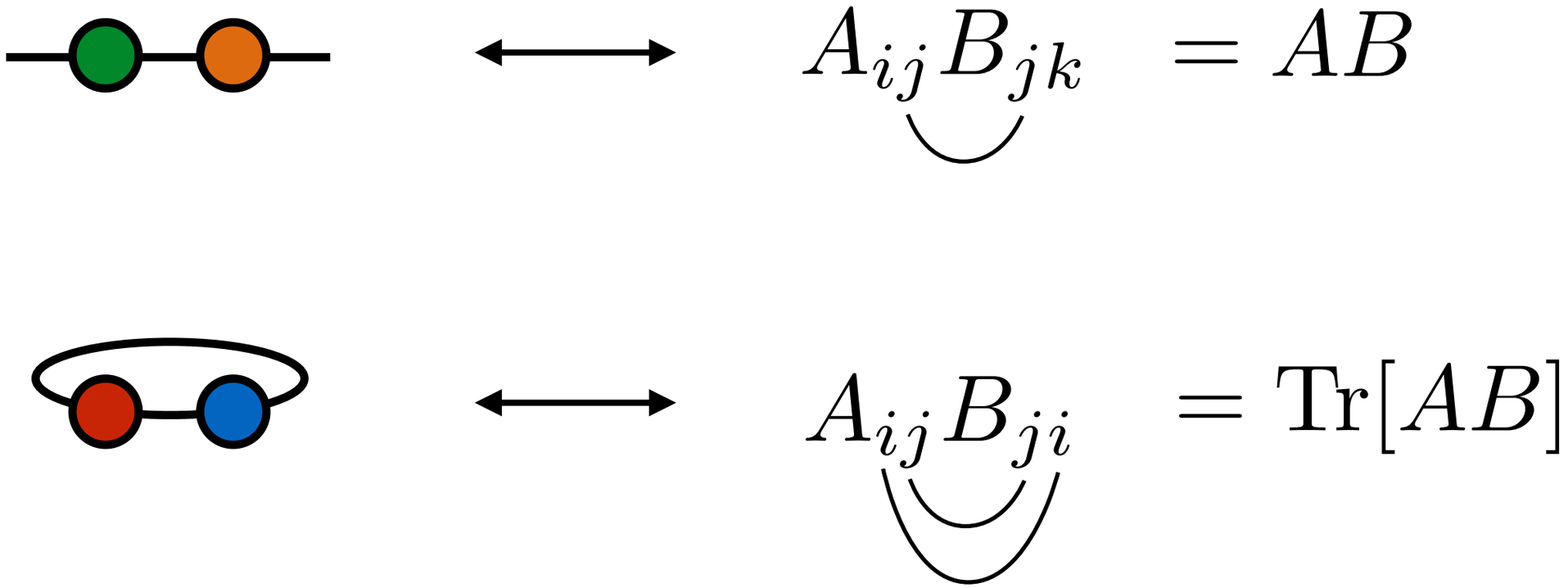




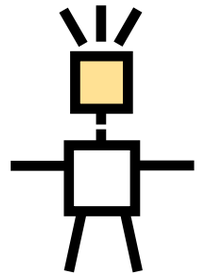


Joining lines means contraction, omit names





ITensor



itensor.org

The **ITensor** software is motivated by tensor diagram notation

The interface of an ITensor is independent of its memory layout (index ordering)

ITensor supports block-sparse tensors and advanced *tensor network* algorithms

ITensor Team

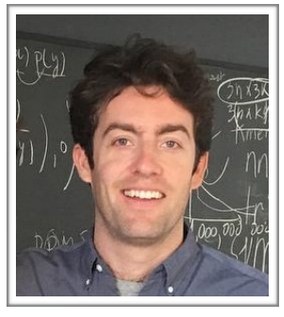
Matt Fishman

Data Scientist, CCQ

Poster

Miles Stoudenmire

Research Scientist, CCQ

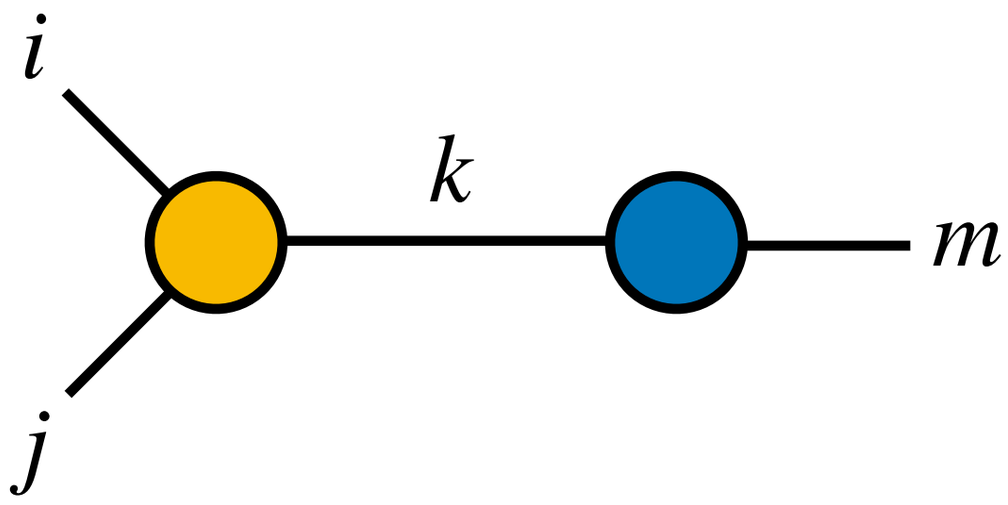


Steven R. White

Professor, UC Irvine

Consultant, CCQ

Conventional tensor library (*not ITensor*)

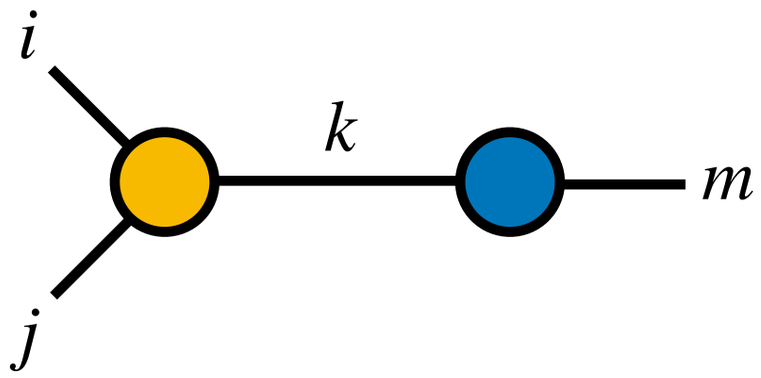



```
C = A["i,j,k"] * B["m,k"]
```

- Index labels are temporary
- Must think about index ordering
- Possible to mistake same-size indices

ITensor introduces "intelligent" indices

which recognize each other:

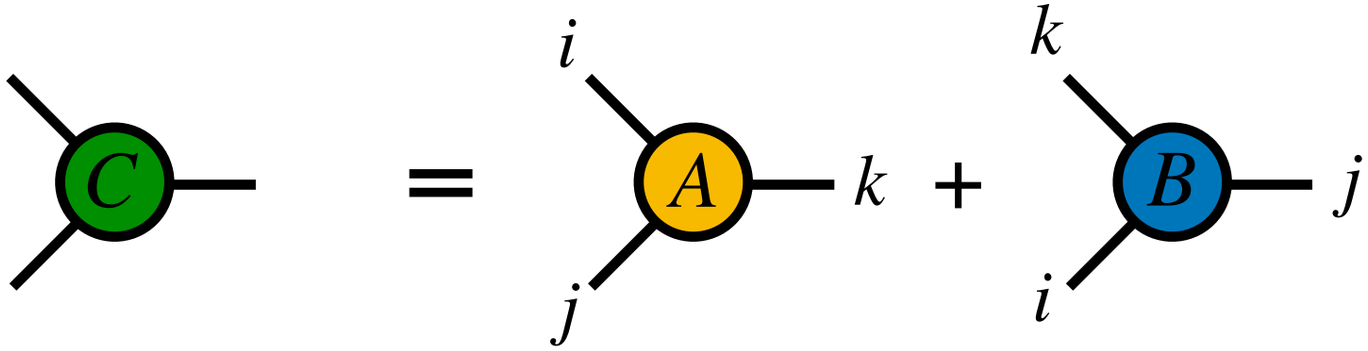



```
i = Index(5)
j = Index(7)
k = Index(3)
m = Index(12)

A = ITensor(i,j,k)
B = ITensor(m,k)

C = A * B
```

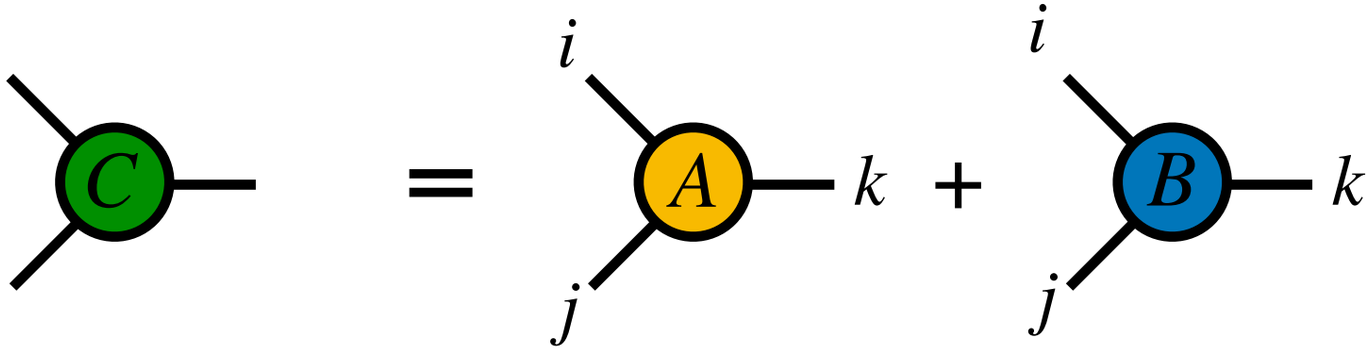
Adding ITensors "just works"




```
C = A + B
```

No thinking about index ordering

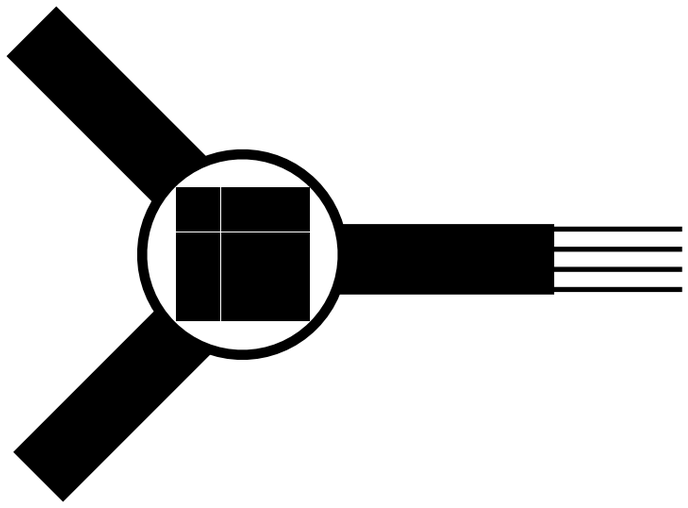
Adding ITensors "just works"




```
C = A + B
```

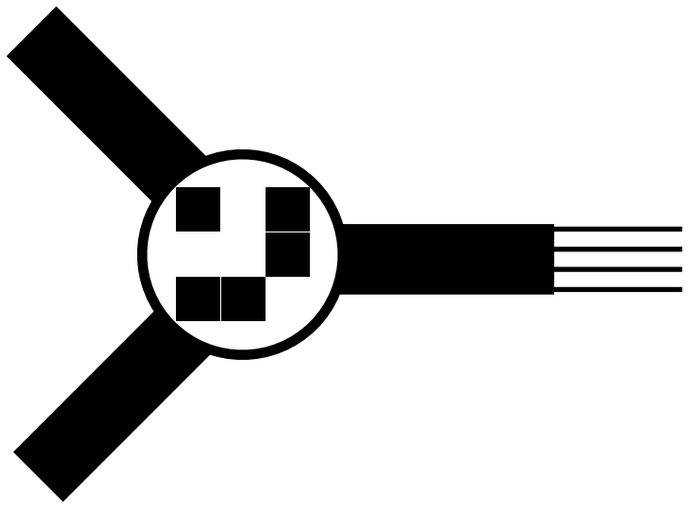
No thinking about index ordering

Index objects also useful because...



} indices can be split into "sectors"

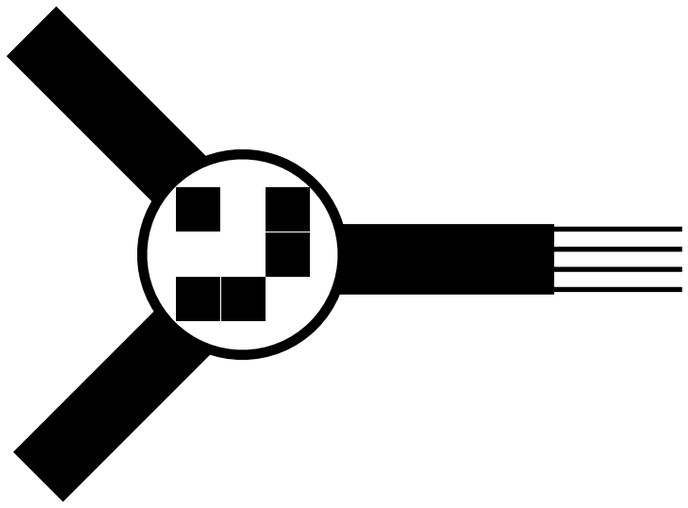
Index objects also useful because...



} indices can be split into "sectors"

Enforces **block sparsity** of tensors = performance improves

Index objects also useful because...

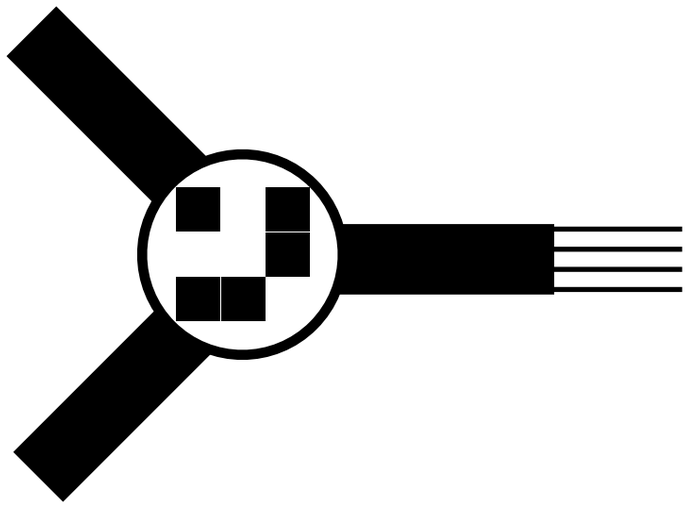


} indices can be split into "sectors"

Enforces **block sparsity** of tensors = performance improves

Indicates e.g. number of particles in a spatial subregion

Index objects also useful because...



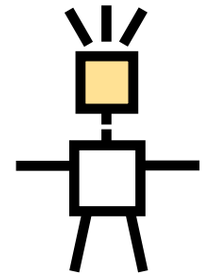
} indices can be split into "sectors"

Enforces **block sparsity** of tensors = performance improves

Indicates e.g. number of particles in a spatial subregion

Very useful for info to reside in indices themselves

Future Directions – ITensor



Moving to Julia language – great fit for us and users

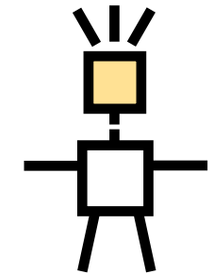
Support for tensor BLAS backends

More sparse tensor formats

Adding more features for application domains:

element-wise operations and auto-differentiation

Wish List / Needs – ITensor



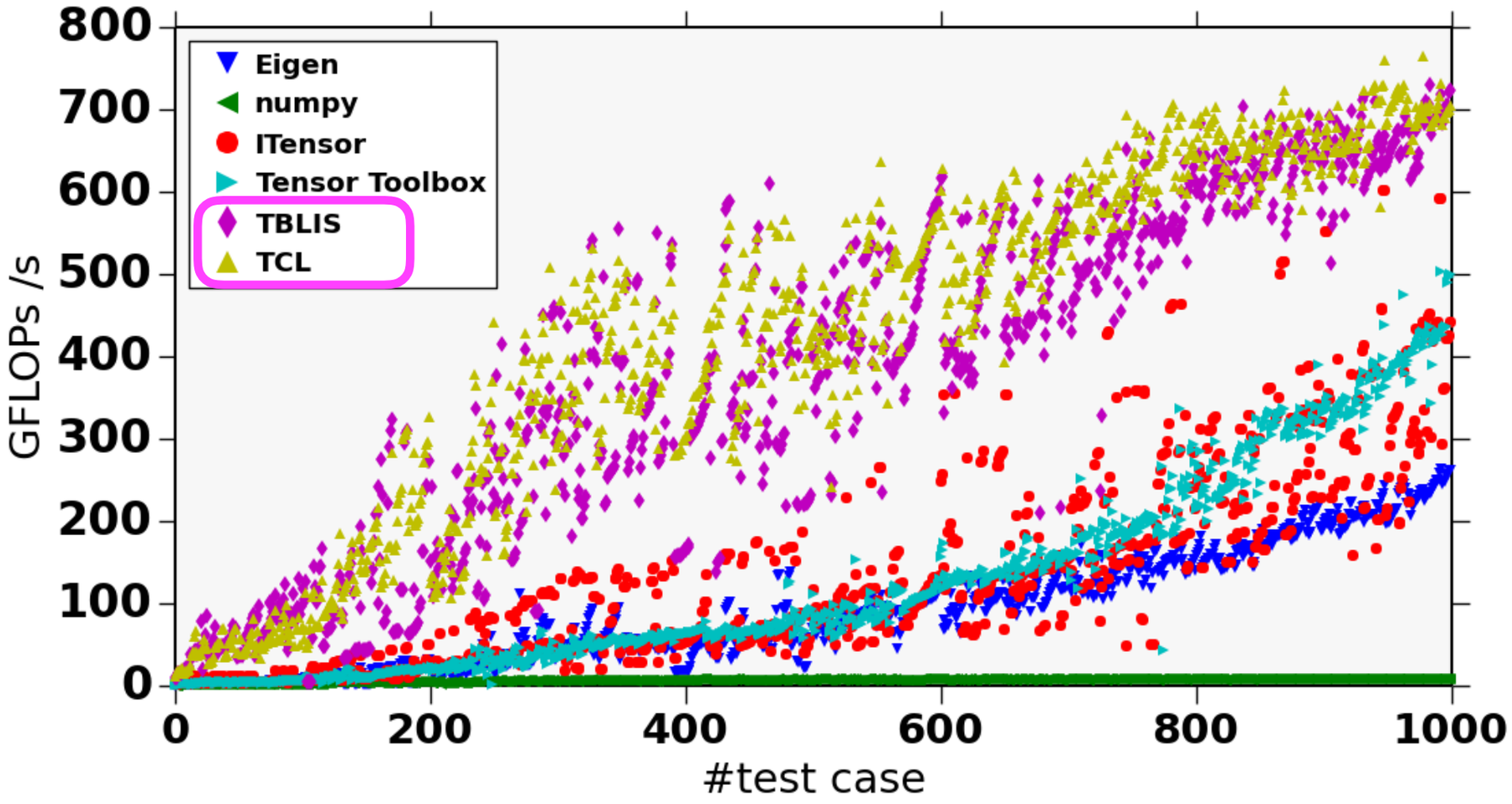
Using GPUs well – memory movement, reuse

Parallelism for block-sparse (& other sparse formats)

Coordination of parallel resources at different levels of lib.

Lots for us at intermediate (tensor), high (algorithm) levels ; would love libraries for above needs

"Tensor BLAS" codes for contracting dense tensors:



Devin Matthews, *High-Performance Tensor Contraction Without Transposition*, arxiv:1607.00291

Paul Springer: *High-Performance Tensor Operations* (Doctoral Thesis), RWTH Aachen University, 2018