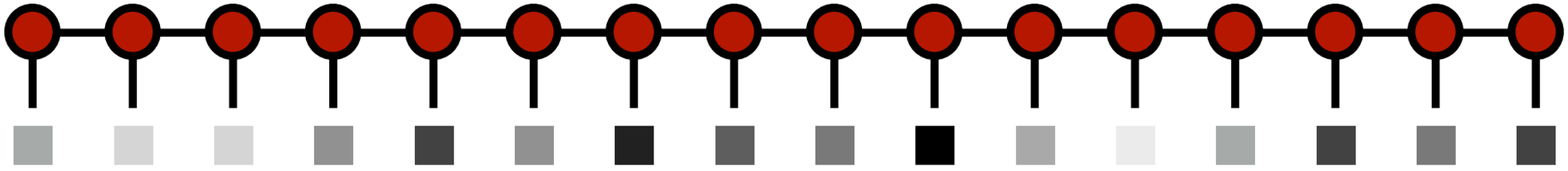
Tensor Networks for Machine Learning and Applications



E.M. Stoudenmire

Mar 2021 - IPAM Tensor Methods

FLATIRON INSTITUTE
SIMONS FOUNDATION

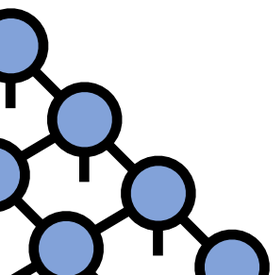
Overview

Tensor networks are a natural way to parameterize interesting and powerful machine learning models

Perspective on this area from physics point of view

Today:

- Overview
- Tensor Networks Architectures & Applications
- Algorithms & Future Directions

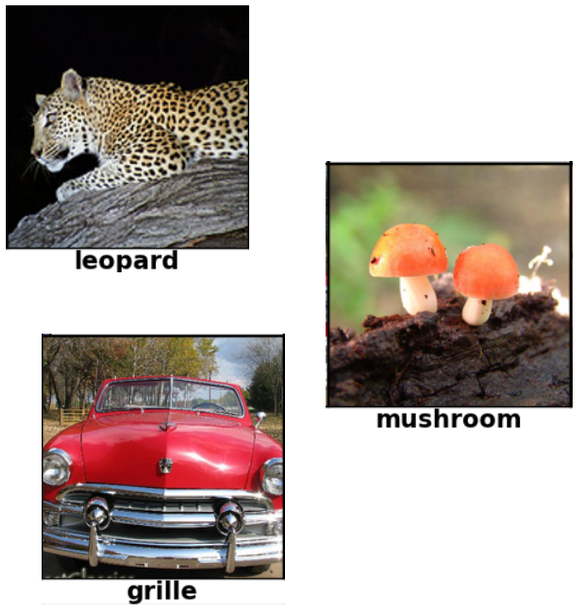


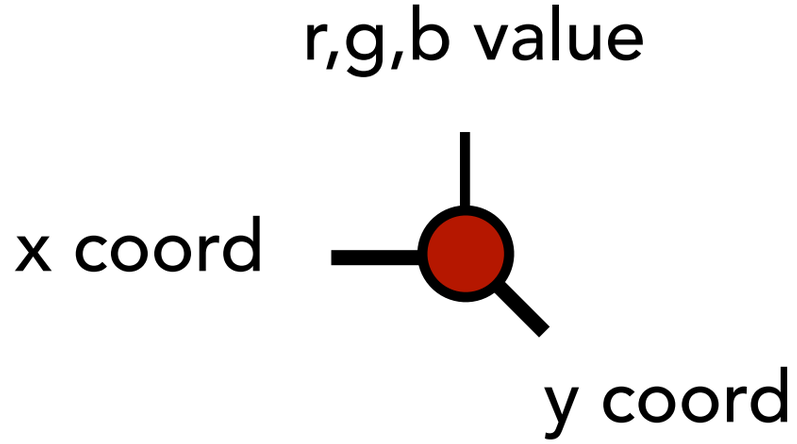
Tensors in Machine Learning

Where can tensors appear in machine learning applications?

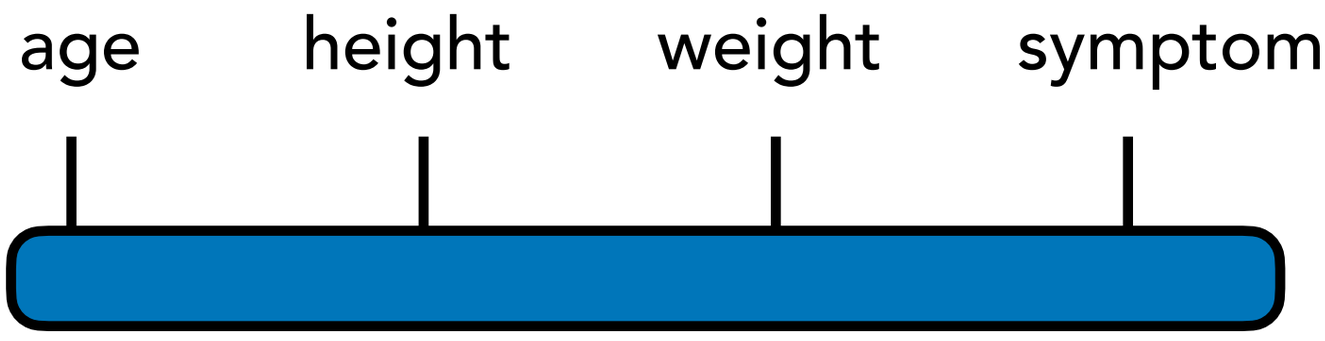
Multi-Dimensional Data

Image Data

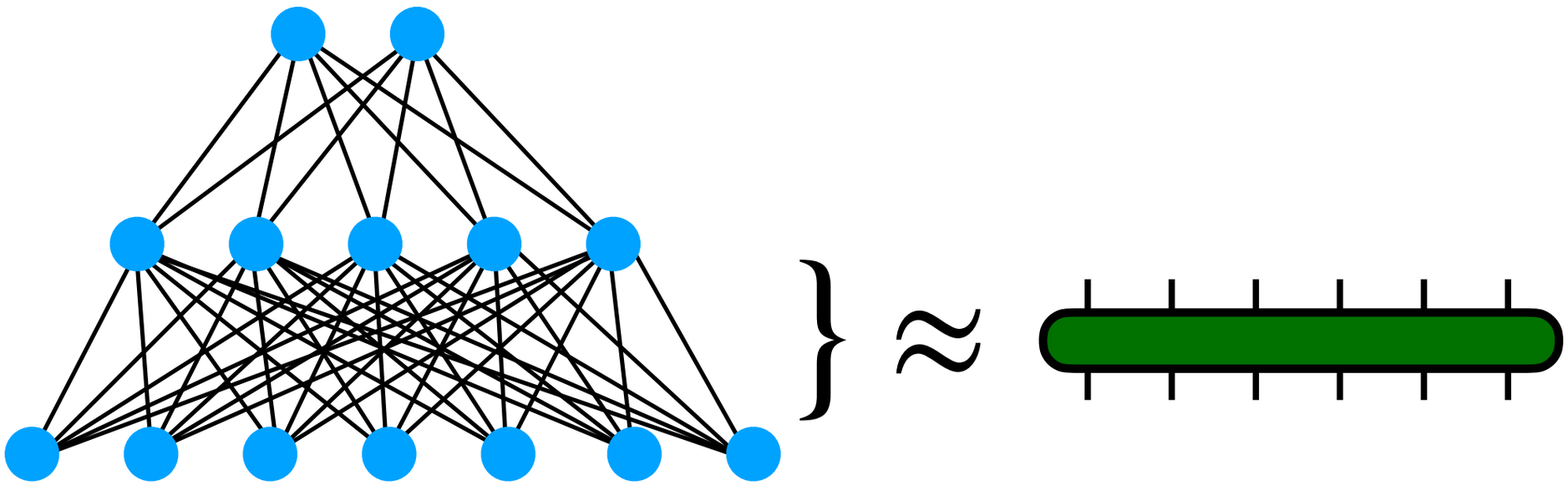




Medical Data



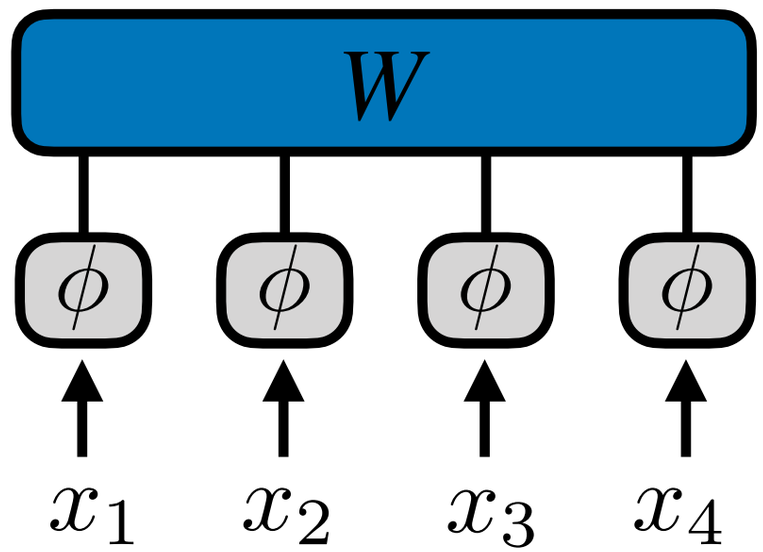
Neural Network Weight Layers



Possible to interpret as a very high-order tensor
(not just a matrix)

Linear Weights of High-Dimensional Models

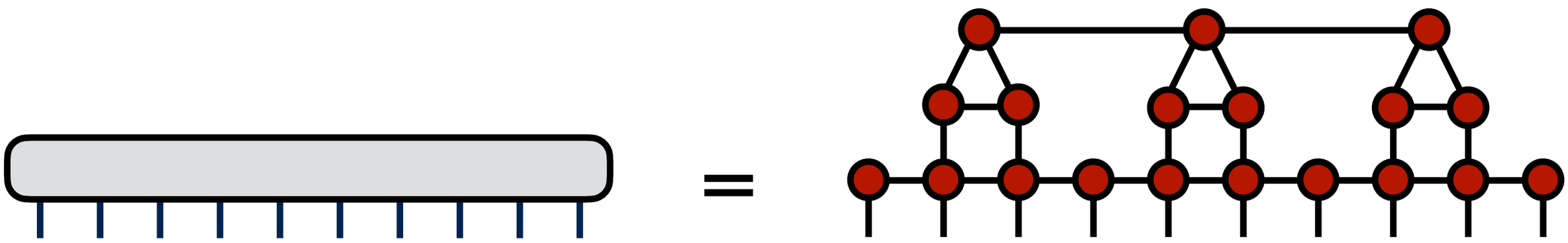
$$f(\mathbf{x}) = W \cdot \Phi(\mathbf{x})$$



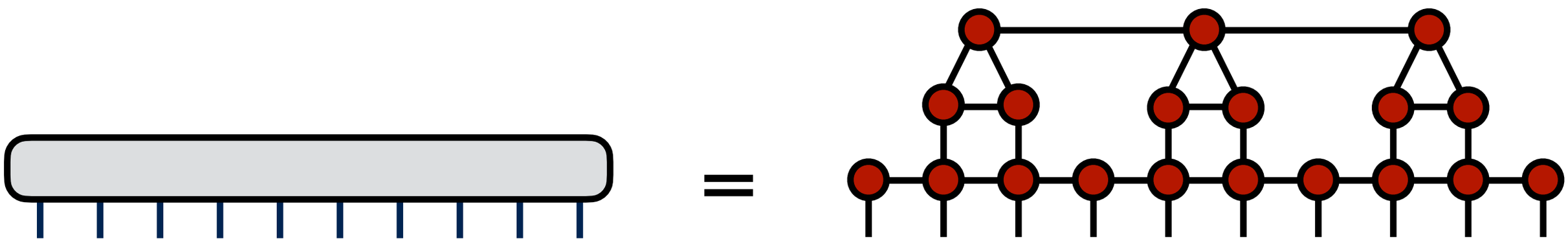
For certain cases of kernel learning and Gaussian processes, weights are naturally a *high-order tensor*

Why Tensor Networks?

Tensor network = factorization of huge tensor into contracted product of smaller tensors



Tensor network = factorization of huge tensor into contracted product of smaller tensors



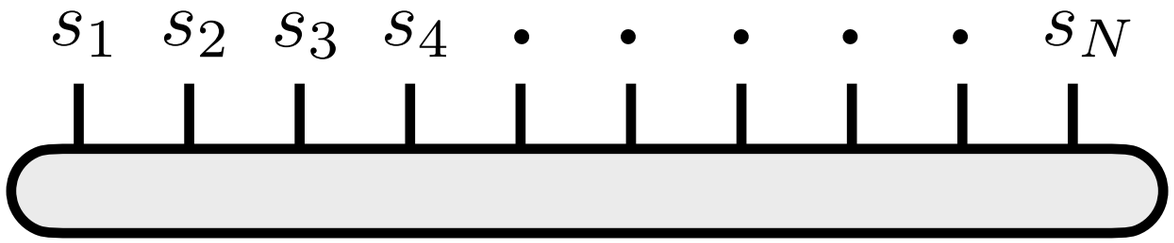
Benefits:

- exponential reduction in **memory** needed
- exponential speedup of **computations** (addition, product)
- theoretical **insight** and **interpretation**
- **estimation** of missing or corrupted entries
- many optimization **algorithms** & strategies

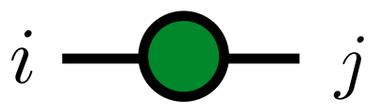
Notation – Tensor Diagrams

N-index tensor = shape with N lines

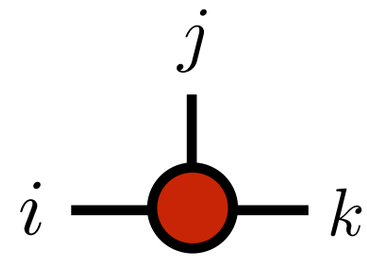
$$T^{s_1 s_2 s_3 \cdots s_N} =$$



v_j



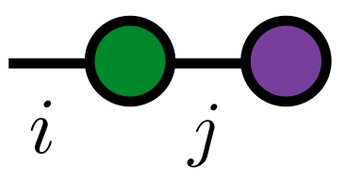
M_{ij}

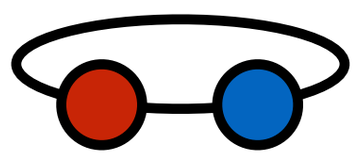


T_{ijk}

Joining lines means contraction

$M_{ij}v_j$

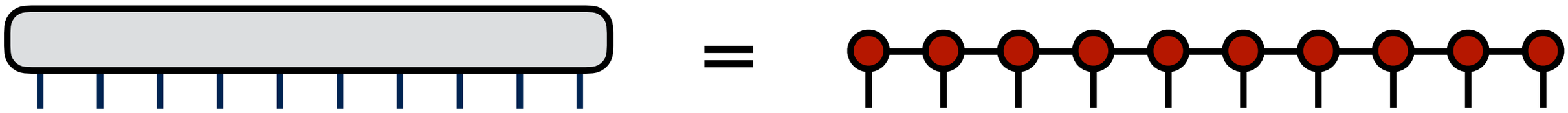




$A_{ij}B_{ji}$

Best understood tensor network is the

matrix product state (MPS)[1,2] or tensor train [3]

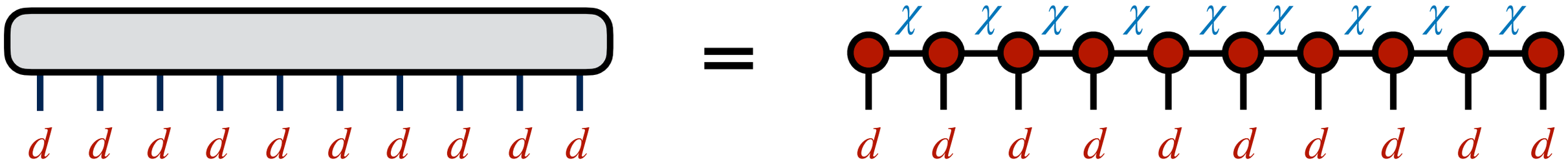


[1] Östlund, Rommer, PRL 75, 3537 (1995)

[2] Vidal, PRL 91, 147902 (2003)

[3] Oseledets, SIAM J. Sci. Comp. 33, 2295 (2011)

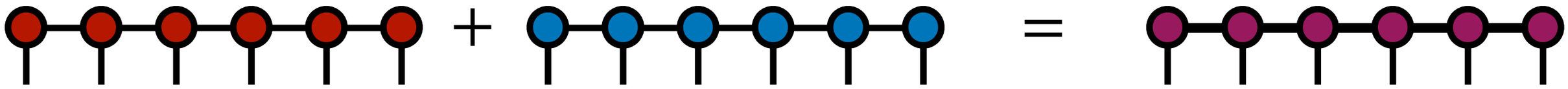
Adjustable parameter of matrix product state (MPS) is bond dimension χ



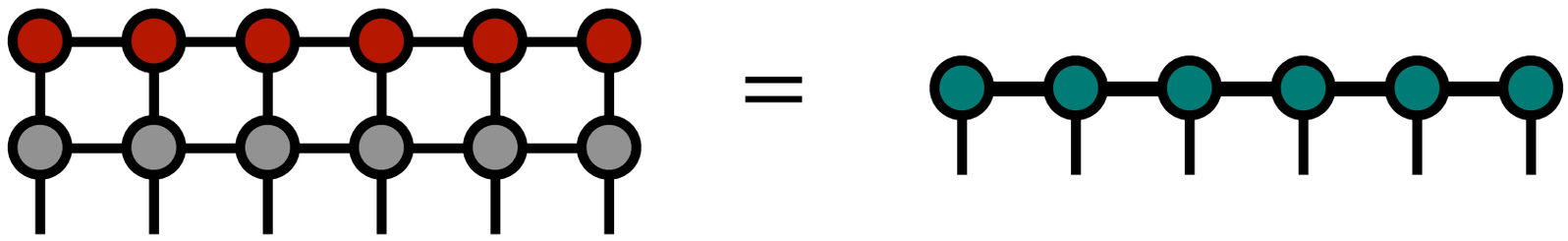
If modest χ yields good approximation, obtain massive compression:

$$d^N \longrightarrow N d \chi^2$$

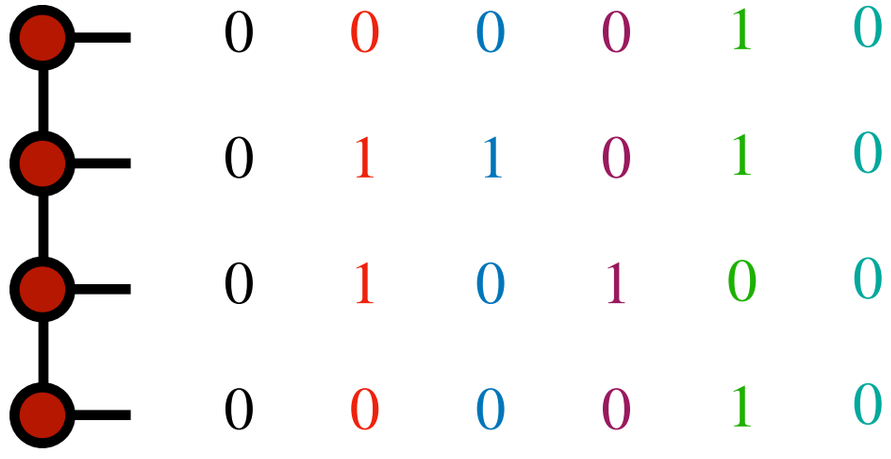
Can efficiently sum MPS in compressed form:



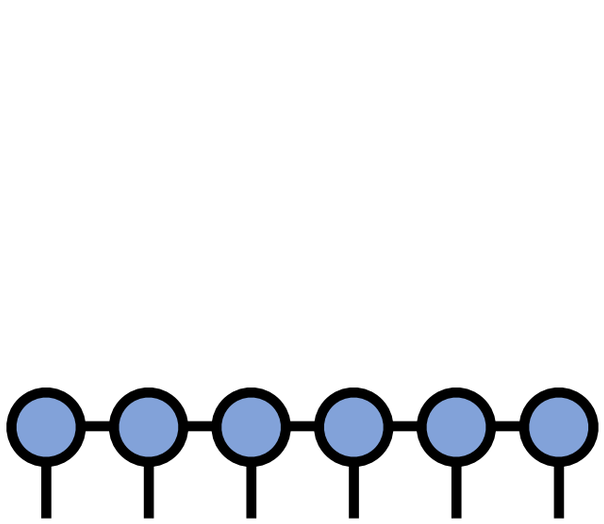
multiply by other networks:



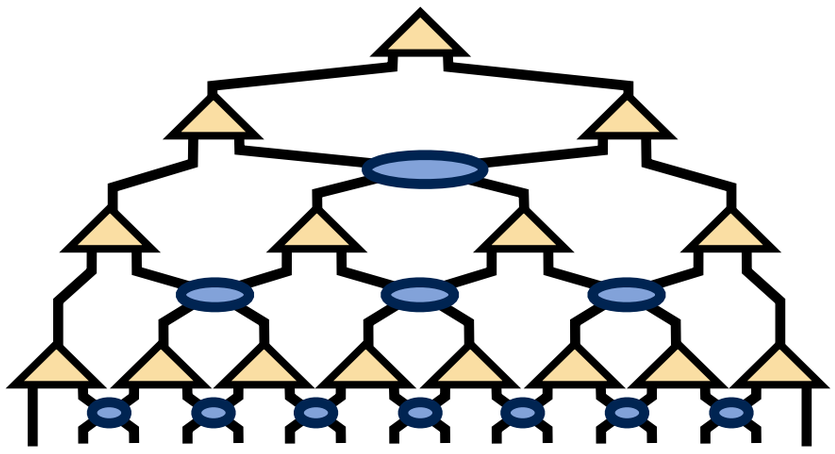
and perfectly sample:



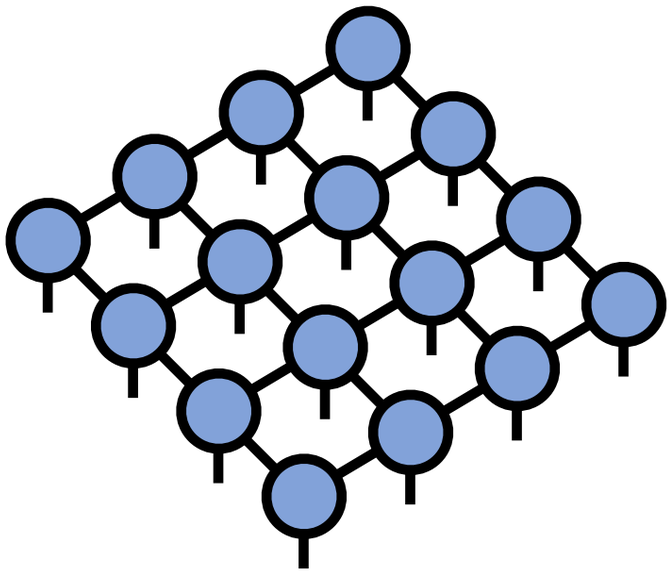
In quantum physics, have rich theory of
which tensor networks are suited for particular "data"



1D system



1D *critical* system



2D system

(Here "data" = samples/measurements of a quantum wavefunction)

Tensor networks a general tool for linear algebra in exponentially high-dimensional spaces

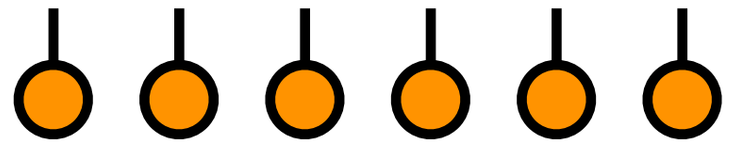
For example, entanglement entropy really just a measure of multilinear tensor rank

Architectures & Applications

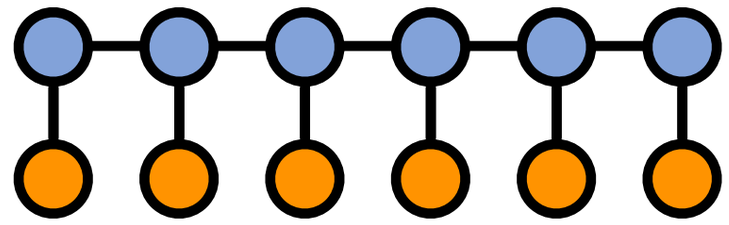
Tensor networks
beyond finite MPS

Most straightforward application of tensor networks to machine learning is using MPS

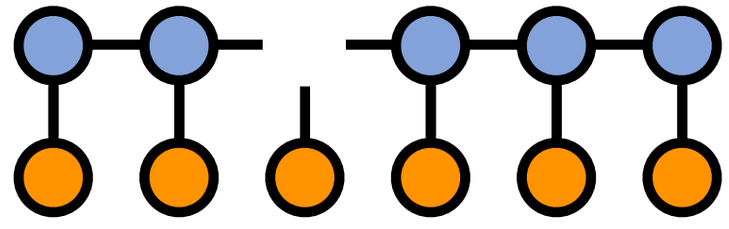
1. map data to tensor product features

$\mathbf{x} \rightarrow$ 

2. evaluate MPS model

$f(\mathbf{x}) =$ 

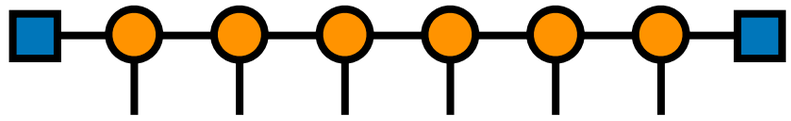
3. optimize MPS tensors for machine learning objective (supervised, unsupervised)

$\frac{\partial f}{\partial A_3} =$ 

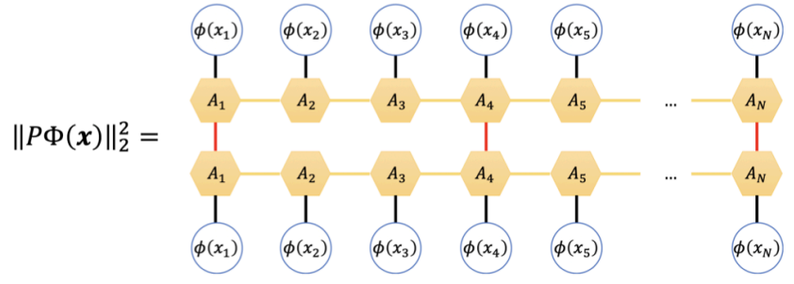
Novikov, Trofimov, Oseledets, arxiv:1605.03795
Stoudenmire, Schwab, Advances in NIPS **29** (2016)
Han et al., PRX **8**, 031012 (2018)

Since 2016, tensor network machine learning now successfully "ported" to other tensor net architectures

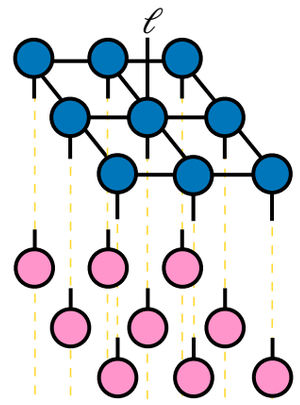
Infinite MPS



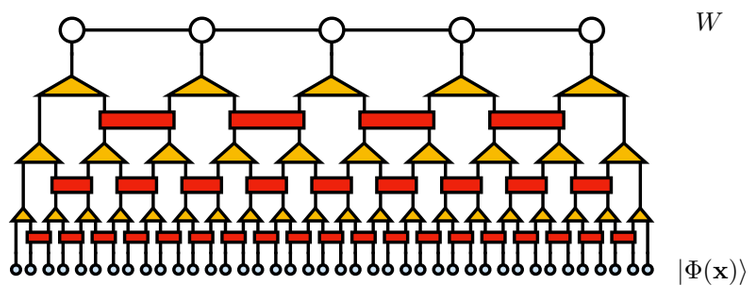
Locally purified states



PEPS

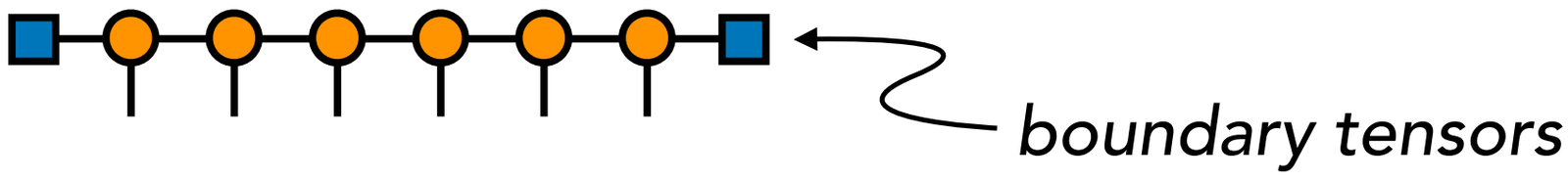


MERA



Infinite MPS

Miller, Rabusseau, Terilla, "Tensor Networks for Probabilistic Sequence Modeling", arxiv:2003.01039



- used to generate model languages with various grammars
- very few parameters and parallel optimization
- superior results to LSTM in many cases, equal in most others
- can *generalize* from training on shorter sequences to correct results on longer sequences (so really learning the grammar)

Locally purified states

Anomaly Detection with Tensor Networks

arxiv:2006.02516

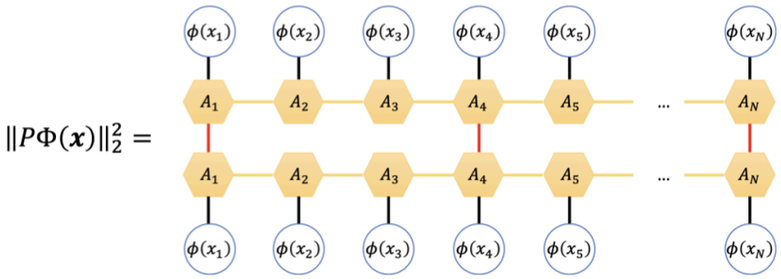


Table 3: Mean AUROC scores (in %) and standard errors on ODDS datasets.

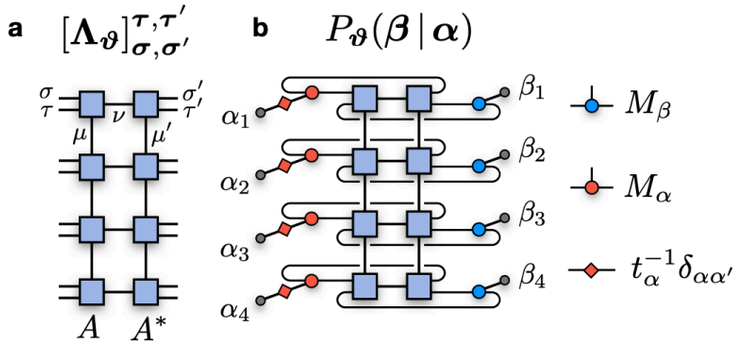
Dataset	OC-SVM	IF	GOAD	DAGMM	TNAD
Wine	**60.0**	46.0 ± 8.4	48.2 ± 24.7	51.7 ± 19.3	$\mathbf{97.3 \pm 4.5}$
Glass	**62.0**	57.2 ± 1.6	53.5 ± 13.6	52.5 ± 12.9	$\mathbf{81.8 \pm 7.3}$
Thyroid	98.8	$\mathbf{99.0 \pm 0.1}$	95.8 ± 1.3	88.8 ± 6.8	$\mathbf{99.0 \pm 0.1}$
Satellite	**79.9**	77.2 ± 0.9	60.6 ± 5.3	72.1 ± 4.7	$\mathbf{81.3 \pm 0.5}$
Forest	**97.7**	71.7 ± 2.6	64.6 ± 4.7	60.9 ± 8.9	$\mathbf{98.8 \pm 0.6}$

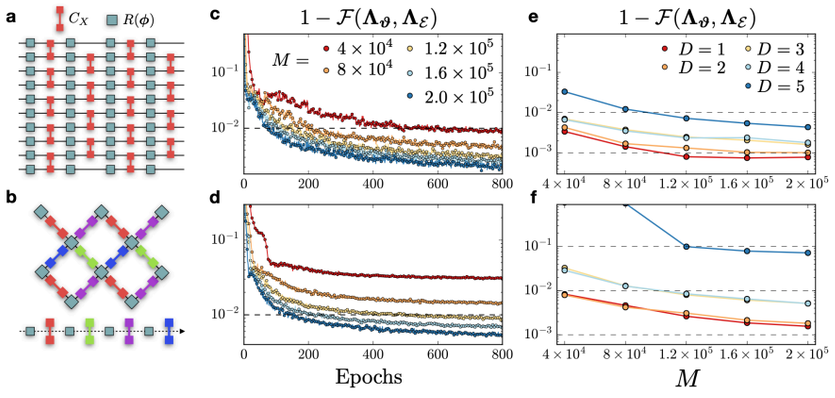
Novel anomaly detection framework

Results better than neural networks for tabular data

Quantum process tomography with ... tensor networks

arxiv:2006.02424

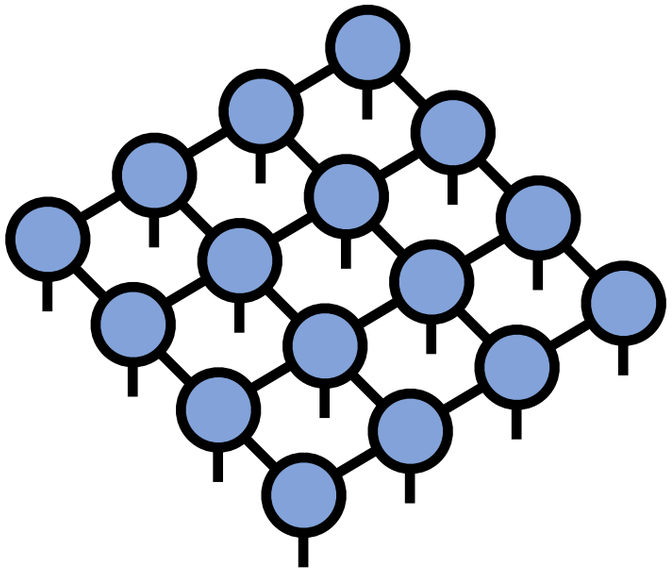




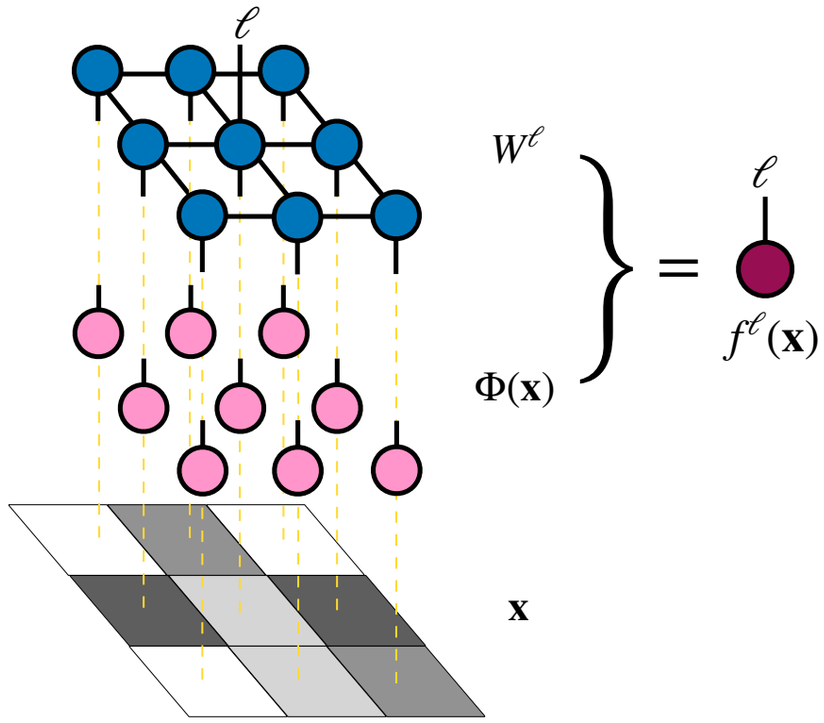
Scalable learning of noisy quantum "channels" or processes from experiments

PEPS (Projected Entangled Pair States)

Cheng, Wang, Zhang, "Supervised Learning with PEPS" arxiv:2009.09932

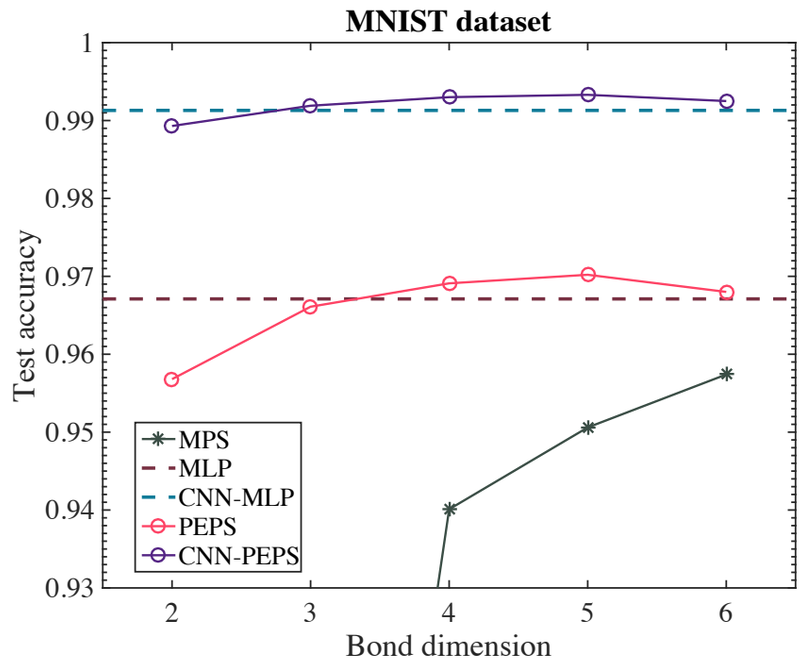


PEPS = 2D analogue of MPS



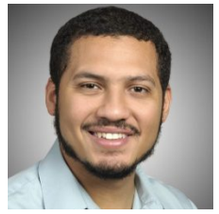
Framework for learning

- comparable results to NN with fewer parameters
- starting to see overfitting at larger D?
- convolutional layers help (91% test correct on FashionMNIST using CNN+PEPS!)
- cost to train is high

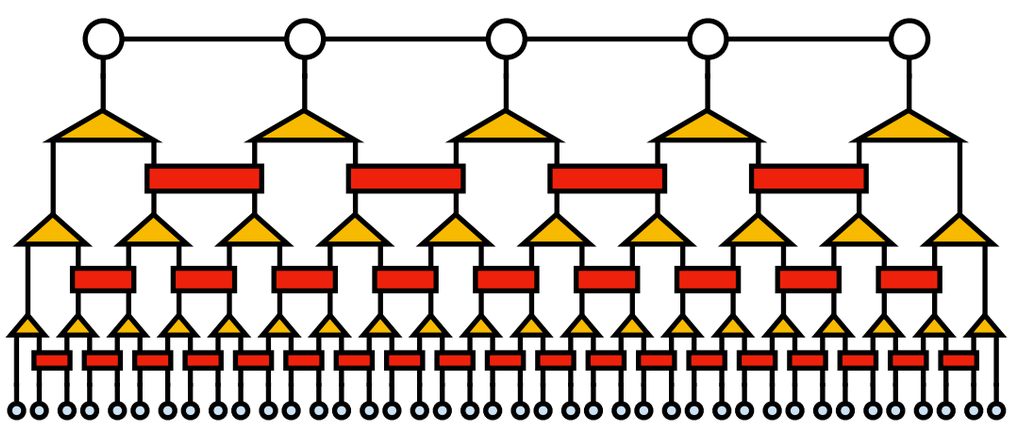


MERA for Audio Classification

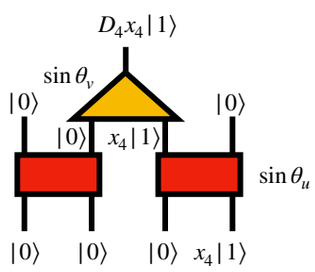
Reyes, Stoudenmire, "A Multi-Scale Tensor Network Architecture for Classification and Regression"
arxiv:2001.08286



Justin Reyes

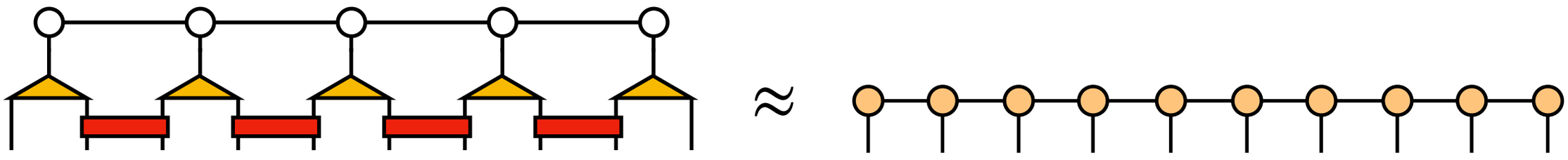


Factorize weights as MERA network with trainable MPS weights on top



Fixed MERA layers encode a discrete wavelet transform

Key capability is fine-graining weights through layers:



Algorithms & Theory

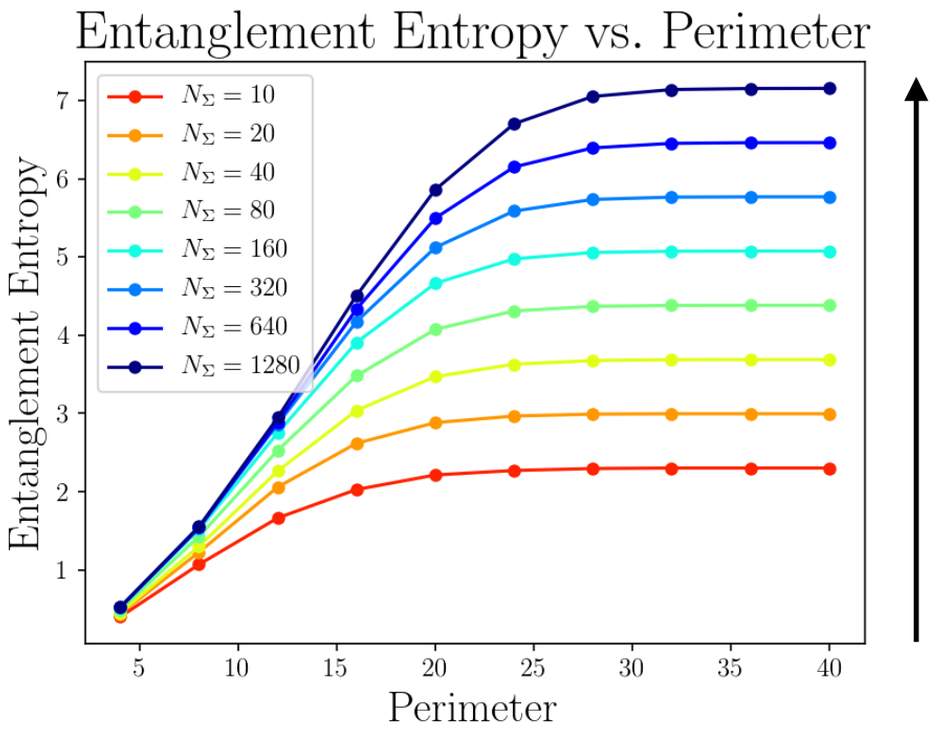
including expressivity, data efficiency,
model adaptation, ...

Why can tensor networks succeed?

Are image datasets comparable to low-entanglement wavefunctions in physics?

Tentative answer: **no!**

Sums of MNIST images:

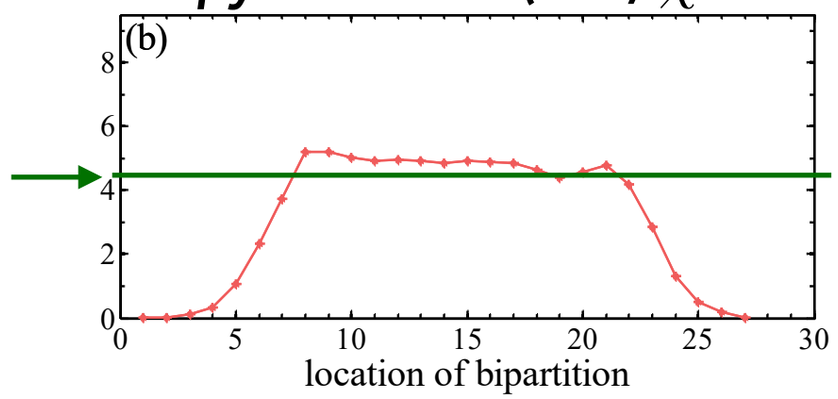


Martyn, Vidal, et al. arxiv:2007.06082

Rapidly increasing entanglement as more images summed

Entropy of MNIST (10k, $\chi \approx 100$)

random MPS of same bond dimension



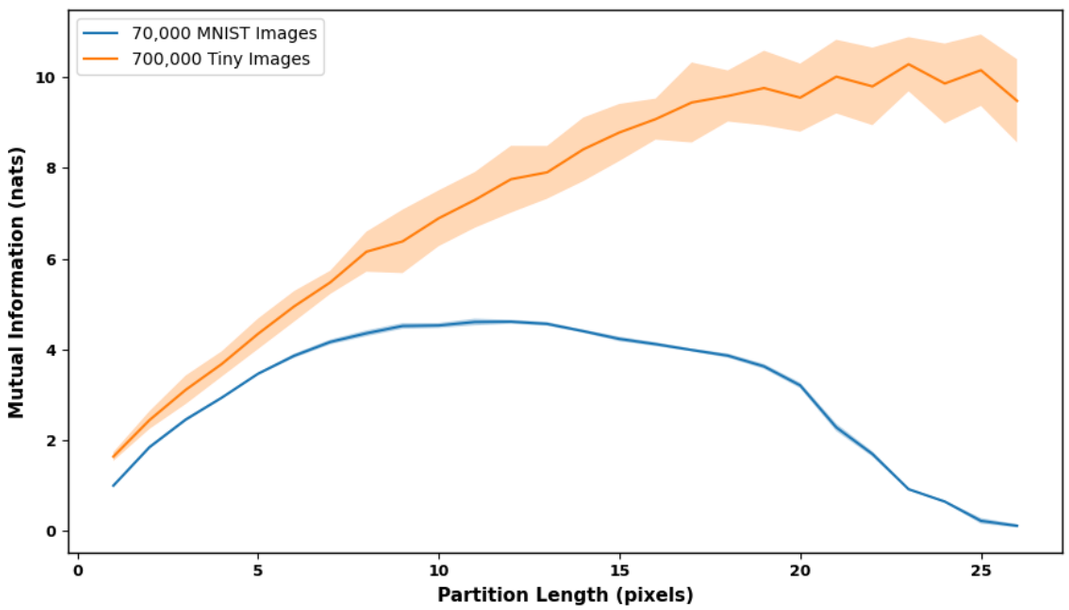
Cheng, Chen, Wang, Entropy 20, 583 (2018)

Why can tensor networks succeed?

Two recent papers give different perspective:

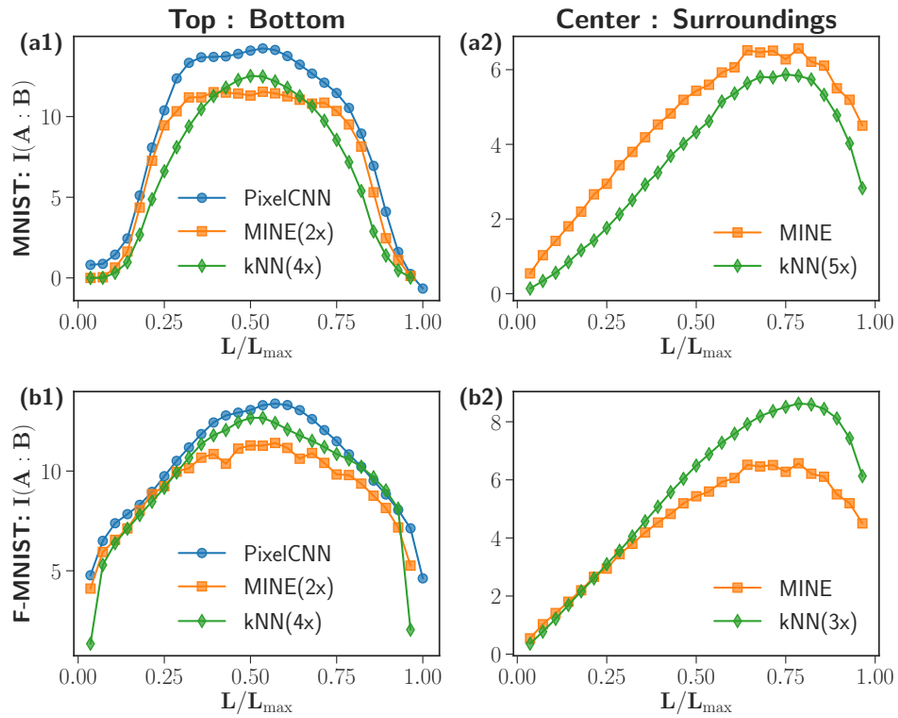
- mutual information (MI) may be best quantity to consider
- strategies to estimate MI

Convy, Huggins, et al. arxiv:2103.00105



- study synthetic Gaussian data
- realistic images have MI like that of near-neighbor correlated Gaussian

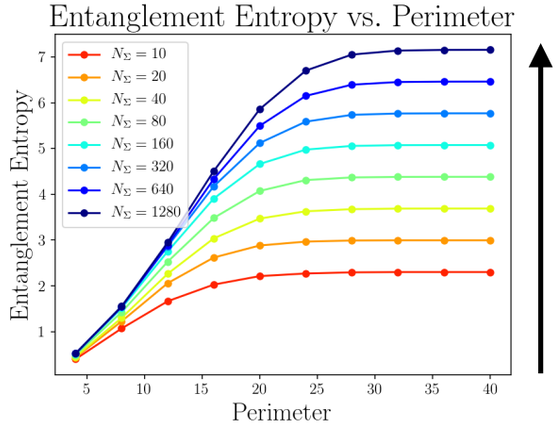
Lu, Kanasz-Nagy, et al. arxiv:2103.06872



- MI of text is beyond area law scaling
- some images are area law, but others may be beyond area law

Why do tensor networks succeed at all?

If images high entanglement,
how do tensor networks succeed?



Rapidly increasing entanglement as more images summed

Martyn, Vidal, et al. arxiv:2007.06082

Possible answers:

- *Convy et al says*: quantity studied by Martyn et al. may be overly sensitive to choice of feature map
- entropy might decrease as more images summed
- supervised learning may require much less resources

Important that we can even pose and answer these questions!

Progress in Training Algorithms

Algorithms could be what sets tensor network models apart

Available algorithms (partial list):

- ALS + gradient descent
- ALS + optimal local update (Stokes, Terilla)
- modified ALS (= 2 site DMRG)
- Riemannian optimization
- TT-cross algorithm
- density matrix algorithm (polynomial-scaling TT-SVD)
- ....

Progress in Training Algorithms

An interesting story of two algorithms for a challenging dataset

The dataset: even-parity bit strings

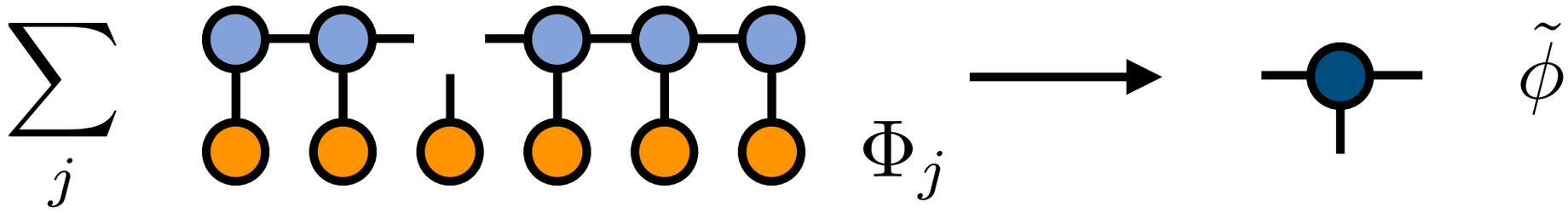
$$\left.\begin{array}{c} 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1 \\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0 \\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0 \\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 1 \\ \vdots \\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \end{array}\right\} \text{even \# of 1's}$$

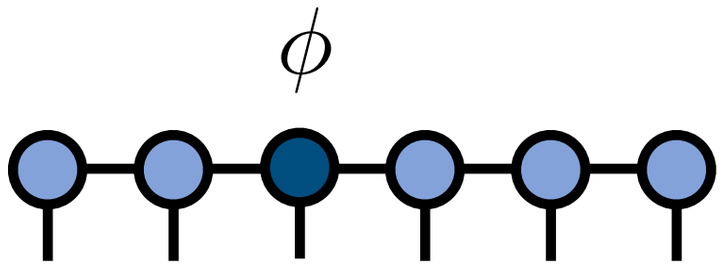
Feature map: $0 \rightarrow$  $= \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $\quad 1 \rightarrow$  $= \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Progress in Training Algorithms

An interesting story of two algorithms for a challenging dataset

Algorithm #1: optimal <u>local</u> update (generative modeling)





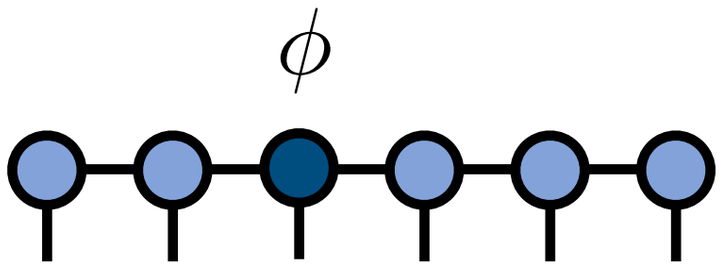
locally improved MPS

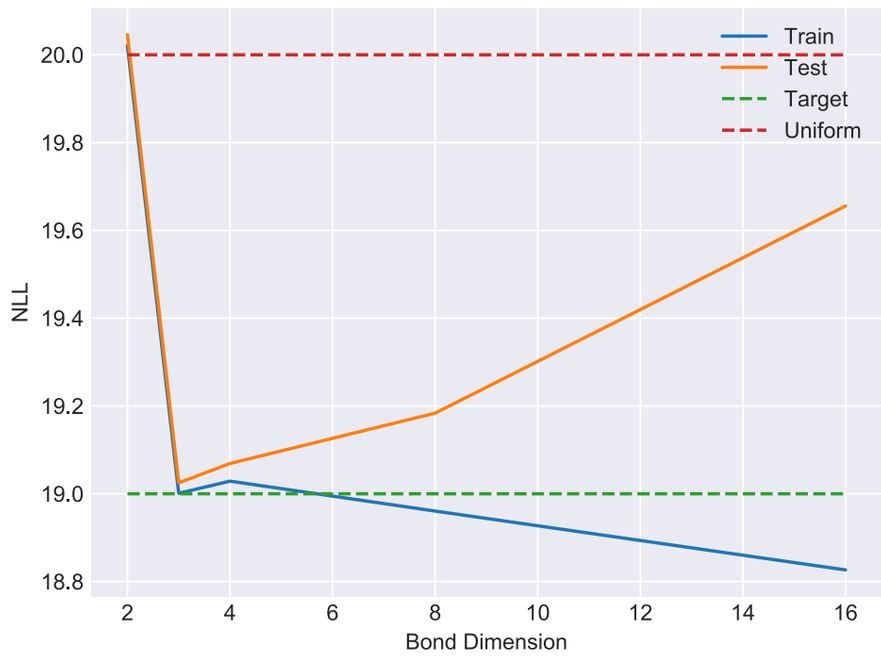
Stokes, Terilla, Entropy **21**, 1236 (2019)

Progress in Training Algorithms

Algorithm #1: optimal <u>local</u> update (generative modeling)

- can use geometric reasoning to develop
- excellent results up to length N=20 bitstrings
- linear scaling in data length and training set size
- can fail for poor choice of initial state

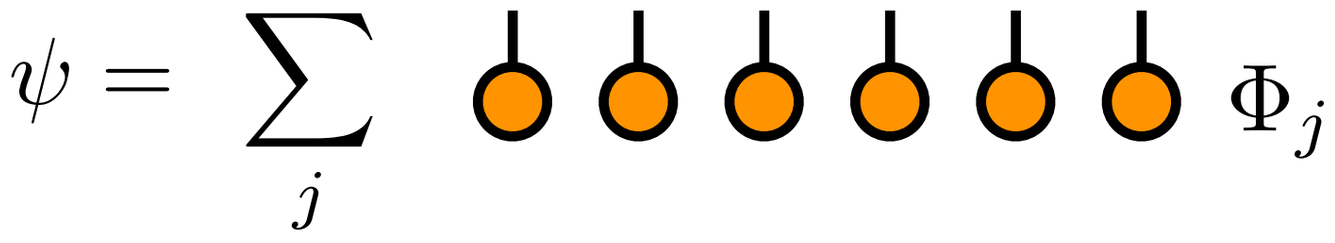




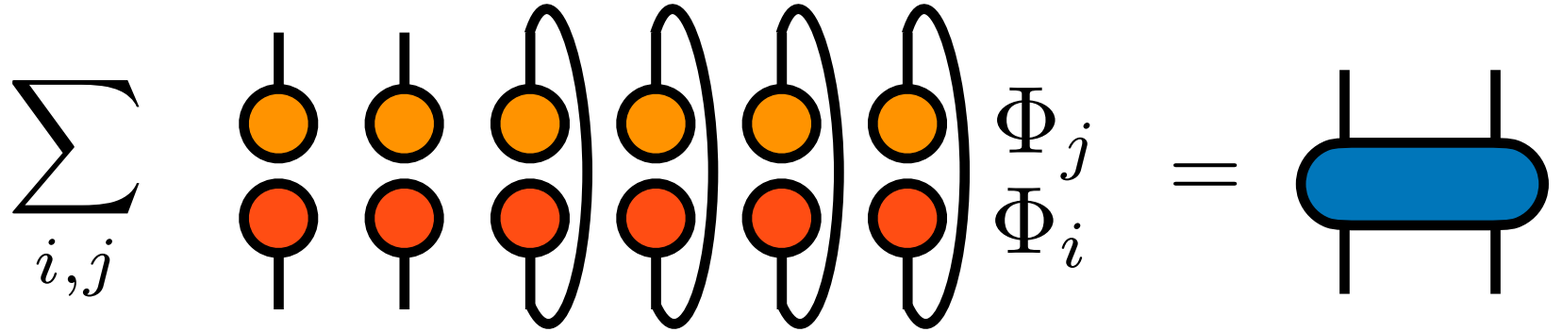
Stokes, Terilla, Entropy **21**, 1236 (2019)

Progress in Training Algorithms

Algorithm #2: direct compression / density matrix algorithm



1. define "target tensor"



2. trace double sum to obtain reduced "density matrix"

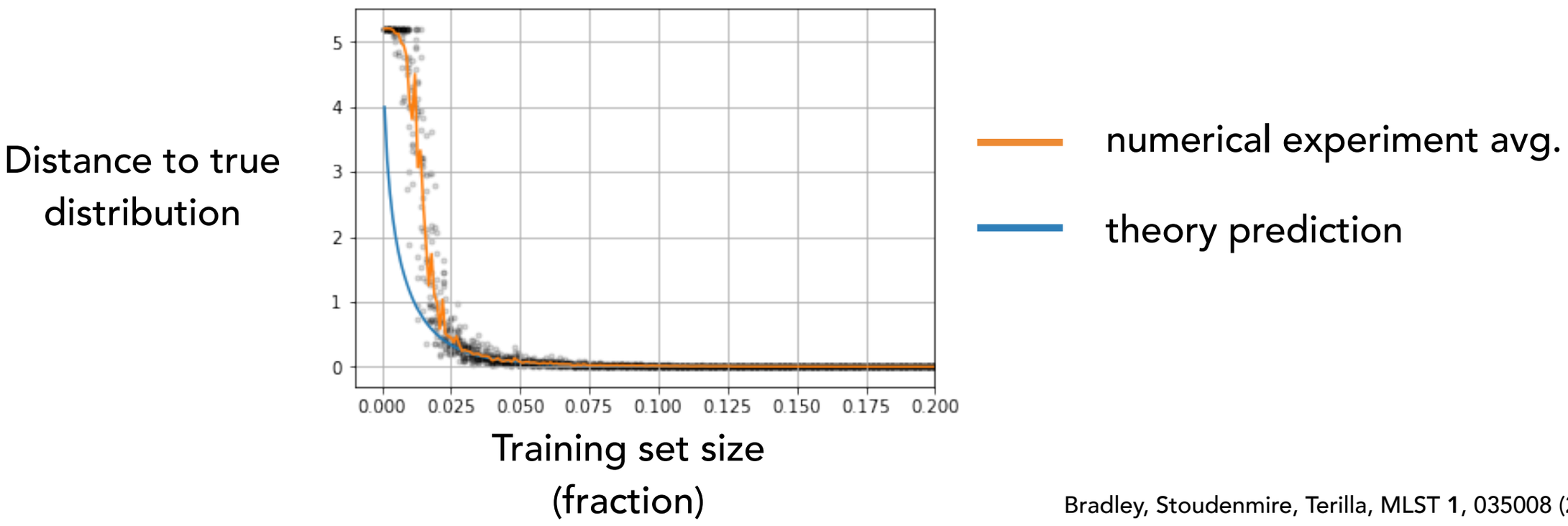
3. diagonalize density matrices to obtain MPS tensors one at a time

Bradley, Stoudenmire, Terilla, MLST **1**, 035008 (2020)

Progress in Training Algorithms

Algorithm #2: direct compression / density matrix algorithm

- cannot get stuck & gives deterministic results
- scales quadratically in training set size
- can develop a theory of generalization:



Bradley, Stoudenmire, Terilla, MLST **1**, 035008 (2020)

Outlook & Future Directions

Many opportunities to fix downsides of tensor network optimization algorithms

Main downsides:

- cost to train (memory + time) can be high
- best architecture choices still being explored

Opportunities to improve:

+ use sparse (e.g. block-sparse) tensors within models

+ use infinite tensor networks more

+ use symmetries more (evidence conv. layers help, infinite TNs for translation symmetry, ...)

+ devise more data-efficient approaches (theory can help)

Push interpretability / understanding / theory frontier

Opportunities here:

+ can characterize which data is learnable by a given type and size of tensor network

+ theory of generalization (Bradley et al.) could be improved:
  - tighter bounds,
  - apply to other data sets
  - estimate using summary statistics of real data

+ progress in theory will feed back into better training algorithms

One final thought:

Tensor networks likely just the "right" way to do linear algebra in very high-dimensional spaces (versus being quantum mechanics oriented)

We haven't even figured out the biggest pieces of tensor networks yet, such as analog of QR or SVD factorizations for matrix-like tensor networks

Some major progress only happened recently, such as canonical forms of PEPS (arxiv:1902.05100) or progress in randomized algorithms (arxiv:2003.05101)