Discovering Optimal Tensor Network Architectures:

Discrete Optimization for Tensor Network Structure Search (TN-SS)

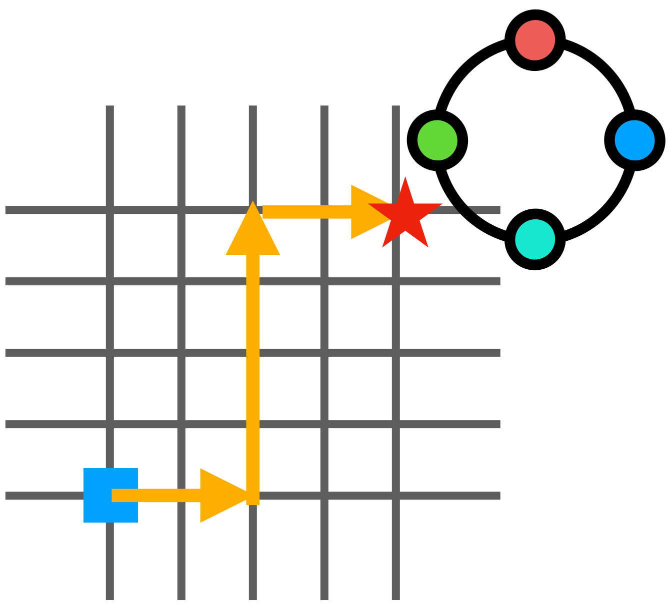
Chao Li

Indefinite-Term Research Scientist

Tensor Learning Team (TLT)

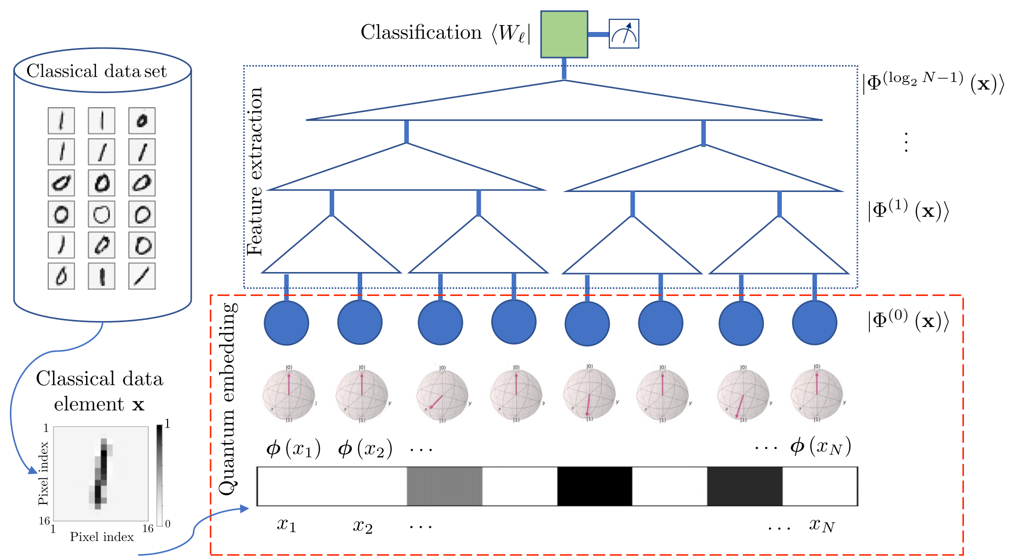
RIKEN-AIP

chao.li@riken.jp

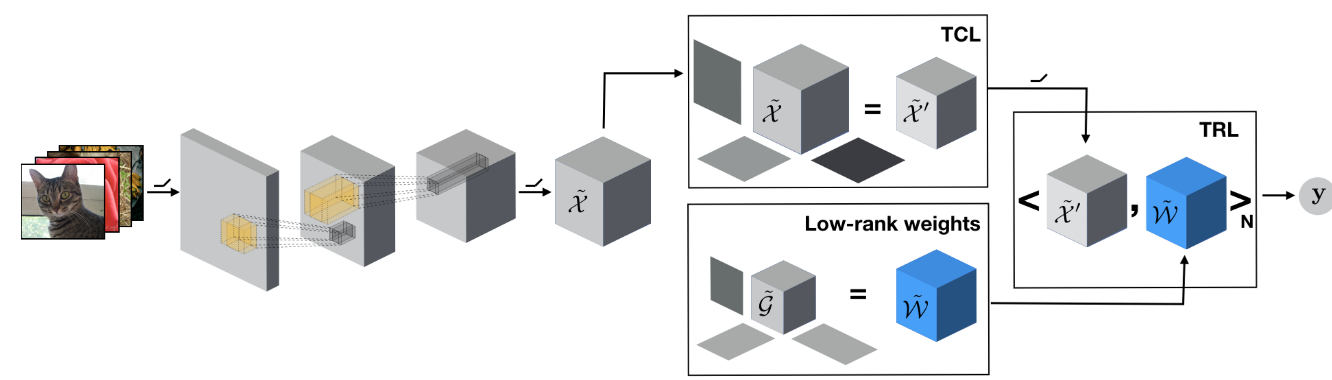


Tensor Network (TN)

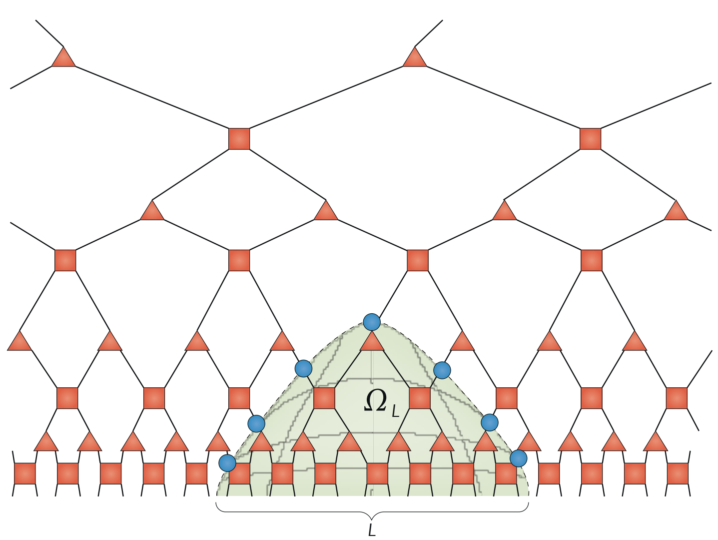
TN is an efficient framework for modeling complex systems by *decomposing it* into simpler, interconnected parts.



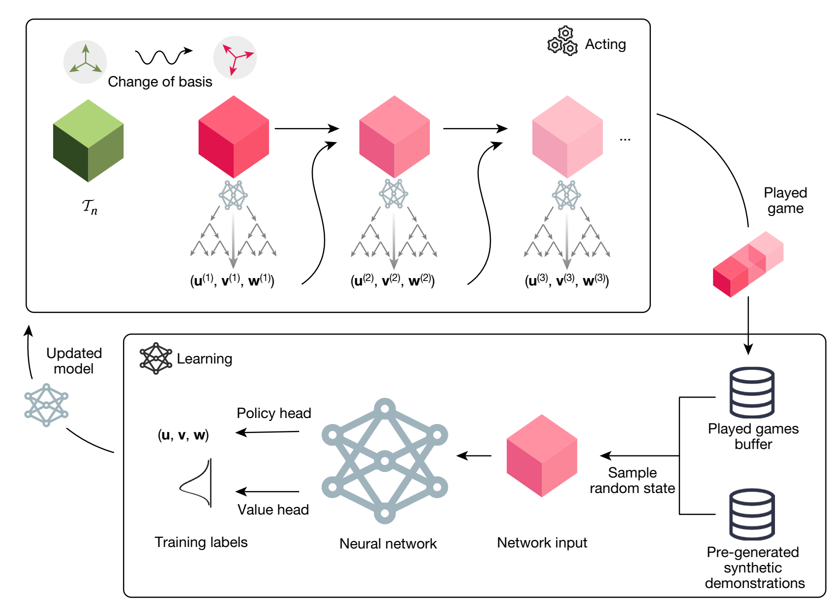
Physic-informed machine learning



Acceleration of ***neural networks*** via TN

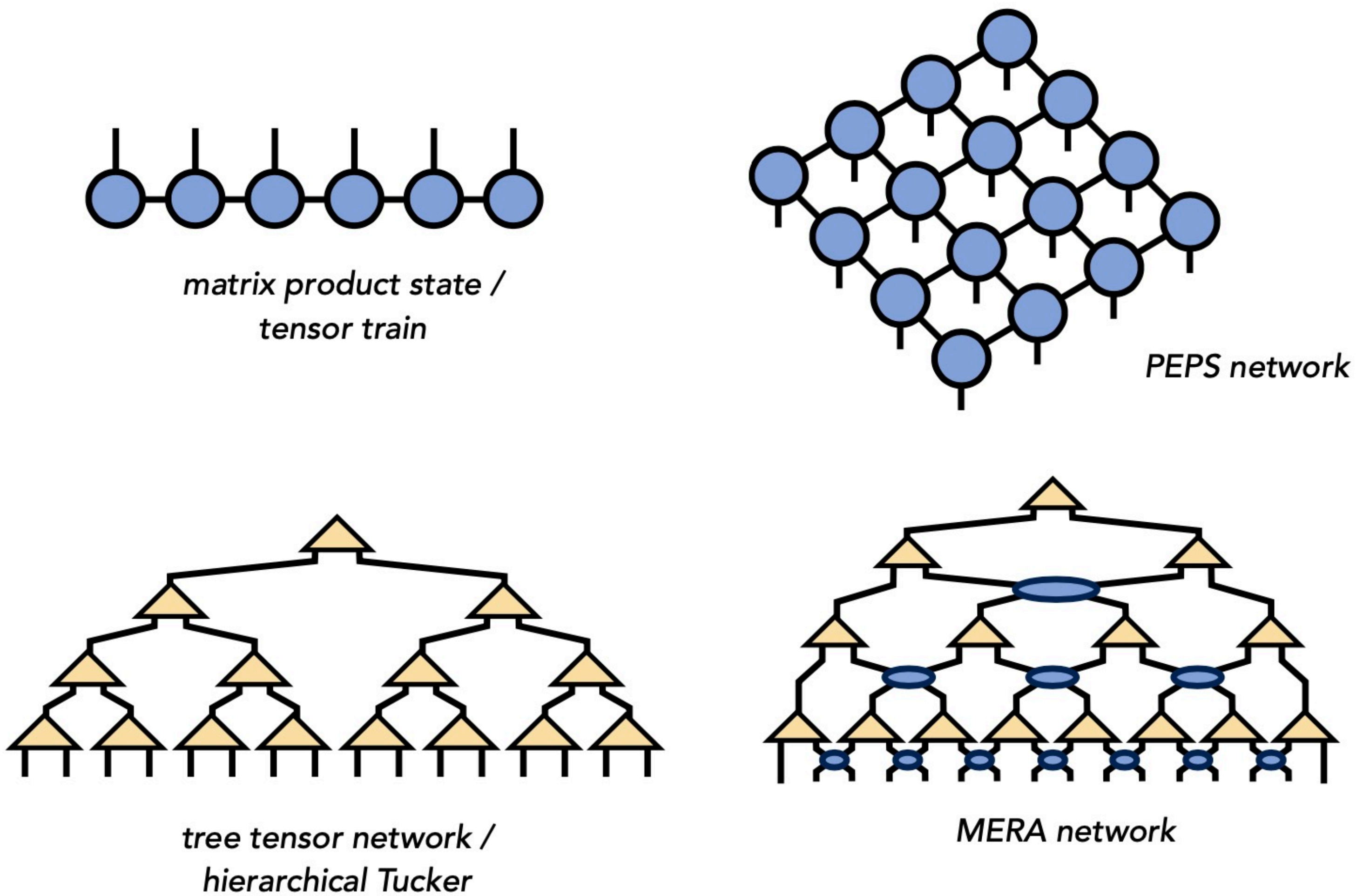


Representation for complex ***quantum systems***
(Orus, Nature Phys.'19)

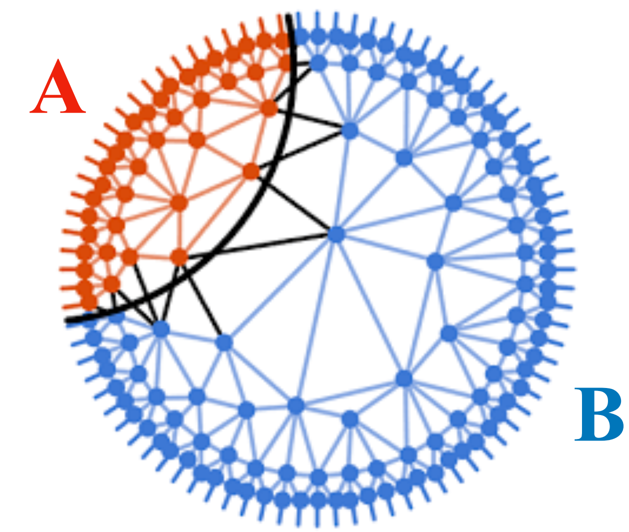


Discovering faster ***matrix multiplication***
(AlphaTensor, Fawzi et al., Nature'22)

Vision: Diversity of Tensor Networks



R. Orus, Ann. of Phys. 349, 117–158 (2014)



Modeling entanglement between A and B

What is the most suitable TN model for our task?

How can we efficiently select the structure-related parameters?

Steps to Attain the Goal

- Formulating TN-SS as discrete optimization
- Solving TN-SS with less computational cost
  - *TNGA*: Genetic Algorithm *(Li and Sun, ICML'20)*
  - *TNLS:* Stochastic Search *(Li et al., ICML'22)*
  - *TnALE:* Alternating Enumeration *(Li et al., ICML'23)*
- Theoretical Analysis
  - Symmetry of TN structures
  - *Search Dynamic in TNLS/TnALE*
- Future works

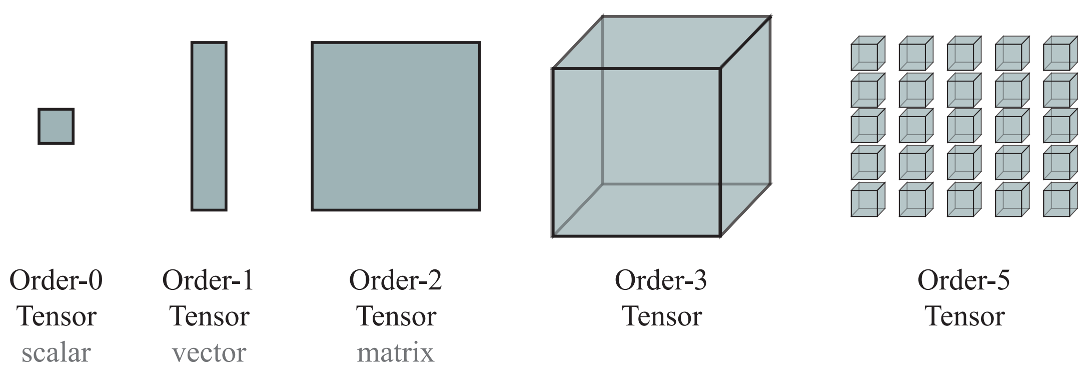
What is TN-SS?

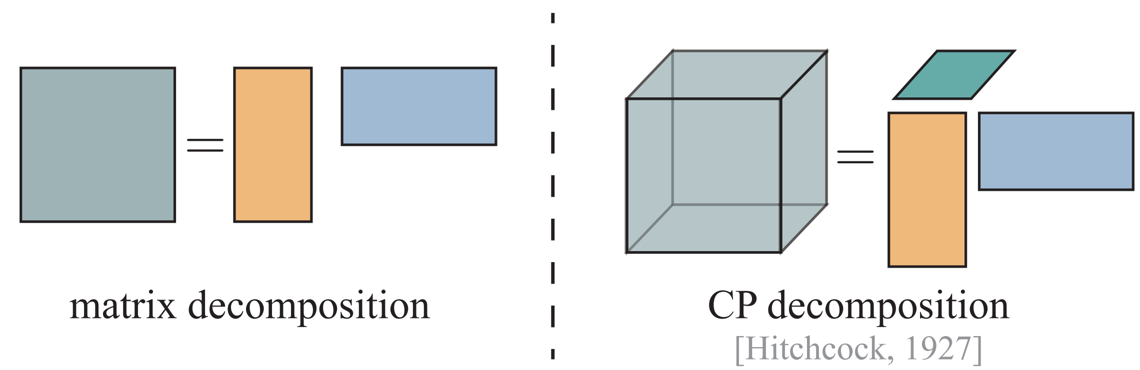
Tensor and TN's Graphical Representation

Tensor is the foundational building block of TNs.

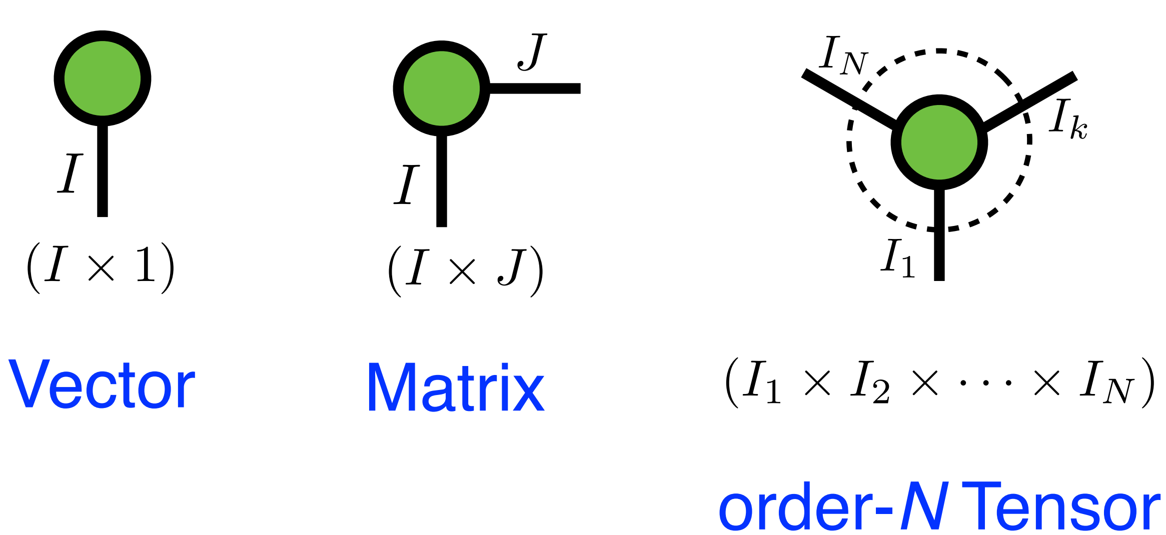
TENSOR is a multi-way number array.

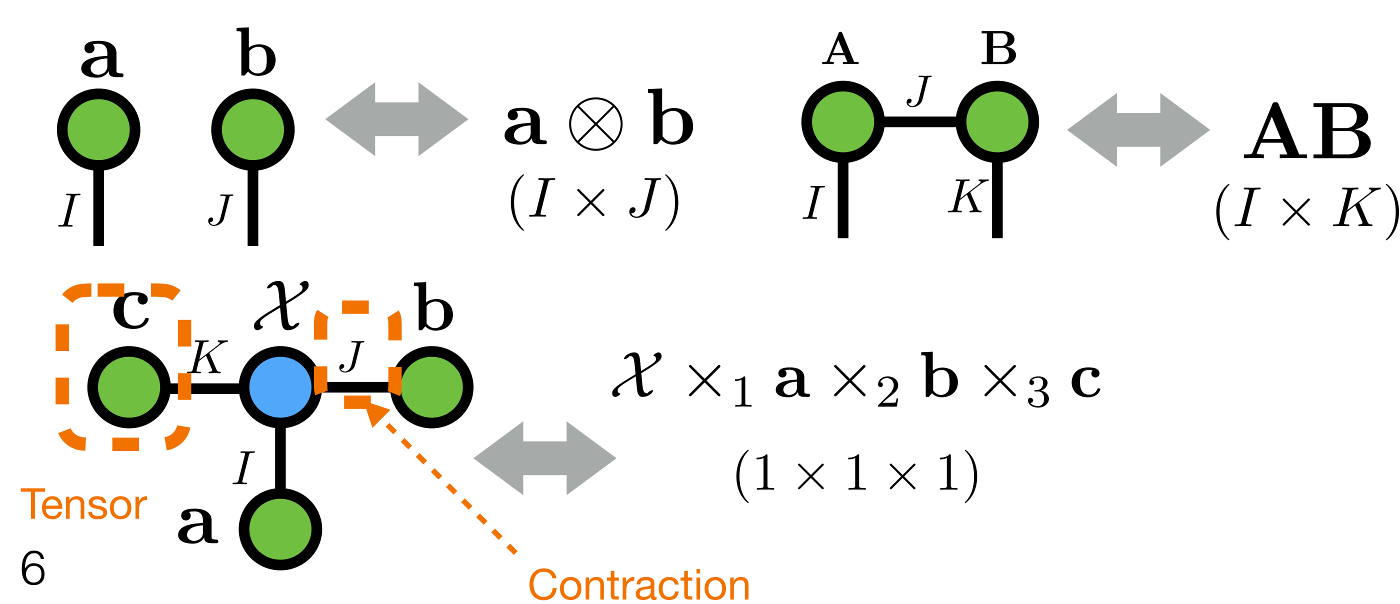
CONTRACTION: "tensor-tensor" multiplication.





A ***TENSOR NETWORK*** (TN) is modeled as an edge-labeled graph depicting a sequence of contractions among many tensors.





The TN structures include ***TN-ranks***, ***vertex-permutation***, and ***TN-topology.***

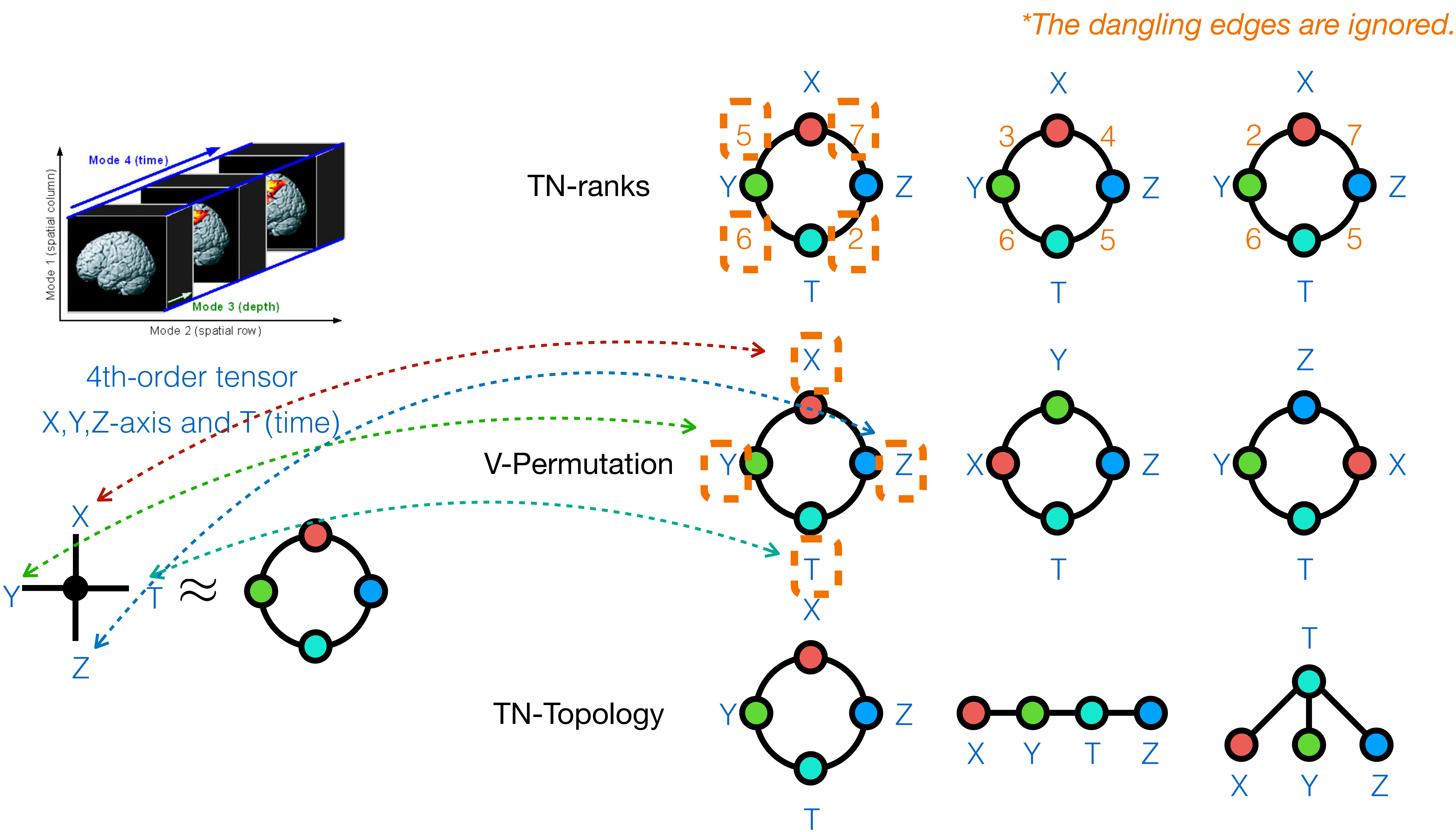
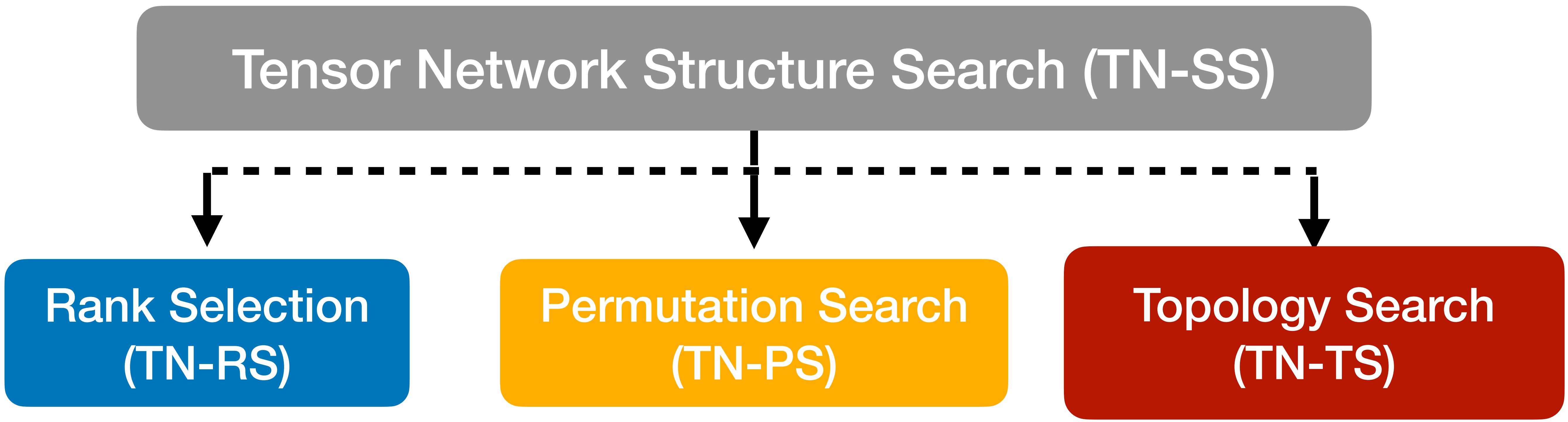


Image source: https://staffwww.dcs.shef.ac.uk/people/H.Lu/feeler.html

TN-SS refers to a process of exploring and identifying ***the optimal combination of those structures*** to represent the complex system using a tensor network.

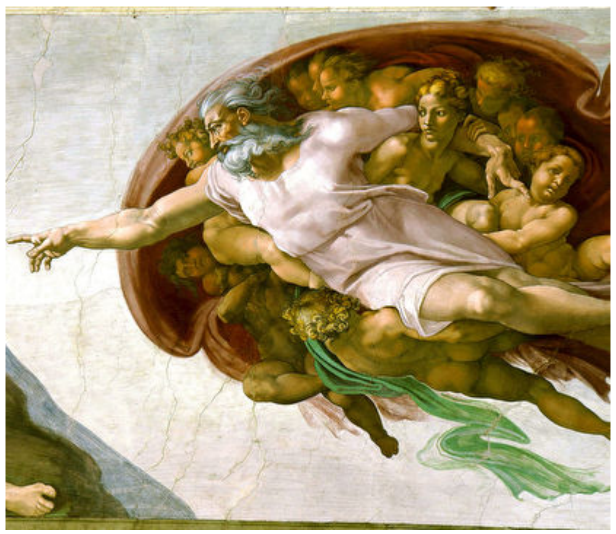


The goal is to *reduce the computational cost* in the search process.

Solving TN-SS is challenging!

- "Most tensor problems are NP-hard." *(CJ Hillar and LH Lim, JACM'13)*
- TN-SS suffers from the ***combinatorial explosion*** issue.

N	*The number of TN-Structure candidates*
2	5
3	125
4	15625
5	9765625
6	30517578125
7	476837158203125
8	37252902984619139072
9	14551915228366852423942144
10	28421709430404005438427049754624
11	277555756156289137946709683182497693696[2]



Matrices

Tensors

Max Noether:

***Matrices* were created by God. *Tensors* were created by the Devil.**

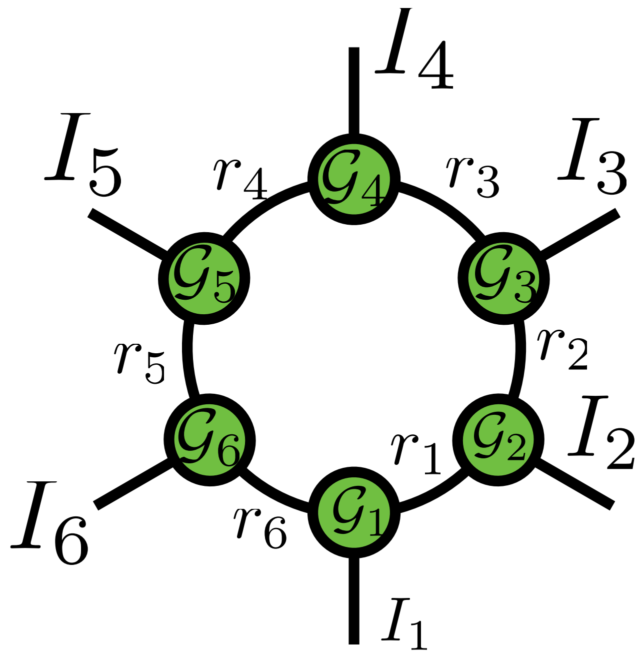
Solving TN-SS via Discrete Optimization

Mathematically, TN-SS is to solve the following optimization problem:

$$\min_{\boxed{(G,r)\in\mathbb{G}\times\mathbb{F}_G}} \left(\underbrace{\phi(G,r)}_{\text{model complexity}} + \lambda \cdot \underbrace{\min_{\mathcal{Z}\in TNS(G,r)} \pi_{\mathcal{X}}(\mathcal{Z})}_{\text{model expressivity}} \right),$$

- $\mathbb{G}$ — *graphs* associated to TN topology and permutation;
- $\mathbb{F}_G$ — positive-integer *vectors* associated to the TN-rank;
- TN-RS/TS/PS tasks correspond to setting different $\mathbb{G}$ and $\mathbb{F}_G$ in the formula.

TN ranks and topology: **adjacency matrix**



order-6 Tensor Ring

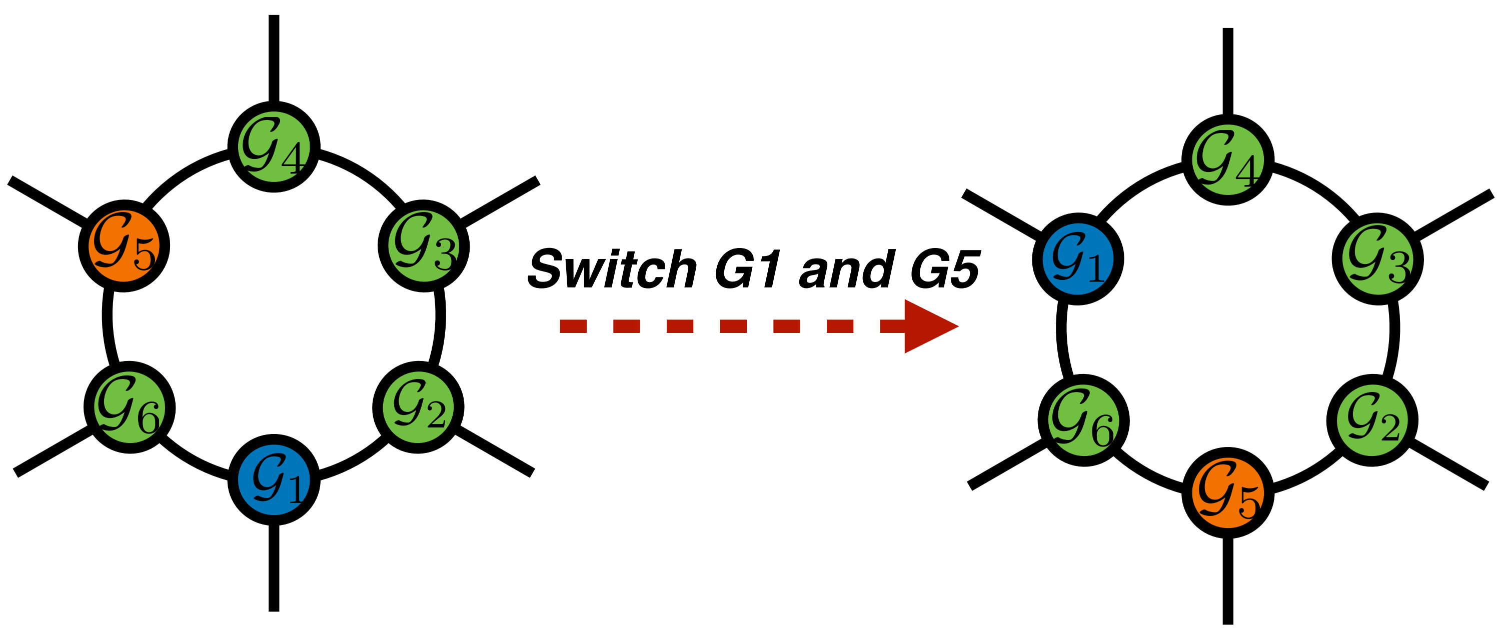
$$\begin{array}{cc} & \begin{array}{cccccc} \mathcal{G}_1 & \mathcal{G}_2 & \mathcal{G}_3 & \mathcal{G}_4 & \mathcal{G}_5 & \mathcal{G}_6 \end{array} \\ \begin{array}{c} \mathcal{G}_1 \\ \mathcal{G}_2 \\ \mathcal{G}_3 \\ \mathcal{G}_4 \\ \mathcal{G}_5 \\ \mathcal{G}_6 \end{array} & \begin{pmatrix} I_1 & r_1 & 0 & 0 & 0 & r_6 \\ r_1 & I_2 & r_2 & 0 & 0 & 0 \\ 0 & r_2 & I_3 & r_4 & 0 & 0 \\ 0 & 0 & r_3 & I_4 & r_4 & 0 \\ 0 & 0 & 0 & r_4 & I_5 & r_5 \\ r_6 & 0 & 0 & 0 & r_5 & I_6 \end{pmatrix} \end{array}$$

Ranks

(Free legs) or zeros

(Augmented) Adjacency matrix

vertex permutation: **permutation matrix**



$$\hat{\mathbf{A}} = \mathbf{P}\mathbf{A}\mathbf{P}^\top$$

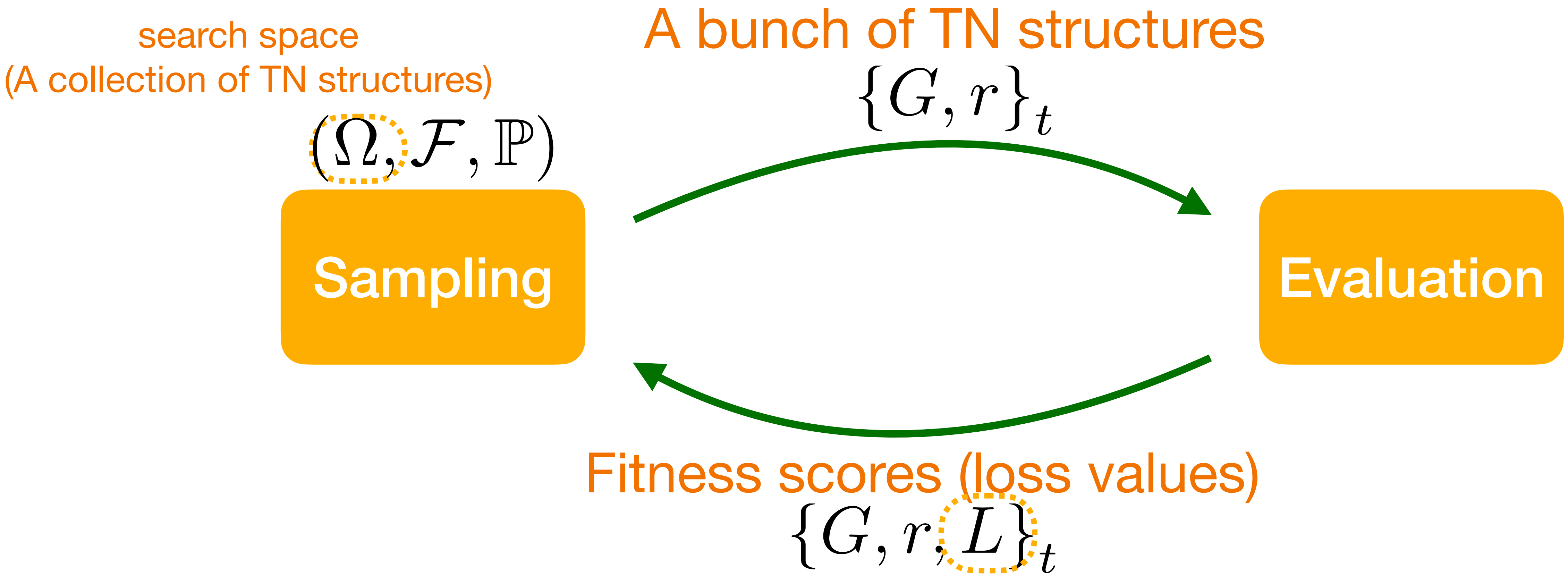
$$\mathbf{P} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Switch

How to solve TN-SS?

with discrete optimization

Big Picture



The sampling distribution is "Markov": $\mathbb{P}\left(\{G, r\}_t \,|\, \{G, r, L\}_{t-1}\right)$

- *TNGA*: Genetic Algorithm *(Li and Sun, ICML'20)*
- *TNLS:* Stochastic Search *(Li et al., ICML'22)*
- *TnALE:* Alternating Enumeration *(Li et al., ICML'23)*

Solution 1: Genetic Algorithm

(Li and Sun, ICML'20)

TNGA: Encoding the TN structures into fixed-length strings.

Adjacency Matrix

$$\begin{bmatrix} 5 & 2 & 0 & 4 \\ 2 & 6 & 3 & 0 \\ 0 & 3 & 6 & 4 \\ 4 & 0 & 4 & 3 \end{bmatrix} \longrightarrow$$

Chromosomes (Population)

2 0 1 3 0 4

2 0 5 3 0 4

⋮

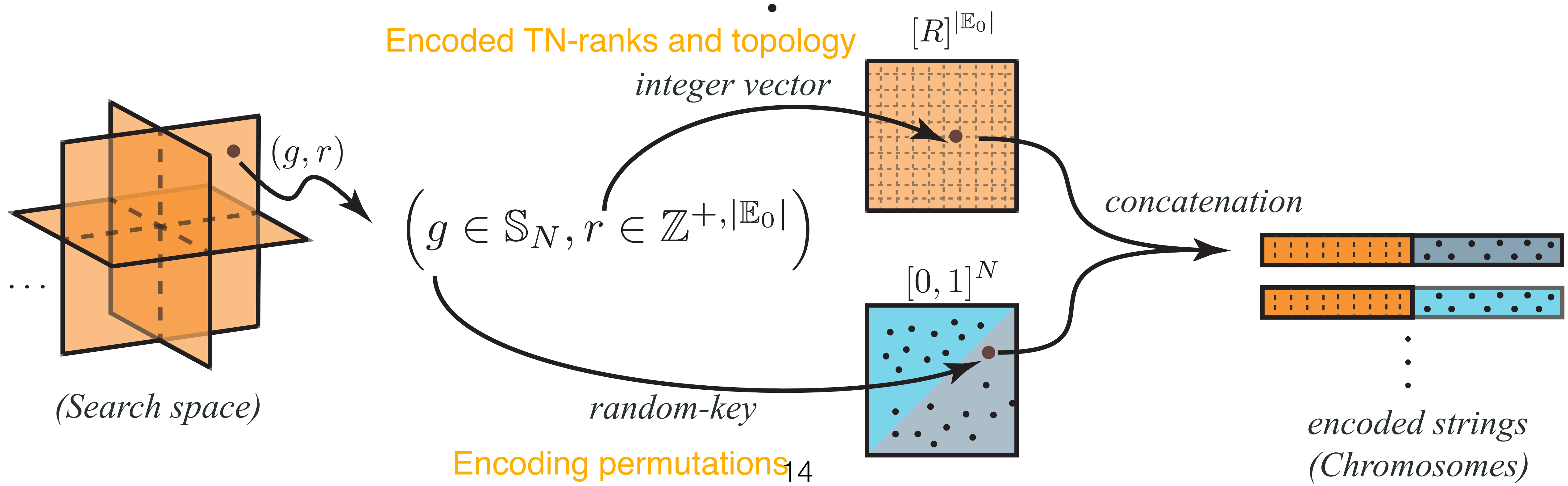
Gene (Allele=0)

Permutation matrix: vertex permutation

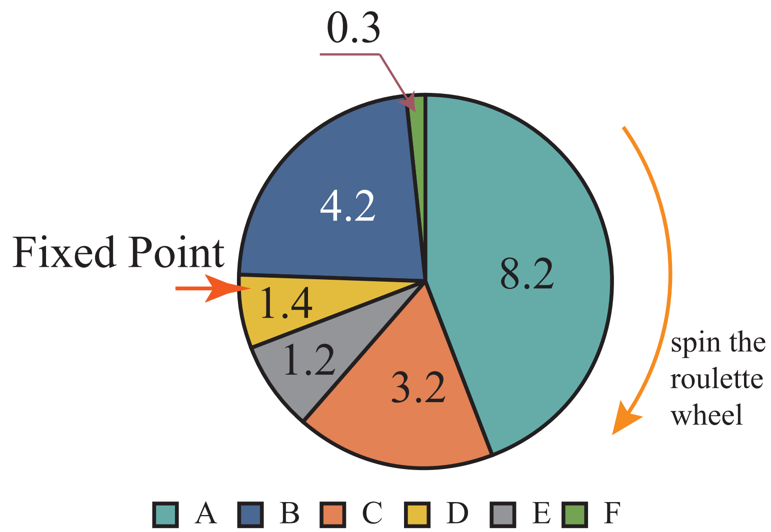
$$\hat{\mathbf{A}} = \mathbf{P}\mathbf{A}\mathbf{P}^{\top}$$

Random key trick *(Bean, 1994)*

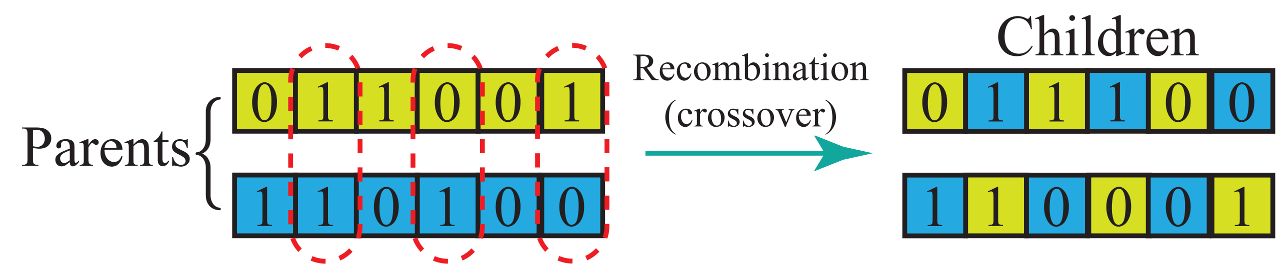
Random key:	0.8	0.5	0.7	0.1	0.4	0.2
Decoded as:	6	4	5	1	3	2



1. Parent selection

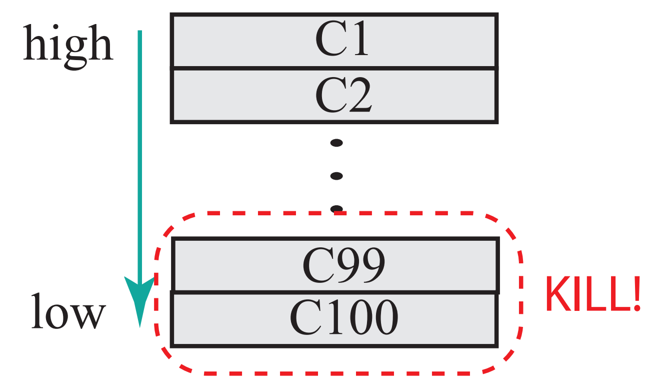


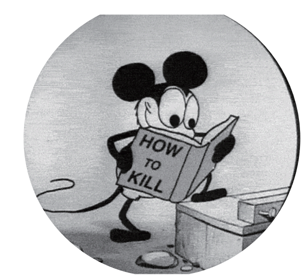
2. Crossover



3. Mutation

4. Elimination





Pros:

- Global convergence
- Multiobjective friendly
- Parallel computation

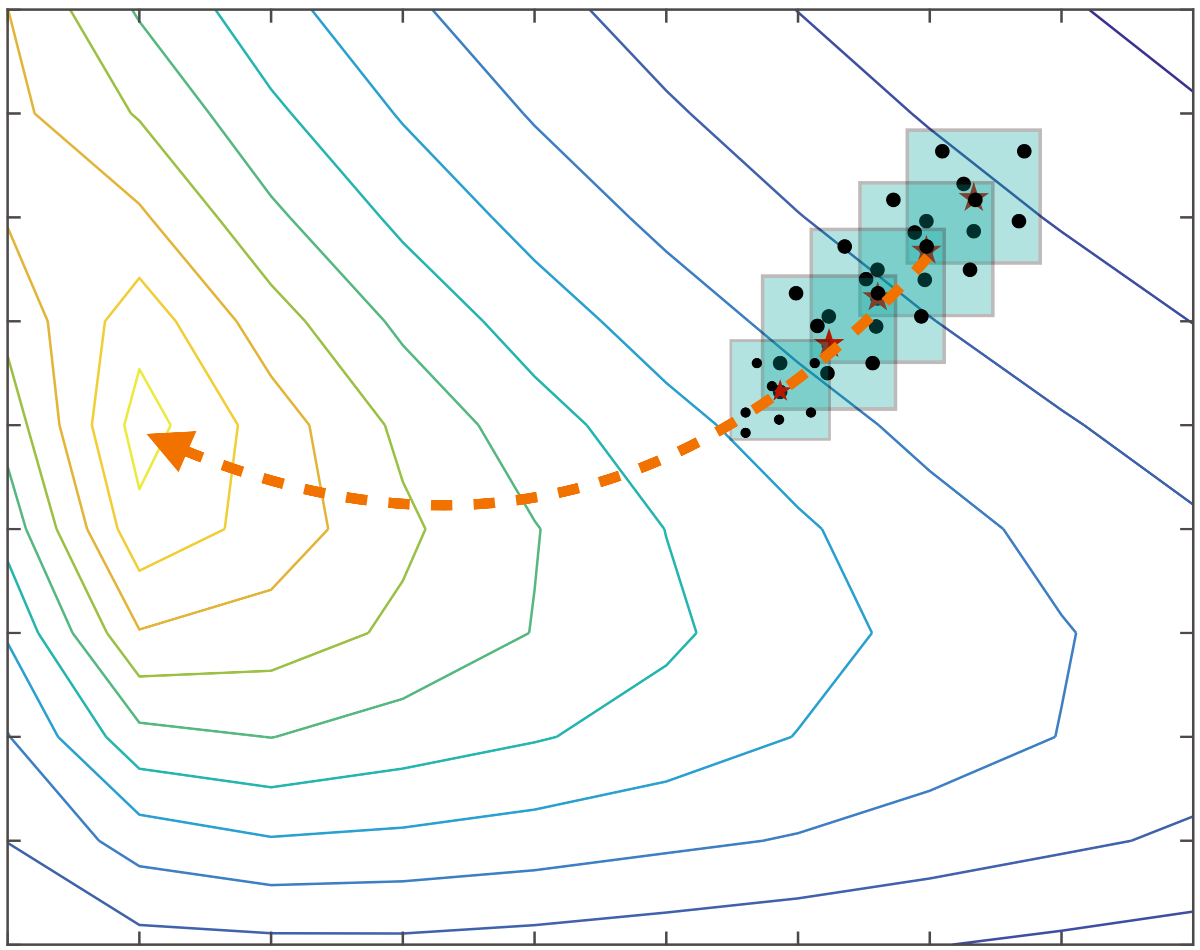
Cons:

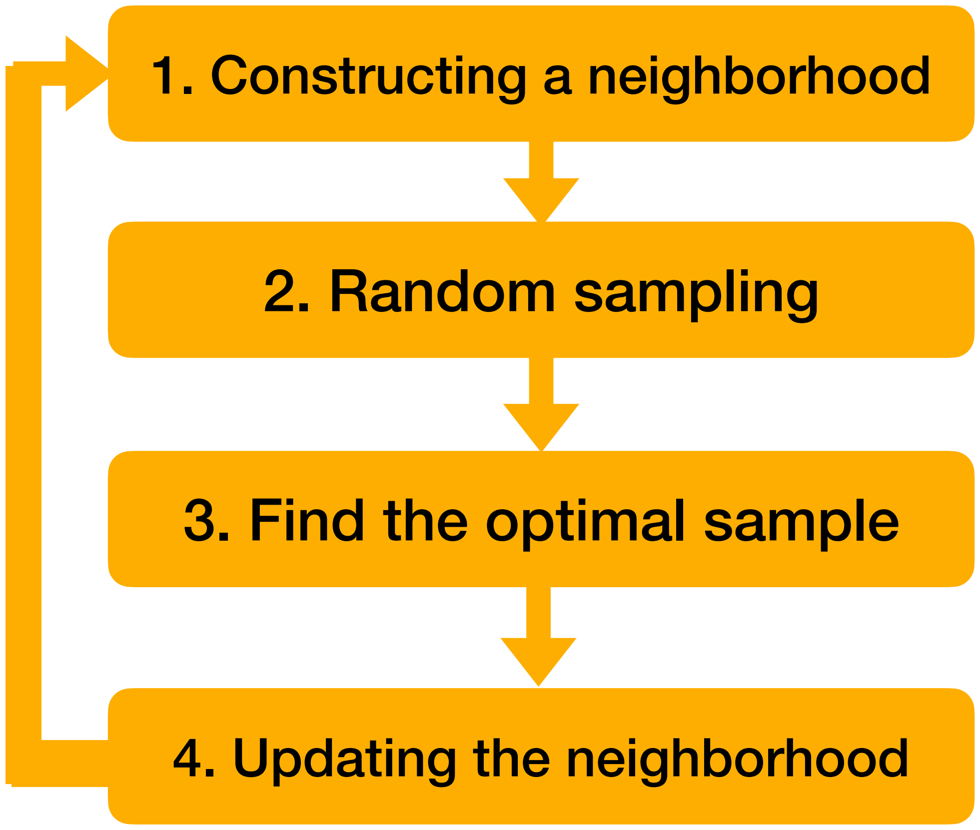
- Low sample efficiency
- No theoretical guarantee
- Too many tuning parameters.

Solution 2: Local Stochastic Search

(Li et al., ICML'22)

- TNLS:"steepest searching direction" by random sampling.





- ***No free lunch:*** the optimization landscape should be smooth.

Theoretical Results

Neighborhood of permutations

Let G_0 be a simple graph and $\mathbb{G}_0$ be the search space. The function $d_{G_0} : \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{R}$ is a predefined semi-metric on $\mathbb{G}_0$. Furthermore, let $\mathbb{I}_d(G)$ be a set constructed as follows:

$$\mathbb{I}_d(G) = \{G' \in \mathbb{G}_0 | G' = q \prod_{i=1}^{d} t_i \cdot G_0, q \in g \cdot Aut(G_0),\ t_i \in \mathbb{T}_N,\ i \in [d]\}. \tag{10}$$

Then $\mathbb{N}_D(G) = \bigcup_{d=0}^{D} \mathbb{I}_d(G)$ is the neighborhood of $G = g \cdot G_0 \in \mathbb{G}_0$ induced by the word metric, with the radius $D \in \mathbb{Z}^+ \cup \{0\}$.

Theorem (convergence rate when p^* is known)

Suppose several assumptions are satisfied, the operator p of (3) is fixed to be p^*, and $0 \le \theta \le 1$. Then, for any $\mathbf{x}$ with $\|\mathbf{x} - \mathbf{x}^*\|_\infty \le c$, we can find a neighborhood $B_\infty(\mathbf{x}, r_\mathbf{x})$ where $r_\mathbf{x} \ge \theta c + \frac{1}{2}$, such that there exists an element $\mathbf{y} \in B_\infty(\mathbf{x}, r_\mathbf{x})$ satisfying

$$f_{p^*}(\mathbf{y}) - f_{p^*}(\mathbf{x}^*) \le (1-\theta)(f_{p^*}(\mathbf{x}) - f_{p^*}(\mathbf{x}^*)) + \boxed{\frac{7}{8}K}. \tag{5}$$

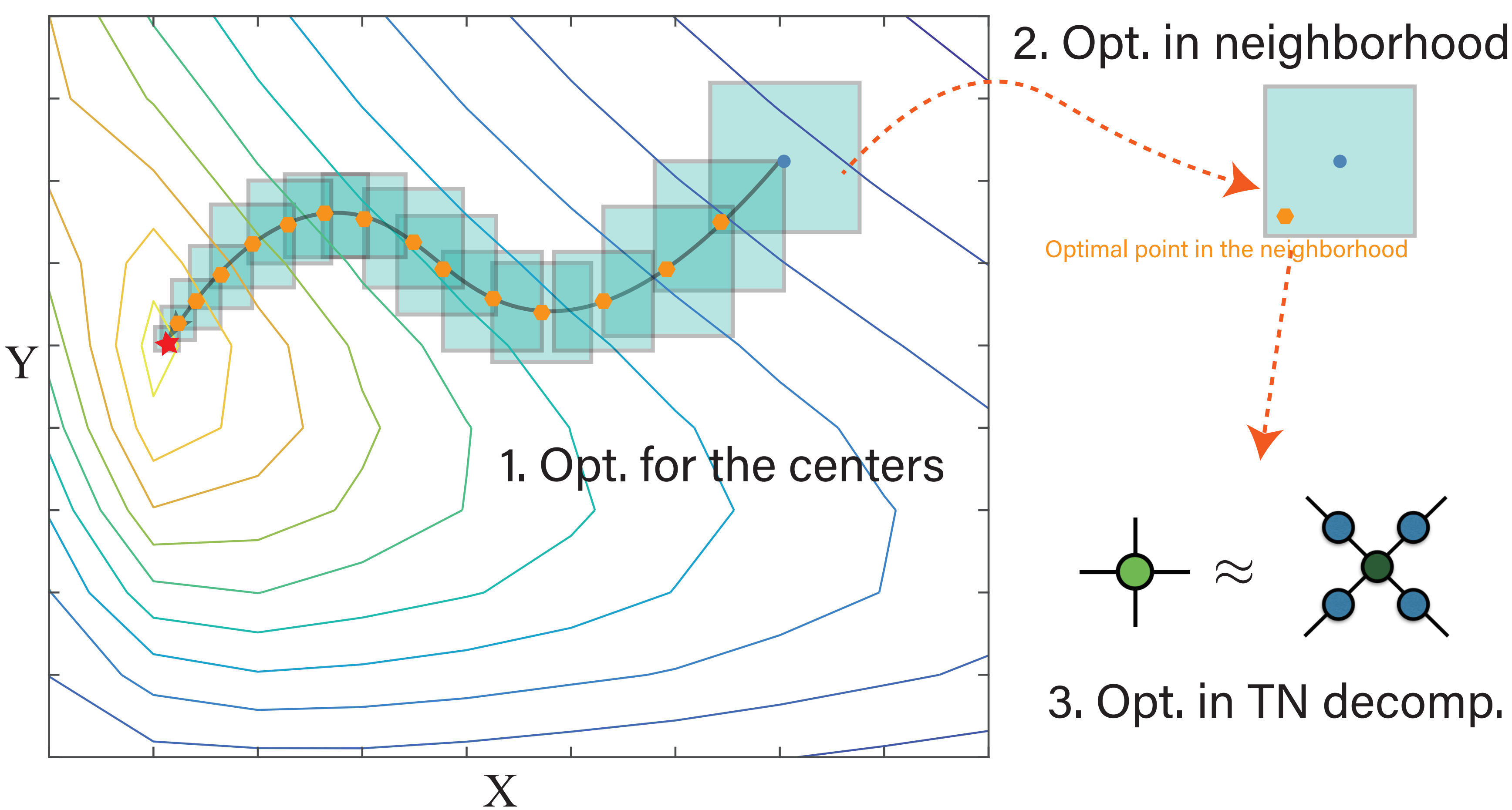
Curse of dimensionality for TNLS (informally)

Theoretically, $\boxed{\mathcal{O}(2^K/\epsilon)}$ samples are required for achieving the probability $Pr \ge \epsilon$ for decreasing the loss function in one iteration.

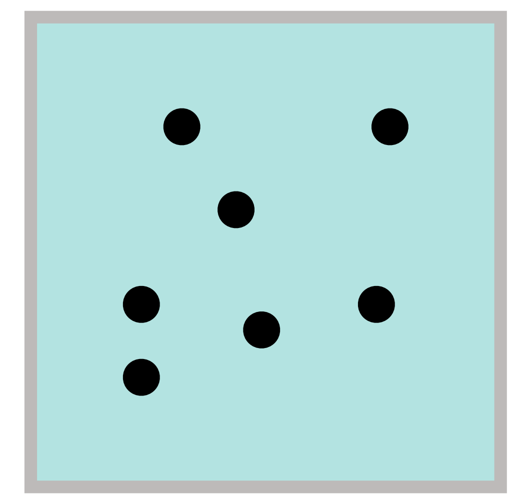
Solution 3: Alternating Local Enumeration

(Li et al., ICML'23)

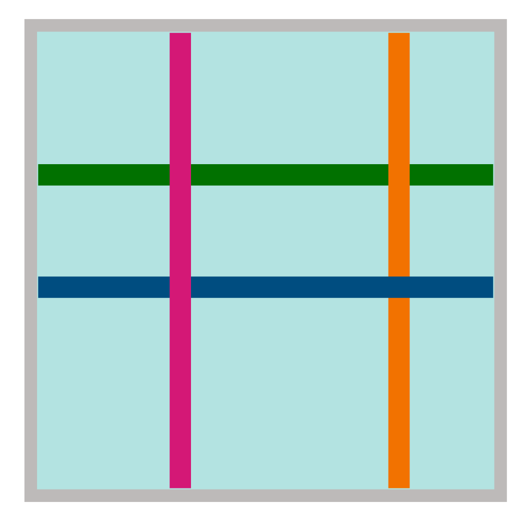
In the new algorithm, called *TnALE*, we follow the fundamental scheme of TNLS, but the ***random sampling*** is replaced by ***alternating enumeration***.

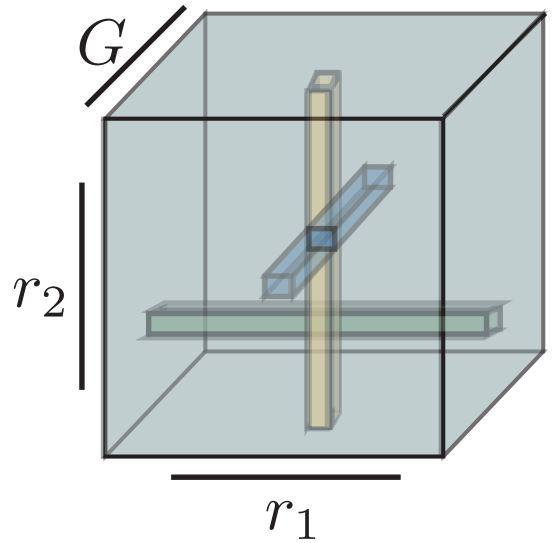


Random sampling



Alternating enumeration





In each neighborhood, the alternating enumeration strongly relates to TT-OPT/cross *(Sozykin et al., Neurips'22, Oseledets, 2010)*

Theorem (A suitable y can be estimated by TT-cross with $\mathcal{O}(KIr)$ samples)

Let $\mathcal{B} \in \mathbb{R}^{I \times I \times \cdots \times I}$ be a tensor of order-K constructed using $B_\infty(x, r_x)$ as above. Then there exists its TT-cross approximation of rank-r as in (Oseledets, 2010), denoted $\hat{\mathcal{B}}$, such that $f\left(x + \mathbf{i}_{\max} - (\lfloor r_x \rfloor + 1)\right) = \min_{y \in B_\infty(x, r_x)} f(y)$ for $\mathbf{i}_{\max} = \arg\max_{\mathbf{i}} \hat{\mathcal{B}}(\mathbf{i})$, provided that

$$f(y^*) \leq f(z) \Big/ \left(1 + 2\frac{(4r)^{\lceil \log_2 K \rceil} - 1}{4r - 1}(r+1)^2 \xi f(z)\right), \; \forall z \in B_\infty(x, r_x), \, z \neq y^*, \qquad (7)$$

where $y^* = \arg\min_{y \in B_\infty(x, r_x)} f(y)$, and ξ denotes the error between $\mathcal{B}$ and its best approximation of TT-rank r in terms of $\|\cdot\|_\infty$. Note that the inequality (7) holds trivially if $\mathcal{B}$ is exactly of the TT format of rank-r, and (Oseledets, 2010) shows that the $f(y^*)$ can be recovered from $\boxed{\mathcal{O}(KIr)}$ entries from B.

No free lunch: the landscape should be ***low-rank***.

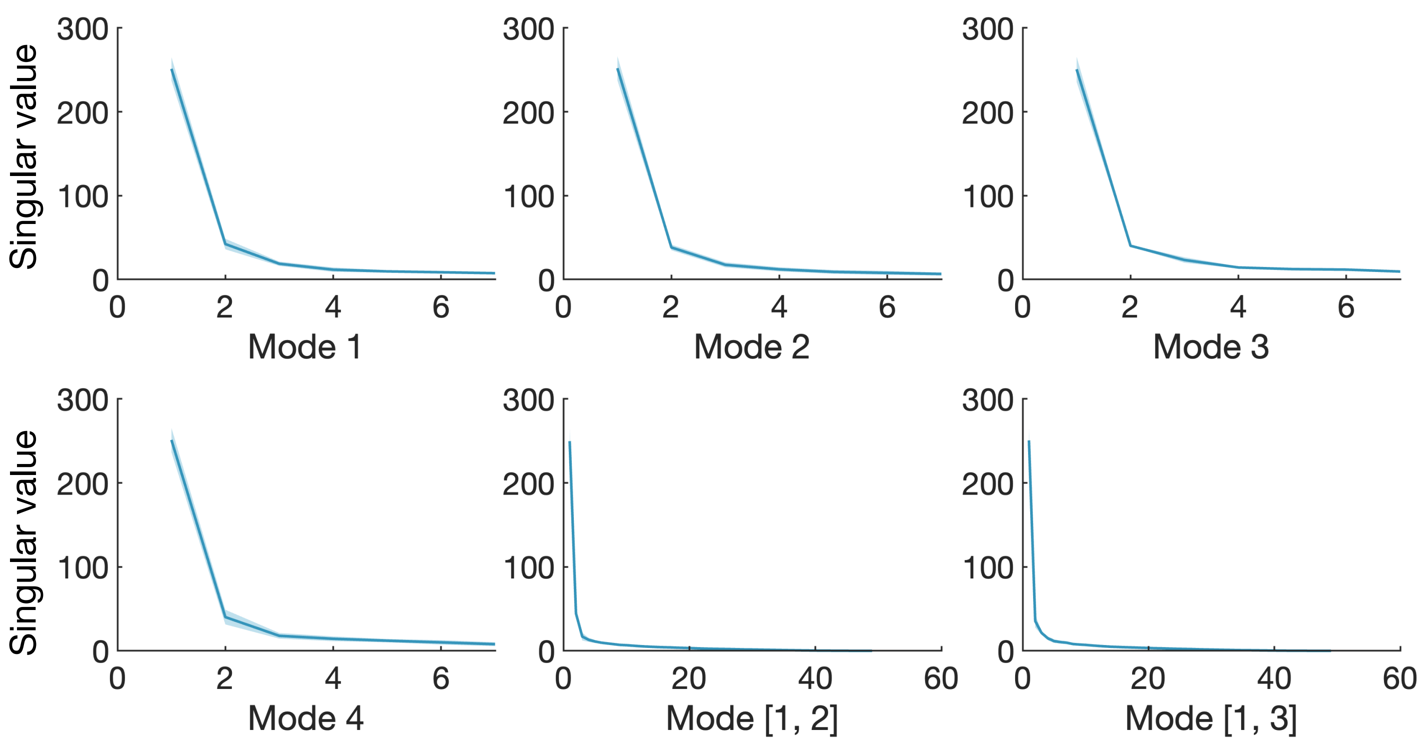
Low-Rank Nature in Landscape

(Li et al., ICML'23)

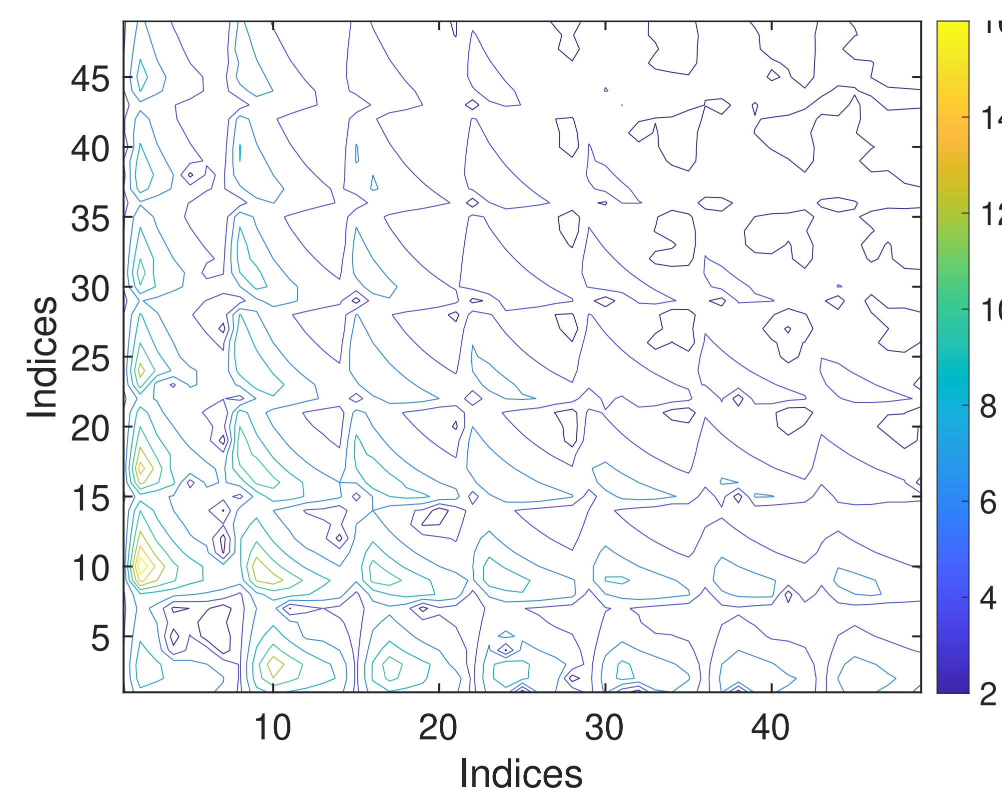
Data: Synthetic tensors in tensor-ring (TR) format.

Setting: order 4~8; mode dimension 3; unknown ranks (in random);

Loss: $F(G, \mathbf{r}) = \underbrace{\frac{1}{\epsilon(G, \mathbf{r})}}_{\text{compression ratio (CR)}} + \lambda \cdot \underbrace{\min_{\mathcal{Z} \in TNS(G, \mathbf{r})} \|\mathcal{X} - \mathcal{Z}\|^2 / \|\mathcal{X}\|^2}_{\text{relative square error (RSE)}},$



Averaged singular values



Landscape visualization

Cont'd: Rank Identification

(Li et al., ICML'23)

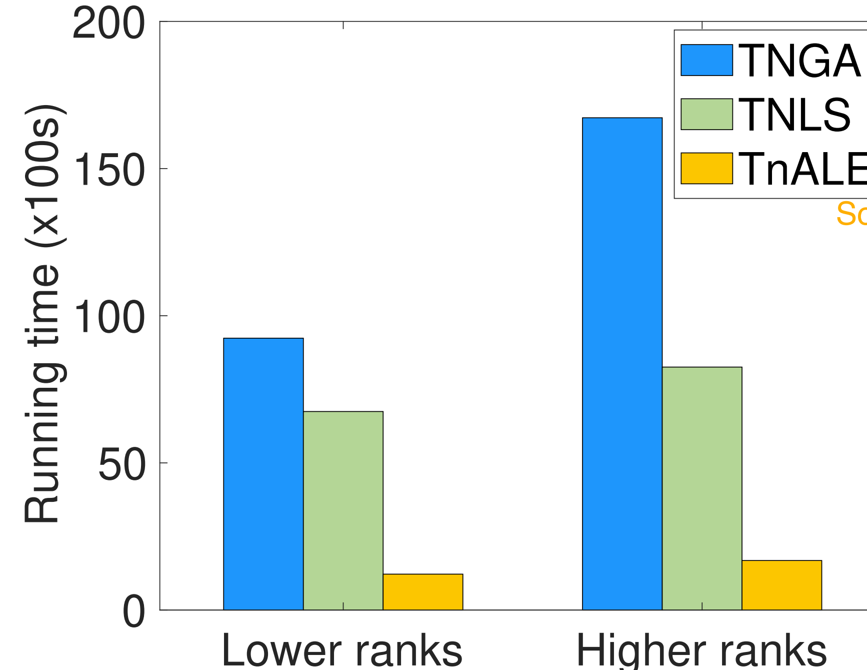
Conditions: order 8; lower-ranks 1~4; higher-ranks 5~8

The ranks are identified if Eff.>=1.

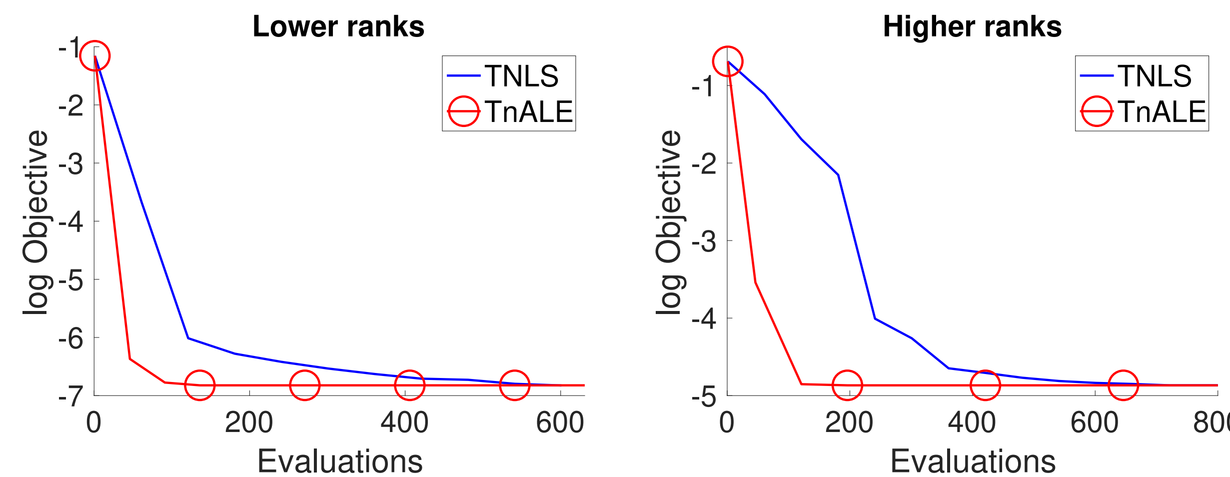
[#Eva.] = Number of evaluations (samples)

Tensor of order 8

Methods	lower-ranks	higher-ranks
Classic rank-selection methods	*Eff.*↑	*Eff.*↑
TR-SVD	0.65±0.46	0.13±0.20
TR-BALS	1.15±0.14	0.19±0.22
TR-LM	1.15±0.14	0.15±0.02
TRAR	0.55±0.10	0.63±0.06
	Eff.↑ *[#Eva.*↓*]*	*Eff.*↑ *[#Eva.*↓*]*
TNGA Solution 1	1.08±0.06 [552]	1.00±0.00 [900]
TNLS Solution 2	1.08±0.06 [492]	1.00±0.00 [588]
TTOpt ($R = 1$)	1.08±0.06 [104]	1.00±0.00 [178]
TTOpt ($R = 2$)	1.02±0.02 [314]	1.00±0.00 [752]
Ours Solution 3	1.08±0.06 [**80**]	1.00±0.00 [**119**]



Running time

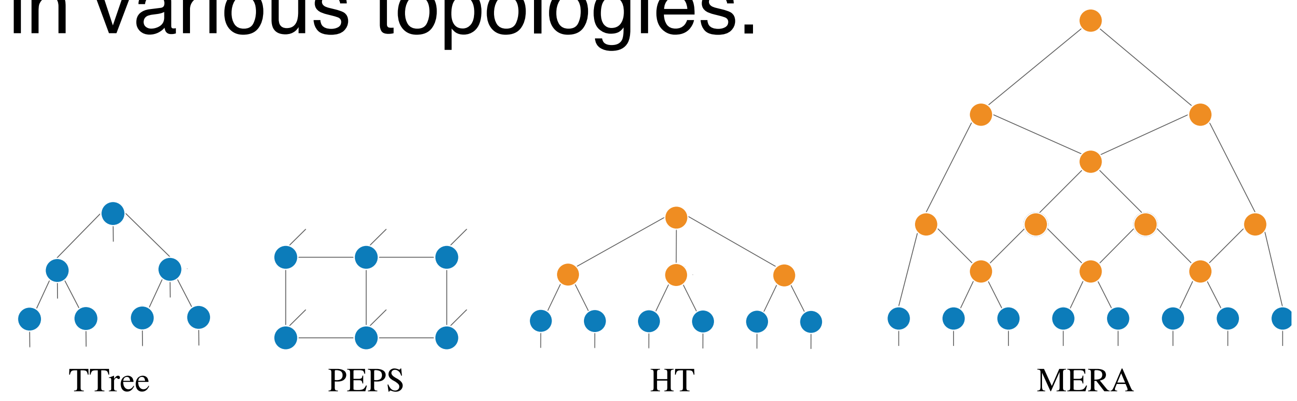


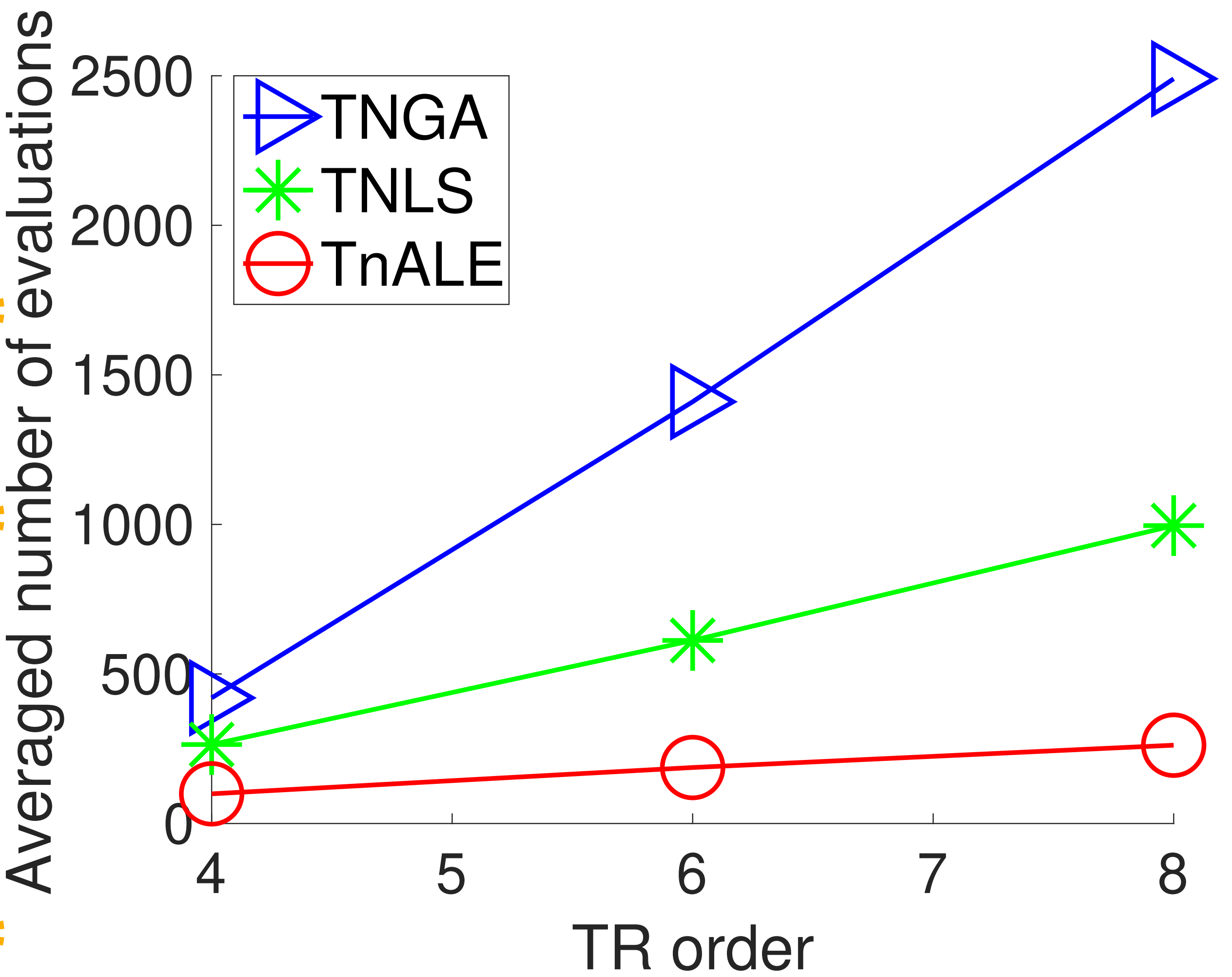
Random sampling vs. ALE

Permutation Search for Various TNs

Goal: How many samples are evaluated to identify the *permutations*?
Data: Synthetic tensors of order four in various topologies.

Topology	Methods	Data – *#Eva.* ↓			
		A	B	C	D
TR	TNGA	2850	2250	3900	1950
	TNLS	1020	960	1320	780
	Ours	**231**	**308**	**308**	**231**
PEPS	TNGA+	1560	-	840	1080
	TNLS	781	781	421	661
	Ours	**407**	**465**	**233**	**175**
HT	TNGA	960	1320	840	1080
	TNLS	841	841	781	721
	Ours	**211**	**281**	**211**	**211**
MERA	TNGA	-	960	2800	3240
	TNLS	1561	841	1441	721
	Ours	**1450**	**484**	**323**	**323**
TW	TNGA	1920	1440	600	720
	TNLS	661	601	601	481
	Ours	**285**	**143**	**285**	**214**

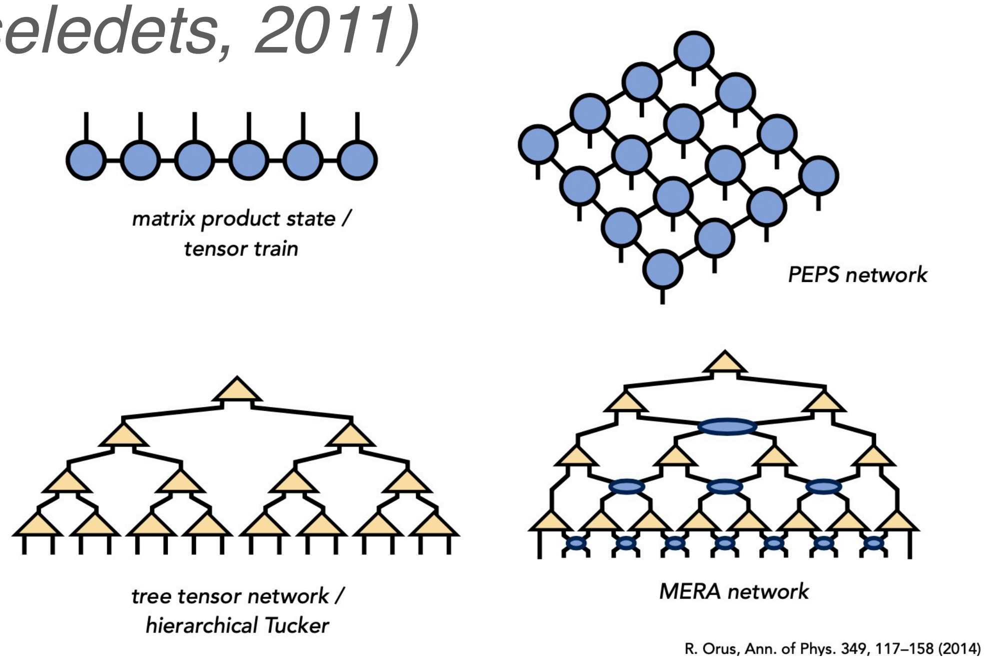


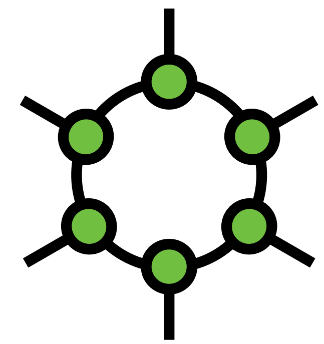


Real-World Data: Topology Search

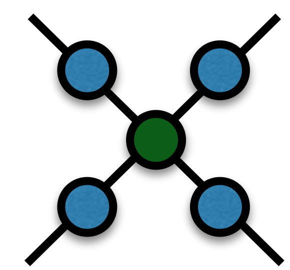
Goal: Search for better TN topologies for natural images using TNGA
Data: 10 images random selected from BSD500 ***(Sheikh et al., 2006)***

(Oseledets, 2011)



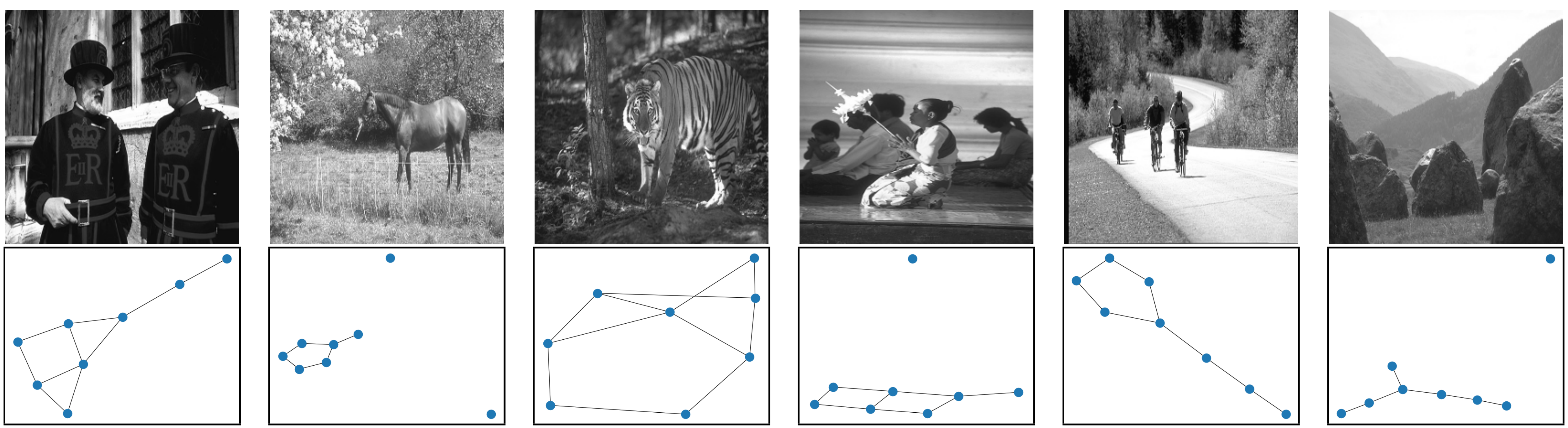


(Zhao et al, 2016)



(Tucker, 1966)

What is the most suitable TN model for our task?



Summary of the Three Algorithms

1. Trade-off between **exploration** and **exploitation**

Global
(e.g., TNGA)

global but slow convergence

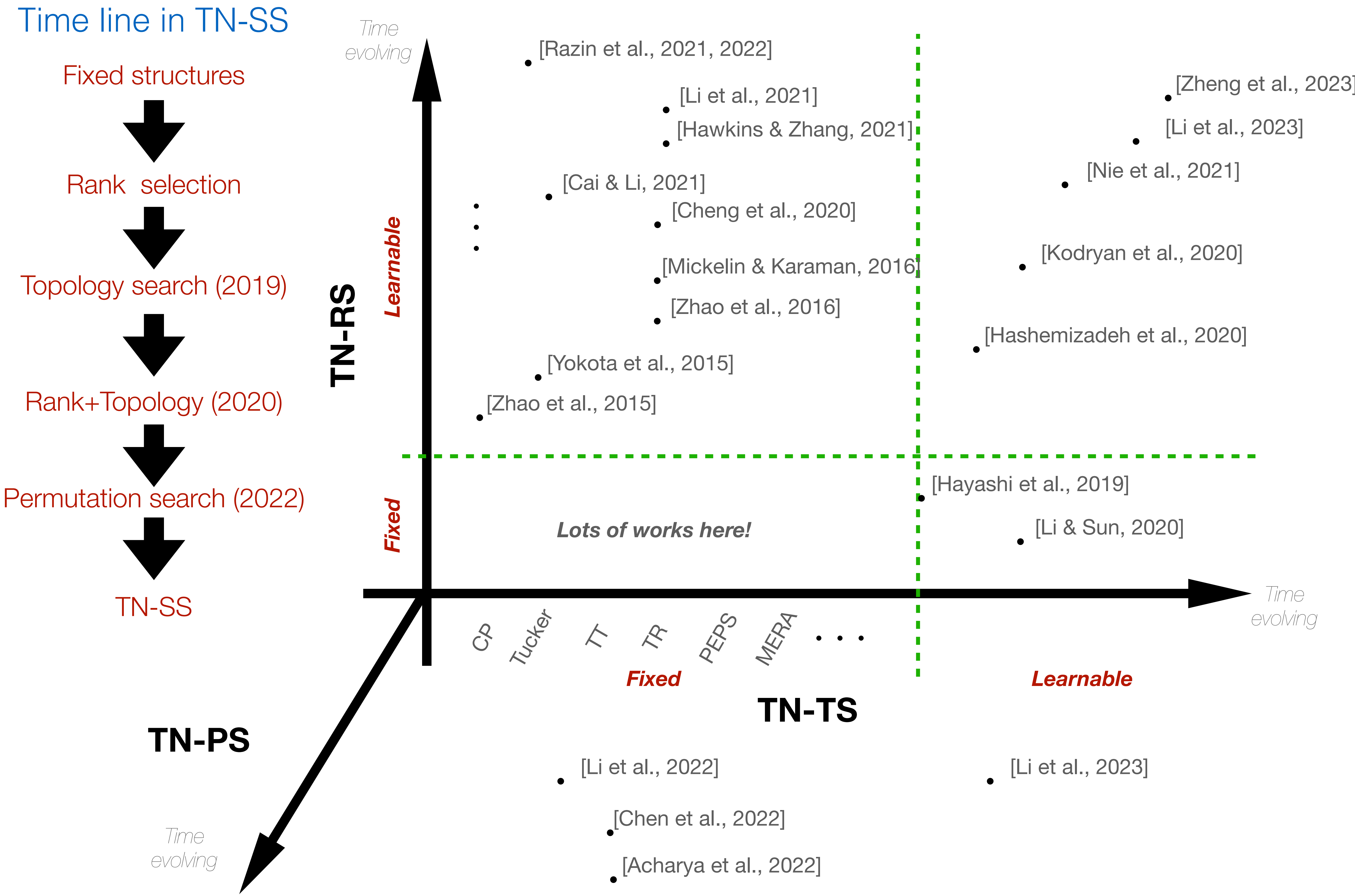
. . .

Local
(e.g., TNLS, TnALE)

local but fast convergence

2. The search acceleration requires additional structural prior to the optimization landscape.

Prior Arts



Time line in TN-SS
Fixed structures
Rank selection
Topology search (2019)
Rank+Topology (2020)
Permutation search (2022)
TN-SS
Time evolving
TN-RS
Learnable
Fixed
[Razin et al., 2021, 2022]
[Li et al., 2021]
[Hawkins & Zhang, 2021]
[Cai & Li, 2021]
[Cheng et al., 2020]
[Mickelin & Karaman, 2016]
[Zhao et al., 2016]
[Yokota et al., 2015]
[Zhao et al., 2015]
[Zheng et al., 2023]
[Li et al., 2023]
[Nie et al., 2021]
[Kodryan et al., 2020]
[Hashemizadeh et al., 2020]
[Hayashi et al., 2019]
[Li & Sun, 2020]
Lots of works here!
Time evolving
CP
Tucker
TT
TR
PEPS
MERA
Fixed
Learnable
TN-TS
TN-PS
[Li et al., 2022]
[Chen et al., 2022]
[Acharya et al., 2022]
[Li et al., 2023]
Time evolving

Techniques in TN-SS

Solving TN-SS in continuous domain

- **Spectrum methods: SVD on unfoldings**
  1. rank search: *(Oseledets, 2011, Zhao et al., 2016, Yin et al., AAAI'22)*;
  2. topology search[3]: *(Nie et al., BMVC'21)*
  3. permutation search: *(Chen et al., arXiv'22)*
- **Regularization-based methods: sparsity/Implicit regularization**
  1. rank search: *(Razin et al., ICML'21,22)*
  2. topology search: *(Kodryan et al., AISTATS'23, Zheng et al., arXiv'23)*
  3. permutation search: *N/A*
- **Bayesian methods: ARD/MGP priors**
  1. rank search: *(Tao and Zhao, IJCAI W'20, Long et al., 2021)*
  2. topology search: *(Zeng et al., ongoing)*
  3. permutation search: *N/A*
- **Discrete optimization: deterministic, stochastic or RL**
  1. rank search: *(Li et al., 2021, Hashemizadel et al., arXiv'20)*
  2. topology search: *(Hayashi et al., Neurips'19, Li and Sun, ICML'20)*
  3. permutation search *(Li et al., ICML'22, 23)*

[3] Most of TN-TS algorithms can solve TN-RS as well.

Comparison within different techniques

	Efficiency	Precision	Flexibility	Scalability	Guarantees
Spectrum	good	bad	bad	good	good
Regularization	good	bad	good	bad	good
Bayesian	bad	good	medium	medium	medium
Discrete opt.	bad	good	good	good	bad

[2] "Flexibility" evaluates if the methods adopt different TNs, tasks, loss, etc..
[3] "Scalability" evaluates if the methods can be deployed for higher-order tensors.
[4] "Guarantees" evaluates if there exist theory on error bounds or convergence, etc..

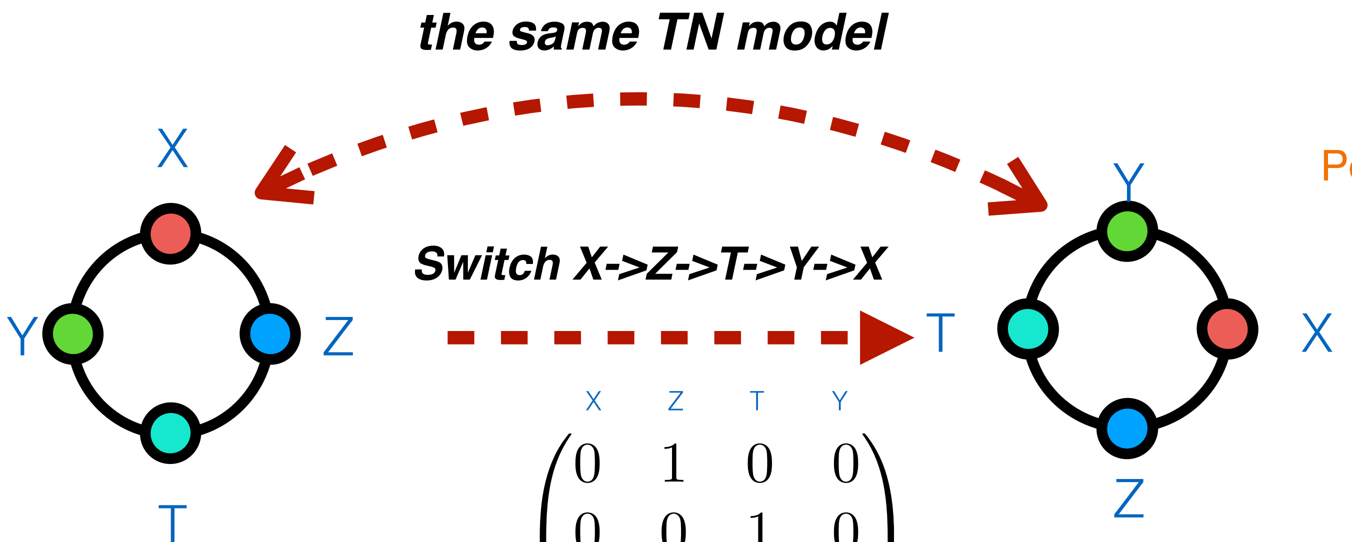
Long-term goal (contributions)

Our works improve the *efficiency* (bad→(medium)→good) and *theoretical understanding* (bad→(medium)→good) for the discrete optimization methods in TN-SS.

Theoretical analysis

- ***Symmetry in TN-SS***
- ***Convergence analysis in TN-SS***

Symmetry in Permutation Search (TN-PS)



$$\mathbf{P} = \begin{array}{c} \begin{array}{cccc} \scriptstyle X & \scriptstyle Z & \scriptstyle T & \scriptstyle Y \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Complexity measure

Permutation

$$\min_{(p,r)\in\mathbb{S}_N\times\mathbb{F}_{G_0}} \phi(pG_0, r),$$

TN topology (cycle)

$$s.t.\ \mathcal{X} \in TNS(pG_0, r)$$

Data

The set of TNs

Identity permutation

$$\Rightarrow \exists p \neq e,\ s.t.\ pG_0 = G_0$$

We can construct a ***smaller*** search space (neighborhood) than $\mathbb{S}_N$

We require a ***quantitative*** analysis tool for this property.

The metric between permutations required to be re-defined.

A Group-Theoretic Framework

Construct the ***isomorphism*** graph set:

TN topology

$$\mathbb{G}_0 = \{G \in \mathbb{G}_N | G \cong G_0\}, \quad (2)$$

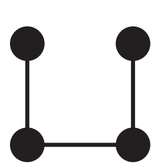
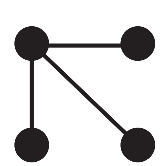
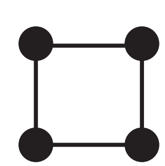
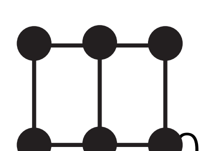
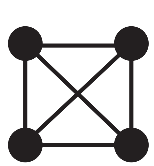
The TN-PS problem can be thus formulated as follows:

$$\min_{(G,r)\in\mathbb{G}_0\times\mathbb{F}_{G_0}} \phi\left(G,r\right), \quad s.t.\, \mathcal{X} \in TNS(G,r). \quad (3)$$

According to the ***Lagrange's theorem*** (in group theory),

Automorphism

$$|\mathbb{S}_N| = |\mathbb{G}_0| \cdot |Aut(G_0)|. \quad (4)$$

TNs	TT	T-Tree	TR	PEPS	CTN
Graphs G_0	Path P_N	Tree T_N	Cycle C_N	Lattice $L_{m,n}$	Complete K_N
$\|V\|$	N	N	N	mn	N
$\|E_0\|$	$N-1$	$N-1$	N	$(m-1)(n-1)$	$N(N-1)/2$
Δ_0	2	$[2, N-1]$	2	$2, 3, 4$	$N-1$
δ_0	1	1	2	2	$N-1$
$\|Aut(G_0)\|$	2	$[2, (N-1)!]$	$2N$	$\leq mn$	$N!$
Examples					

What if arbitrary TN-topology?

Lemma. Upper bound of $|Aut(G_0)|$ for any graph.

Let G_0 be a simple graph of N vertices, and $Aut(G_0)$ be the set containing automorphisms of G_0. Assume that G_0 is connected and its maximum degree Δ satisfies $N/\Delta = d > 1$, then the following inequality holds:

$$|Aut(G_0)| \leq N! \cdot e^{-\gamma(d)\cdot N + \frac{1}{2}\log d + 1/24},$$

where $\gamma(d) = \log d + \frac{1}{d} - 1$ is a positive and monotonically increasing function for $d > 1$.

Theorem. Bounding #TN-structures

Assume G_0 to be a simple and connected graph of N vertices, and $\mathbb{G}_0$ is constructed as (2). Let $\delta = N/d_1$ and $\Delta = N/d_2$, $d_1 \geq d_2 > 1$, be the minimum and maximum degree of G_0, respectively. The size of the search space of (3), written $\mathbb{L}_{G_0,R} := \mathbb{G}_0 \times \mathbb{F}_{G_0,R}$, is bounded as follows:

$$R^{\frac{N^2}{2d_2}} \cdot N! \geq |\mathbb{L}_{G_0,R}| \geq R^{\frac{N^2}{2d_1}} \cdot e^{\gamma(d_2)\cdot N - \frac{1}{2}\log d_2 - 1/24},$$

where $\gamma(d) = \log d + \frac{1}{d} - 1$ is defined as the lemma above.

Distance of Vertex Permutation

Suppose $G_i = g_i G_0$ for $i=1,2$, We construct the following function:

$$d_{G_0}(G_1, G_2) = \min_{p_i \in g_i \cdot Aut(G_0), i=1,2} d_{\mathbb{T}_N}(p_1, p_2), \qquad (7)$$

word metric

Lemma. Metric and neighborhood

Let G_0 be a simple graph and $\mathbb{G}_0$ be the set defined as (2). The function $d_{G_0} : \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{R}$ defined by (7) is a **semi-metric** on $\mathbb{G}_0$. Furthermore, let $\mathbb{I}_d(G)$ be a set constructed as follows:

$$\mathbb{I}_d(G) = \{G' \in \mathbb{G}_0 | G' = q \prod_{i=1}^{d} t_i \cdot G_0, \\ q \in g \cdot Aut(G_0),\ t_i \in \mathbb{T}_N,\ i \in [d]\}.$$

Then $\mathbb{N}_D(G) = \bigcup_{d=0}^{D} \mathbb{I}_d(G)$ is the neighborhood of $G = g \cdot G_0 \in \mathbb{G}_0$ induced by (7), with the radius $D \in \mathbb{Z}^+ \cup \{0\}$.

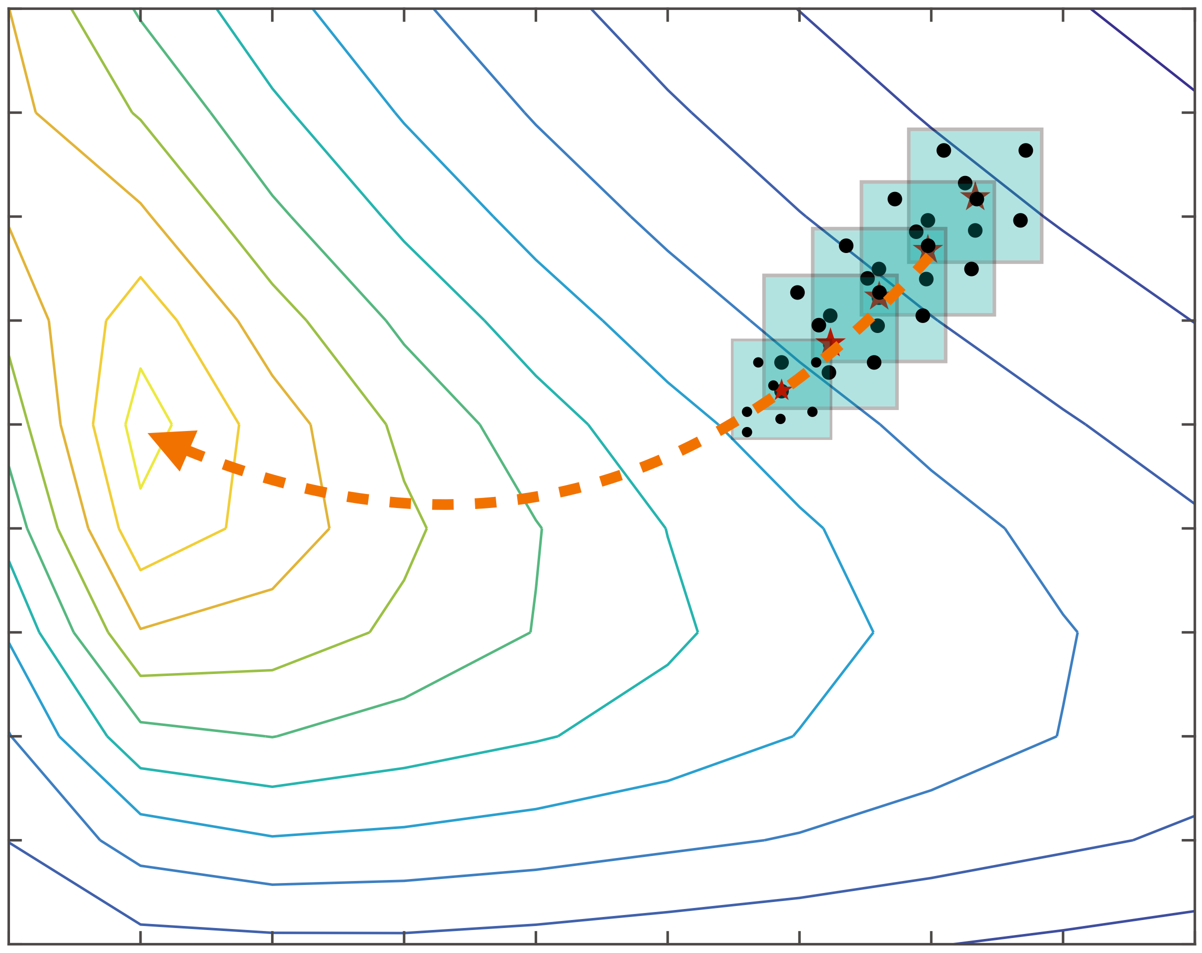
Sampling all possible q is computationally hard (NP-intermediate).

q is omitted in our algorithm, since "***most graphs are not symmetric***".

(C Godsil, GF Royle, 2001)

Searching Dynamic for Local sampling

Local sampling is deployed in both TNLS and TnALE.



The ***convergence*** is unknown due to the discrete nature of the problem.

How do ***sampling strategies*** affect the search efficiency?

Convex analysis in discrete domain

Problem Reformulation

We consider a general form for optimization in TN-SS:

Objective $f : \mathbb{Z}_{\geq 0}^{L} \rightarrow \mathbb{R}_{+}$

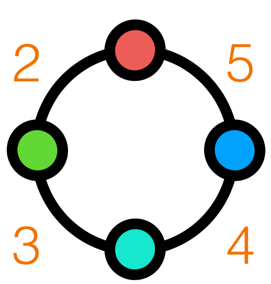
TN-ranks $\mathbf{x} \in \mathbb{Z}_{+}^{K}$

$$\min_{\mathbf{x}\in\mathbb{Z}_{+}^{K},\, p\in\mathbb{P}} f_{p}(\mathbf{x}) := f \circ p(\mathbf{x}),$$

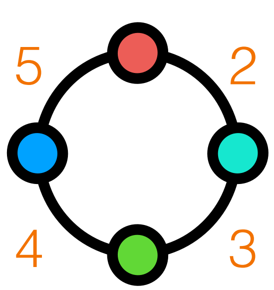
TN-topology (including permutation)

$p : \mathbb{Z}_{+}^{K} \rightarrow \mathbb{Z}_{\geq 0}^{L}$

Any TN-structures (edge-labelled graph) can be represented by $p(\mathbf{x})$.



$$\mathbf{A}_1 = \begin{pmatrix} 0 & 0 & 2 & 3 \\ 0 & 0 & 4 & 5 \\ 2 & 3 & 0 & 0 \\ 4 & 5 & 0 & 0 \end{pmatrix} \Longleftrightarrow p_1(\mathbf{x}) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \end{pmatrix},$$



$$\mathbf{A}_2 = \begin{pmatrix} 0 & 2 & 0 & 5 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 0 & 4 \\ 5 & 0 & 4 & 0 \end{pmatrix} \Longleftrightarrow p_2(\mathbf{x}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 5 \\ 3 \\ 0 \\ 4 \end{pmatrix}$$

Gradient in Discrete Domain

Definition (finite gradient)

For any function $f : \mathbb{Z}_{\geq 0}^{L} \to \mathbb{R}$, its finite gradient $\Delta f : \mathbb{Z}_{\geq 0}^{L} \to \mathbb{R}^{L}$ at the point $\mathbf{x}$ is defined as the vector

$$\Delta f(\mathbf{x}) = \left[f(\mathbf{x} + \mathbf{e}_1) - f(\mathbf{x}), \ldots, f(\mathbf{x} + \mathbf{e}_L) - f(\mathbf{x}) \right]^{\top}, \tag{2}$$

where $\mathbf{e}_i \, \forall i \in [L]$ denote the unit vectors with the i-th entry being one and other entries being zeros.

Finite gradient

Strong convexity

Smoothness

Convexity in Discrete Domain

Definition (α-strong convexity with finite gradient)

We say f is α-strongly convex for $\alpha \geq 0$ if $f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \Delta f(\mathbf{x}) - \frac{\alpha}{2}\mathbf{1}, \mathbf{y} - \mathbf{x} \rangle + \frac{\alpha}{2}\|\mathbf{y} - \mathbf{x}\|^2$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^L_{\geq 0}$, where $\mathbf{1} \in \mathbb{R}^L$ denotes the vector with all entries being one. We simply say f is convex if it is α-strongly convex and $\alpha = 0$.

Lemma

If f is α-strongly convex in $\mathbb{Z}^L_{\geq 0}$, then the following inequalities are held:

1. $g(\mathbf{x}) = f(\mathbf{x}) - \frac{\alpha}{2}\|\mathbf{x}\|^2$ *is convex in the discrete scenario for all* $\mathbf{x} \in \mathbb{Z}^L_{\geq 0}$, *and vice versa;*
2. $\langle \Delta f(\mathbf{x}) - \Delta f(\mathbf{x}), \mathbf{x} - \mathbf{y} \rangle \geq \alpha\|\mathbf{x} - \mathbf{y}\|^2$ *for any* $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^L_{\geq 0}$;
3. $\|\Delta f(\mathbf{x}) - \Delta f(\mathbf{y})\| \geq \alpha\|\mathbf{x} - \mathbf{y}\|$ *for any* $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^L_{\geq 0}$;

Here $\| \cdot \|$ denotes the l_2 norm for vectors.

Smoothness in Discrete Domain

Definition ((β_1, β_2)-smoothness with finite gradient)

We say f is (β_1, β_2)-smooth for $\beta_1, \beta_2 > 0$ if

1. $|f(\mathbf{x}) - f(\mathbf{y})| \leq \beta_1 \|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^L_{\geq 0}$; bound the changing rate of the function
2. The function $l(\mathbf{x}) := \frac{\beta_2}{2}\|\mathbf{x}\|^2 - f(\mathbf{x})$ is convex. bound the changing rate of the gradient

Lemma

If $|f(\mathbf{x}) - f(\mathbf{y})| \leq \beta \|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^L_{\geq 0}$, then the norm of the finite gradient with respective to $\mathbf{x}$ is bounded, i.e., $\|\Delta f(\mathbf{x})\|_\infty \leq \beta$.

Lemma

If $l(\mathbf{x}) = \frac{\beta}{2}\|\mathbf{x}\|^2 - f(\mathbf{x})$ is convex, then for all $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^L_{\geq 0}$

1. $f(\mathbf{y}) \leq f(\mathbf{x}) + \left\langle \Delta f(\mathbf{x}) - \frac{\beta}{2}\mathbf{1}, \mathbf{y} - \mathbf{x} \right\rangle + \frac{\beta}{2}\|\mathbf{y} - \mathbf{x}\|^2$ *and vise versa;*
2. $\langle \Delta f(\mathbf{x}) - \Delta f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \leq \beta \|\mathbf{x} - \mathbf{y}\|^2$.

Properties for Strongly Convex, Smooth Function in Discrete Domain

Lemma (convex combination in the discrete domain)

Suppose $\mathbf{q} = \theta \mathbf{x} + (1-\theta)\mathbf{y}$, $\forall \theta \in [0,1]$, *and there is* $\hat{\mathbf{q}} \in \mathbb{Z}^L_{\geq 0}$ *where* $\boldsymbol{\Lambda} = \mathbf{q} - \hat{\mathbf{q}}$. *If* f *is* α*-strongly convex, then*

$$\theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y}) \geq f(\hat{\mathbf{q}}) + \left\langle \Delta f(\hat{\mathbf{q}}) - \frac{\alpha}{2}\mathbf{1}, \boldsymbol{\Lambda} \right\rangle + \frac{\alpha}{2}\|\boldsymbol{\Lambda}\|^2. \tag{3}$$

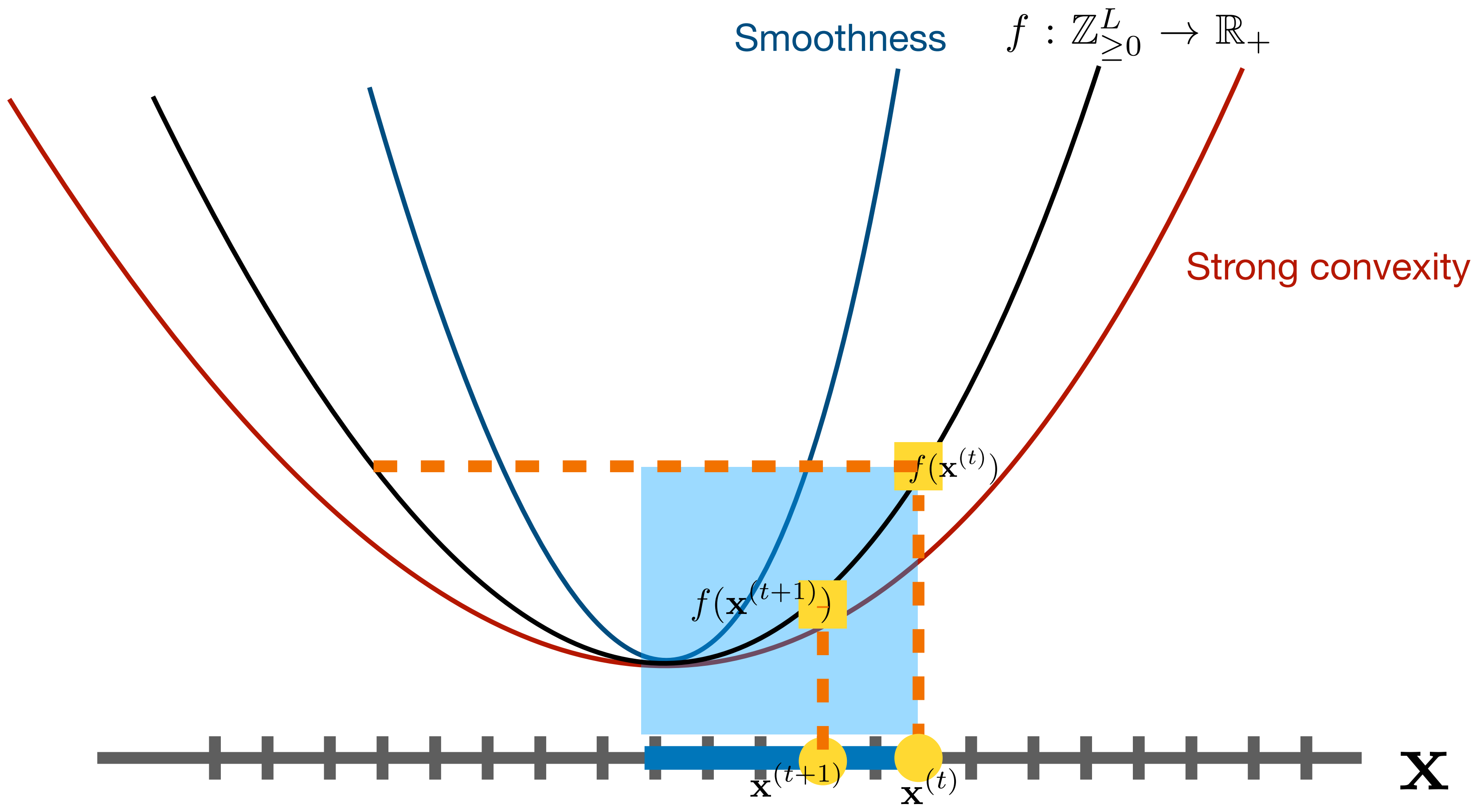
Definition (sub-level set)

The level set of f at point $\mathbf{x} \in \mathbb{Z}^L_{\geq 0}$ is $\mathbb{L}_{\mathbf{x}}(f) = \left\{\mathbf{y} \in \mathbb{Z}^L_{\geq 0} : f(\mathbf{y}) = f(\mathbf{x})\right\}$. The sub-level set of f at point $\mathbf{x} \in \mathbb{Z}^L_{\geq 0}$ is $\mathbb{L}^{\downarrow}_{\mathbf{x}}(f) = \left\{\mathbf{y} \in \mathbb{Z}^L_{\geq 0} : f(\mathbf{y}) \leq f(\mathbf{x})\right\}$.

Lemma (the sub-level cube)

Assume that $f : \mathbb{Z}^L_{\geq 0} \to \mathbb{R}$ *is* α*-strongly convex,* (β_1, β_2)*-smooth, and its minimum, denoted* $f(\mathbf{x}^*)$*, satisfies* $\|\frac{\beta_2}{2}\mathbf{1} - \Delta f(\mathbf{x}^*)\| \leq \gamma$ *where* γ *is a constant and* $0 \leq \gamma < \alpha$*. Then, for all* $\mathbf{x} \in \mathbb{Z}^L_{\geq 0}$*, there is a L-dimensional cube, which is of the edge length* $\frac{2(\alpha-\gamma)}{\beta_2\sqrt{L}}\|\mathbf{x} - \mathbf{x}^*\|$*, tangent at* $\mathbf{x}$*, and inside the sub-level set* $\mathbb{L}^{\downarrow}_{\mathbf{x}}(f)$*.*

Intuition in One-Dimensional Case



Convergence for Strongly Convex, Smooth Function in Discrete Domain

Assumption

Assume that $f : \mathbb{Z}^L_{\geq 0} \to \mathbb{R}_+$ of is α-strongly convex, (β_1, β_2)-smooth, and its minimum, denoted $(p^*, \mathbf{x}^*) = \arg\min_{p,\mathbf{x}} f \circ p(\mathbf{x})$, satisfies $\|\Delta f_{p^*}(\mathbf{x}^*) - \frac{\beta_2}{2}\mathbf{1}\| \leq \gamma$ where $0 \leq \gamma < \alpha \leq \beta_1 \leq \beta_2 \leq 1$.

Theorem (**convergence rate**)

Suppose the Assumption above is satisfied, the operator p [redacted] *is fixed to be p^*, and $0 \leq \theta \leq 1$. Then, for any $\mathbf{x}$ with $\|\mathbf{x} - \mathbf{x}^*\|_\infty \leq c$, we can find a neighborhood $B_\infty(\mathbf{x}, r_\mathbf{x})$ where $r_\mathbf{x} \geq \theta c + \frac{1}{2}$, such that there exist a element $\mathbf{y} \in B_\infty(\mathbf{x}, r_\mathbf{x})$ satisfying*

$$f_{p^*}(\mathbf{y}) - f_{p^*}(\mathbf{x}^*) \leq (1-\theta)(f_{p^*}(\mathbf{x}) - f_{p^*}(\mathbf{x}^*)) + \frac{7}{8}K. \quad (4)$$

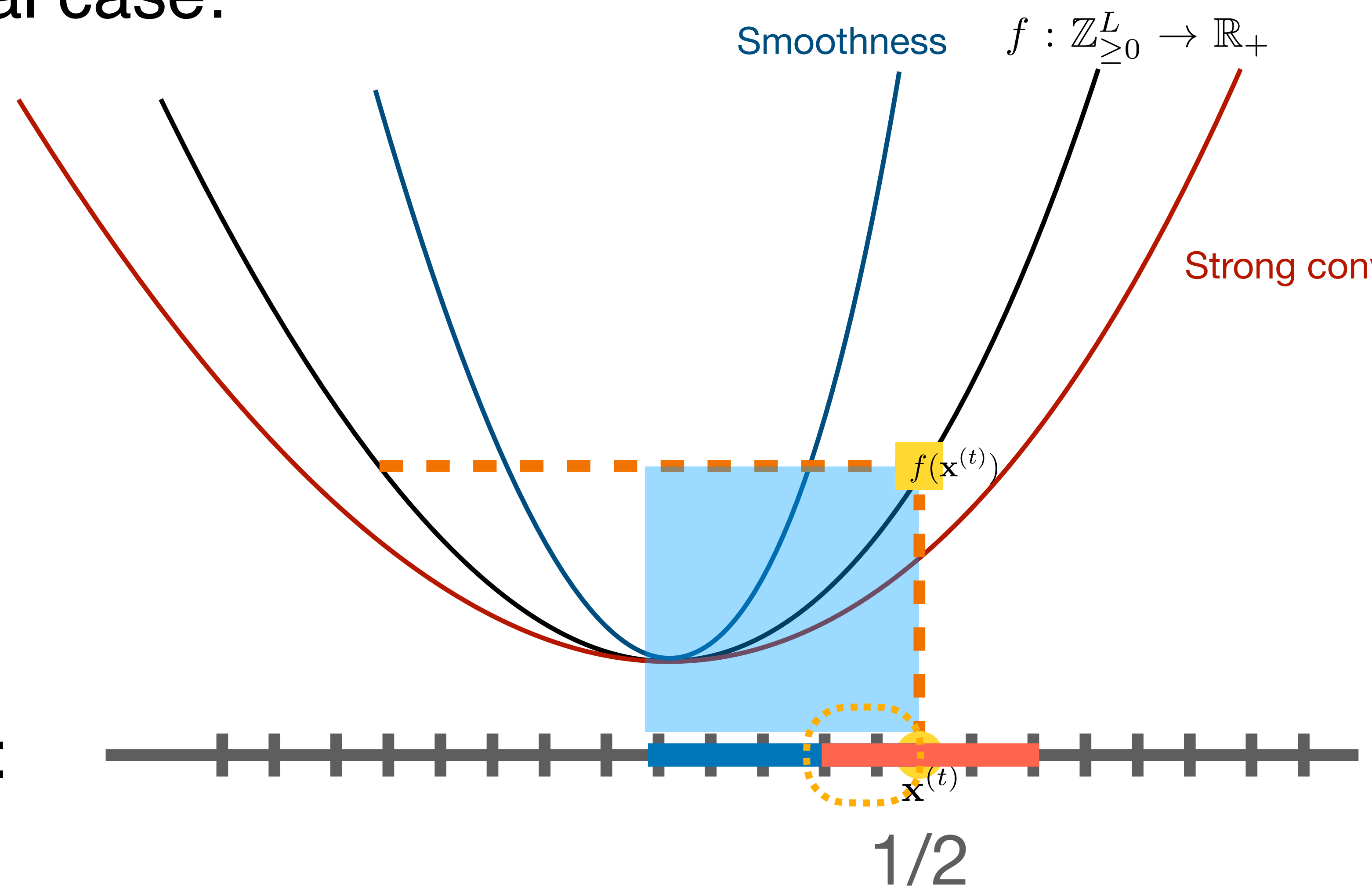
Corollary (**convergence guarantee**)

Suppose p^* is known and a series $\{\mathbf{x}_n\}_{n=0}^\infty$, where $\mathbf{x}_0$ is randomly chosen in $\mathbb{Z}_+^K$, and for each $n > 0$, $\mathbf{x}_n$ is equal to the $\mathbf{y}$ in Theorem 10. Then we can achieve the following limit when $\Omega(1/K) \leq \theta \leq 1$,

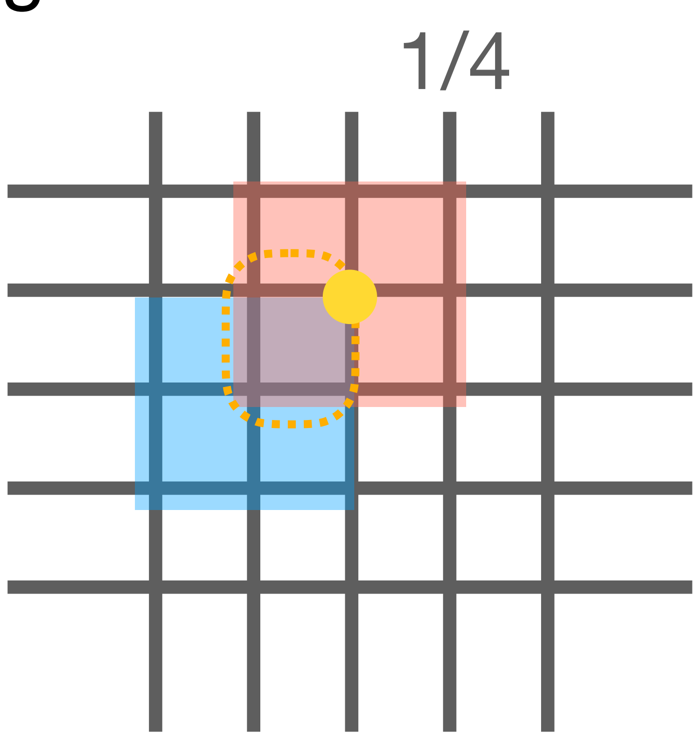
$$\lim_{n\to\infty} (f_{p^*}(\mathbf{x}_n) - f_{p^*}(\mathbf{x}^*)) = O(1) \quad (5)$$

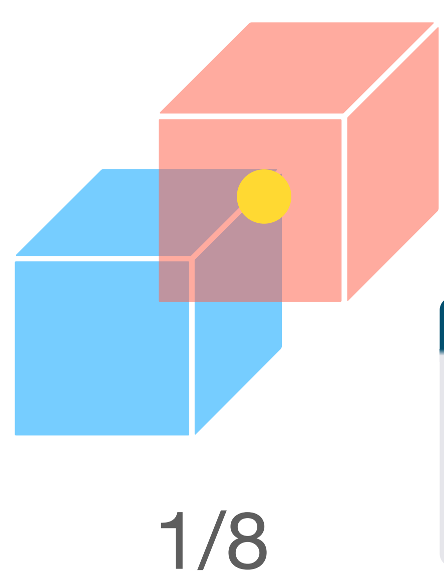
Curse of Dimensionality in TNLS

One-dimensional case:



Higher dimension:





The ratio of the overlapped area gets smaller exponentially with increasing the dimension.

Proposition (curse of dimensionality for TNLS)

Let the assumptions in the Theorem be satisfied. Furthermore, assume that $\mathbf{x}^*$ is sufficiently smaller (or larger) than $\mathbf{x}$ entry-wisely except for a constant number of entries. Then the probability of achieving a suitable $\mathbf{y}$ as mentioned in Theorem 10 by uniformly randomly sampling in $B_\infty(\mathbf{x}, r_\mathbf{x})$ with $r_\mathbf{x} \geq \theta c + \frac{1}{2}$ equals $O(2^{-K})$.

Conclusion

chao.li@riken.jp

- **TN-SS can boost the performance of tensor learning.**
- **TN-SS can be solved by genetic algorithm, stochastic search, and alternating enumeration.**
- **TN-SS algorithms can explore unknown and more efficient tensor networks than the ones proposed in the literature.**

*1. **TNGA**: Li, Chao, and Sun, Zhun. "Evolutionary topology search for tensor network decomposition." International Conference on Machine Learning (ICML). PMLR, 2020.*

*2. **TNLS**: Li, Chao, et al. "Permutation search of tensor network structures via local sampling." International Conference on Machine Learning (ICML). PMLR, 2022.*

*3. **TnALE**: Li, Chao, et al. "Alternating Local Enumeration (TnALE): Solving Tensor Network Structure Search with Fewer Evaluations ." International Conference on Machine Learning (ICML). PMLR, 2023.*

Contributors