Efficient Machine Learning with Tensor Networks

Qibin Zhao

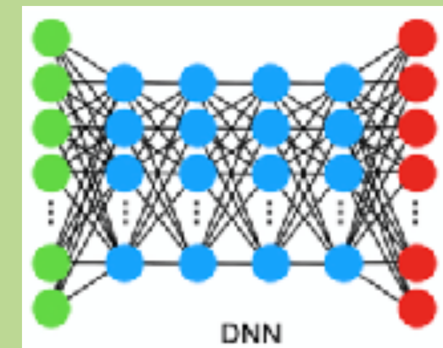
Tensor Learning Team
RIKEN AIP
https://qibinzhao.github.io

Success of Deep Learning

Big Data

Computation

Model & Algorithm



Tensor Networks

Mathematics

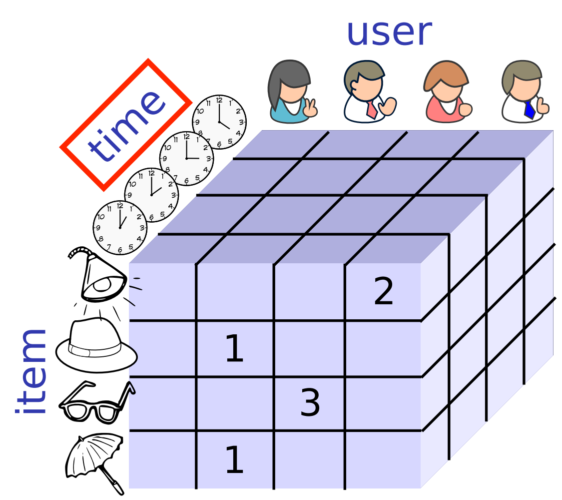
Quantum Physics

Outline

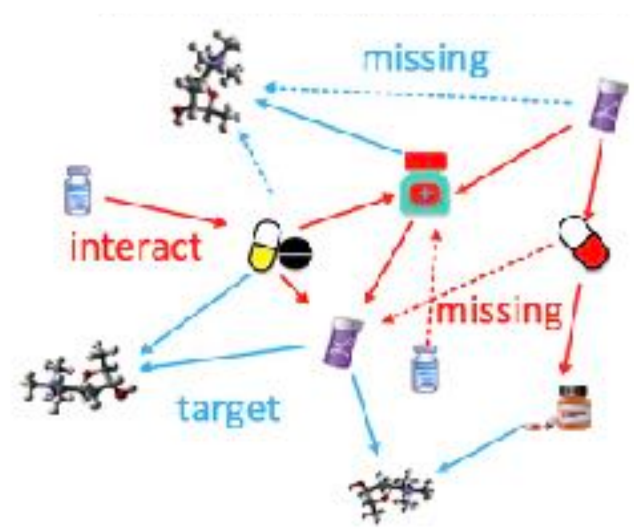
- Learning from incomplete or limited data
- Parameter efficient machine learning models

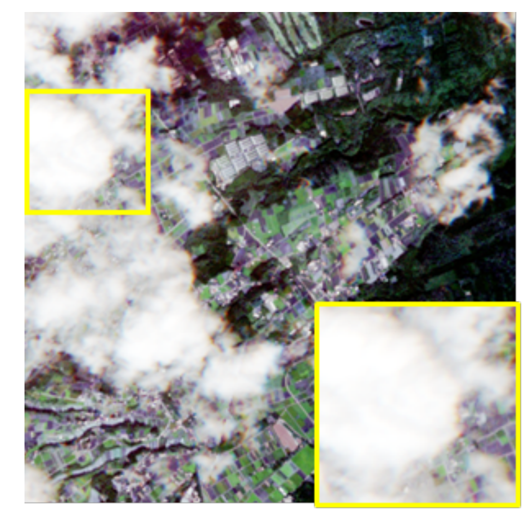
Learning from Imperfect Data

- Recommender system, social network
- knowledge graph prediction, drug repositioning
- Image or video inpainting/denoising



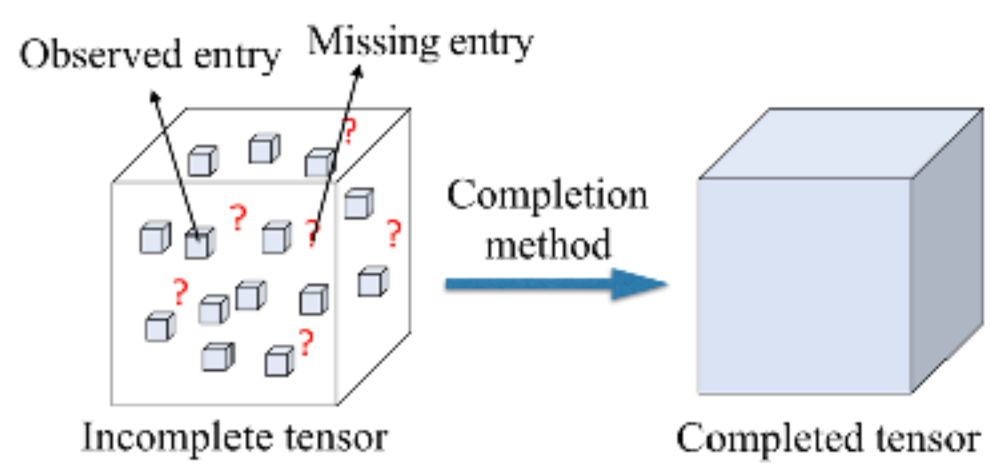
3 order tensor





- Task: learning from an incomplete tensor to predict values for unobserved positions

$$\mathcal{Y}_{\Omega} \to \mathcal{Y}_{\bar{\Omega}}$$



Tensor Completion

Objective:

$$\min_{\mathcal{X}} \|\Omega * (\mathcal{Y} - \mathcal{X})\| + R(\mathcal{X})$$

Fitting error Structure Regularizer

Challenges: data efficiency & efficient optimization

Approaches:

- Low-rankness assumption (convex, not scalable)

$$R(\mathcal{X}) = \|\mathcal{X}\|_*$$

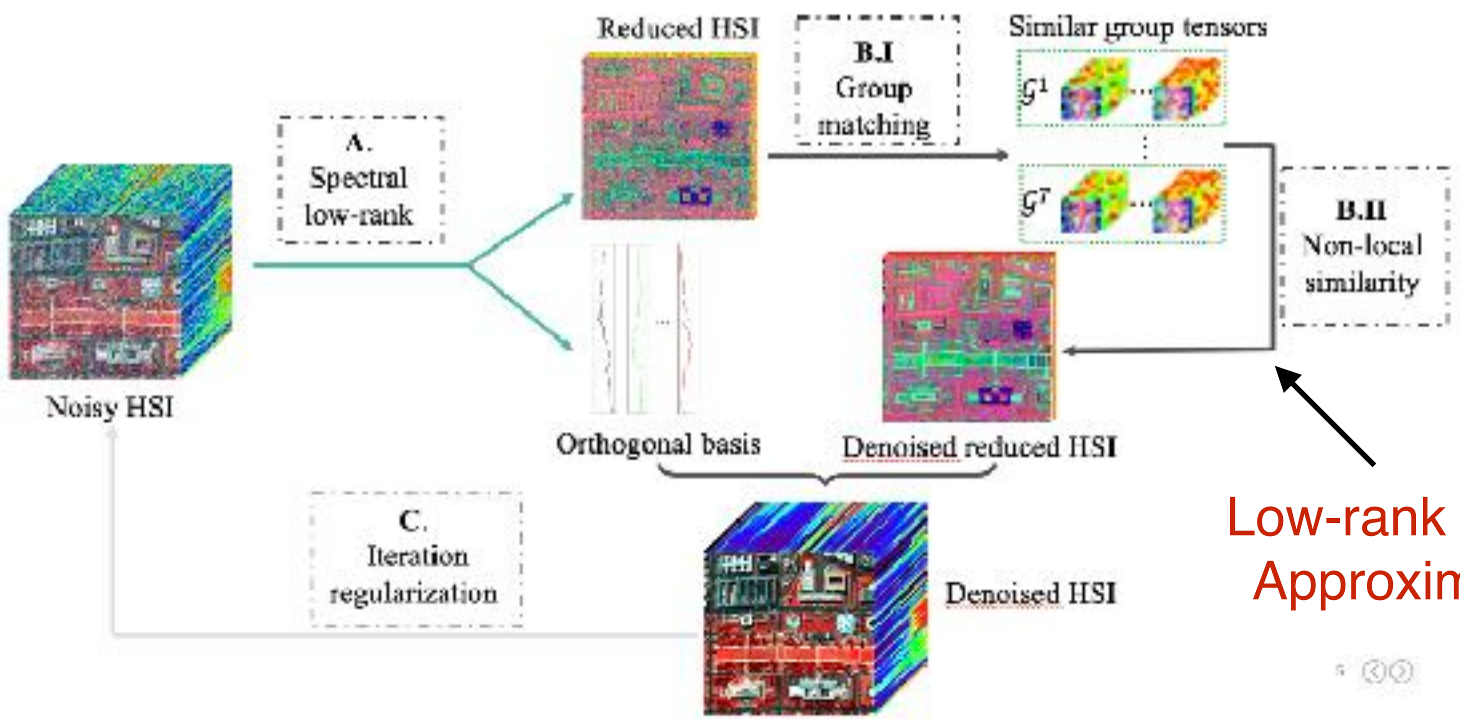
- Decomposition based approach (optimal rank selection)

$$R(\mathcal{X}) = \|\mathcal{X} - \mathrm{TN}(\mathcal{G}_1, \ldots, \mathcal{G}_N)\|$$

- Prior knowledge (smoothness, non-negative), side information

Low-rankness Under Multiple Transformation

- Image is not always globally low-rank (He et al., CVPR 2019)



Low-rank Tensor Approximation

- Non-uniform missing patterns (slice, fiber missing) (Li et al, CVPR 2019)

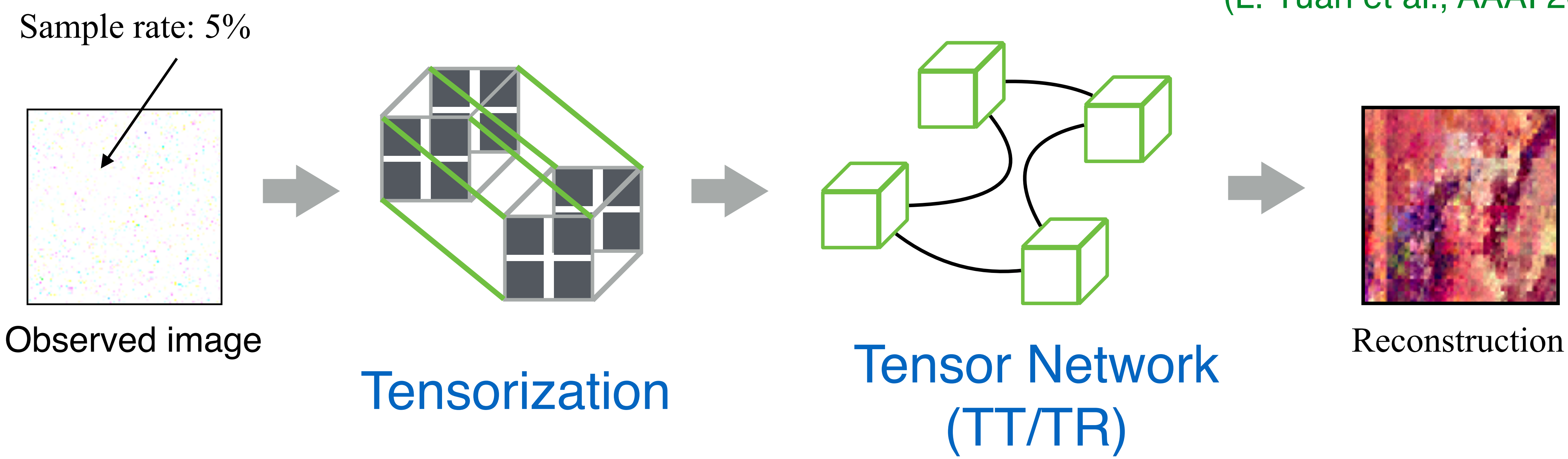
$$\min_{\mathbf{X}\in\mathbb{R}^{m_1\times m_2}} \|\mathcal{Q}(\mathbf{X})\|_* \quad s.t. \|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{Y})\|_F \leq \delta,$$

Linear transformation

Error bound is theoretically guaranteed

Tensor Networks with Low-rank Cores

(L. Yuan et al., AAAI 2019)



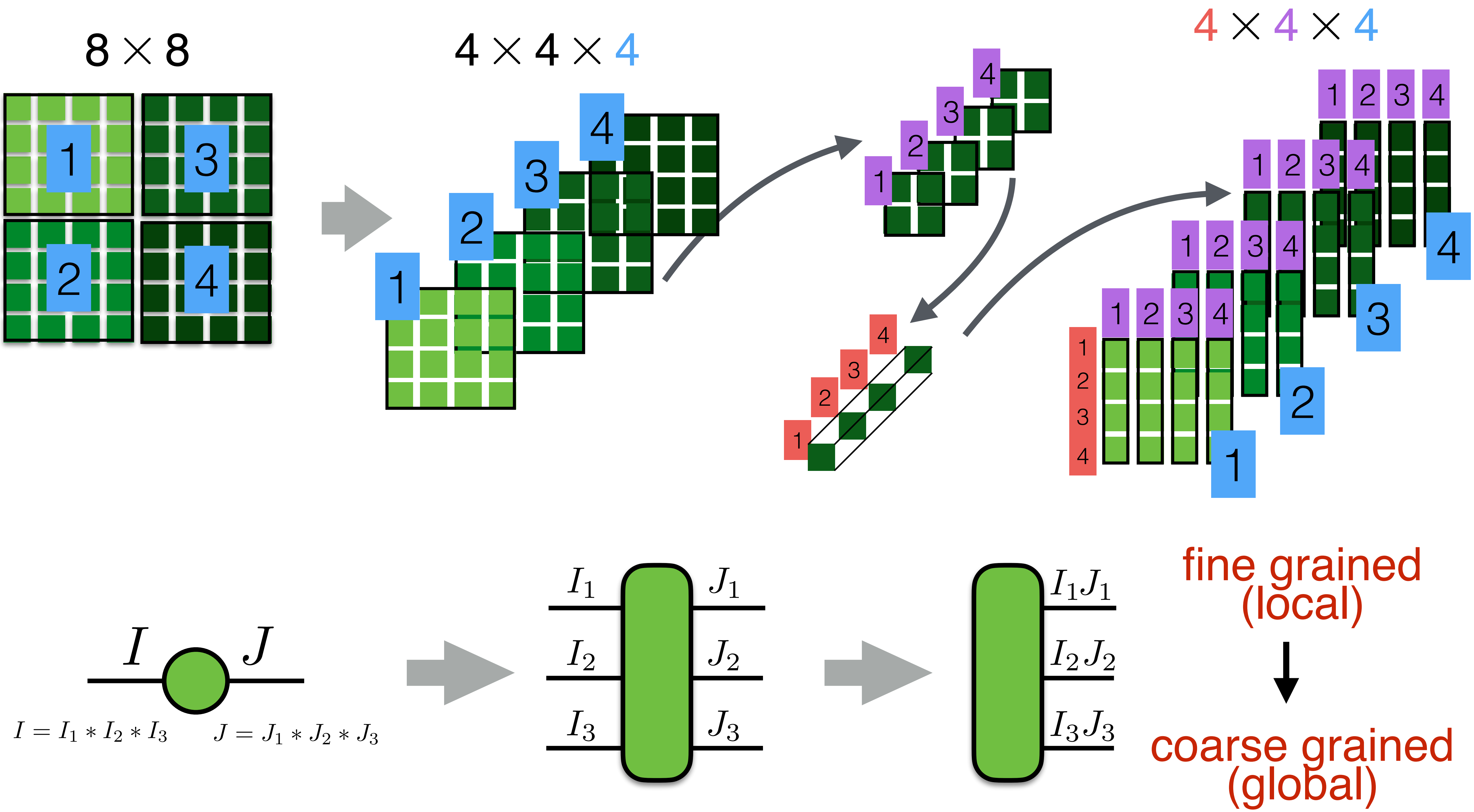
Fitting error | Nuclear norm on core tensor | TT/TR decomposition

$$\min_{\mathcal{G}} \quad \left\|\Omega * (\mathcal{Y} - \hat{\mathcal{Y}})\right\|_F^2 + \lambda \sum_{n=1}^{d} \sum_{i=1}^{3} \left\| \mathbf{G}_{(i)}^{(n)} \right\|_*, \quad s.t. \quad \hat{\mathcal{Y}} = \mathrm{TR}(\mathcal{G}^{(1)}, \cdots, \mathcal{G}^{(d)}).$$

- Tensorization preprocess allows for capturing complex structural dependency
- Efficient optimization by combining decomposition and nuclear norm regularization

"Tensor ring decomposition with rank minimization on latent space: An efficient approach for tensor completion "

Image Tensorization

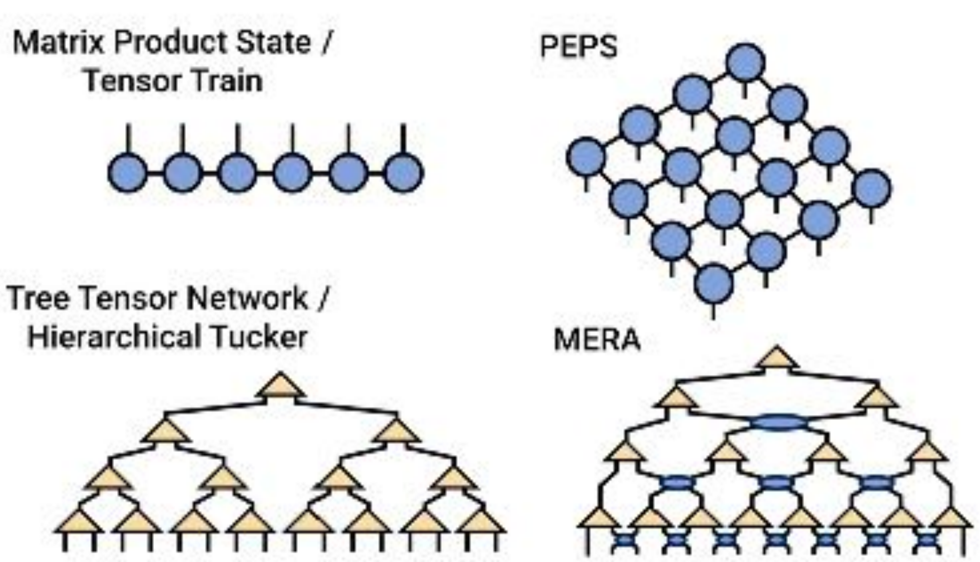


- Explore correlations of patches in multi-scales

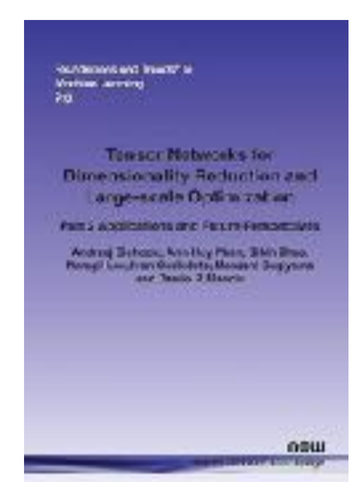
What Is Tensor Network?

2-order		3-order		*N*-order
Matrix Factorization	→	Tensor Factorization	→	Tensor Networks

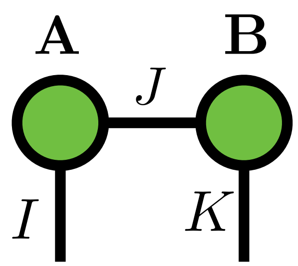
- Representation of N-order tensor as contractions of O(N) smaller tensors
- Physics: to describe entangled quantum many-body systems



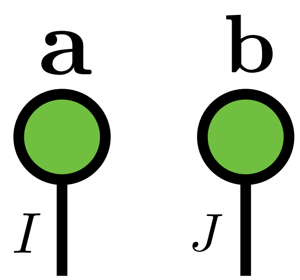
https://tensornetwork.org



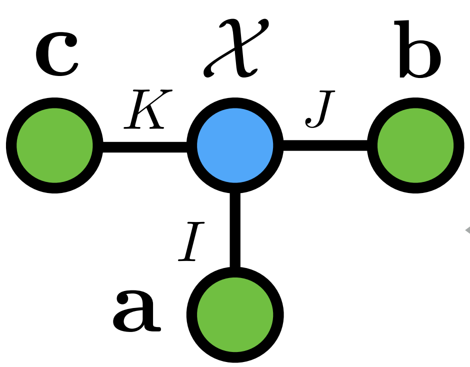
Tensor Network Operations



$\Longleftrightarrow$ $\mathbf{AB}$ $(I \times K)$ $\Longleftrightarrow$ $\displaystyle\sum_{j=1}^{J} A_{ij} B_{jk}$



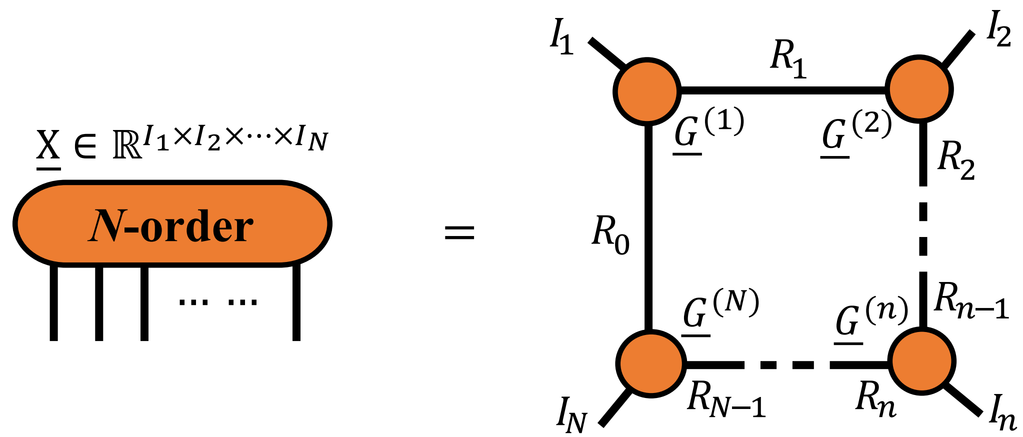
$\Longleftrightarrow$ $\mathbf{a} \otimes \mathbf{b}$ $(I \times J)$

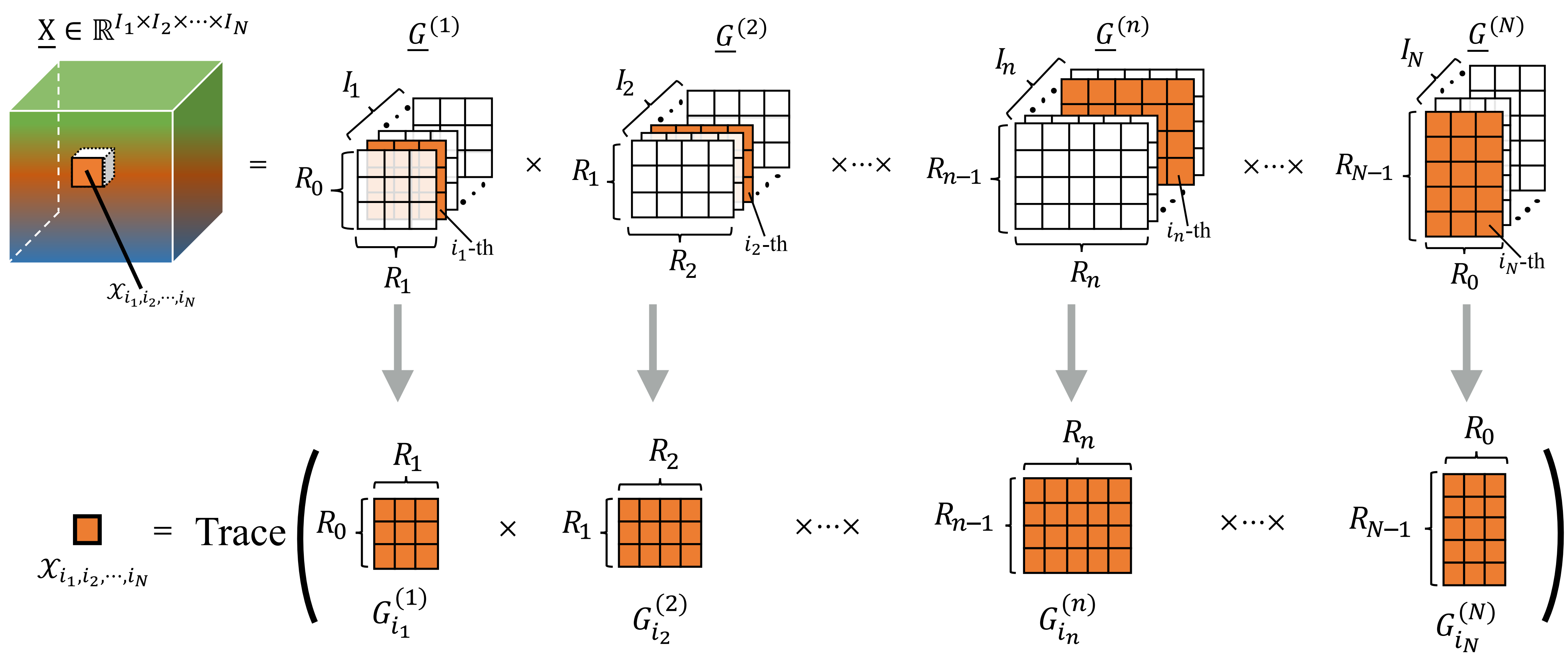


$\Longleftrightarrow$ $\mathcal{X} \times_1 \mathbf{a} \times_2 \mathbf{b} \times_3 \mathbf{c}$ $(1 \times 1 \times 1)$ $\Longleftrightarrow$ $\displaystyle\sum_{i,j,k} \mathcal{X}_{i,j,k} \mathbf{a}_i \mathbf{b}_j \mathbf{c}_k$

Tensor Ring Decomposition

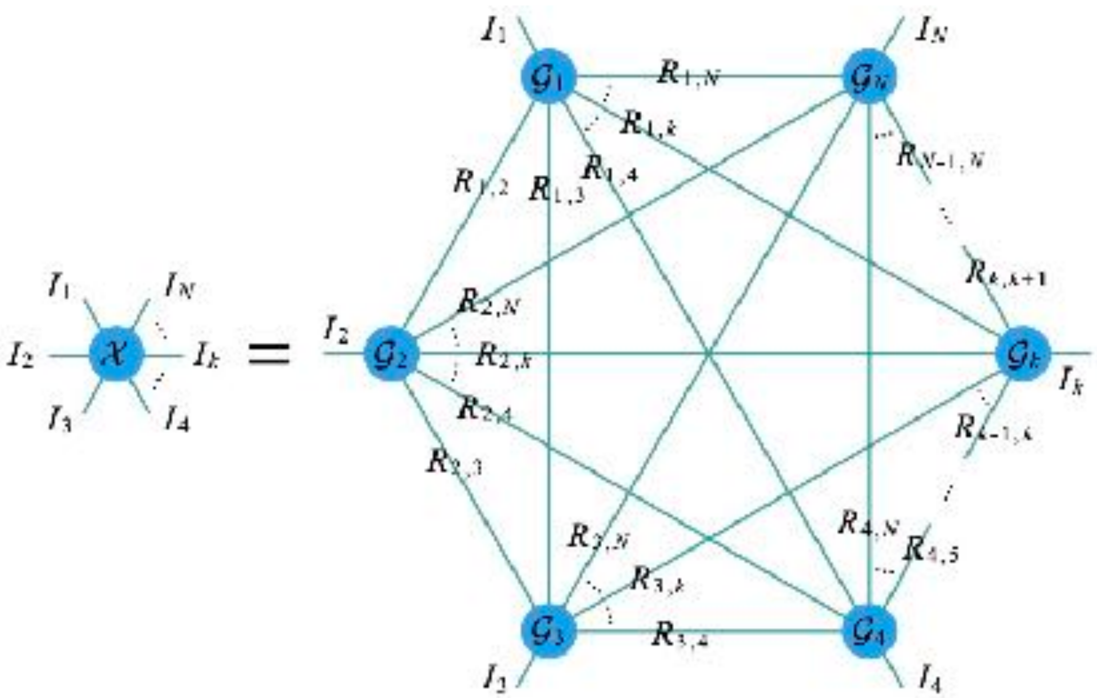
(Zhao et al., arXiv 2016, ICASSP 2019)





Fully Connected TN (FCTN)

(Zheng et al., AAAI 2021)



$$\mathcal{X}(i_1,i_2,\cdots,i_N)=\sum_{r_{1,2}=1}^{R_{1,2}}\sum_{r_{1,3}=1}^{R_{1,3}}\cdots\sum_{r_{1,N}=1}^{R_{1,N}}\sum_{r_{2,3}=1}^{R_{2,3}}\cdots\sum_{r_{2,N}=1}^{R_{2,N}}\cdots\sum_{r_{N-1,N}=1}^{R_{N-1,N}} \{\mathcal{G}_1(i_1,r_{1,2},r_{1,3},\cdots,r_{1,N}) \mathcal{G}_2(r_{1,2},i_2,r_{2,3},\cdots,r_{2,N})\cdots \mathcal{G}_k(r_{1,k},r_{2,k},\cdots,r_{k-1,k},i_k,r_{k,k+1},\cdots,r_{k,N})\cdots \mathcal{G}_N(r_{1,N},r_{2,N},\cdots,r_{N-1,N},i_N)\}.$$

Transpositional Invariance

- Number of Parameters

$$\begin{aligned}\text{CPD:}\quad & \mathcal{O}(NIR)\\ \text{Tucker:}\quad & \mathcal{O}(NIR+R^N)\\ \text{TT/TR:}\quad & \mathcal{O}(NIR^2)\\ \boxed{\text{FCTN:}\quad \mathcal{O}(NIR^{N-1})}\end{aligned}$$

- Tensor Network Ranks

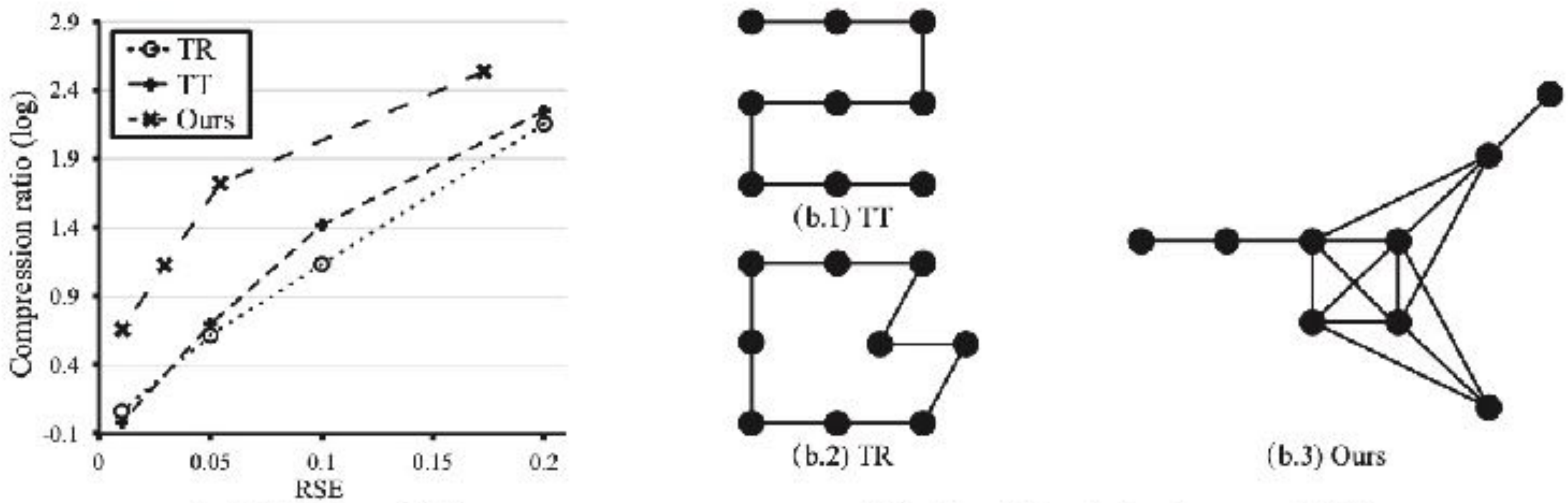
Comparison:

▷ TT-rank: $\mathrm{Rank}(\mathbf{X}_{[1:d;d+1:N]}) \leq R_d$;

▷ TR-rank: $\mathrm{Rank}(\mathbf{X}_{[1:d;d+1:N]}) \leq R_d R_N$;

▷ FCTN-ank: $\mathrm{Rank}(\mathbf{X}_{[1:d;d+1:N]}) \leq \prod_{i=1}^{d}\prod_{j=d+1}^{N} R_{i,j}$.

Learning Tensor Network Structure



(a) RSE *vs.* CR

(b) Graphical strctures of TN

Standard models may not be the most compressive one

- Can we learn an optimal TN structure from data?
- Difficulty: given an 9-order tensor, there are more than 68 BILLION candidates

Optimization of TN topology

(Li et al., ICML 2020)

- The TN structures can be fully described by its adjacency matrice

$$\mathcal{X} = TN(\mathbb{V}; \mathbf{A})$$

Collection of core tensors

Adjacency matrix

Adjacency Matrix

$$\begin{pmatrix} 0 & 2 & 0 & 0 \\ 2 & 0 & 2 & 0 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix}$$

Order-4 tensor with TT-format

- Find an optimal A such that

$$\min_{\mathbf{A} \in \mathbb{A}} \frac{1}{\epsilon(\mathbf{A})}, \quad s.t. \, \exists \widehat{\mathbb{V}} \text{ which satisfies } \left\| \mathcal{X} - TN(\widehat{\mathbb{V}}; \mathbf{A}) \right\|_F^2 \leq \delta,$$

$$\epsilon(\mathbf{A}) = \frac{\text{Uncompressed size of } \mathcal{X}}{\text{Parameter size of } \mathbb{V} \text{ under } \mathbf{A}}.$$

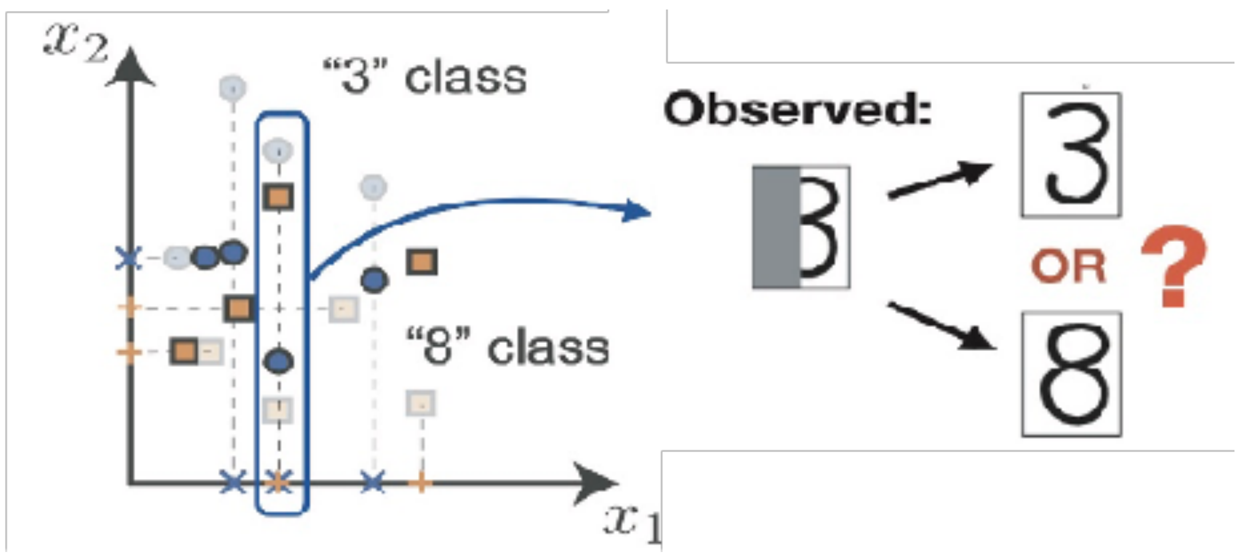
- Learn (near-)optimal TN topology via evolutionary algorithm (EA).

Classification of incomplete data

Problem: learning classification model from incomplete data $(x_n^{miss}, y_n), n = 1, \ldots, N$

Objective: $\hat{f}(\boxed{g(x^{miss})}, \hat{\theta}) \approx f(x, \theta)$

Reconstruction of incomplete data



Sequential approach (completion + classification), but cannot ensure statistical consistence of classifier

- Exact recovery is not guaranteed
- Label information is ignored for reconstruction

Simultaneous reconstruction and classification

(Caiafa et al., CVPR workshop 2021)

- Learning sparse representation and classifier collaboratively (NNs + sparse coding)

Data model: $\hat{\mathbf{x}}_i = \mathbf{D}\mathbf{s}_i$ → Sparse vector $\|\mathbf{s}_i\|_0 \leq K$

Dictionary matrix

Optimization problem: Minimize over the classifier's parameters and data representation

$$J(\Theta, \mathbf{D}, \mathbf{s}_i) = \frac{1}{I}\sum_{i=1}^{I}\{J_0(\Theta, \hat{\mathbf{x}}_i, y_i) + \lambda_1 J_1(\mathbf{D}, \mathbf{s}_i) + \lambda_2 J_2(\mathbf{s}_i)\}$$

Classification loss (e.g. crossentropy) for any classifier (deep network)

Representation error

$$J_1(\mathbf{D}, \mathbf{s}_i) = \frac{M}{N}\|\mathbf{m}_i * (\mathbf{x}_i - \mathbf{D}\mathbf{s}_i)\|^2.$$

Promotes sparsity

$$J_2(\mathbf{s}_i) = \frac{1}{N}\|\mathbf{s}_i\|_1$$

Alternated minimization training algorithm: alternate between $\{\Theta, \mathbf{D}\}$ and $\mathbf{s}_i$

Sufficient condition

(Caiafa et al., CVPR workshop 2021)

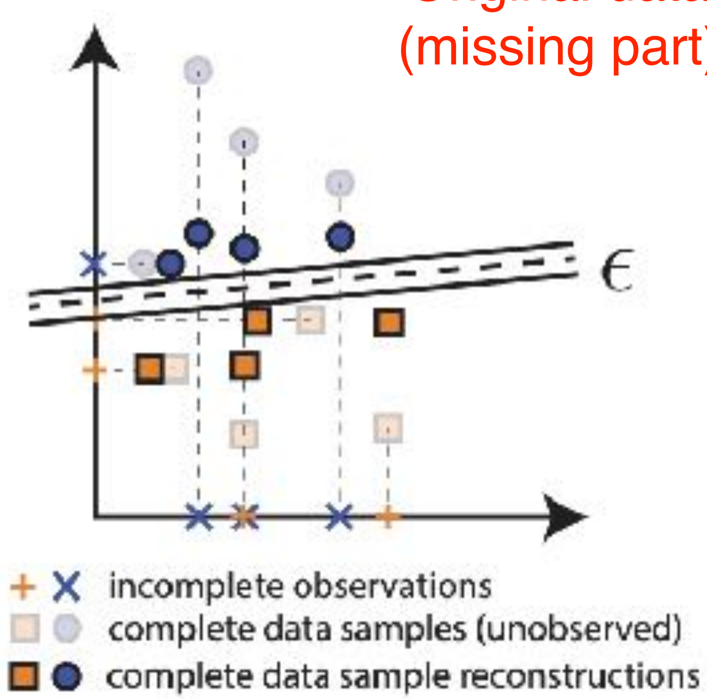
- If the reconstructed data points are well separated by a hyperplane, then the same classifier also correctly separates the original (unobserved) data points.

Weights of classifier

$$\epsilon > |\langle \mathbf{w}^m, \mathbf{x}^m \rangle| + |\langle \mathbf{w}^m, \hat{\mathbf{x}}^m \rangle|$$

Original data (missing part)

Reconstructed data (Missing part)



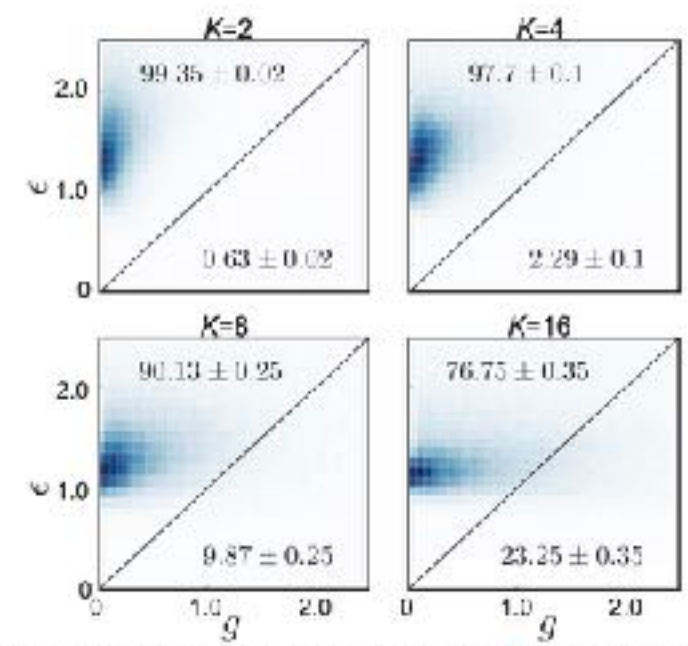


Figure 3. Verification of the sufficient condition (6) for various levels of sparsity K: 2D-histogram of ϵ versus $g = |\langle \mathbf{w}^m, \mathbf{x}^m \rangle| + |\langle \mathbf{w}^m, \hat{\mathbf{x}}^m \rangle|$. Mean + s.e.m ($n = 10$) percentage of correctly classified data samples are shown for $\epsilon > g$ and $\epsilon < g$.

Results

MNIST (CNN4)								
Miss.	ZF	MS	KNN10	KNN20	KNN50	KNN100	NN-GMM	Simult.
75%	84.86 ± 0.02	83.79 ± 0.01	88.16 ± 0.01	87.94 ± 0.01	87.03 ± 0.002	86.52 ± 0.01	96.36 ± 0.12	$\mathbf{98.09 \pm 0.04}$
50%	90.13 ± 0.06	88.55 ± 0.01	91.36 ± 0.02	91.11 ± 0.02	90.87 ± 0.01	90.82 ± 0.01	97.57 ± 0.37	$\mathbf{98.23 \pm 0.10}$
CIFAR10 (Resnet18)								
Miss.	ZF	MS	KNN10	KNN20	KNN50	KNN100	NN-GMM	Simult.
75%	32.22 ± 2.09	21.30 ± 0.40	22.84 ± 0.87	25.67 ± 0.80	26.52 ± 0.70	26.01 ± 0.52	12.10 ± 0.61	$\mathbf{54.81 \pm 0.47}$
50%	46.37 ± 1.93	17.90 ± 0.94	30.94 ± 0.54	29.68 ± 0.46	30.01 ± 0.51	26.23 ± 1.01	14.02 ± 0.75	$\mathbf{62.50 \pm 0.95}$

Reconstruction
of test examples

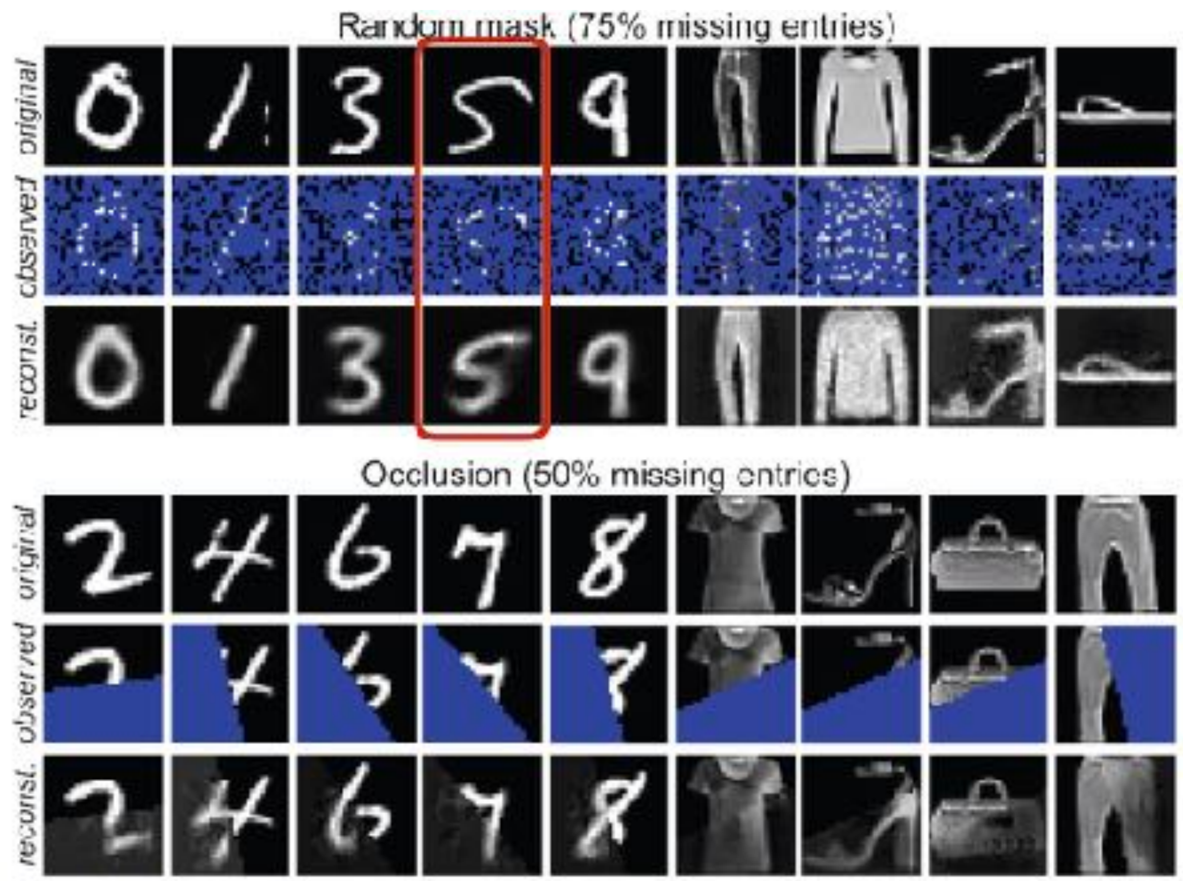


Figure 4. Original (top), observed (middle) and reconstructed (bottom) MNIST and Fashion test images.

Latent factor analysis of limited data

Problem: Learning correlations and hidden patterns of higher-order data require large sample sizes, which may be unavailable.

Given $\boldsymbol{y} \in \mathbb{R}^P$, suppose it has $K \ll P$ common factors,

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{\eta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}), \tag{1}$$

where $\boldsymbol{W} \in \mathbb{R}^{P\times K}$, $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ are latent factors and $\boldsymbol{\Sigma}$ is diagonal.

Marginalize $\boldsymbol{\eta}$, then we have $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{V})$,

$$\boxed{\boldsymbol{V} = \underbrace{\boldsymbol{W}\boldsymbol{W}^\intercal}_{\text{low-rank}} + \underbrace{\boldsymbol{\Sigma}}_{\text{noise}} .}$$

Higher-order latent factor analysis

(Tao et al., ACML 2021)

- Given higher-order data $\mathcal{Y} \in \mathbb{R}^{P_1 \times \cdots \times P_D}$, marginalize η gives $\mathcal{Y} \sim \mathcal{N}(\mathbf{0}, \mathcal{V})$

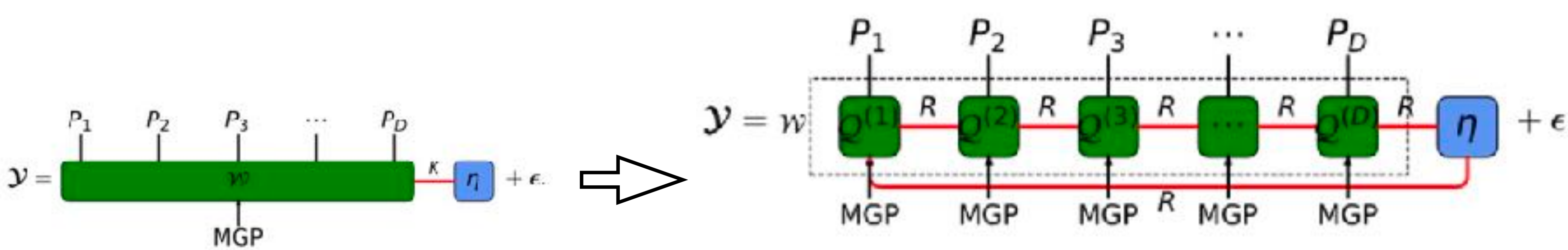
Covariance of vectors: $\boldsymbol{V}_{ij} = \mathrm{cov}(\boldsymbol{y}_i, \boldsymbol{y}_j)$.

Covariance of tensors: $\mathcal{V}_{i_1 i_2 i_3 j_1 j_2 j_3} = \mathrm{cov}(\mathcal{Y}_{i_1 i_2 i_3}, \mathcal{Y}_{j_1 j_2 j_3})$.

Core tensors

$$\mathcal{V}_{p_1 \cdots p_D p'_1 \cdots p'_D} =$$
$$\underbrace{\mathrm{tr}(\boldsymbol{Q}^{(1)}[p_1] \cdots \boldsymbol{Q}^{(D)}[p_D](\boldsymbol{Q}^{(D)}[p'_D])^{\intercal} \cdots (\boldsymbol{Q}^{(1)}[p'_1])^{\intercal})}_{\text{low-rank TR}} + \underbrace{\tau^{-1}}_{\text{noise}},$$

- TN representation of parameter W

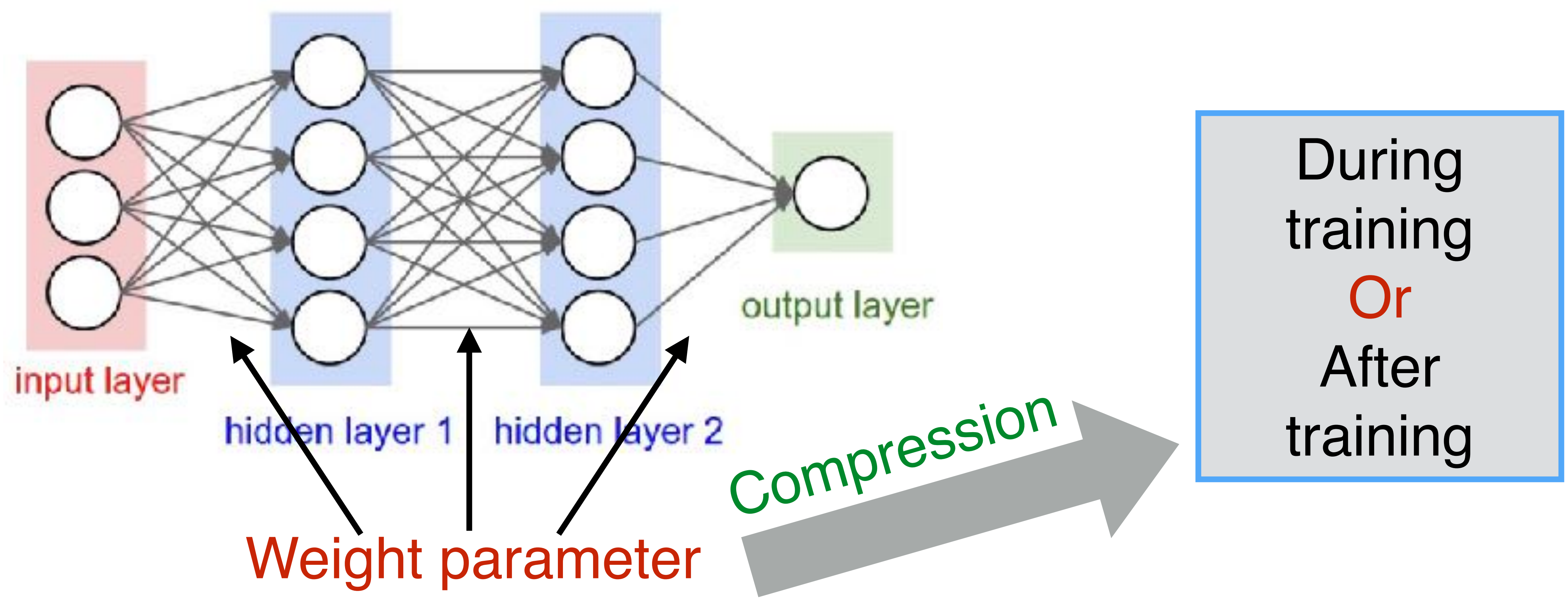


$$\mathcal{Y} = \ll \mathcal{Q}^{(1)}, \ldots, \mathcal{Q}^{(D)}, \boldsymbol{\eta} \gg + \mathcal{E},$$

Tensor Networks for Efficient Modeling

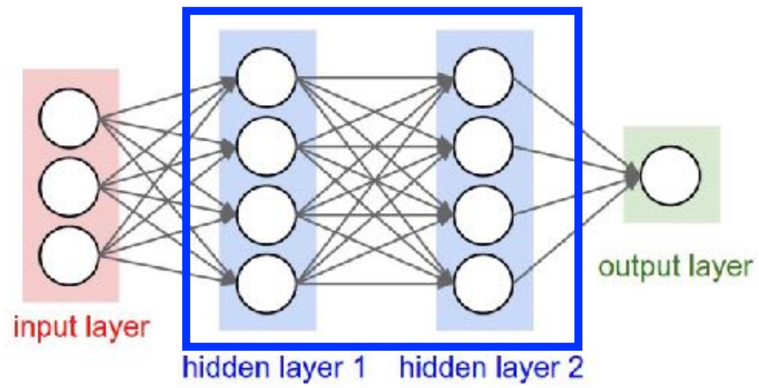
Model Compression

Goal: Make a lightweight model that is fast, memory-efficient and energy-efficient



- Reduce number of parameters but keeping comparable performance
- Compatible with SGD based optimization algorithm
- Computation efficiency

Reparametrization via TN



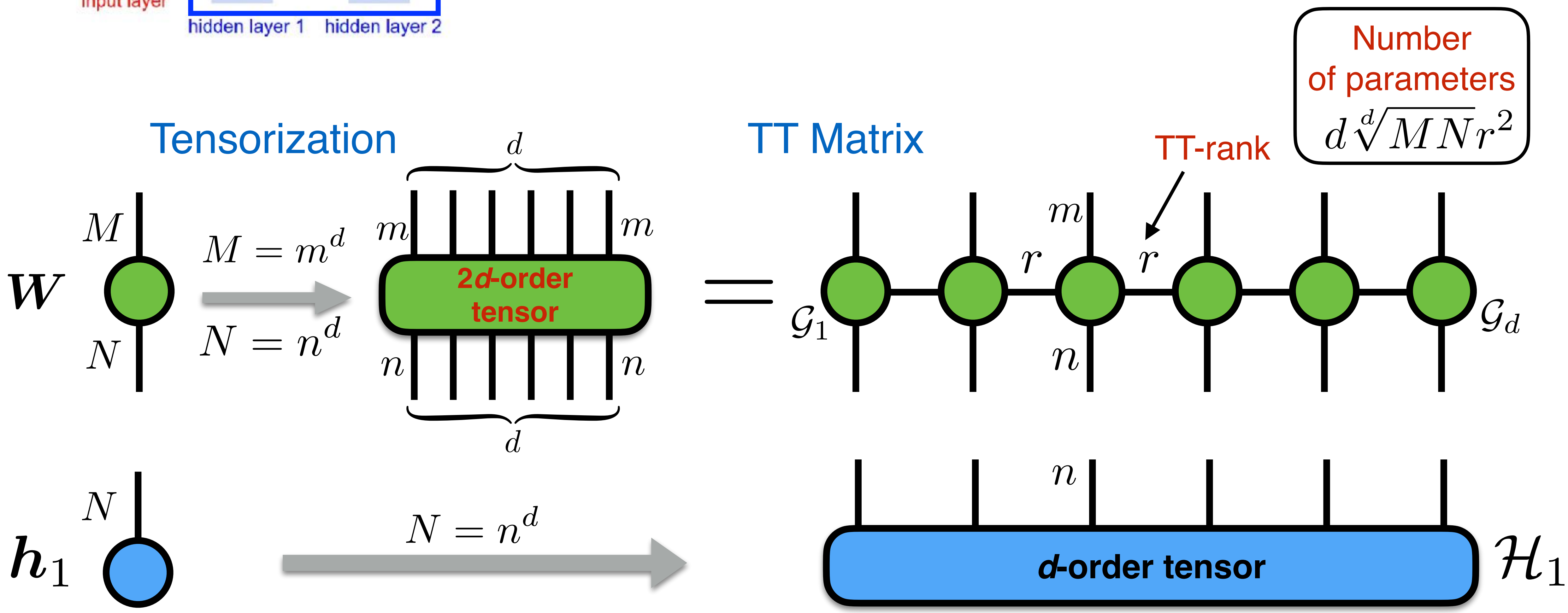
Hidden layer

$$\boldsymbol{h}_2 = \sigma(\boldsymbol{W}^T \boldsymbol{h}_1 + \boldsymbol{b})$$

Hidden layer

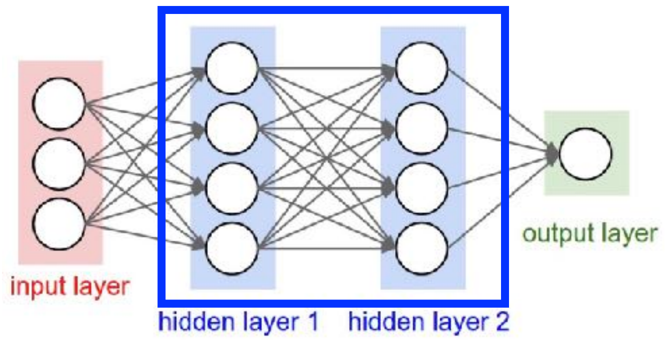
$(N \times M)$

Weights



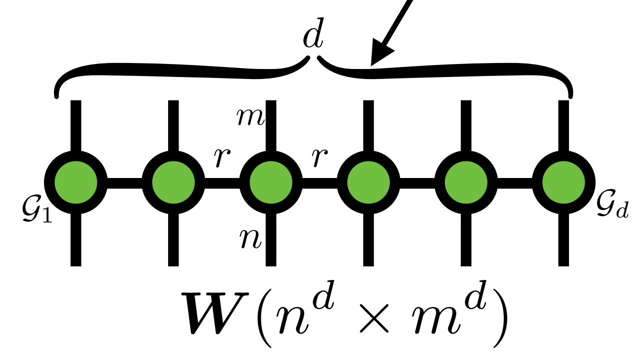
$$\boldsymbol{W}^T \boldsymbol{h}_1 = \mathrm{TT}(\mathcal{G}_1, \cdots, \mathcal{G}_d) \times_{1,\ldots,d} \mathcal{H}_1$$

Learning of TN Weights

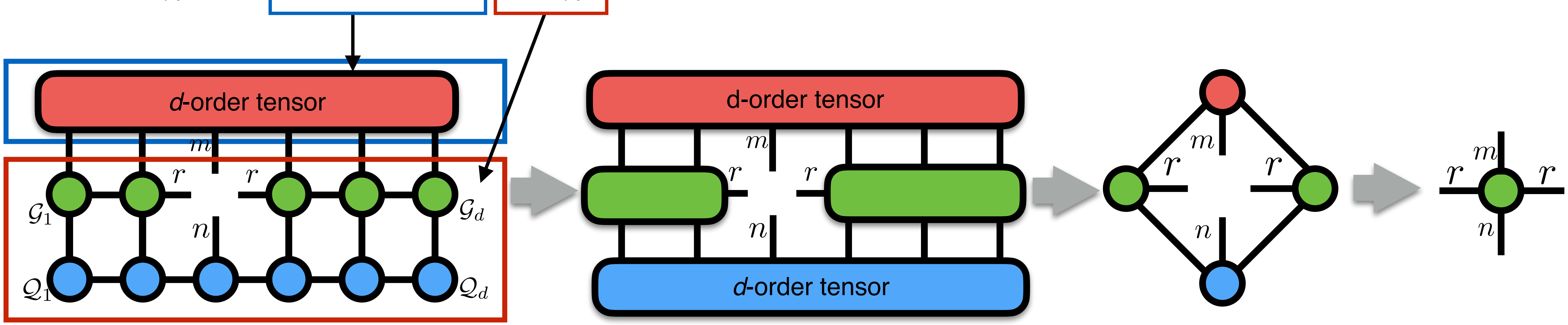


$$\boldsymbol{h}_2 = \sigma(\underbrace{\boxed{\boldsymbol{W}^T\boldsymbol{h}_1 + \boldsymbol{b}}}_{\boldsymbol{o}_2})$$

- Loss: $\min_{\boldsymbol{W}} \mathcal{L}(\boxed{\boldsymbol{W}}, \boldsymbol{x}, y) \Rightarrow \min_{\mathcal{G}_1,\cdots,\mathcal{G}_d} \mathcal{L}(\boxed{\{\mathcal{G}_1, \cdots, \mathcal{G}_d\}}, \boldsymbol{x}, y)$
- Gradients over TN core tensors

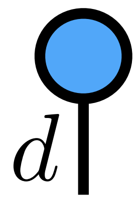


$$\frac{\partial \mathcal{L}}{\partial \mathcal{G}_k} = \boxed{\frac{\partial \mathcal{L}}{\partial \boldsymbol{h}_2}\frac{\partial \boldsymbol{h}_2}{\partial \boldsymbol{o}_2}}\boxed{\frac{\partial \boldsymbol{o}_2}{\partial \mathcal{G}_k}}$$



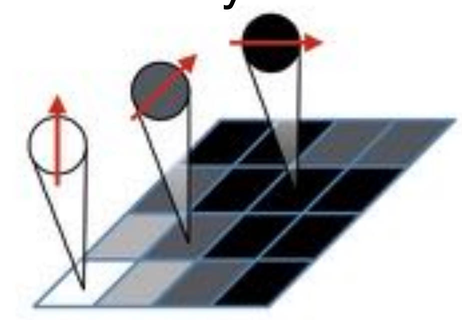
TN representation of inputs

- Mapping input data into TN representation



$$\mathbf{x} = [x_1, x_2, \ldots, x_d]^T$$

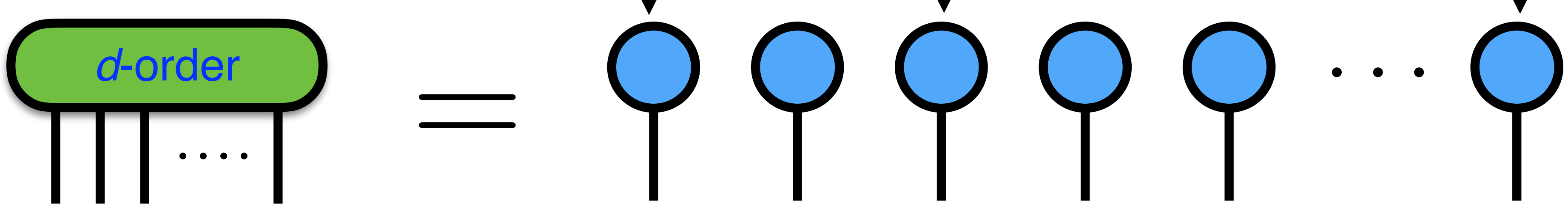
Inspired by "spin" vectors in quantum system



$$\phi(x_i) = \left[\cos\left(\frac{\pi}{2}x_i\right), \sin\left(\frac{\pi}{2}x_i\right)\right]^T$$

$$d \mapsto 2^d \qquad \boxed{\Phi(\mathbf{x})} = \boxed{\phi(x_1)} \otimes \boxed{\phi(x_2)} \otimes \cdots \otimes \boxed{\phi(x_d)}$$

Local feature map



$(2 \times 2 \times \cdots \times 2)$

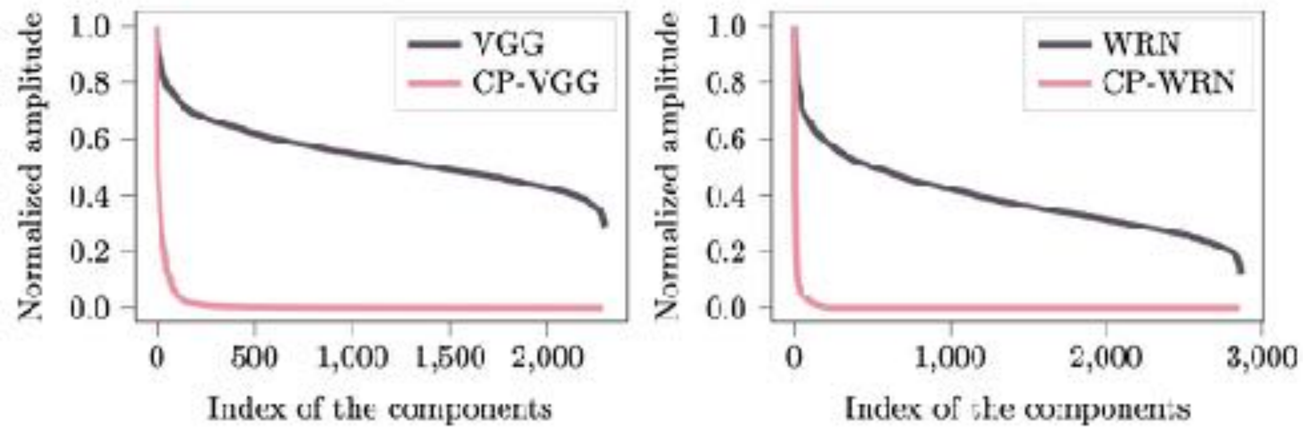
Rank-1 Tensor

- Accuracy of 99.03% on MNIST by one layer

Supervised Learning with Quantum-Inspired Tensor Networks (Stoudenmire et al., NIPS 2016)

Generalization of Compressed CNN

- Weight matrices/kernels of well-trained models are not necessarily low-rank



- Re-parametrizes weight tensors as CPD reduces the generalization error

$$\mathcal{K} = \sum_{r=1}^{R} \lambda_r \mathbf{v}_r^{(1)} \otimes \cdots \otimes \mathbf{v}_r^{(N)}$$

- Higher compression -> smaller generalization error bound

Generalization Error

Empirical Loss

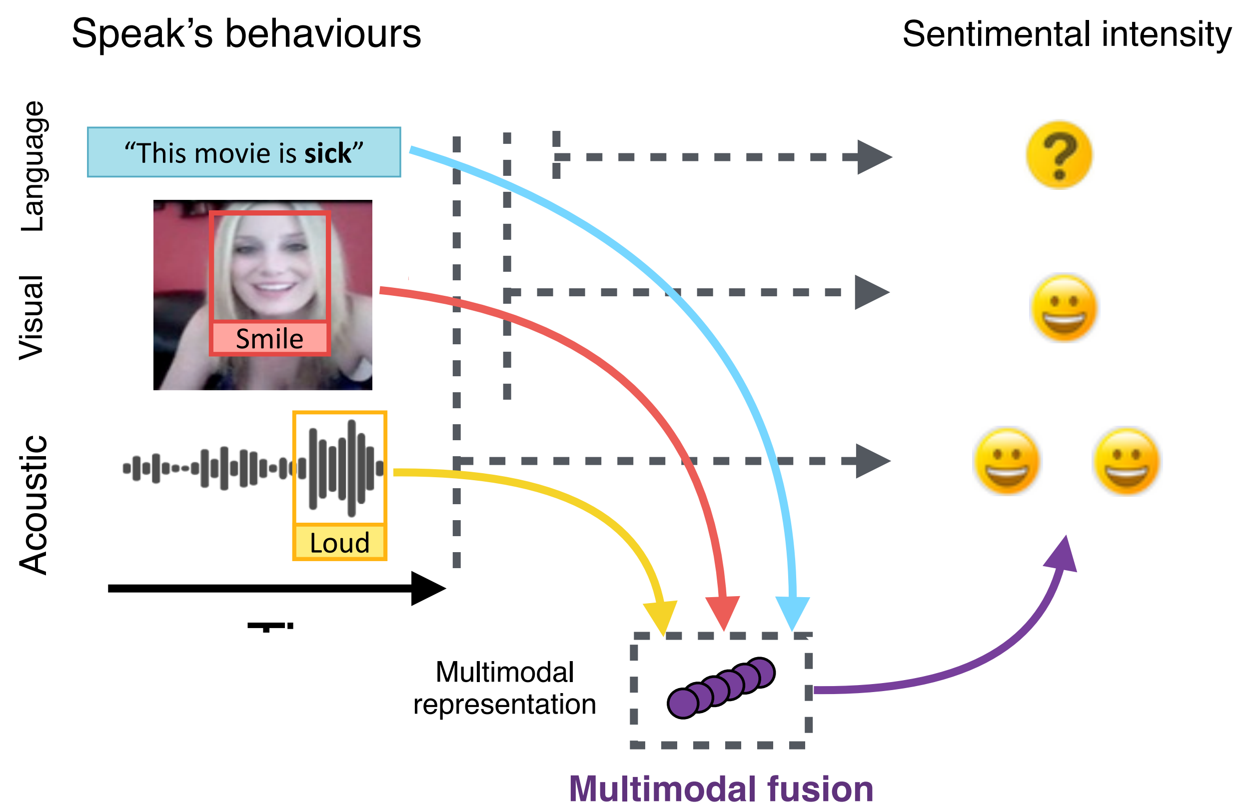
CP Rank

$$L_0(\hat{\mathbb{M}}) \leq \hat{L}_\gamma(\mathbb{M}) + \tilde{O}\left(\sqrt{\frac{\sum_{k=1}^{n} \hat{R}^{(k)}(s^{(k)} + o^{(k)} + k_x^{(k)} k_y^{(k)} + 1)}{m}}\right)$$

Understanding Generalization in Deep Learning via Tensor Methods (Li et al., AISTATS 2020)

Multimodal Learning

- Multimodal sentimental classification (Acoustic, Visual, Language)

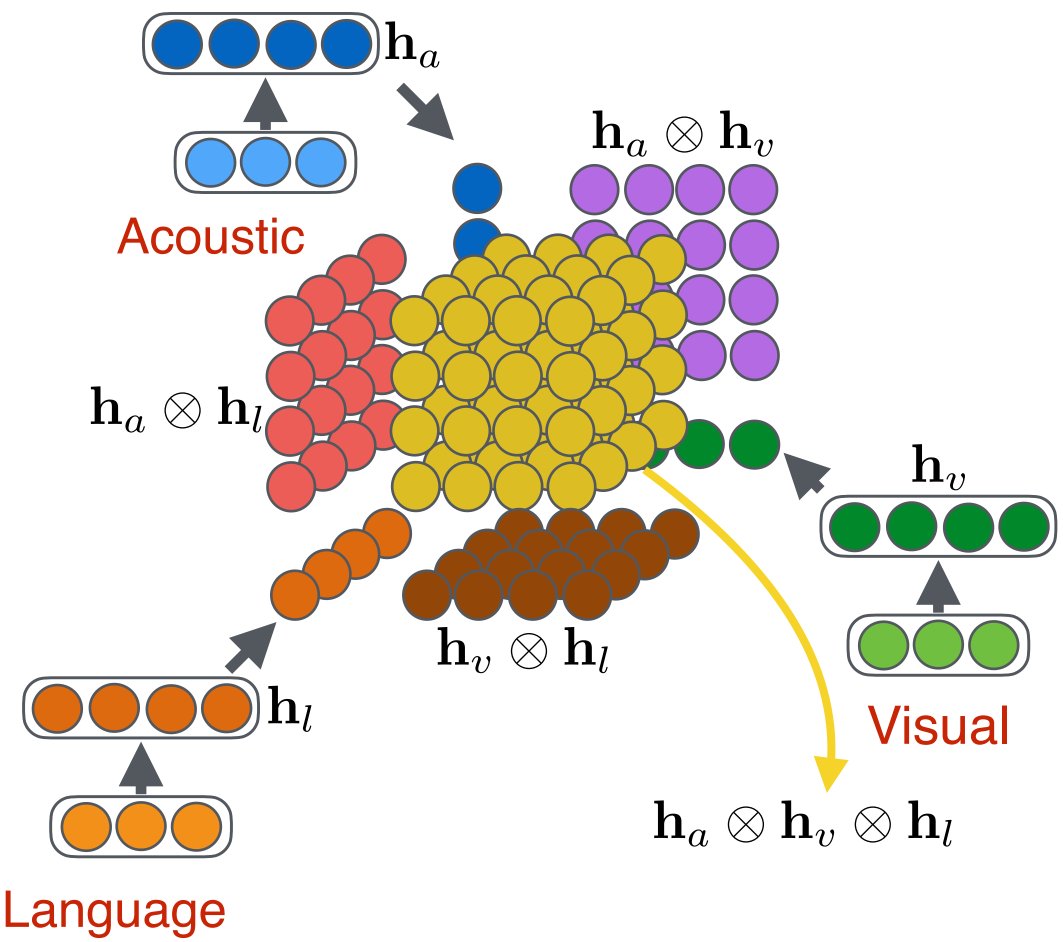


- Visual question answering (Image + Language)

Q : “What do you see?” (Ground Truth : a_3)
a_1 : “A courtyard with flowers”
a_2 : “A restaurant kitchen”
a_3 : “A family with a stroller, tables for dining”
a_4 : “People waiting on a train”

Tensor Fusion Network for Multimodal Learning

- Trilinear fusion: linear, bilinear and trilinear interactions



$$\mathbf{z}^m = \begin{bmatrix} \mathbf{z}^l \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \mathbf{z}^v \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \mathbf{z}^a \\ 1 \end{bmatrix}$$

Baseline	Binary		5-class	Regression	
	Acc(%)	F1	Acc(%)	MAE	r
TFN$_{language}$	74.8	75.6	38.5	0.99	0.61
TFN$_{visual}$	66.8	70.4	30.4	1.13	0.48
TFN$_{acoustic}$	65.1	67.3	27.5	1.23	0.36
TFN$_{bimodal}$	75.2	76.0	39.6	0.92	0.65
TFN$_{trimodal}$	74.5	75.0	38.9	0.93	0.65
TFN$_{notrimodal}$	75.3	76.2	39.7	0.919	0.66
TFN	**77.1**	**77.9**	**42.0**	**0.87**	**0.70**
TFN$_{early}$	75.2	76.2	39.0	0.96	0.63

- Exponential increase of dimensionality and complexity

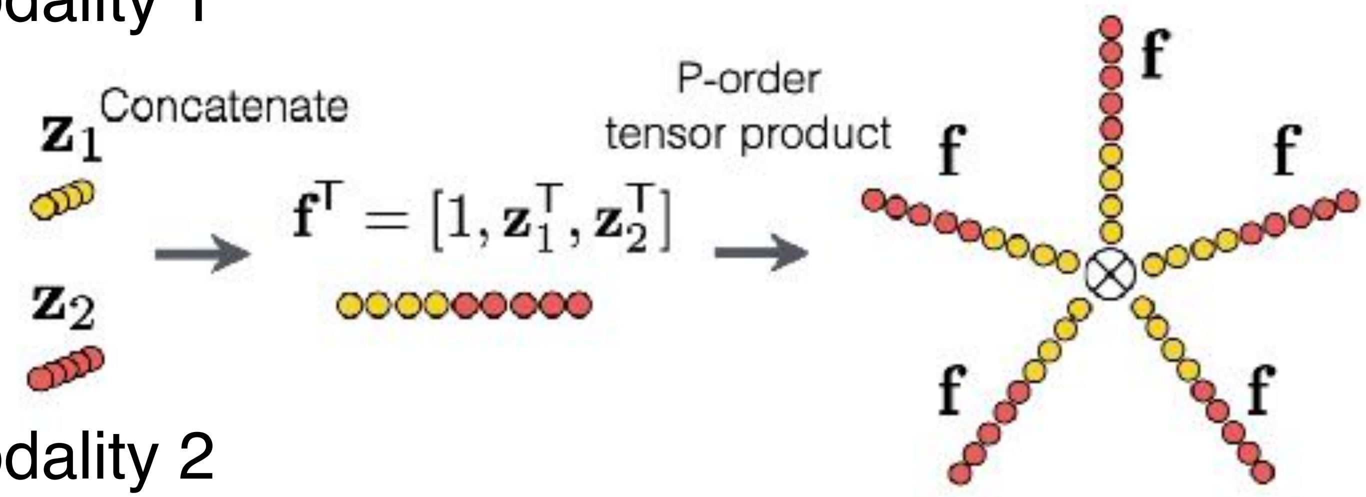
Tensor Fusion Network for Multimodal Sentiment Analysis (Zadeh et al., EMNLP 2017)

High-order Tensor Fusion

(Hou et al., NeurIPS 2019)

- Expressive power of tensor fusion is limited
- High-oder intra-modal and cross-modal feature interactions
- Tensor Polynomial Pooling (PTP)

Modality 1



Modality 2

Feature Interactions

- Linear
- Bilinear
- Trilinear
- Intra-modal
- High-order

$$\mathcal{F} = \underbrace{\mathbf{f} \otimes \mathbf{f} \otimes \cdots \otimes \mathbf{f}}_{\text{P-order}}$$

Example: $\mathcal{F}_{1,1,1} = f_1^3,\ \mathcal{F}_{1,2,1} = f_1^2 f_2,$
$\mathcal{F}_{1,2,3} = f_1 f_2 f_3,\ \mathcal{F}_{1,2,2} = f_1 f_2^2$

Dimensionality increases exponentially with *P*

Order

$$(md)^P$$

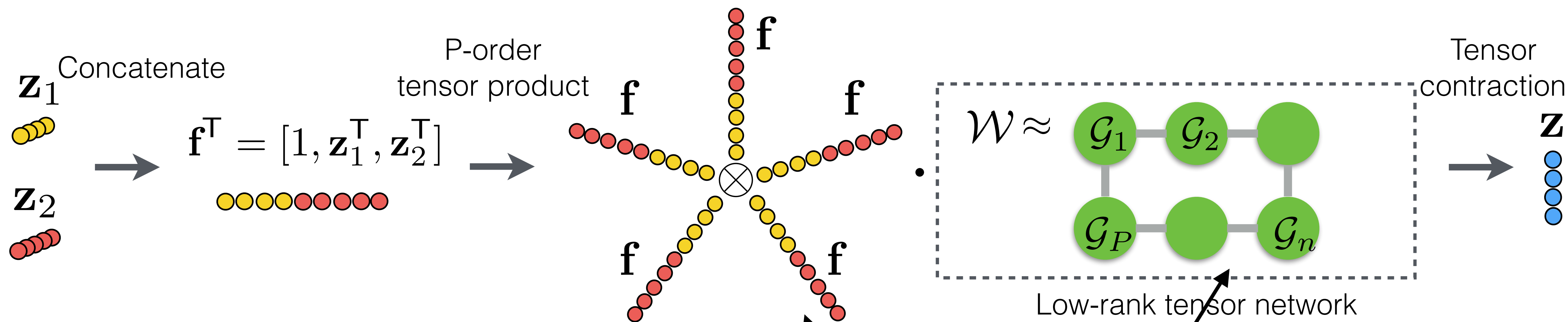
Number of modality

Dimension of feature

Deep Multimodal Multilinear Fusion with High-order Polynomial Pooling, NeurIPS 2019

Tensor Polynomial Pooling (PTP)

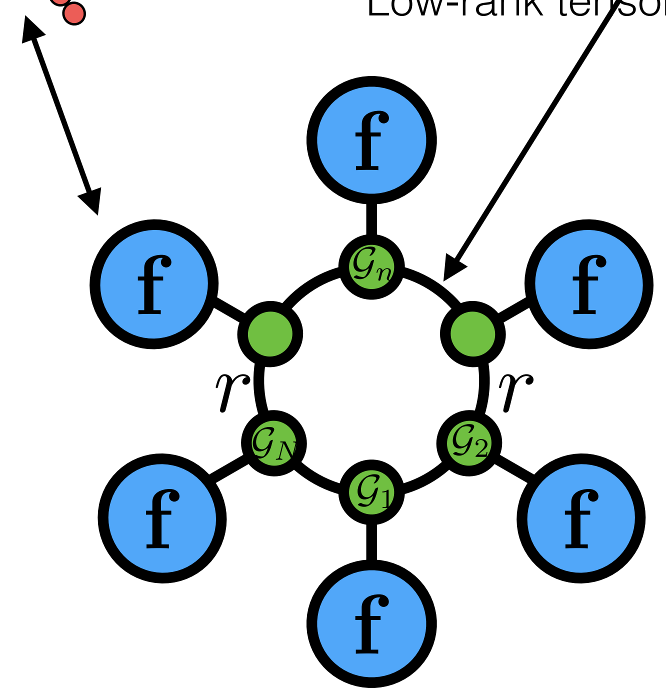
(Hou et al., NeurIPS 2019)



Number of Parameters

$$\mathcal{O}(md)^P$$

$$\mathcal{O}(mdr^2P)$$



- Highly enhanced expressive without much increasing number of parameters

Imperfect Multimodal Time Series Data

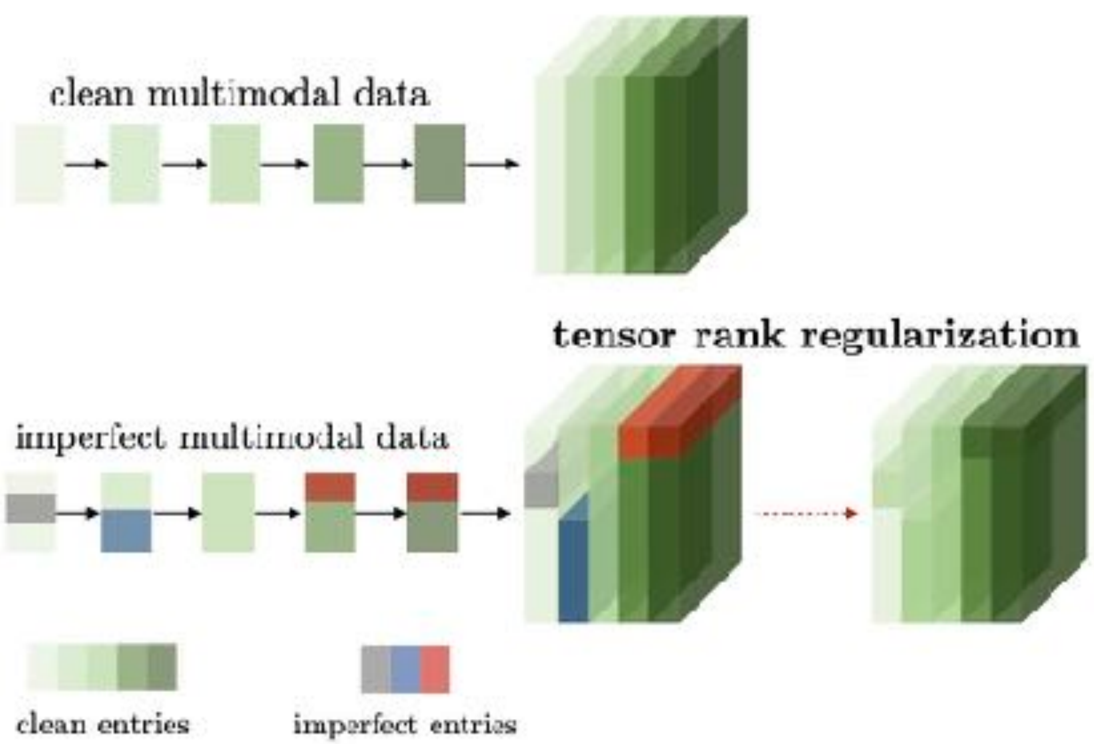
(Liang et al. ACL 2019)

Imperfect data:

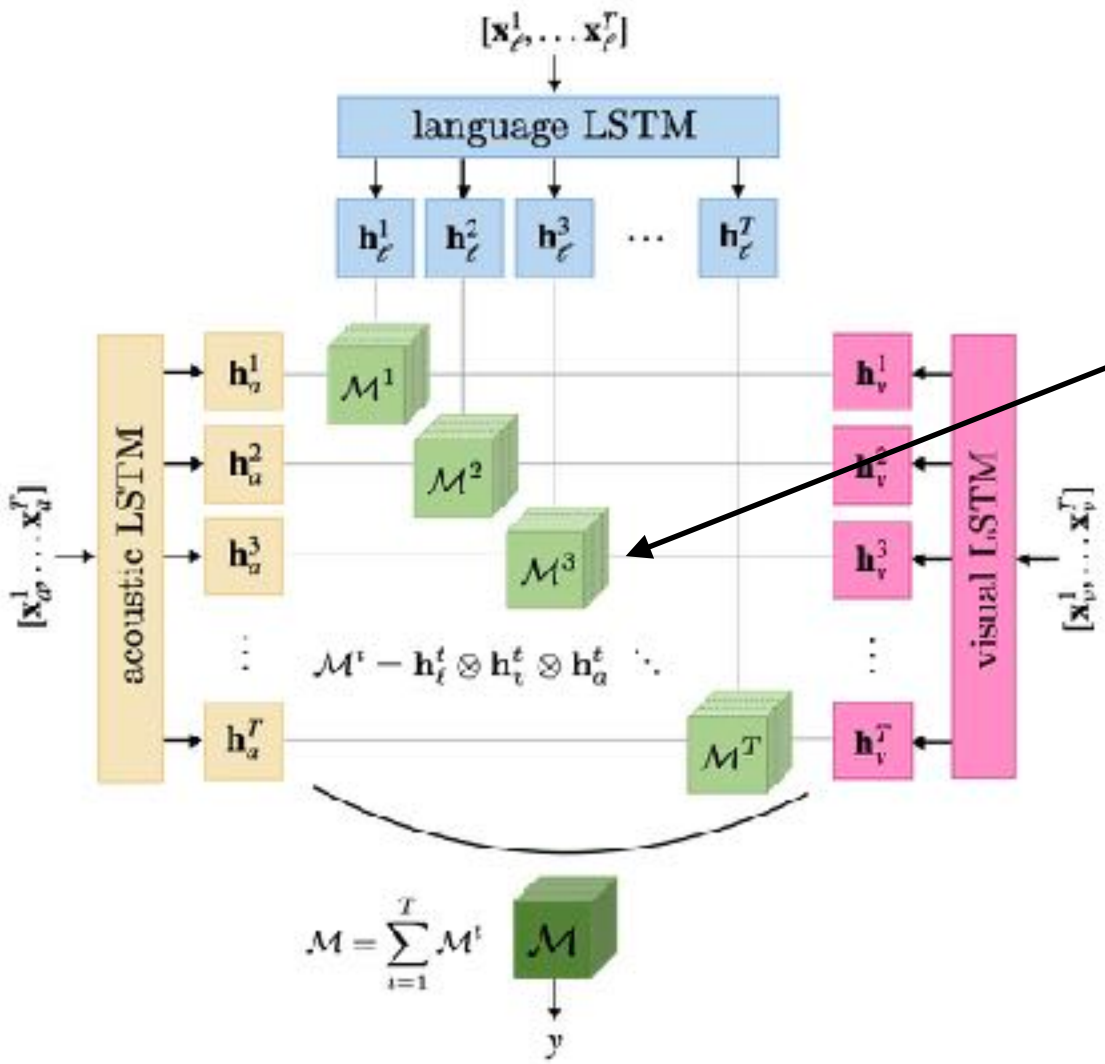
- Incomplete due to sensor failure
- Corrupted by random or structured noises

How to learn robust representation from imperfect multimodal data?

- Clean data: multimodal fused tensor exhibits low-rankness across time and modality
- Noisy and incomplete data breaks low-rank structure



Temporal Tensor Fusion Network (T2FN)



$$\mathcal{M} = \sum_{t=1}^{T} \boxed{\begin{bmatrix} \mathbf{h}_\ell^t \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \mathbf{h}_v^t \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \mathbf{h}_a^t \\ 1 \end{bmatrix}}$$

Tensor fusion (Rank-1 tensor)

Low-rank regularizer

Upper bounds on nuclear norm

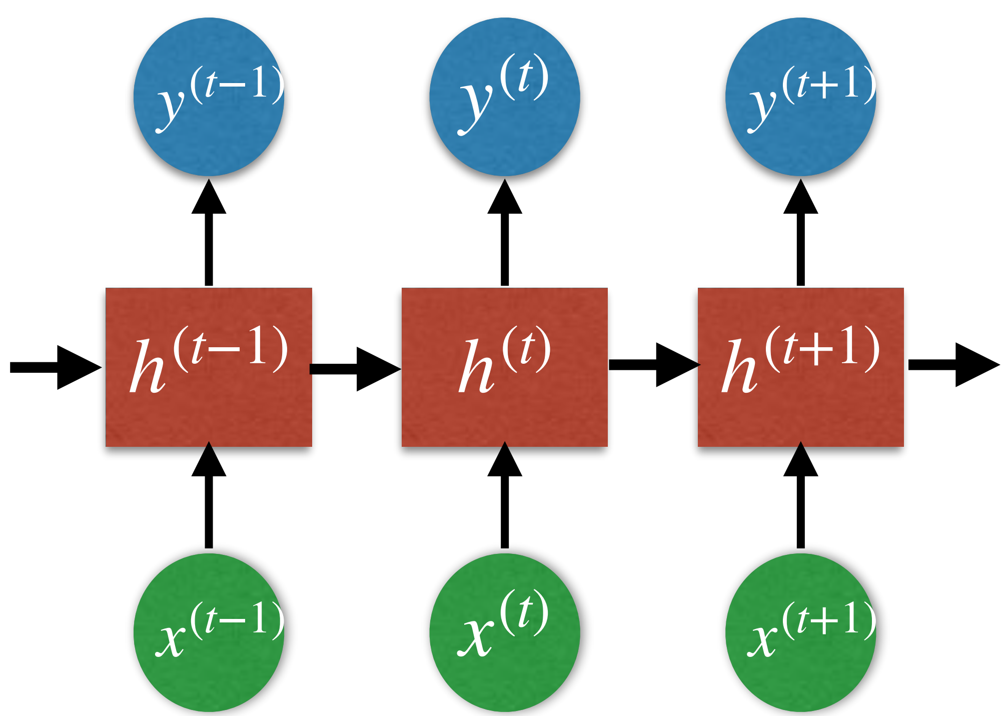
$$\|\mathcal{M}\|_* \le \sqrt{\frac{\prod_{i=1}^{M} d_i}{\max\{d_1, \ldots, d_M\}}} \|\mathcal{M}\|_F.$$

Low-rankness regularizer improves robustness to imperfect data

Learning Representations from Imperfect Time Series Data via Tensor Rank Regularization (Liang et al., ACL 2019)

Recurrent Neural Networks

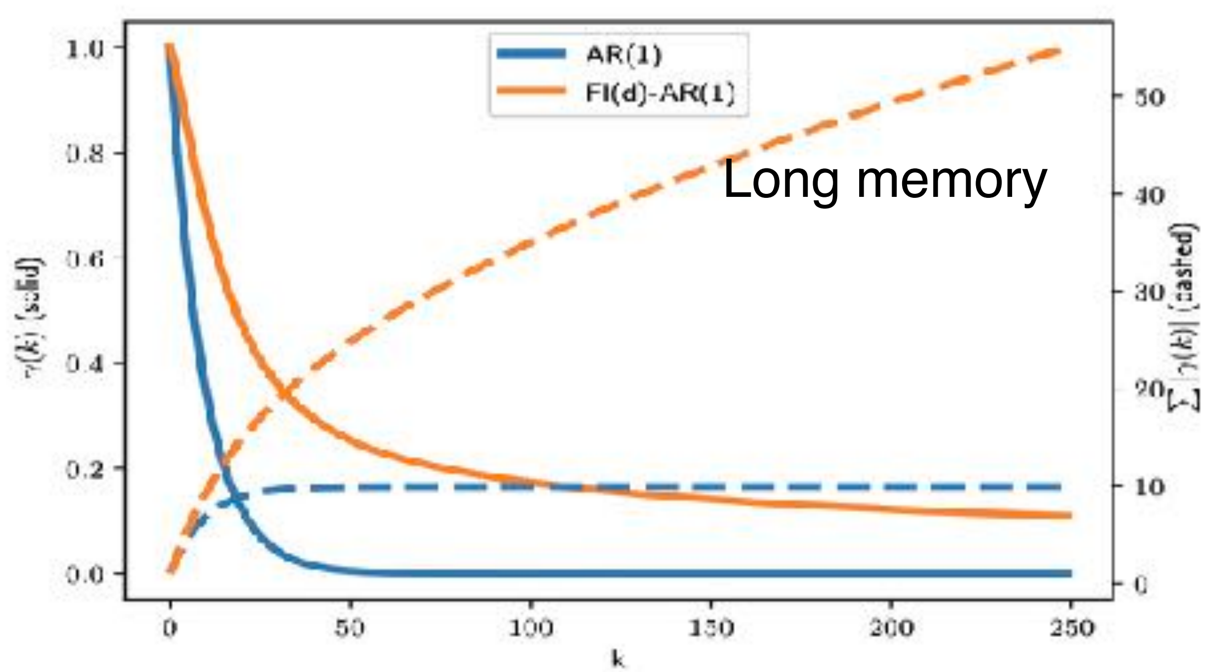
- RNN and LSTM do not have long memory from a statistical perspective [Zhao et al., ICML 2020]
- How to achieve long memory?



RNN

$$h^{(t)} = \sigma(Wh^{(t-1)} + Ux^{(t)} + b)$$

$$\gamma(k) = \mathrm{Cov}(X_t, X_{t+k}), \quad k \in \mathbb{Z}$$



(Greaves-Tunnell et al., ICML 2019)

Tensor-Power Recurrent Models

(Li et al., AISTATS 2021)

Transition function

$(p+1)$-order weight tensor

$$\mathbf{h}^{(t)} = \mathcal{G} \underbrace{\times_1 \begin{pmatrix} \mathbf{x}^{(t)} \\ \mathbf{h}^{(t-1)} \end{pmatrix} \times_2 \cdots \times_p \begin{pmatrix} \mathbf{x}^{(t)} \\ \mathbf{h}^{(t-1)} \end{pmatrix}}_{p\text{-fold tensor product with itself}} = \mathcal{G} \cdot \begin{pmatrix} \mathbf{x}^{(t)} \\ \mathbf{h}^{(t-1)} \end{pmatrix}^{\otimes p}$$

Theorem (Long memory requires a high model degree.)

Under mild assumptions, with high probability, if the tensor power (TP)-induced RNP has the long memory under Def. 1, then the following inequality obeys:

$$p \geq \frac{p_0}{2}\left(1 + \sqrt{1 + \frac{C_1}{n\sigma^2} - \frac{C_2}{n}}\right) - 1, \quad (3)$$

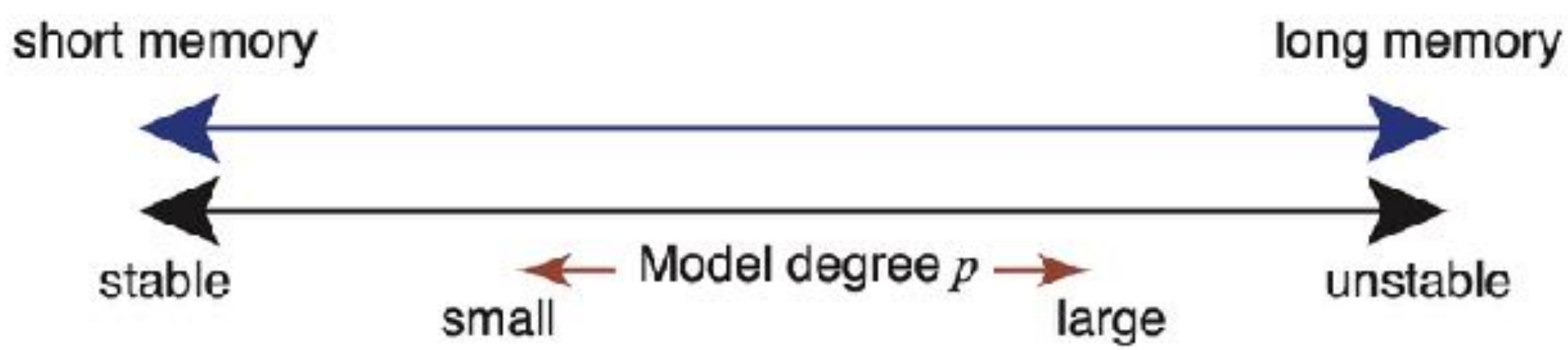
where $p_0 = \log(3/2)$*, and* C_1*,* C_2 *denote two positive constants.*

Large p leads to long memory, small p leads to short memory

Learnable degree p

(Li et al., AISTATS 2021)

- Long memory with increasing p but unstable



- Symmetric tensor decomposition (STD) of weight tensor

$$\mathbf{h}^{(t)} = \mathcal{G} \times_1 \begin{pmatrix} \mathbf{x}^{(t)} \\ \mathbf{h}^{(t-1)} \end{pmatrix} \times_2 \cdots \times_p \begin{pmatrix} \mathbf{x}^{(t)} \\ \mathbf{h}^{(t-1)} \end{pmatrix}$$

STD factors of weight tensor G

$$\mathbf{h}^{(t)}[j] = \sum_{r=1}^{R} \left\langle \mathbf{w}_{j,r}, \begin{pmatrix} \mathbf{x}^{(t)} \\ \mathbf{h}^{(t-1)} \end{pmatrix} \right\rangle^{p} + \mathbf{b}[j],$$

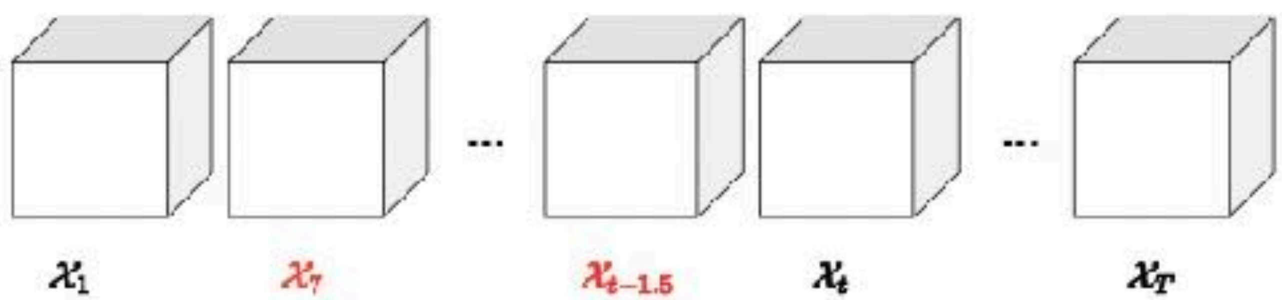
Update p

$$p^{(t)} = MLP\left(p^{(t-1)}, \mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}\right)$$

Tensorial time series data with irregular time step

Task: Given tensorial time series with irregular time steps, how to train a model for prediction on continuous time points and extrapolation for future.

Examples: videos with missing frames, relations between stock market prices of many companies, etc



Challenges:

- Tensorial NN/RNN (Bai et al. 2017): Incapable of handling irregular time steps, and prediction on decimal time points
- Neural ODE (Chen et al. NeurIPS 2018): Ignoring spatial structure information, large number of parameters

Tensor Neural ODE

(Bai et al., IJCNN 2021)

We directly process the tensorial time series $\{\mathcal{Y}_t\}_{t\in[0,T]}, \mathcal{Y}_t \in \mathbb{R}^{I_1\times\cdots\times I_N}$, proposing tensor neural ODE (TENODE)

$$\frac{\mathrm{d}\mathcal{Y}(t)}{\mathrm{d}t} = f_{\Theta}(\mathcal{Y}(t), \mathcal{X}(t), t)$$

with the control input $\mathcal{X}(t)$ and the initial condition $\mathcal{Y}(0) = \mathcal{Y}_0$. Parameter size: from $O(I^{2N})$ of neural ODE to $O(NI^2)$

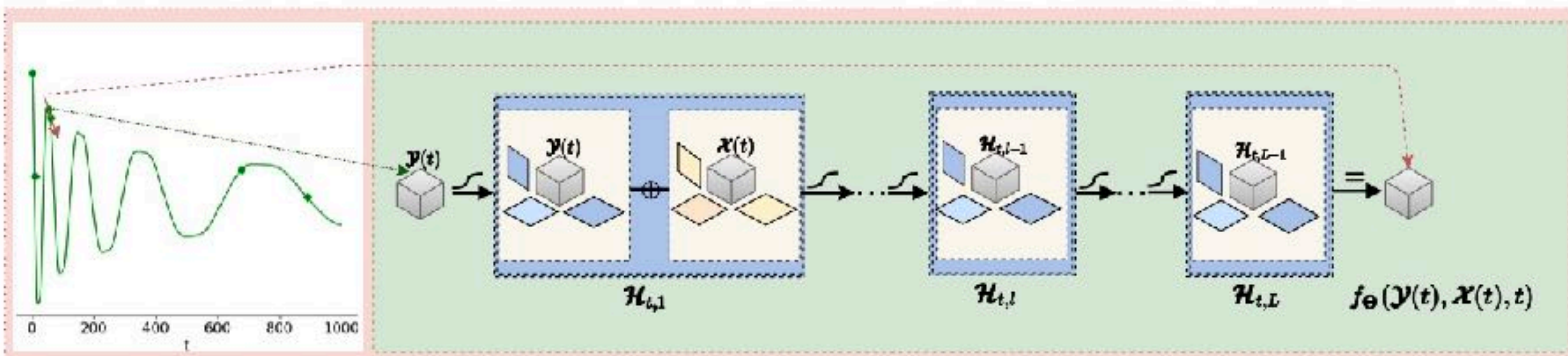


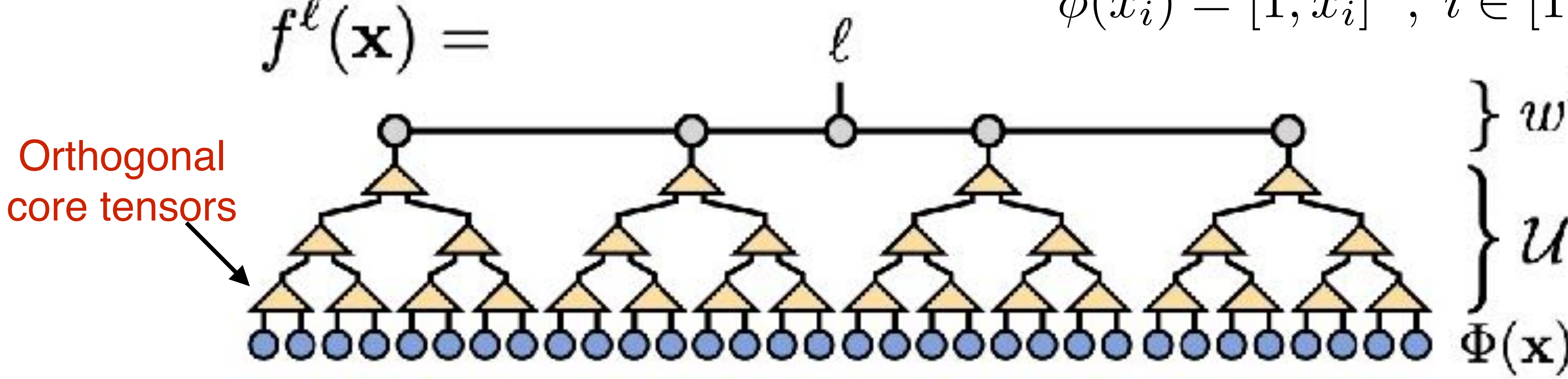
Figure 5: Architecture Overview: Tensor neural ODE (TENODE)

Trends and Directions

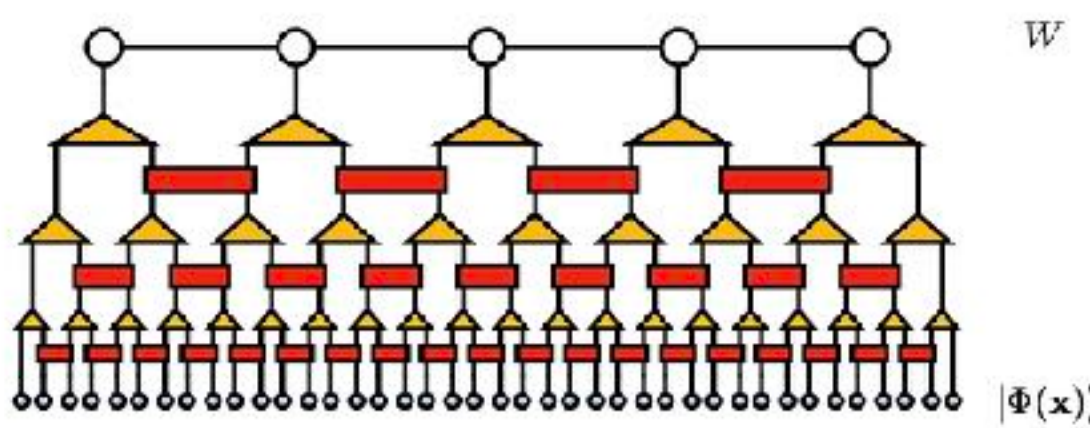
Multi-Scale Tensor Network Architecture

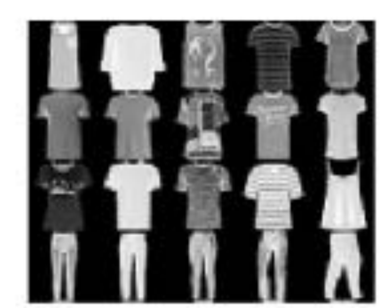
- Unsupervised learning with reduced order of TN representation
- Supervised learning for the top classification layer

$$\phi(x_i) = [1, x_i]^T,\ i \in [1, d]$$



98.08% accuracy on MNIST



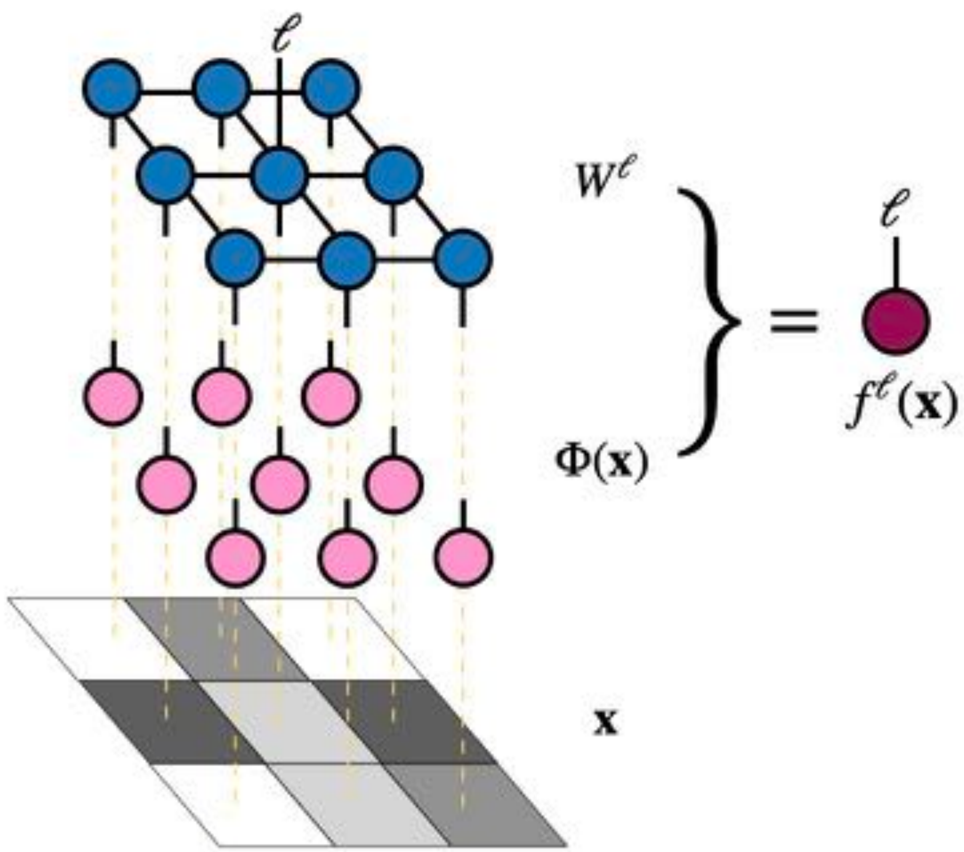


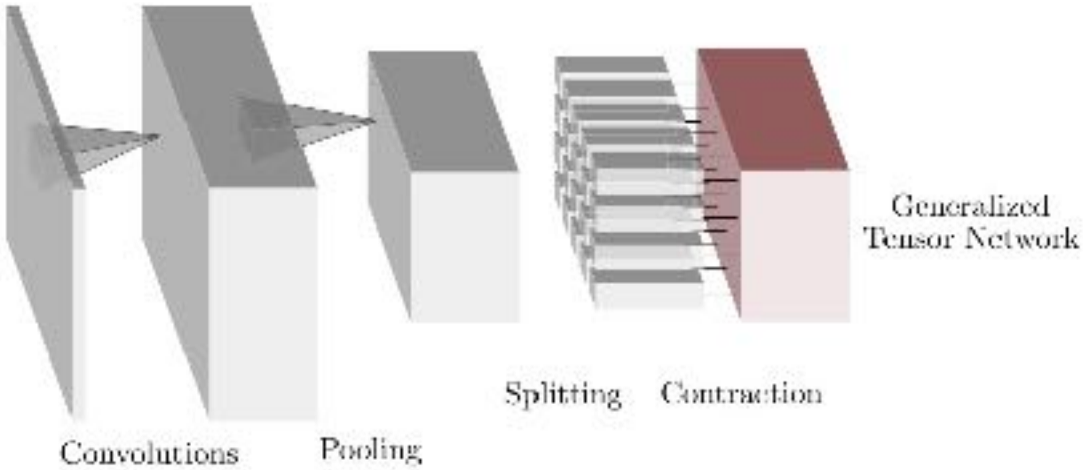
89% accuracy on Fashion MNIST data set

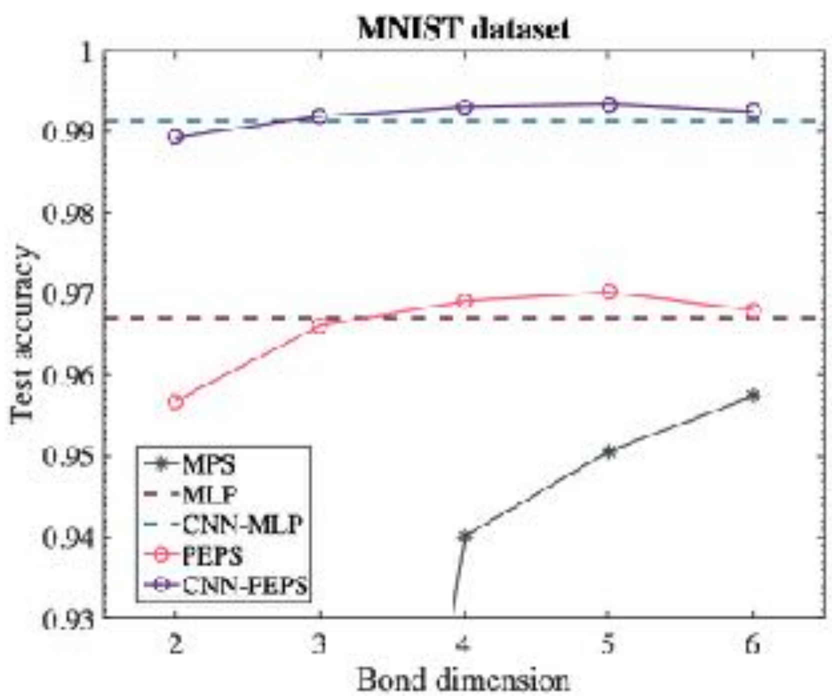
Learning Relevant Features of Data with Multi-scale Tensor Networks (Stoudenmire et al. 2018)
A Multi-Scale Tensor Network Architecture for Classification and Regression (Reyes et al., 2020)

Supervised Learning with Projected Entangled Pair States

- Hybrid model (CNN + PEPS)



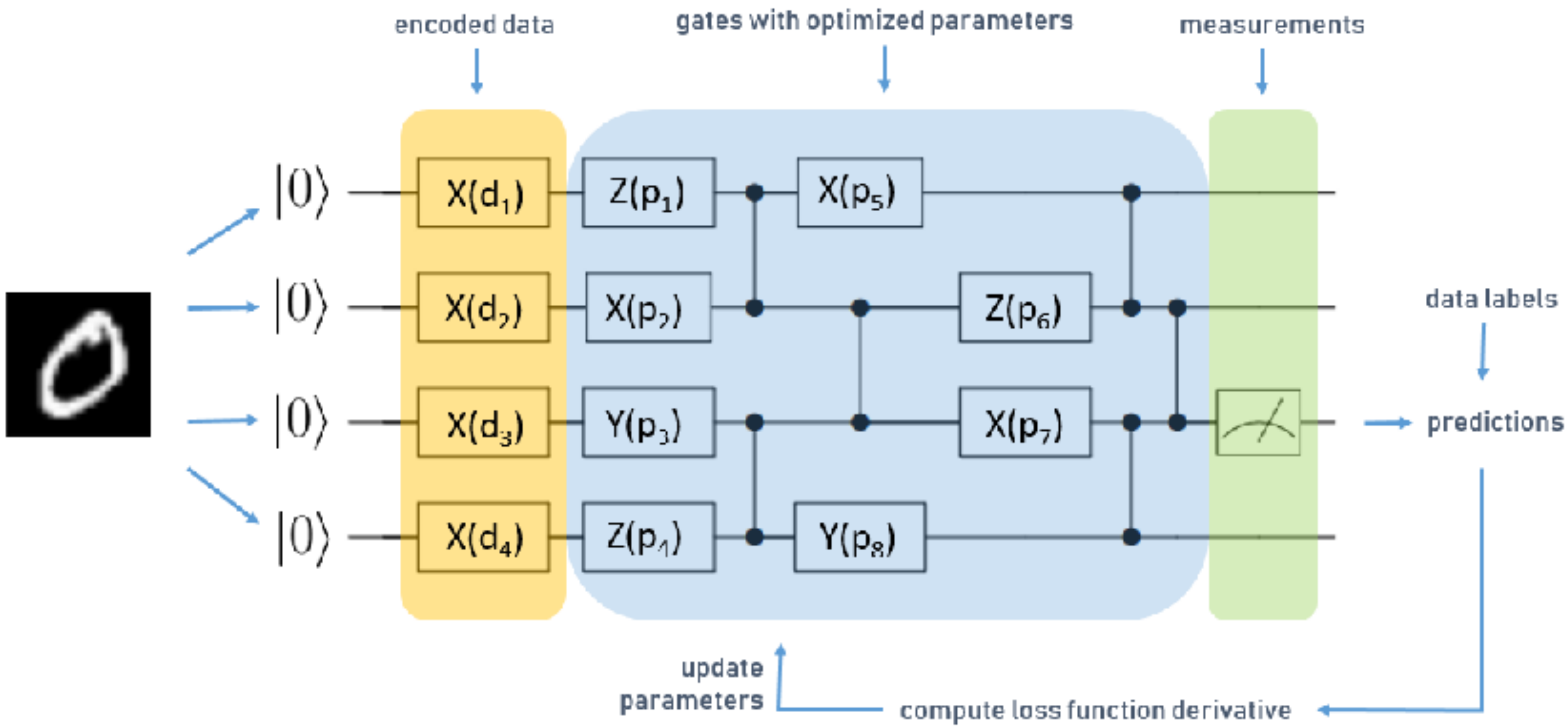




From Probabilistic Graphical Models to Generalized Tensor Networks for Supervised Learning (Glasser, arXiv 2019)

Supervised Learning with Projected Entangled Pair States (Chen et al., arXiv 2020)

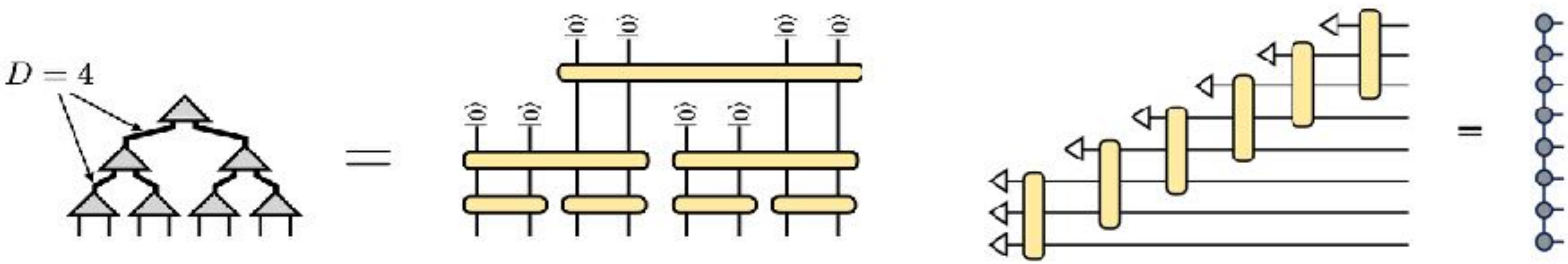
Quantum Neural Networks



https://blog.tensorflow.org/2020/08/layerwise-learning-for-quantum-neural-networks.html

TensorFlow Quantum

Quantum Machine Learning with Tensor Networks



- Robustness to noise
- Tensor network circuits provide qubit-efficient schemes

Towards Quantum Machine Learning with Tensor Networks (Huggins et al., 2019)

Summary

- TNs are useful tools for representation of high order structured data, and efficient reparameterization of deep NN models
- Theory shows TNs have expressive power similar to DNNs
- Robustness to adversarial attacks and interpretability of TN based ML models

Acknowledgements

- Team Members

Chao Li

Jianfu Zhang

Andong Wang

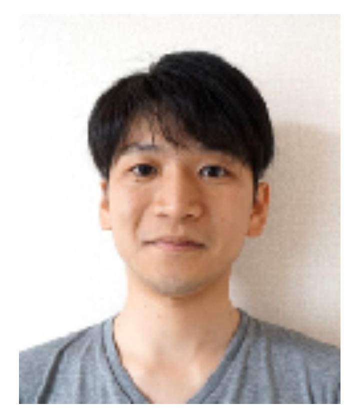
Zerui Tao

Yubang Zheng

Andrzej Cichocki

Toshihisa Tanaka

Cesar F. Caiafa



Tatsuya Yokota

Jianting Cao

- Part-timer, interns, and collaborators