Recent Progress on Tensor PCA: From Kikuchi Spectral Algorithm to Tensor Networks

Zhangsong Li

School of Mathematical Sciences, Peking University

June 18, 2026

AMSS Stochastic Analysis Seminar

High-dimensional statistical inference

- Example: spiked Wigner model

$$\boldsymbol{Y} = \lambda \boldsymbol{x}\boldsymbol{x}^\top + \boldsymbol{Z}\,.$$

  $\boldsymbol{x} \in \{-1, +1\}^n$: signal vector ("spike"); $\boldsymbol{Z}$: Gaussian noise matrix.
- Connection to statistical physics: posterior distribution is a Gibbs measure.
- Algorithms: Belief propagation (BP) [Pearl '86]; approximate message passing (AMP) [Donoho-Maleki-Montanari '09].
- Beliefs:
  - Optimality of BP/AMP;
  - "Sharp" threshold;
- This talk: case study on tensor PCA – a problem where beliefs in statistical physics seem to fail (!!!)

Tensor PCA (principle component analysis)

Definition (Spiked tensor model)

$\boldsymbol{x} \in \{-1, +1\}^n$: signal;
$p \in \{2, 3, 4, \ldots\}$: tensor order;
For each subset $U \subset [n]$ with $|U| = p$, observe

$$\boldsymbol{Y}_U = \lambda \prod_{i \in U} \boldsymbol{x}_i + \mathcal{N}(0, 1)\,;$$

$\lambda \geq 0$: signal-to-noise ratio;
Goal: estimate $\boldsymbol{x}$ given $\{\boldsymbol{Y}_U\}$ (with high probability as $n \to \infty$).

- "For every p variables, get a noisy observation of their parity".
- In tensor notation: $\boldsymbol{Y} = \lambda \boldsymbol{x}^{\otimes p} + \boldsymbol{Z}$ where $\boldsymbol{Z}$ is symmetric noise.
- Case $p = 2$ is the spiked Wigner matrix model.

Algorithms for tensor PCA

Maximum Likelihood Estimator (MLE):

$$\mathbb{P}(\boldsymbol{x} \mid \boldsymbol{Y}) \propto \exp\left(\sum_{|U|=p} \lambda \boldsymbol{Y}_U \prod_{i \in U} \boldsymbol{x}_i\right) = \exp\left(\lambda \langle \boldsymbol{Y}, \boldsymbol{x}^{\otimes p} \rangle\right)$$

$$\text{MLE} : \widehat{\boldsymbol{x}} := \arg \max_{v \in \{-1,+1\}^n} \langle \boldsymbol{Y}, v^{\otimes p} \rangle .$$

- Succeeds when $\lambda \overset{\log}{\gg} n^{\frac{1-p}{2}}$ [Richard-Montanari '14].
- Statistically optimal (up to constant factors).
- Problem: requires exponential running time 2^n.

Algorithms for tensor PCA

Local algorithms: keep track of a "guess" $v \in \mathbb{R}^n$ and locally maximize the log-likelihood $\mathcal{L}(v) = \langle \boldsymbol{Y}, v^{\otimes p} \rangle$.

- Gradient descent [Ben Arous-Gheissari-Jagannath '18].
- Tensor power iteration [Richard-Montanari '14].
- Langevin dynamics [Ben Arous-Gheissari-Jagannath '18].
- Approximate message passing (AMP) [Richard-Montanari '14].

These only succeed when $\lambda \overset{\log}{\gg} n^{-\frac{1}{2}}$.

Recall: MLE works for $\lambda \approx n^{-\frac{1-p}{2}}$.

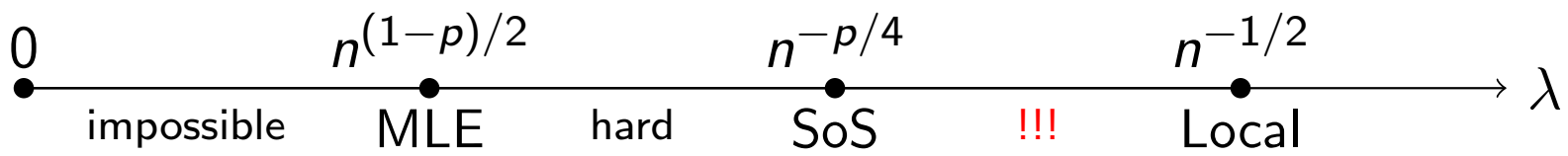
Algorithms for tensor PCA

Sum-of-Squares based method (SoS): Systematic way to obtain convex relaxations of polynomial optimization problems.

- SoS definite program [Hopkins-Shi-Steurer '15].
- Spectral SoS [Hopkins-Shi-Steurer '15, Hopkins-Schramm-Shi-Steurer '15].
- Tensor unfolding [Richard-Montanari '14].

These are poly-time and succeed when $\lambda \overset{\log}{\gg} n^{-\frac{p}{4}}$.

SoS lower bounds suggest no poly-time algorithm when $\lambda \overset{\log}{\ll} n^{-\frac{p}{4}}$.



Local algorithms (gradient descent, AMP...) are suboptimal when $p \geq 3$!

Algorithms for tensor PCA

Kikuchi spectral algorithm: a hierarchy of increasingly powerful algorithms (level ℓ requires running time $n^{O(\ell)}$) [Wein-El Alaoui-Moore '19].

- Refines and "redeems" the statistical physics approach to algorithm design.
- Motivation: Want to understand posterior $\mathbb{P}(\boldsymbol{x} \mid \boldsymbol{Y})$ (a Gibbs measure).
- Find distribution μ over $\{-1, +1\}^n$ minimizing free energy $\mathcal{F}(\mu) = \mathcal{E}(\mu) - \mathcal{S}(\mu)$ (energy−entropy).
- Problem: need exponentially-many parameters to describe μ.
- BP/AMP: just keep track of marginals $m_i = \mathbb{E}[\boldsymbol{x}_i]$ and minimize a proxy, Bethe free energy $\mathcal{B}(m)$.
  - Locally minimize $\mathcal{B}(m)$ via iterative update.

Generalized BP and Kikuchi Free Energy

Recall: BP/AMP keeps track of marginals $m_i = \mathbb{E}[\boldsymbol{x}_i]$ and minimizes Bethe free energy $\mathcal{B}(m)$.

Natural high-order variant:

- Keep track of $m_i = \mathbb{E}[\boldsymbol{x}_i]$, $m_{ij} = \mathbb{E}[\boldsymbol{x}_i \boldsymbol{x}_j]$, ... (up to degree ℓ).
- Minimizes Kikuchi free energy $\mathcal{K}_\ell(m)$ [Kikuchi '51].
- Algorithm: compute the bottom eigenvector of Hessian of $\mathcal{K}(m)$ with respect to moments $m = \{m_i, m_{ij}, \ldots\}$.
- Intuition: best direction of local (up to order-ℓ interactions) improvement.
- This talk: even $p \geq 4$ only (for simplicity).

Kikuchi spectral algorithm

Definition (Kikuchi matrix)

Input: an order-p tensor $\boldsymbol{Y} = (\boldsymbol{Y}_U)_{|U|=p}$ (with p even) and an integer ℓ in the range $p/2 \le \ell \le n - p/2$. Define the $\binom{n}{\ell} \times \binom{n}{\ell}$ matrix (indexed by ℓ-subsets of $[n]$)

$$M_{S,T} := \begin{cases} Y_{S \triangle T}\,, & |S \triangle T| = p\,; \\ 0\,, & \text{otherwise}\,. \end{cases}$$

- Approximately a submatrix of the Kikuchi Hessian.
- Algorithm: compute the top eigenpair of M.
- Runtime: $n^{O(\ell)}$.
- The case $\ell = p/2$ is "tensor unfolding", which is poly-time and succeeds up to the SoS threshold (up to poly($\log n$)).

Analysis of the algorithm

Simplest statistical task: detection;

- Distinguish between $\lambda = \overline{\lambda}$ (spiked) and $\lambda = 0$ (noise).

Algorithm: given $\boldsymbol{Y}$, build matrix $M_{S,T} = \mathbf{1}_{|S \triangle T| = p} \boldsymbol{Y}_{S \triangle T}$, threshold maximum eigenvalue.

Split M into two matrices $M^{(1)}_{S,T} = \lambda \mathbf{1}_{|S \triangle T| = p} \prod_{i \in S \triangle T} \boldsymbol{x}_i$ (the signal part) and $M^{(2)}_{S,T} = \mathbf{1}_{|S \triangle T| = p} \boldsymbol{W}_{S \triangle T}$ (the noise part).

- Observation: the signal part has certain algebraic structure, so $\|M^{(1)}\|$ can be explicitly calculated: $\|M^{(1)}\| = \lambda \binom{n-\ell}{p/2} \binom{\ell}{p/2}$.
- Key challenge: bound spectral norm $\|M^{(2)}\|$.

From detection to recovery: standard matrix perturbation bounds.

Spectral norm of the noise matrix

Idea of [Wein-El Alaoui-Moore '19]: apply matrix Chernoff bound [Oliveira '10, Tropp '10] to $M^{(2)}_{S,T} = \mathbf{1}_{|S \triangle T| = p} \boldsymbol{W}_{S \triangle T}$.

Theorem (Matrix Chernoff bound)

Let $M = \sum_i z_i A_i$ where $z_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ and $\{A_i\}$ is a sequence of $d \times d$ symmetric matrices. Then, for all $t \geq 0$

$$\mathbb{P}(\|M\| \geq t) \leq 2de^{-t^2/\sigma^2} \text{ where } \sigma^2 = \left\| \sum_i A_i^2 \right\|.$$

- Recall that $\|M^{(1)}\| = \lambda d_\ell$ where $d_\ell = \binom{n-\ell}{p/2}\binom{\ell}{p/2}$.
- Applying it to $M^{(2)}$ yields $\mathbb{P}(\|M^{(2)}\| \geq \frac{1}{2}\lambda d_\ell) \leq 2n^\ell e^{-\lambda^2/d_\ell}$.
- Conclusion: Succeeds when $\lambda \gg n^{-\frac{p}{4}}/\sqrt{\log n}$.

Sharp norm bounds?

- The poly$(\log n)$ factor in the analysis seems unnecessary.
- Conjecture [Wein-El Alaoui-Moore '19, Bandeira '24]: Actually $\|M^{(2)}\| \le O_p(1) \cdot n^{\frac{p}{4}} \ell^{\frac{p+2}{4}}$.
- Would imply the ℓ-order Kikuchi algorithm succeeds when $\lambda \ge O_p(1) \cdot n^{-\frac{p}{4}} \ell^{\frac{2-p}{4}}$.
  - Smooth phase transition: increase ℓ (computational cost) $\longrightarrow$ decrease λ (critical signal-to-noise ratio needed).
  - Explains the sub-optimality of BP/AMP: optimal algorithm cannot have near-linear running time.
  - Crucial for the quartic speedup phenomenon in quantum computing [Schmidhuber-O'Donnell-Kothari-Babbush '24].
  - Evidence within low-degree polynomials [Bandeira-Kunisky-Wein '22] suggests it to be optimal among all efficient algorithms.
- Extends to $\ell = n^{1-\Omega(1)}$ (sub-exponential time algorithms).

Proving sharp norm bounds

Conjecture: $\|M^{(2)}\| \leq O_p(1) \cdot n^{\frac{p}{4}} \ell^{\frac{p+2}{4}}$.

Attempt 1: the dimension-dependent factor in matrix Chernoff bound (or equivalently, non-commutative Khintchine's inequality) can be removed when the matrices A_i are "sufficiently non-commutative".

- [Bandeira-Boedihardjo-van Handel '23, Bandeira-Cipolloni-Schöder-van Handel '24+]: attack the conjecture via free probability.
- Result: prove the conjecture for $\frac{p}{2} \leq \ell \leq \frac{3p}{4}$.
- Advantage: sharp norm bounds; can be applied beyond the case of Kikuchi matrix.
- Limitation: has intrinsic difficulty toward the case $\ell \gg p$.
  - [Kothari-Xu '25]: the Kikuchi matrices are **NOT** "intrinsic free" when $\ell \gg p$.

Proving sharp norm bounds

Conjecture: $\|M^{(2)}\| \le O_p(1) \cdot n^{\frac{p}{4}} \ell^{\frac{p+2}{4}}$.

Attempt 2: [Kothari-Xu '25]: directly apply trace method

$$\|M^{(2)}\|^{2q} \le \mathrm{tr}\left((M^{(2)})^{2q}\right).$$

Theorem (Kothari-Xu '25)

For any $\ell, q = n^{o(1)}$, we have

$$\mathbb{E}\left[\mathrm{tr}\left((M^{(2)})^{2q}\right)\right] \le n^{\ell} \cdot \left(O_p(1) \cdot n^{\frac{p}{4}} \ell^{\frac{p+2}{4}}\right)^{2q}.$$

- Thus solves the conjecture for all $\ell = n^{o(1)}$.

Matrix trace calculation

Recall that $M^{(2)}_{S,T} = \mathbf{1}_{|S \triangle T| = p} \boldsymbol{W}_{S \triangle T}$.

$$\mathbb{E}\left[\mathrm{tr}\left((M^{(2)})^{2q}\right)\right] = \sum_{\substack{S_1,\ldots,S_{2q} \\ |S_i \triangle S_{i+1}| = p}} \mathbb{E}\left[\prod_{1 \le i \le 2q} \boldsymbol{W}_{S_i \triangle S_{i+1}}\right]. \quad (\star)$$

- From Gaussianity, $(\star)$ cancels out unless $\{S_i : 1 \le i \le 2q\}$ are "admissible": $\{S_i \triangle S_{i+1} : 1 \le i \le 2q\} = \{A_1, A_1, \ldots, A_q, A_q\}$.
- Equivalent view: First choose S_1 (n^ℓ choices), then choose "leaving vertices" $L_i = S_i \setminus S_{i+1}$ and "coming vertices" $C_i = S_{i+1} \setminus S_i$. Vertices in $S_i \cap S_{i+1}$ are "dormant".
- [Kothari-Xu '25]: given S_1, the choices of $(L_1, C_1; \ldots, L_{2q}, C_{2q})$ are at most $(O_p(1) \cdot n^{\frac{p}{4}} \ell^{\frac{p+2}{4}})^{2q}$.

Our contribution: other algorithms?

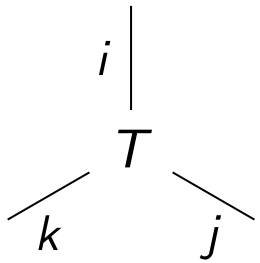
Question: is there a way to find (a family of) algorithms for tensor PCA that are:

- easier to analyze than the Kikuchi spectral algorithm;
- achieves same statistical guarantee (smooth transition);
- easier to generalize?

It turns out there is a systematic way to solve tensor based problems using tensor networks.

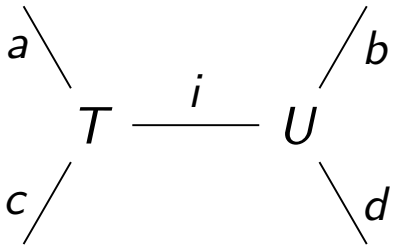
Tensor network notation

A graphical representation for tensors:



$$\Leftrightarrow \qquad T = (T_{i,j,k})$$

Two (or more) tensors can be attached by contracting indices:



$$\Leftrightarrow \qquad B = (B_{a,b,c,d})$$
$$B_{a,b,c,d} = \sum_i T_{a,c,i}\, U_{b,d,i}$$

Rule: sum over "fully connected" indices (in this case, i).

Algorithms from tensor networks

General framework for solving tensor problems:

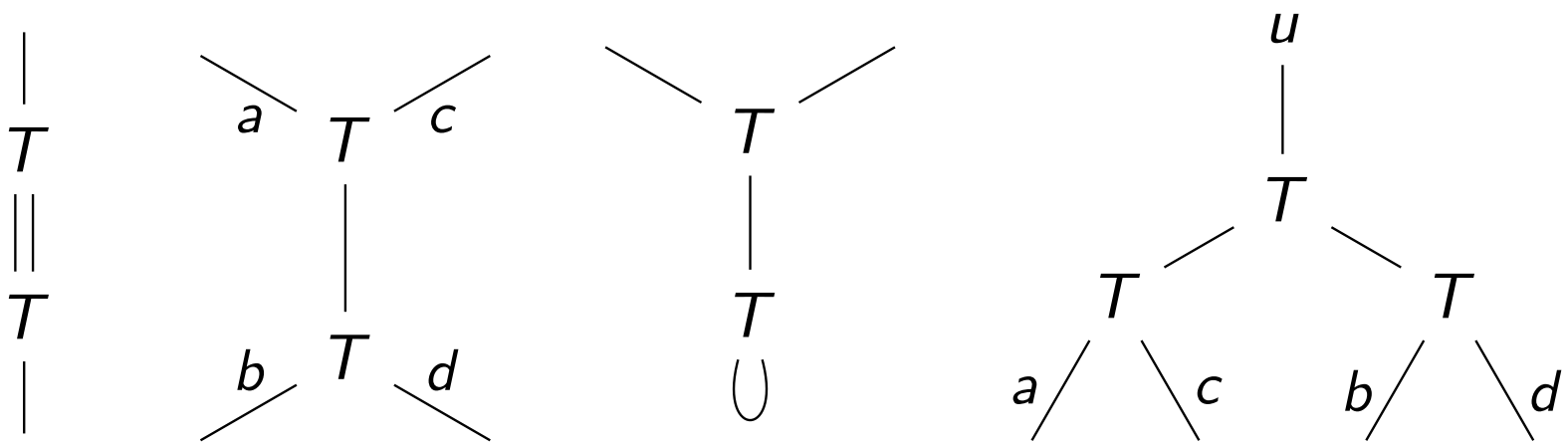
1. Given input tensor $\boldsymbol{T}$.

2. Build a matrix $\boldsymbol{B}$ by connecting copies of $\boldsymbol{T}$ in a tensor network (and flattening).

3. Compute the leading eigenvector of $\boldsymbol{B}$.

Previous tensor networks

Prior work has (implicitly) used this framework:



- [Richard-Montanari '14, Hopkins-Shi-Steurer '15]: tensor unfolding.
- [Hopkins-Shi-Steurer '15]: spectral SoS.
- [Hopkins-Schramm-Shi-Steurer '16]: spectral SoS with partial trace.
- [Hopkins-Schramm-Shi-Steurer '16]: spectral tensor decomposition.

Our results

Theorem (L. 25+)

Consider the order-p tensor-PCA $\boldsymbol{Y} = \lambda \boldsymbol{x}^{\otimes p} + \boldsymbol{W}$ *for* $p \geq 3$. *Suppose* $\lambda = \ell^{\frac{p+2}{4}} n^{-\frac{p}{4}}$ *for some* $\ell = \Omega(1)$. *Then there exists a family of tensor networks* $\mathcal{N}$ *such that the leading eigenvector of*

$$\sum_{B \in \mathcal{N}} B(\boldsymbol{Y}) \qquad (\star)$$

has non-vanishing correlation with $\boldsymbol{x}\boldsymbol{x}^\top$. *In addition,* $(\star)$ *can be calculated in time* $n^{O(\ell)}$.

- Achieves same theoretical guarantee as the Kikuchi algorithm.

Construction of tensor networks

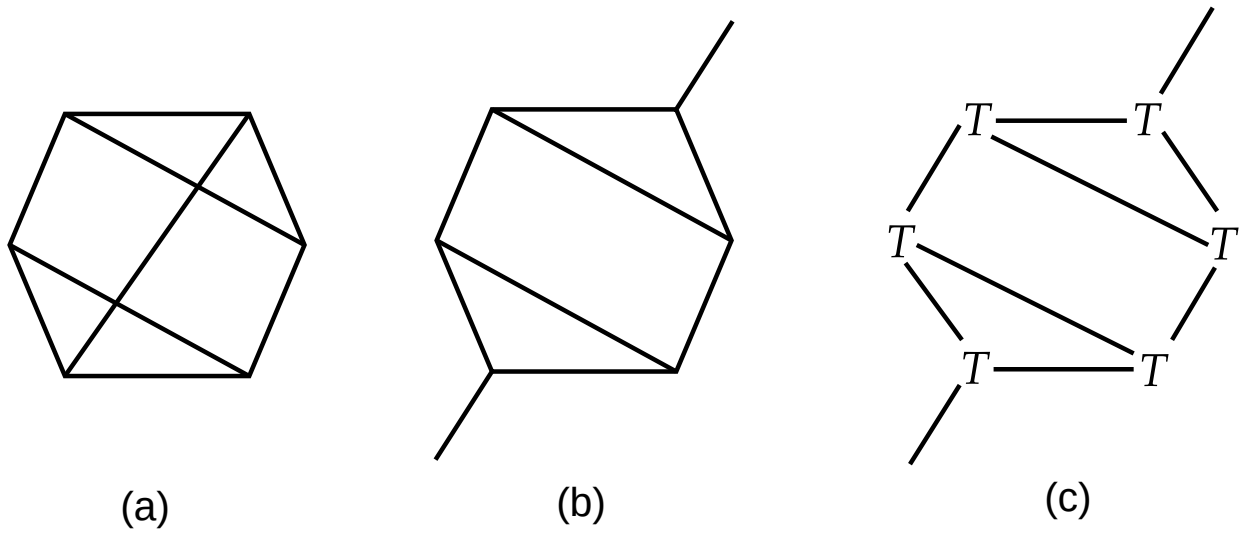
Desired properties of our tensor network $B \in \mathcal{N}$:

- $B_{i,j}$ should correlate with $\boldsymbol{x}_i \boldsymbol{x}_j$.
  - B should have two legs.
  - $B_{i,j}$ should "compress" to $\boldsymbol{x}_i \boldsymbol{x}_j$ if we assign $\boldsymbol{x}_a \boldsymbol{x}_b \boldsymbol{x}_c$ to each $T_{a,b,c}$.
- Should be "as large as possible".
  - Intuition: maximize the enumeration of the number of copies.
  - Each edge attaches two tensors.
- Should be "rich enough".
  - Intuition: different tensor networks B, B' are "almost uncorrelated".
  - Use the averaging effect when summing up different tensor networks.
- Can be enumerated in polynomial-time.
  - [Gottlob-Leone-Scarcello '02] Such tensor network should have bounded fractional tree-width.

Construction of tensor networks

Our construction: Let $m = O(\ell)$, first construct "gadget" tensor networks of size m.

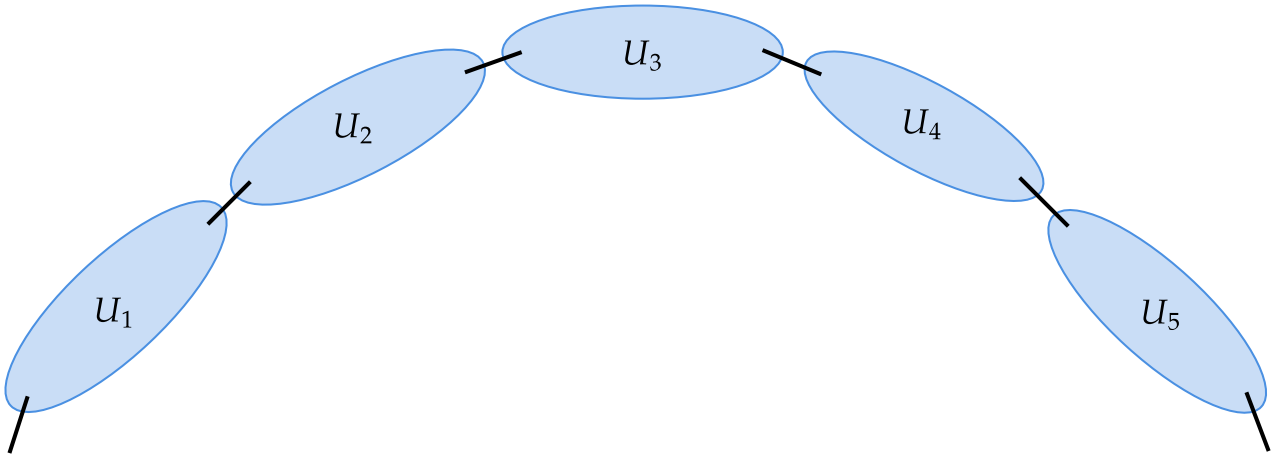
- Choose a 3-regular graph on m vertices (see (a)).
- Remove one arbitrary edge and replace it with two "half edges" (see (b)).
- Place a tensor notation T in each vertex (see (c))



(a) (b) (c)

Construction of tensor networks

Our construction: Let $w = O(\log n)$, we will finally combine w gadget tensor networks into a larger tensor network.



- Get a family of tensor networks $\mathcal{N}$; able to show that $\sum_{B \in \mathcal{N}} B(\boldsymbol{Y})$ correlates with $\boldsymbol{x}\boldsymbol{x}^\top$.
- The tensor networks in $\mathcal{N}$ has tree width bounded by $O(\ell) \Longrightarrow$ can be enumerated in time $n^{O(\ell)}$.

Summary and perspectives

- Local algorithms are suboptimal for tensor PCA. Reasons:
  - smooth transition $\Longrightarrow$ optimal algorithm cannot be nearly-linear time;
  - guess: for p-way data, need p-way algorithm?
- "Redemption" for local algorithms and AMP:
  - keep track of beliefs about higher-order correlations;
  - minimize Kikuchi free energy.
- A systematic way of solving tensor based problems: tensor networks.
  - matches the smooth transition given by Kikuchi algorithm;
  - appears to have broader applications beyond tensor PCA. e.g., refuting noisy k-XOR [Mao '26+]; learning Gaussian single index models (?); tensor completion (?) ...
- Future directions:
  - Unify different approaches?
  - Systematically predict/explain optimality of local algorithms?

Thanks!