# Complexity of slightly positive tensor networks

CCQ Tensor Meeting 04/30/2025

Chris (Jielun) Chen
(Caltech)

Jiaqing Jiang
(Caltech)

Dominik Hangleiter
(Simons Institute)

Norbert Schuch
(University of Vienna)

Based on:

Sign problem in tensor network contraction (arXiv:2404.19023)

Positive bias makes tensor network contraction tractable (arXiv:2410.05414)

Caltech SIMONS INSTITUTE for the Theory of Computing universität wien

# Sign problem and tensor networks

In quantum Monte Carlo (QMC) simulations, especially for fermions, "sign problem" exponentially increases the number of samples needed.

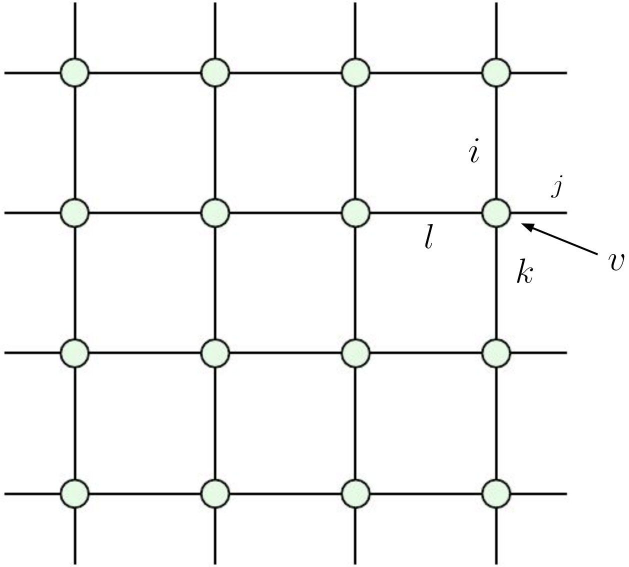
It has often been narrated that tensor networks can circumvent the sign problem, since by construction they do not depend on local basis choices.

**We try to understand this aspect rigorously by studying:**

**Random tensor networks with controlled sign structure.**

# Tensor networks (TNs)

**2D square-lattice graph with n vertices**

$\mathcal{T} =$ 


**Edges:** $i \in \{1, ..., d\}$ , d is called bond-dimension

**Vertices:** $M^v_{ijkl}$ , order-4 tensor

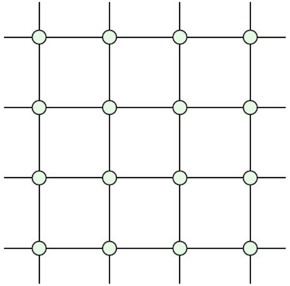
**Edge labeling:** an assignment of values to all edges

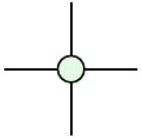
e.g. $c = (..., i = 3, j = 2, k = 1, l = 1, ...)$

**Contraction of a TN yields a number:**

$$\mathcal{C}(\mathcal{T}) = \sum_{\text{edge labeling } c} \prod_{v} M^v_c$$

# The problems we study





$\sim \mathcal{N}(\mu, 1)$

i.i.d. for all entries & all tensors.

Contracting a tensor network with random entries. Larger mean → more positive.

- Is the contraction easier when the mean increases?
- If yes, when does the contraction become easy/hard?

# Part I: Sign problem in tensor network contraction (arXiv:2404.19023)

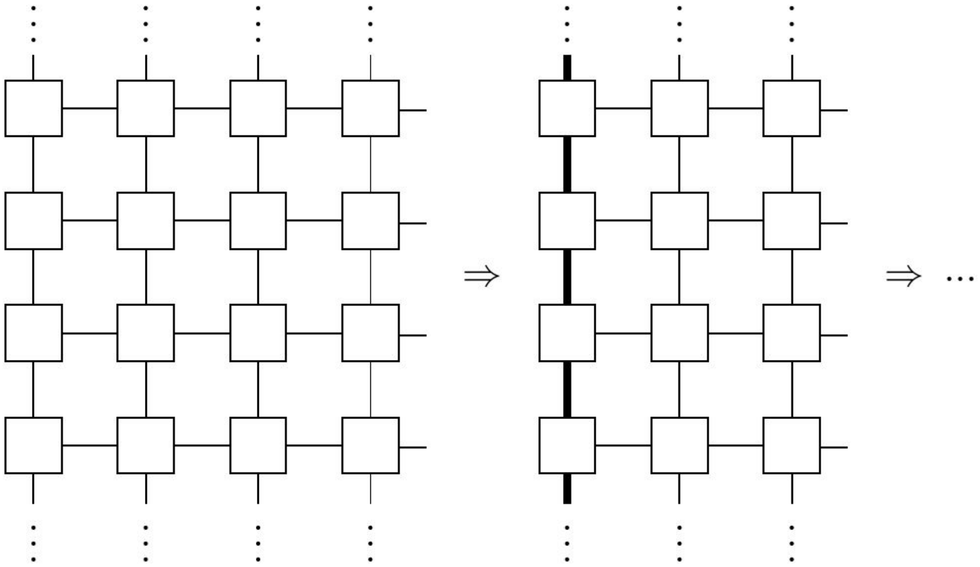
We predicted the complexity **transition point** through an effective classical statistical model, and numerically verified the transition.

Transition point: **mean = 1/bond-dimension**

This transition happens **much earlier** than "sign problem" disappears.
In other words, the entries only need to be **slightly positive**.

# Effective stats model

## Contractability ≈ entanglement in the TN



Increased bond-dimension

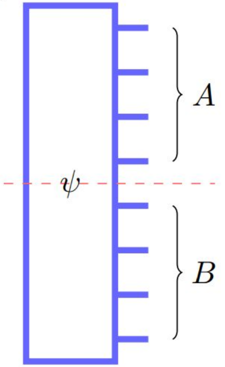
# Effective stats model

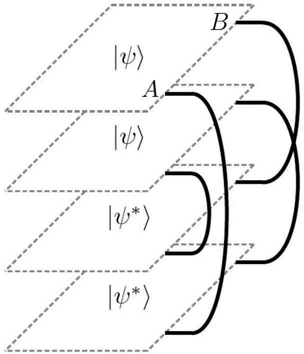
**Contractability ≈ entanglement in the TN ⇔ Average of copies of random TNs**

Renyi-2 entropy

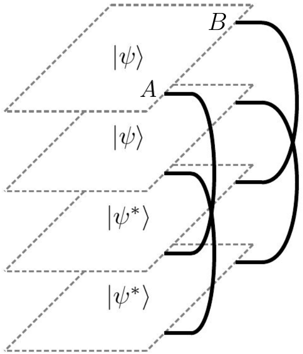
$$\mathbb{E}\left[-\log(\rho_A^2)\right] \approx -\log\left(\mathbb{E}\left[\frac{Tr\left(Tr_B\left(|\psi\rangle\langle\psi|\right)^2\right)}{|\langle\psi|\psi\rangle|^2}\right]\right)$$

$$\approx -\log\left(\mathbb{E}\left[Tr\left(Tr_B\left(|\psi\rangle\langle\psi|\right)^2\right)\right]\right) + \log\left(\mathbb{E}\left[|\langle\psi|\psi\rangle|^2\right]\right)$$




“Mixed”
boundary condition




“Fixed”
boundary condition

# Effective stats model

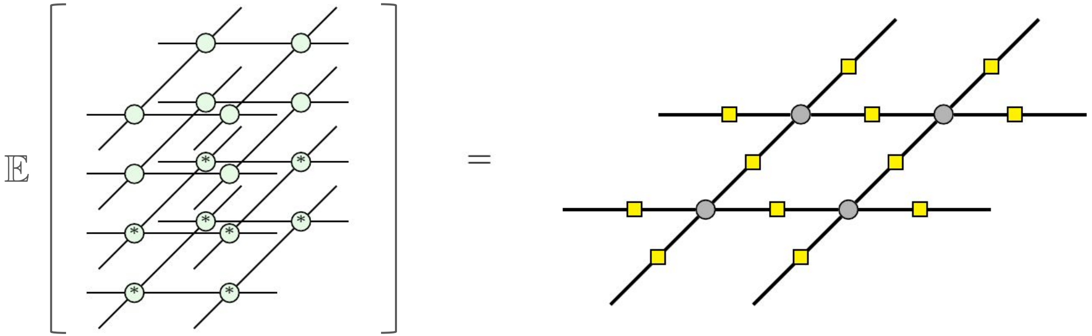
**Contractability ≈ entanglement in the TN ⇔ Average of copies of random TNs**

**⇕**

**Partition function of classical stats model**

Average of copies of random tensors can be computed analytically.

E.g. Isserlis/Wick's theorem



Mean/positivity

⇕

Local field

Partition function of a **ferromagnetic** Potts model with external field

$$\sum_{\{\sigma^{(s)}\}} e^{-\sum_s h(\sigma^{(s)}) - \sum_{\langle s,s'\rangle} k(\sigma^{(s)},\sigma^{(s')})}$$

# Effective stats model

**Contractability ≈ entanglement in the TN ⇔ Average of copies of random TNs**

⇕

**Phase transition point:** $\mu d = 1$ ⇔ **Partition function of classical stats model**

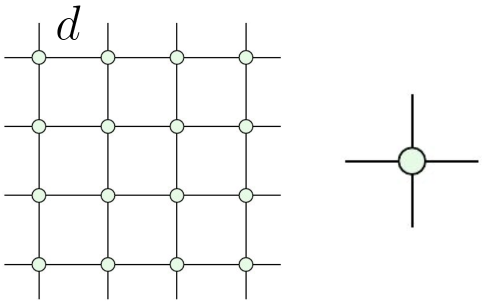
$$\mu d = 1$$

⇕

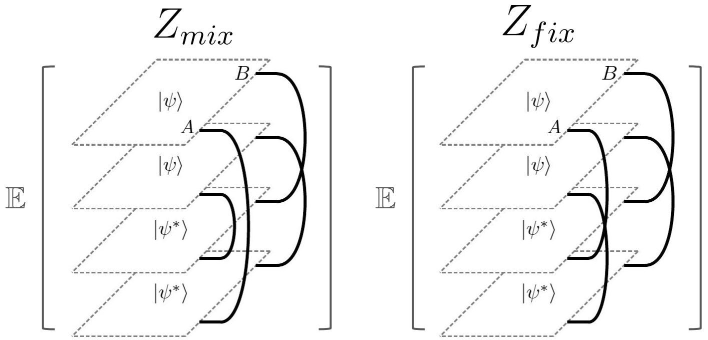
**No local field**

$\mu d > 1 : Z_{mix} \approx Z_{fix}$ $\quad -\log\left(\frac{Z_{mix}}{Z_{fix}}\right) \approx 0$ $\quad$ low entanglement

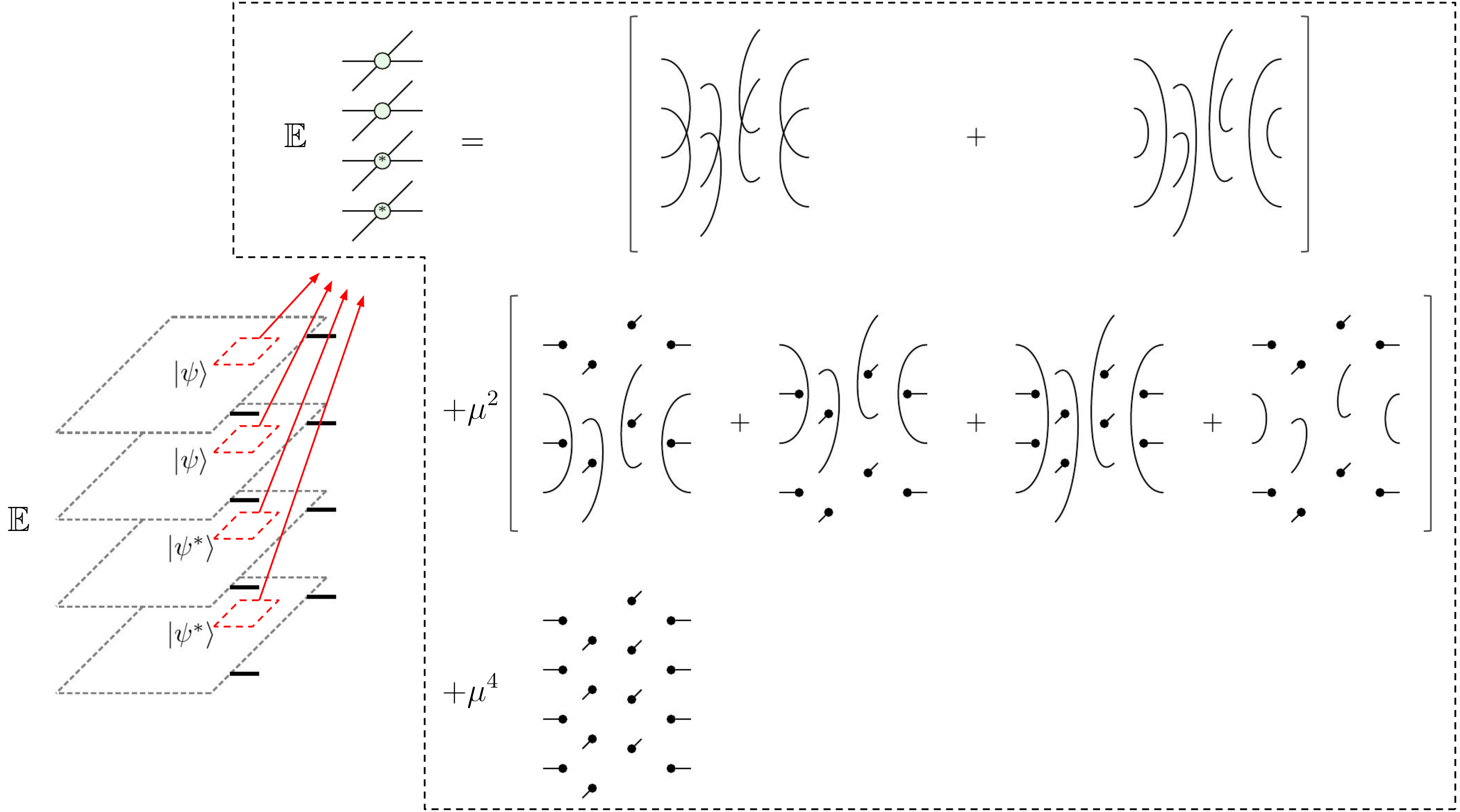
$\mu d < 1 : Z_{mix} \ll Z_{fix}$ $\quad -\log\left(\frac{Z_{mix}}{Z_{fix}}\right) \gg 0$ $\quad$ high entanglement




$\sim \mathcal{N}(\mu, 1)$

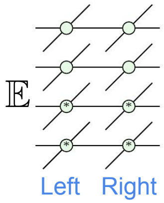


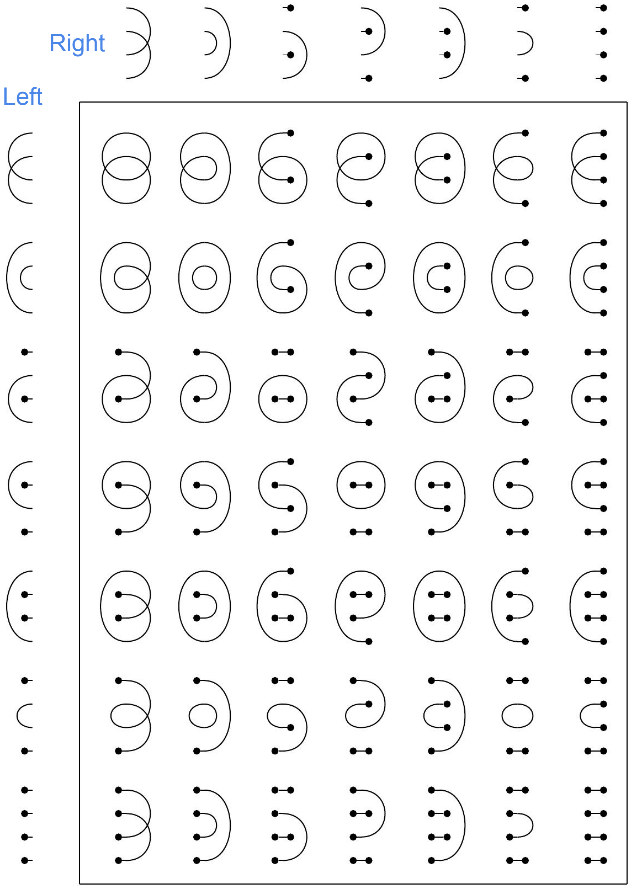
# Details for averaging over tensors



# Details for averaging over tensors

Contracting adjacent tensors introduces a scalar depending on the number of loops & lines

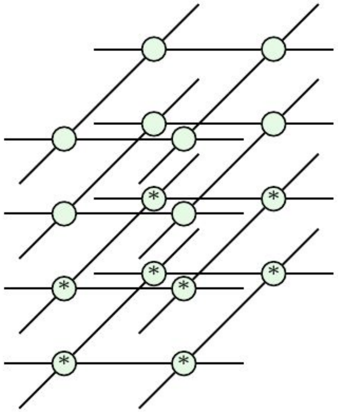



$$= \begin{bmatrix} d^2 & d & d^2 & d^2 & d & d & d^2 \\ d & d^2 & d & d & d^2 & d^2 & d^2 \\ d^2 & d & d^3 & d^2 & d^3 & d^2 & d^3 \\ d^2 & d & d^2 & d^3 & d^2 & d^2 & d^3 \\ d & d^2 & d^3 & d^2 & d^3 & d^2 & d^3 \\ d & d^2 & d^2 & d^2 & d^2 & d^3 & d^3 \\ d^2 & d^2 & d^3 & d^3 & d^3 & d^3 & d^4 \end{bmatrix}$$

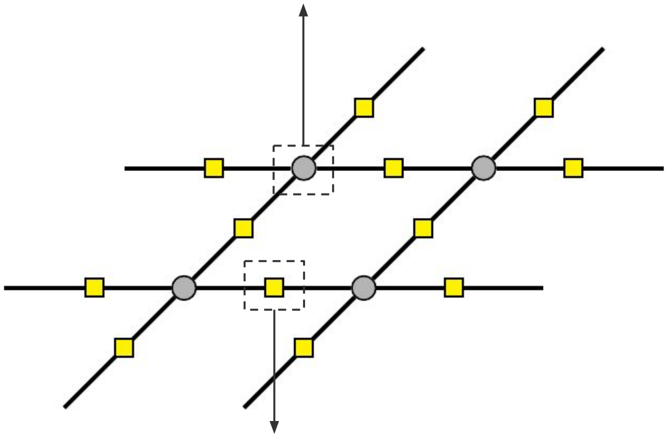
# Details for averaging over tensors

$\mathbb{E}$



$=$

$$\delta_{ijkl}\begin{bmatrix}1 & 1 & \mu^2 & \mu^2 & \mu^2 & \mu^2 & \mu^4\end{bmatrix}_i$$

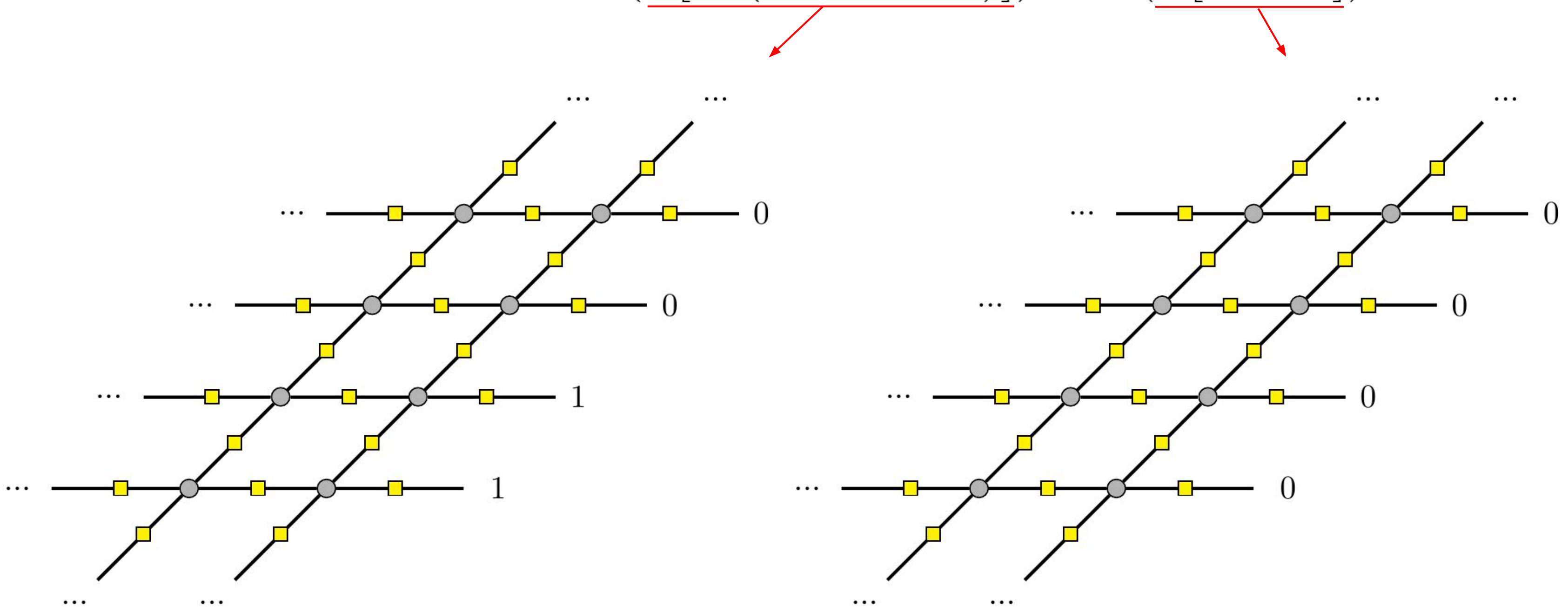


$$\begin{bmatrix}
d^2 & d & d^2 & d^2 & d & d & d^2 \\
d & d^2 & d & d & d^2 & d^2 & d^2 \\
d^2 & d & d^3 & d^2 & d^3 & d^2 & d^3 \\
d^2 & d & d^2 & d^3 & d^2 & d^2 & d^3 \\
d & d^2 & d^3 & d^2 & d^3 & d^2 & d^3 \\
d & d^2 & d^2 & d^2 & d^2 & d^3 & d^3 \\
d^2 & d^2 & d^3 & d^3 & d^3 & d^3 & d^4
\end{bmatrix}$$

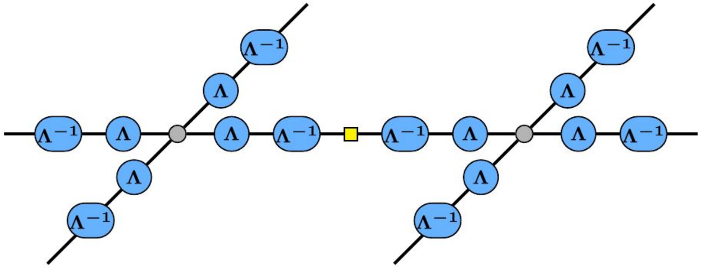
# Details for averaging over tensors

$$\mathbb{E}\left[-\log(\rho_A^2)\right] \approx -\log\left(\mathbb{E}\left[\frac{Tr\left(Tr_B\left(|\psi\rangle\langle\psi|\right)^2\right)}{|\langle\psi|\psi\rangle|^2}\right]\right)$$

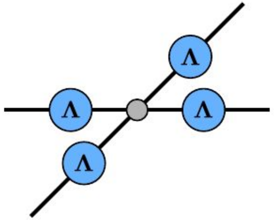
$$\approx -\log\left(\mathbb{E}\left[Tr\left(Tr_B\left(|\psi\rangle\langle\psi|\right)^2\right)\right]\right) + \log\left(\mathbb{E}\left[|\langle\psi|\psi\rangle|^2\right]\right)$$



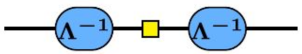
# Details for averaging over tensors



$$\Lambda = \begin{pmatrix} d & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & d & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & d^{3/2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & d^{3/2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & d^{3/2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & d^{3/2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & d^2 \end{pmatrix}$$



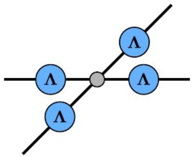
$$= d^4 \delta_{ijkl} \begin{bmatrix} 1 & 1 & \mu^2 d^2 & \mu^2 d^2 & \mu^2 d^2 & \mu^2 d^2 & \mu^4 d^4 \end{bmatrix}_i$$



$$= \begin{bmatrix} 1 & 1/d & 1/d^{0.5} & 1/d^{0.5} & 1/d^{1.5} & 1/d^{1.5} & 1/d \\ 1/d & 1 & 1/d^{1.5} & 1/d^{1.5} & 1/d^{0.5} & 1/d^{0.5} & 1/d \\ 1/d^{0.5} & 1/d^{1.5} & 1 & 1/d & 1/d & 1/d & 1/d^{0.5} \\ 1/d^{0.5} & 1/d^{1.5} & 1/d & 1 & 1/d & 1/d & 1/d^{0.5} \\ 1/d^{1.5} & 1/d^{0.5} & 1/d & 1/d & 1 & 1/d & 1/d^{0.5} \\ 1/d^{1.5} & 1/d^{0.5} & 1/d & 1/d & 1/d & 1 & 1/d^{0.5} \\ 1/d & 1/d & 1/d^{0.5} & 1/d^{0.5} & 1/d^{0.5} & 1/d^{0.5} & 1 \end{bmatrix}$$

Becomes ferromagnetic as D → ∞!

# Details for averaging over tensors

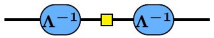



$$= d^4\delta_{ijkl}\begin{bmatrix}1 & 1 & \mu^2 d^2 & \mu^2 d^2 & \mu^2 d^2 & \mu^2 d^2 & \mu^4 d^4\end{bmatrix}_i$$

$\mu d > 1$ → Prefer the latter five configurations → Competition between boundary condition and magnetic field → disorder

$\mu d < 1$ → Prefer first two configurations → Mixed boundary condition → domain wall

Transition point: $\mu d = 1$ (no local field)

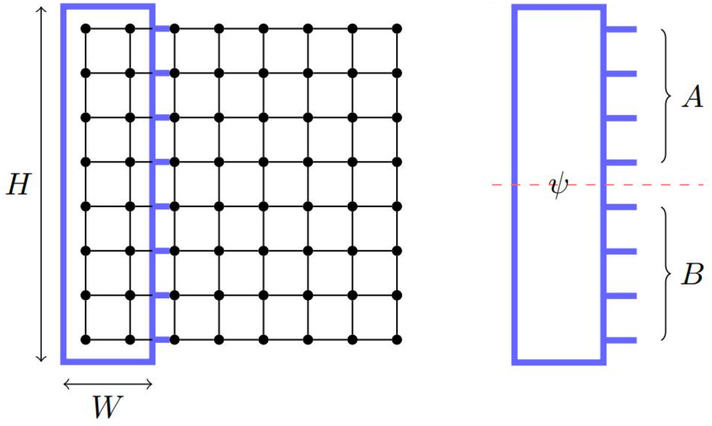


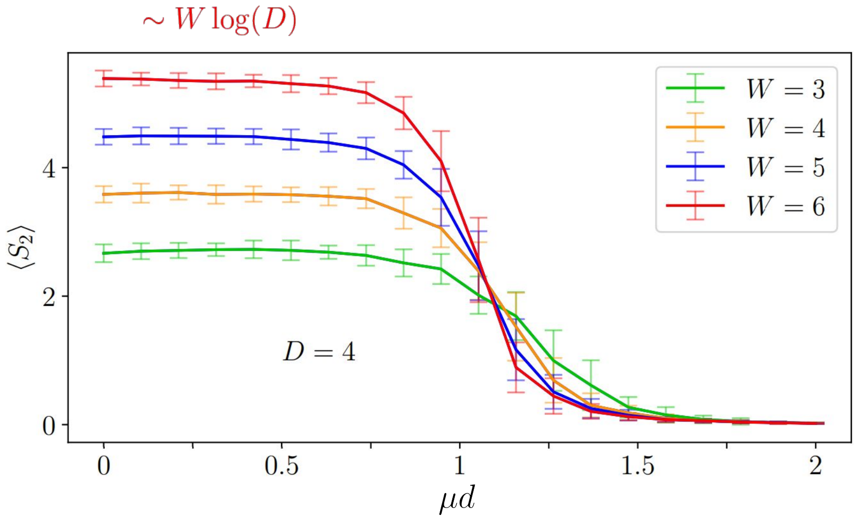

$$= \begin{bmatrix}
1 & 1/d & 1/d^{0.5} & 1/d^{0.5} & 1/d^{1.5} & 1/d^{1.5} & 1/d \\
1/d & 1 & 1/d^{1.5} & 1/d^{1.5} & 1/d^{0.5} & 1/d^{0.5} & 1/d \\
1/d^{0.5} & 1/d^{1.5} & 1 & 1/d & 1/d & 1/d & 1/d^{0.5} \\
1/d^{0.5} & 1/d^{1.5} & 1/d & 1 & 1/d & 1/d & 1/d^{0.5} \\
1/d^{1.5} & 1/d^{0.5} & 1/d & 1/d & 1 & 1/d & 1/d^{0.5} \\
1/d^{1.5} & 1/d^{0.5} & 1/d & 1/d & 1/d & 1 & 1/d^{0.5} \\
1/d & 1/d & 1/d^{0.5} & 1/d^{0.5} & 1/d^{0.5} & 1/d^{0.5} & 1
\end{bmatrix}$$

→ Tend to be ferromagnetic (spins aligned)

# Numerical simulation

We observed the same transition in finite-size simulation. We choose H >> W so the entropy saturates (H = 4W in our simulation).

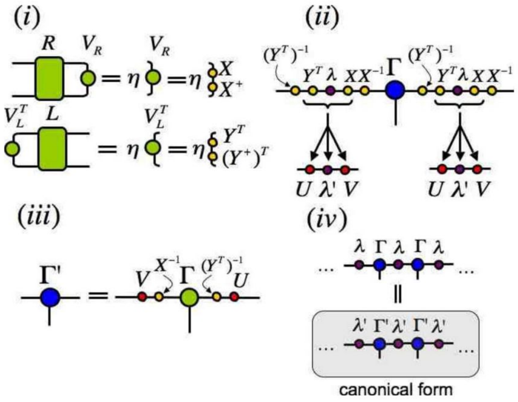


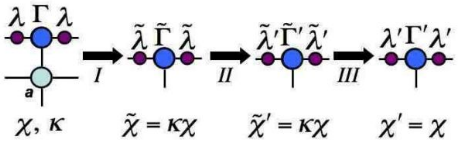



# iMPS simulation of the effective model

## iMPS - iMPO algorithm

Roman Orus and Guifre Vidal, The iTEBD algorithm beyond unitary evolution, Phys. Rev. B 78, 155117 (2008)

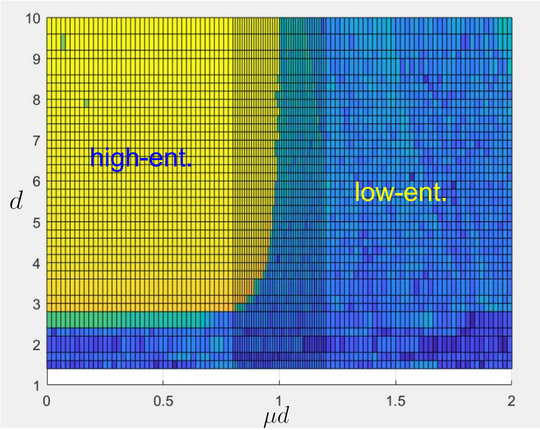


Overlap between left/right dominant eigenvectors of fixed-point iMPS

$$\left|\frac{\langle l_1|\lambda|r_0\rangle}{\langle l_0|\lambda|r_0\rangle}\right|$$

$$|r_i\rangle = \mathrm{v}^r_{\max}(\Gamma_i\lambda)$$
$$\langle l_i| = \mathrm{v}^l_{\max}(\lambda\Gamma_i)$$



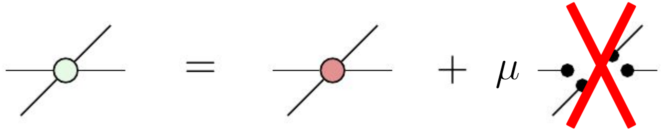

$\mu$ = 0 case also relates to previous results:


[1] Romain Vasseur, Andrew C. Potter, Yi-Zhuang You, Andreas W. W. Ludwig, Entanglement Transitions from Holographic Random Tensor Networks, Phys. Rev. B 100, 134203 (2019)

[2] Ryan Levy, Bryan K. Clark, Entanglement Entropy Transitions with Random Tensor Networks, arXiv:2108.02225

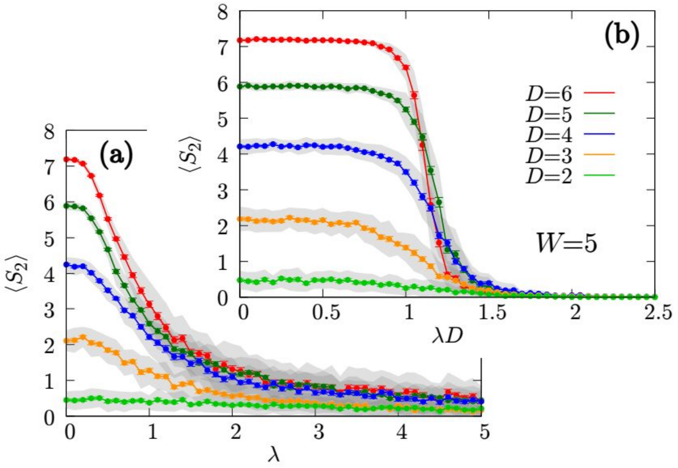
# Role of positivity?

Does “rank-one”ness cause the complexity transition instead of positiveness?




(a) random rank-one (but not positive)

(b) random positive (but not rank-one)




The “positivity” part is important to observe the transition!

Or in other words, the rank-one states need to be “aligned”.

# Sign problem in QMC

$$\langle O\rangle = \frac{1}{Z}\mathrm{Tr}[Oe^{-\beta H}] = \frac{1}{Z}\mathrm{Tr}[O(e^{-\beta H/M})^M]$$
$$= \frac{1}{Z}\sum_{\{x_i\}}\langle x_0|O|x_1\rangle\langle x_1|e^{-\beta H/M}|x_2\rangle\langle x_2|...|x_M\rangle\langle x_M|e^{-\beta H/M}|x_0\rangle$$
$$= \frac{1}{Z}\sum_{x} O(x)T(x)$$

$$T(x) = \mathrm{sign}(T(x))|T(x)|$$

$$\langle \mathrm{sign}\rangle = \frac{\sum_x T(x)}{\sum_x |T(x)|} = e^{-\beta N\Delta f}$$

$T(x) \geq 0$

sample $x_i^* \sim \frac{T(x)}{\sum_x T(x)}$

estimate by $\frac{1}{K}\sum_{i=1}^{K} O(x_i^*)$

error $\sim \frac{1}{\sqrt{K}}$

$T(x)$ varying signs

sample $x_i^* \sim \frac{|T(x)|}{\sum_x |T(x)|}$

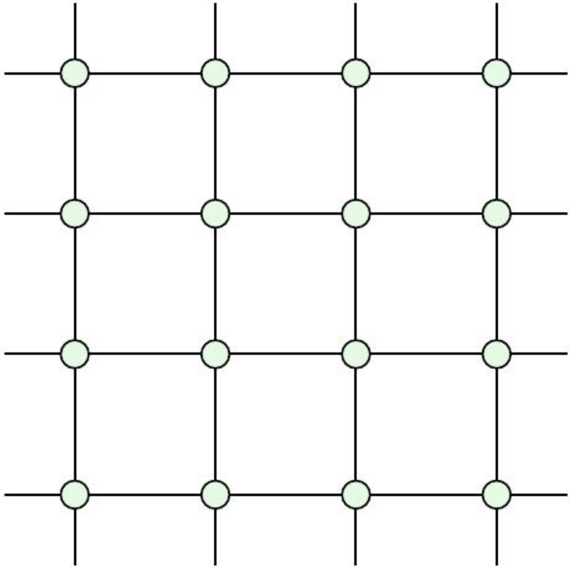
estimate by $\frac{1}{K}\frac{1}{\langle \mathrm{sign}\rangle}\sum_{i=1}^{K} O(x_i^*)\mathrm{sign}(T(x_i^*))$

error $\sim \frac{e^{\beta N\Delta f}}{\sqrt{K}}$

“sign problem”: exponential dependence on N caused by $\langle \mathrm{sign}\rangle$

# Sign problem in random TN

"Sign problem" in TN: $$e^{-N\Delta f} := \frac{\sum_{\text{edge labeling } c} \prod_v M_c^v}{\sum_{\text{edge labeling } c} \prod_v |M_c^v|}$$

$\mathcal{T} =$ 

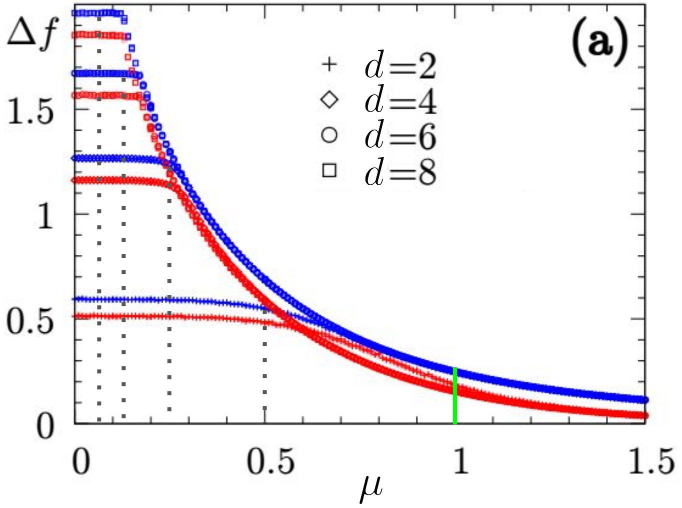
$$\mathcal{C}(\mathcal{T}) = \sum_{\text{edge labeling } c} \prod_v M_c^v$$

# Sign problem in random TN

$$e^{-N\Delta f} := \frac{\sum_{\text{edge labeling } c} \prod_v M_c^v}{\sum_{\text{edge labeling } c} \prod_v |M_c^v|}$$

Sign problem is

| | |
|---|---|
| $\mu \lesssim 1/d$ : | worse for large d |
| $1/d \lesssim \mu \lesssim 1$ : | independent of d |
| $1 \lesssim \mu$ : | rapidly vanishing |




**Sign problem only disappears when $\mu \gtrsim 1$, where entries are mostly positive**

# Part II: Positive bias makes tensor network contraction tracatable (arXiv:2410.05414)

We give a series of more rigorous results, including a provably "efficient" algorithm to contract slightly positive random tensor networks (same transition point).

**Complexity of (2D) tensor network contraction**

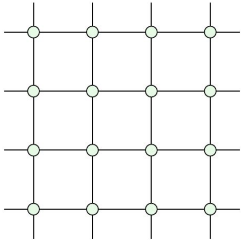
| | Exact | Approximate |
|---|---|---|
| Worst-case | #P-hard [SWVC07] | Empirically hard |
| Average-case ( $\mu = 0$ ) | #P-hard [HHEG20] | |
| Average-case + small positive bias ( $\mu \gtrsim 1/d$ ) | #P-hard [our result] | Quasi-poly time algorithm [our result] |

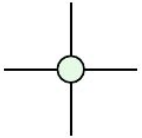
# Main theorem

**Theorem** (informal): For a random 2D tensor network, if

$$\mu \gtrsim 1/d, \qquad d \gtrsim n$$

then with high probability, there exists a quasi-polynomial time algorithm which approximates the TN contraction value up to arbitrary 1/poly(n) multiplicative error.





$\sim \mathcal{N}(\mu, 1)$

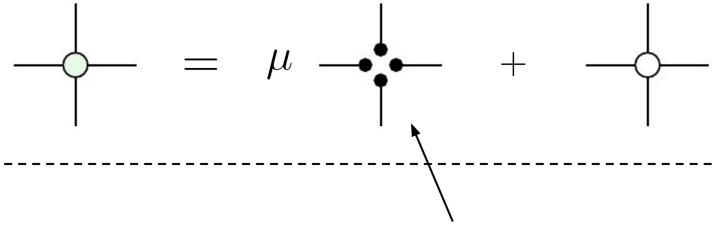
i.i.d. for all entries & all tensors.

# Method overview

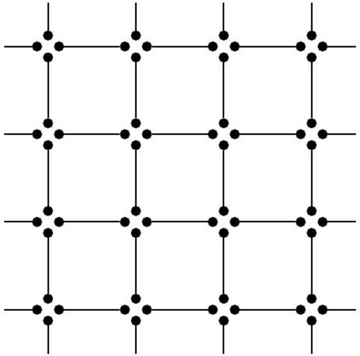
Easy — Interpolate → Hard

$\mu \to \infty$ → $\mu = 1/d$

$$\mathcal{N}(\mu, 1) = \mu + \mathcal{N}(0, 1)$$




all-one tensor



Easy to contract!

# Method overview

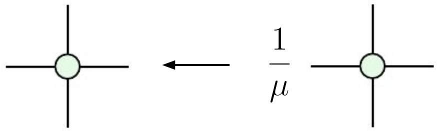
Easy — Interpolate → Hard

$\mu \to \infty$ $\quad\quad$ $\mu = 1/d$

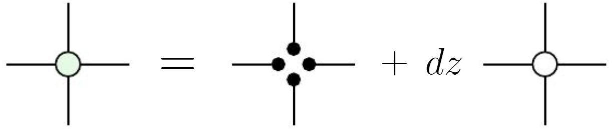
$z = 0$ $\quad\quad$ $z = 1$

For convenience later, introduce change of variable $z = \frac{1}{\mu d}$

and rescale the tensors. No effect on multiplicative error.

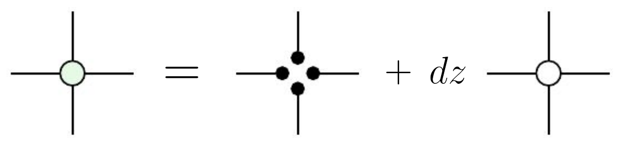



New definition



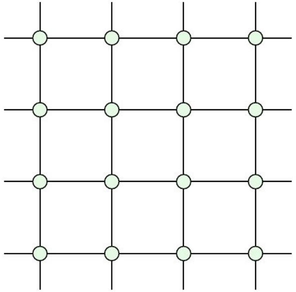
# Method overview

To interpolate from z = 0 to 1, we uses **Barvinok's method [Bar16].** It relies on two observations.




## Observation 1

Contracted TN is a degree-n random polynomial on z, denoted as g(z).



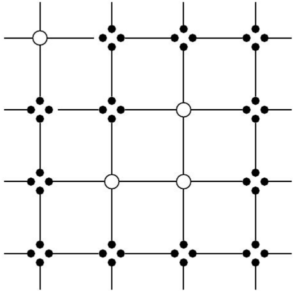
$= g(z)$

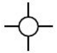
## Observation 2

k-th derivative of g(z) at z = 0 can be computed brute-forcely in $n^{O(k)}$ time.

e.g.

$$\frac{1}{4!}\frac{\partial^4 g(z)}{\partial z^4}\bigg|_{z=0} =$$



\+ all other configs with four 

# Barvinok's method [Bar16]

**Originally designed for permanent**

**Input:** degree-n polynomial $g(z)$

$g^{(k)}(0)$ accessible in $n^{O(k)}$ time

**Output:** $g(1)$, ε multiplicative error

**Algorithm:** $f(z) = \ln(g(z))$, now ε additive error

$f(1) \approx \hat{f}(1) = \sum_{k=0}^{M} \frac{f^{(k)}(0)}{k!}$, output $e^{\hat{f}(1)}$, that's it!

Effectiveness depends on the roots of $g(z)$

# Barvinok's method [Bar16]

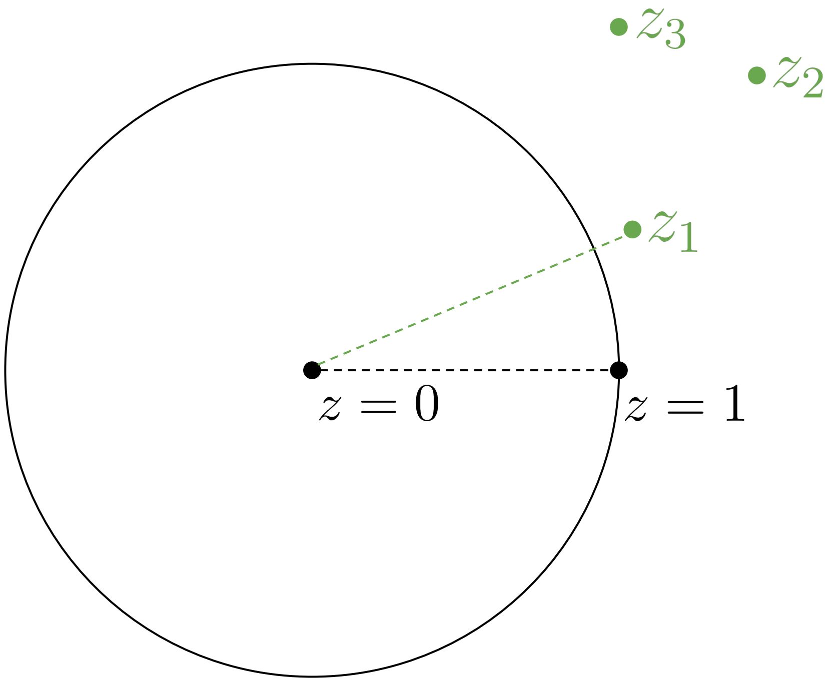
Roots of $g(z)$ are outside the $|z| = 1$ disk → small error

$$f(z) = \ln(g(z))$$

$$f(1) \approx \hat{f}(1) = \sum_{k=0}^{M} \frac{f^{(k)}(0)}{k!}$$

$$M = O\left(\ln\left(\frac{n}{\epsilon}\right)\right)$$

Run time $n^{O(M)}$, quasi-poly

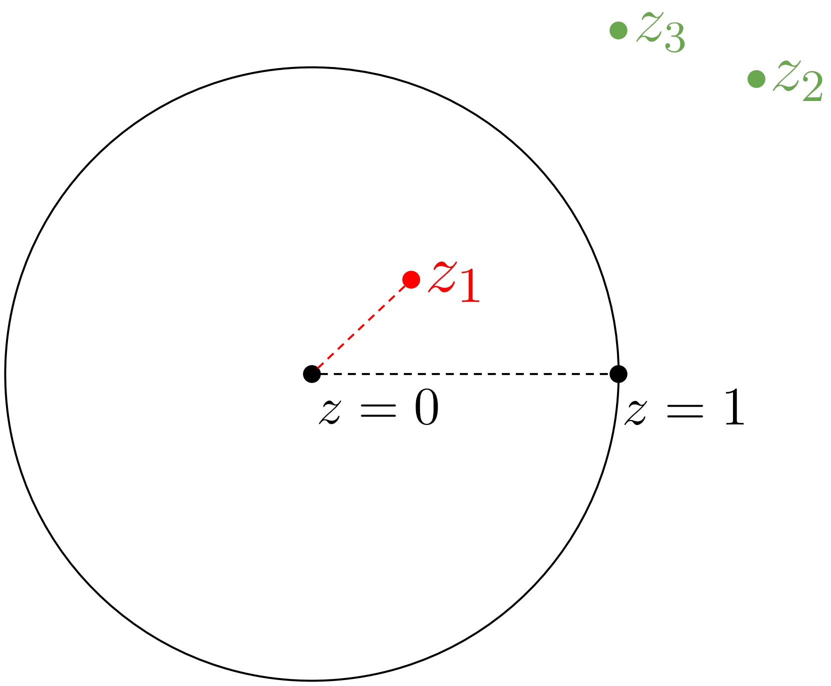


# Barvinok's method [Bar16]

A root of $g(z)$ is inside the $|z| = 1$ disk $\longrightarrow$ error blows up!

$$f(z) = \ln(g(z))$$

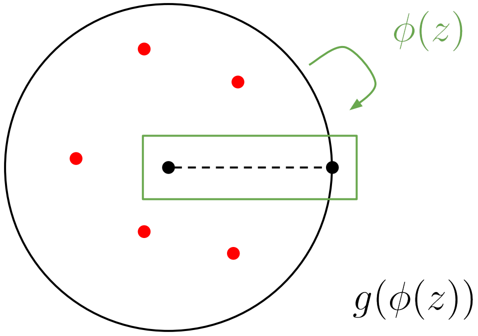
$$f(1) \approx \hat{f}(1) = \sum_{k=0}^{M} \frac{f^{(k)}(0)}{k!}$$



# Barvinok's method via root-free path

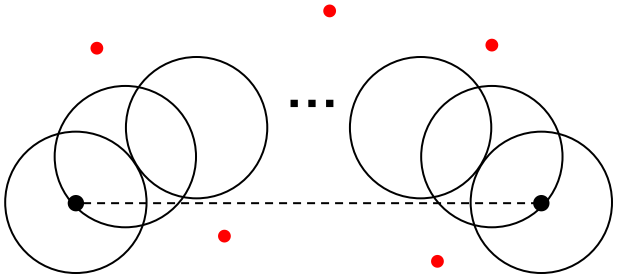
Not many roots (e.g. O(1)) → can interpolate along a root-free path!

### Method 1 [Bar16]




Mapping the disk to a strip

### Method 2 [EM18]



"Algorithmic" analytic continuation

Both remain quasi-polynomial time.

# Barvinok's method on random tensor network

$$f(z) = \ln(g(z))$$

$$f(1) \approx \hat{f}(1) = \sum_{k=0}^{M} \frac{f^{(k)}(0)}{k!}$$

$\longrightarrow$ few roots $\longrightarrow$ find a root-free path $\longrightarrow$ quasi-poly algorithm

# Barvinok's method on random tensor network

$$f(z) = \ln(g(z))$$

$$f(1) \approx \hat{f}(1) = \sum_{k=0}^{M} \frac{f^{(k)}(0)}{k!}$$

→ few roots on average → find a root-free path with high probability → quasi-poly algorithm that successes with high probability

# Barvinok's method on random tensor network

$f(z) = \ln(g(z))$

$f(1) \approx \hat{f}(1) = \sum_{k=0}^{M} \frac{f^{(k)}(0)}{k!}$ → few roots on average → find a root-free path with high probability → quasi-poly algorithm that successes with high probability

How to bound average number of roots? →

$$\mathbb{E}_g[N_{r(1-\delta)}] \leq \frac{1}{2\delta}\mathbb{E}_g\left[\oint_r \log\left(\left|\frac{g(z)}{g(0)}\right|^2\right) dz\right]$$

Jensen's formula

$$\leq \frac{1}{2\delta}\log\left(\oint_r \mathbb{E}_g\left|\frac{g(z)}{g(0)}\right|^2 dz\right)$$

Jensen's inequality

[EM18]

# Barvinok's method on random tensor network

$f(z) = \ln(g(z))$

$f(1) \approx \hat{f}(1) = \sum_{k=0}^{M} \frac{f^{(k)}(0)}{k!}$ → few roots on average → find a root-free path with high probability → quasi-poly algorithm that successes with high probability

How to bound average number of roots? → by upper-bounding $E_g \left| \frac{g(z)}{g(0)} \right|^2$ [EM18]

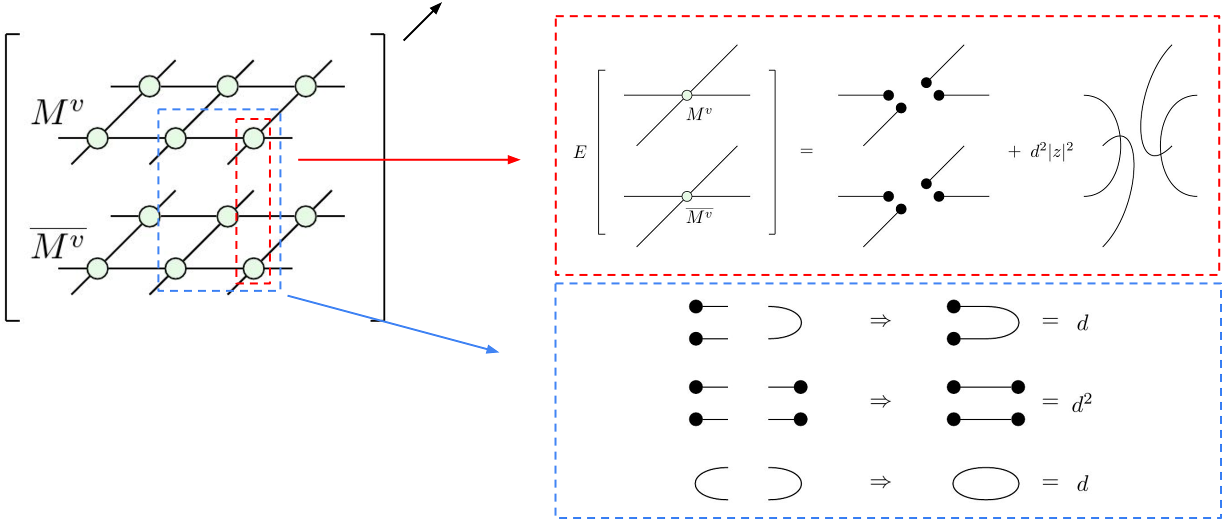
# Barvinok's method on random tensor network

$f(z) = \ln(g(z))$

$f(1) \approx \hat{f}(1) = \sum_{k=0}^{M} \frac{f^{(k)}(0)}{k!}$ → few roots on average → find a root-free path with high probability → quasi-poly algorithm that successes with high probability

How to bound average number of roots? → by upper-bounding $E_g \left| \frac{g(z)}{g(0)} \right|^2$ [EM18]

$E_g |g(z)|^2 = E_g$



# Barvinok's method on random tensor network

$$f(z) = \ln(g(z))$$

$$f(1) \approx \hat{f}(1) = \sum_{k=0}^{M} \frac{f^{(k)}(0)}{k!}$$

→ few roots on average → find a root-free path with high probability → quasi-poly algorithm that successes with high probability

How to bound average number of roots? → by upper-bounding $E_g \left| \frac{g(z)}{g(0)} \right|^2$ [EM18]

$E_g |g(z)|^2 =$ partition function of 2D classical ising model with local magnetic field!

$z = 1 \longleftrightarrow$ zero magnetic field

# Barvinok's method on random tensor network

$f(z) = \ln(g(z))$

$f(1) \approx \hat{f}(1) = \sum_{k=0}^{M} \frac{f^{(k)}(0)}{k!}$ → few roots on average → find a root-free path with high probability → quasi-poly algorithm that successes with high probability

How to bound average number of roots? → by upper-bounding $E_g \left| \frac{g(z)}{g(0)} \right|^2$ [EM18]

$E_g |g(z)|^2 =$ partition function of 2D classical ising model with local magnetic field!

$z = 1$ ⟷ zero magnetic field

→ Onsager's solution [Ons44] for zero-field 2D ising model (finite-size variant [Kau49])

# Barvinok's method on random tensor network

$f(z) = \ln(g(z))$

$f(1) \approx \hat{f}(1) = \sum_{k=0}^{M} \frac{f^{(k)}(0)}{k!}$ → few roots on average $O(1)$ → find a root-free path with high probability → quasi-poly algorithm that successes with high probability

How to bound average number of roots? → by upper-bounding $E_g \left| \frac{g(z)}{g(0)} \right|^2$ [EM18]

$O(1)$ for z ≲ 1

$E_g |g(z)|^2 =$ partition function of 2D classical ising model with local magnetic field! → Onsager's solution [Ons44] for zero-field 2D ising model (finite-size variant [Kau49])

$z = 1$ ⟷ zero magnetic field

# Outlook

**(Part I)**

- Besides the stats model mapping, how to more intuitively understand the complexity transition?

**(Part II)**

- How to prove tractability with constant bond-dimension?

- Can one improve the interpolation method?
    - Interpolate from other rank-one tensors? (e.g. BP fixed point)
    - Use better interpolations?

- Can the interpolation method be used in practice? (with some modifications & heuristics)