

# Grassmann tensor network study of multi-flavor gauge theory



NIIGATA  
UNIVERSITY

Atis Yosprakob

KEK Theory Workshop  
November 30<sup>th</sup>, 2023



Based on JHEP11(2023)187, with **Jun Nishimura** (KEK) and **Kouichi Okunishi** (Niigata U)

# Introduction

- Lattice fermions are known to be difficult to simulate because of its large computational cost

$$\int \exp[-\theta^T A \eta] d\theta d\eta = \det(A)$$

Dirac matrix grows with lattice volume!

- In many systems, the fermions give the **Numerical Sign Problem**  
==> Hubbard model, Chiral fermion, Finite density
- **Tensor network (Grassmann TRG)** offers a solution

# Introduction

Selected works that use 'Grassmann tensor network'

- ◆ Lattice Schwinger model
  - Shimizu-Kuramashi (<sup>`14</sup>,<sup>`14</sup>,<sup>`18</sup>)
- ◆ 2D QCD
  - Bloch-Lohmayer (<sup>`23</sup>)
- ◆ Hubbard model
  - Akiyama-Kuramashi (<sup>`21</sup>)
  - Akiyama-Kuramashi-Yamashita (<sup>`22</sup>)
- ◆ and many others

# Introduction

- Tensor size grows with the dimension of local Hilbert space  
==> Grows exponentially with  $N_f$  !!!  
  
→ Akiyama (23) proposes a way to reduce the CG cost based on matrix product decomposition.
- Gauge theory with multiple flavors is even more difficult because of the gauge d.o.f.  
==> The topic of this talk

# Content

- Introduction
- Multi-layer construction
- Results
- Summary

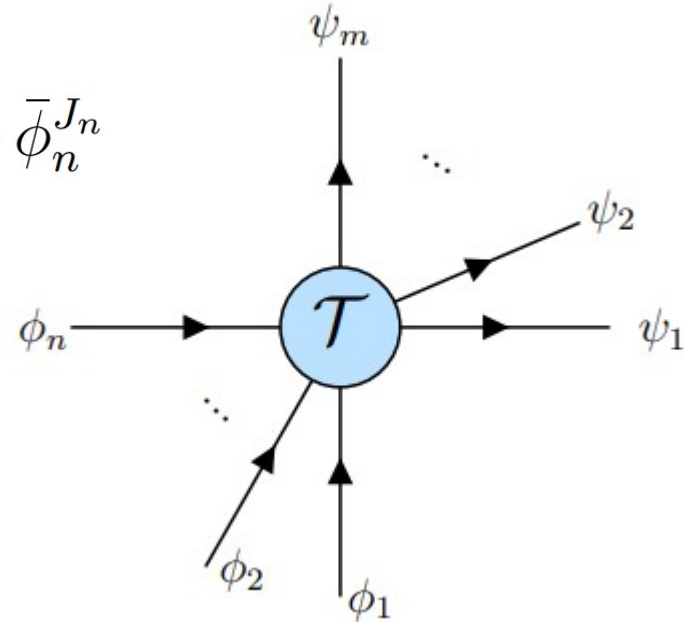
# Grassmann tensor

$$\mathcal{T}_{\psi_1 \cdots \psi_m \bar{\phi}_1 \cdots \bar{\phi}_n}$$

$$= \sum_{I_1, \dots, I_m, J_1, \dots, J_n} T_{I_1 \dots I_m J_1 \dots J_n} \psi_1^{I_1} \cdots \psi_m^{I_m} \bar{\phi}_1^{J_1} \cdots \bar{\phi}_n^{J_n}$$

Coefficient tensor

Multi-component  
Grassmann numbers



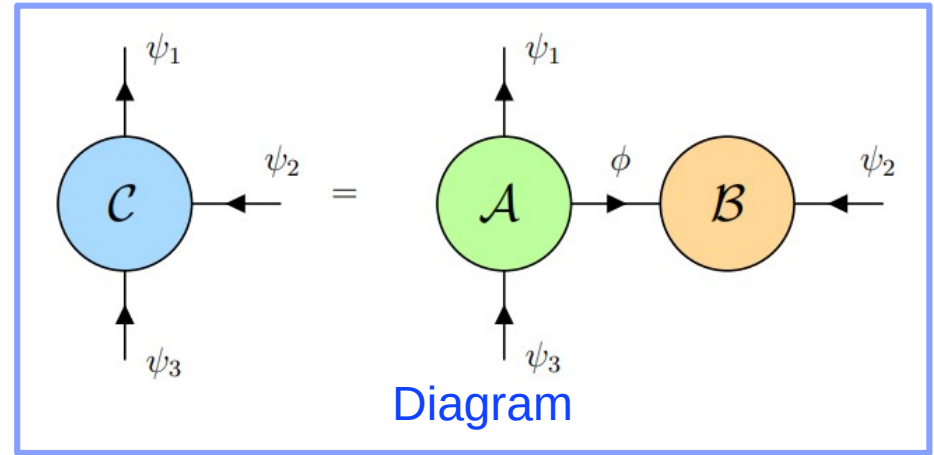
- Grassmann numbers are anti-commuting
- **Non-conjugated** and **Conjugated** fermions are distinguished !

# Grassmann tensor contraction

$$C_{\psi_1 \bar{\psi}_2 \bar{\psi}_3} = \int_{\bar{\phi} \phi} \mathcal{A}_{\psi_1 \phi \bar{\psi}_3} \mathcal{B}_{\bar{\psi}_2 \bar{\phi}}$$

**Symbolic**

$$\int_{\bar{\eta} \eta} \equiv \int d\bar{\eta} d\eta e^{-\bar{\eta} \eta}$$



$$C_{IJK} = \sum A_{ILK} B_{JLS} S_{JKL}$$

$$S_{JKL} = \sigma_L \times (-)^{p(L)(p(J)+p(K))+p(J)p(K)}$$

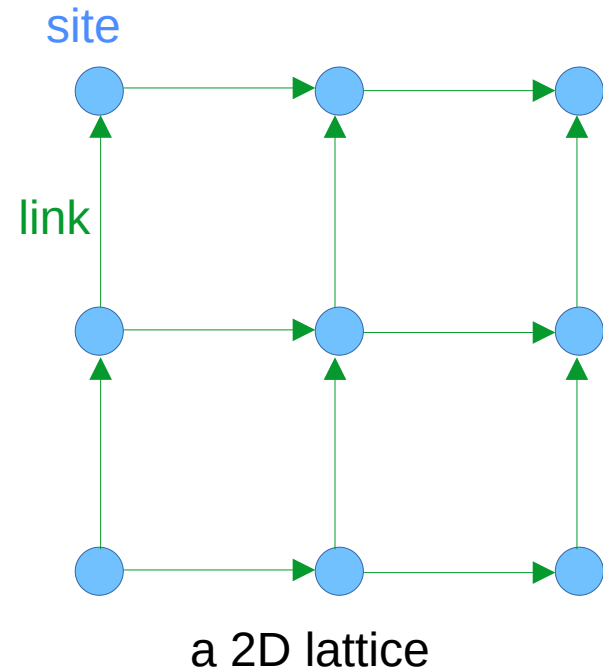
**Numerical coefficient tensor**

# Lattice gauge theory

$$S = S_{\text{gauge}}[\varphi] + \sum_{x \in \Lambda_2} \sum_{\alpha=1}^{N_f} \bar{\psi}_x^{(\alpha)} \mathcal{D}^{(\alpha)} \psi_x^{(\alpha)} ,$$

$$S_{\text{gauge}}[\varphi] = \beta \sum_{x \in \Lambda_2} \{1 - \cos(\varphi_{x,1} + \varphi_{x+\hat{1},2} - \varphi_{x+\hat{2},1} - \varphi_{x,2})\}$$

- Gauge fields live on the links
- Fermions live on the sites
- All (Wilson) fermion flavors live on the same site  
Local Hilbert space is very big for large  $N_f$





# Multi-layer construction

1) Separate the action for different flavors

$$S^{(\alpha)} = \frac{1}{N_f} S_{\text{gauge}}[\varphi^{(\alpha)}] + \sum_{x \in \Lambda_2} \bar{\psi}_x^{(\alpha)} \not{D}^{(\alpha)} \psi_x^{(\alpha)}$$

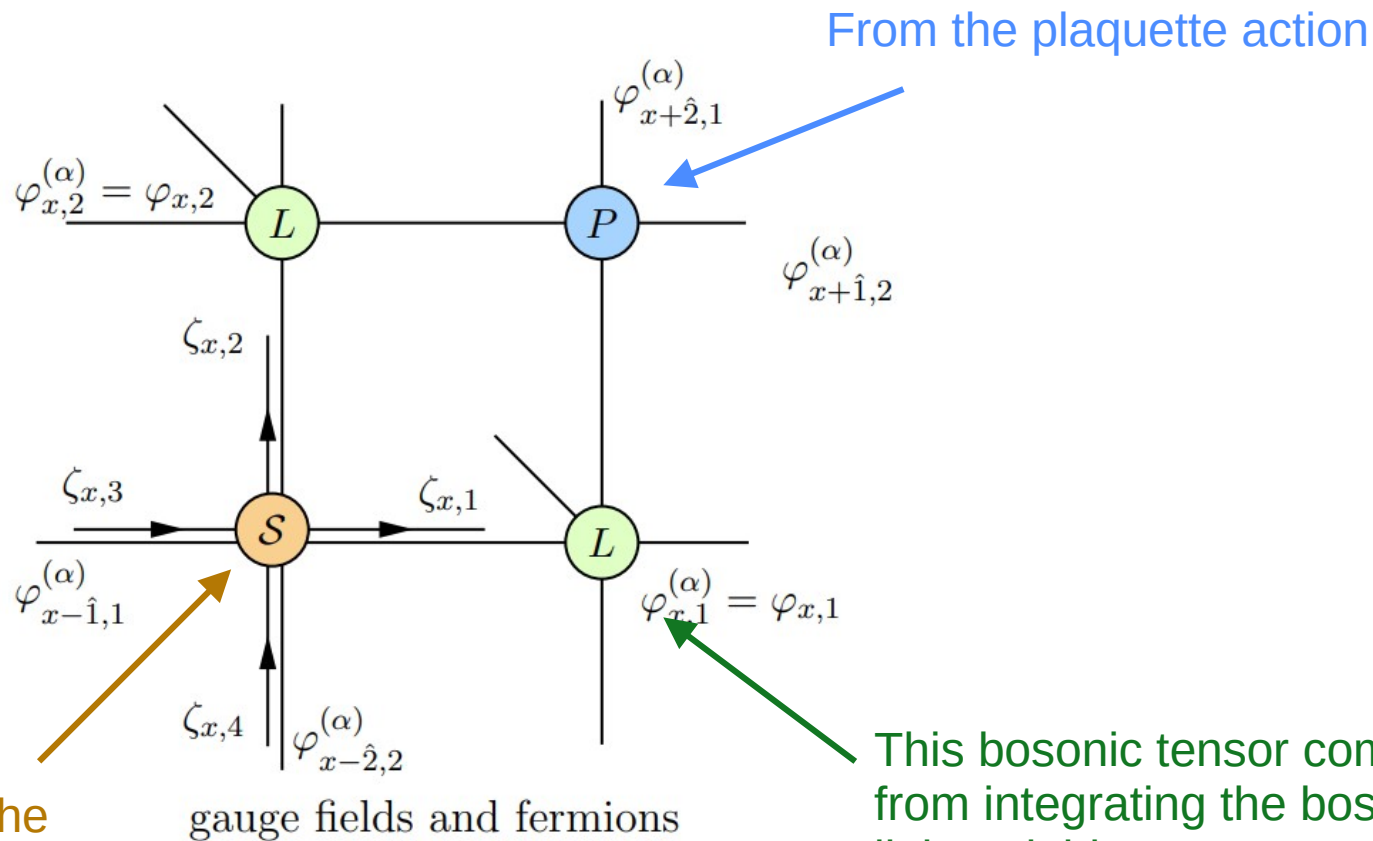
2) Identify all the gauge field from different layers in the partition function

$$Z = \int D\varphi \prod_{\alpha=1}^{N_f} (D\varphi^{(\alpha)} D\psi^{(\alpha)} D\bar{\psi}^{(\alpha)}) \delta(\varphi^{(\alpha)} - \varphi) e^{-\sum_{\alpha} S^{(\alpha)}}$$

In this way, the flavor index acts as an extra dimension.

Inspiration: Domain wall fermion

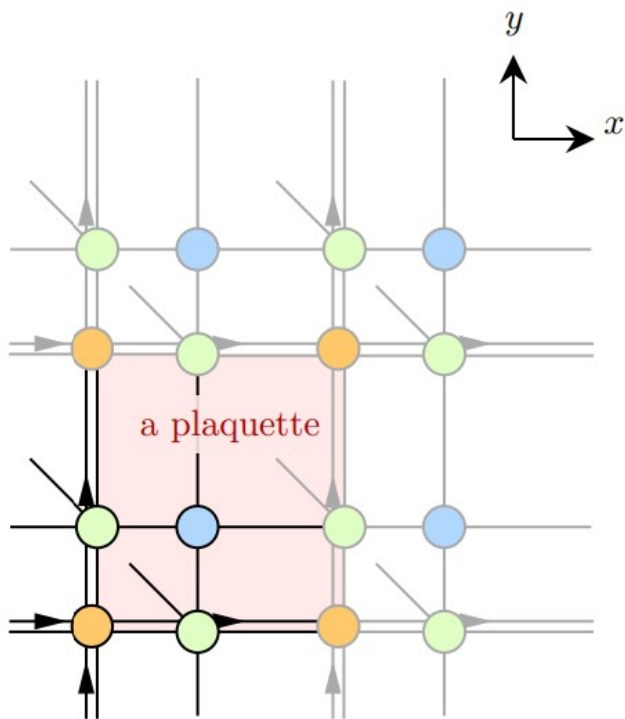
# The tensor network (one site, one layer)



This Grassmann tensor comes from integrating the site fermions.

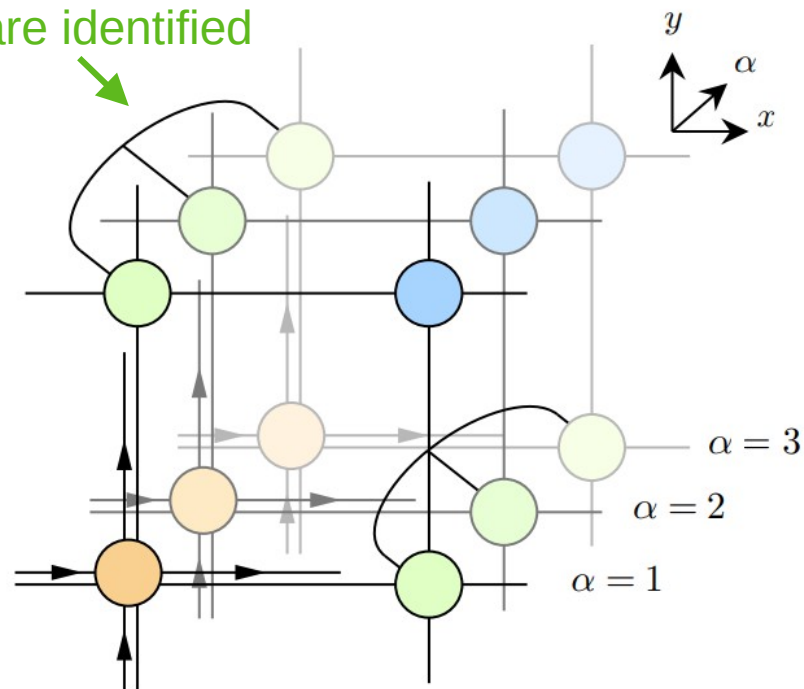
This bosonic tensor comes from integrating the bosonic link variables.

# The tensor network



2x2 lattice, 1 layer

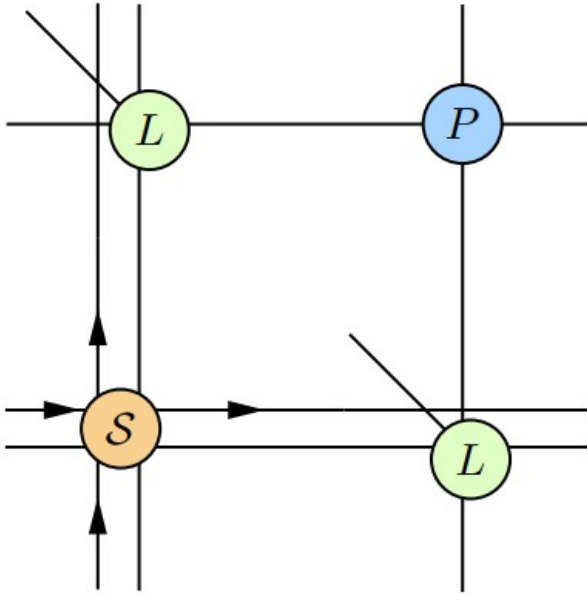
Gauge fields from different layers are identified



1 site, 3 layers

# Tensor compression

The initial tensor is still too big

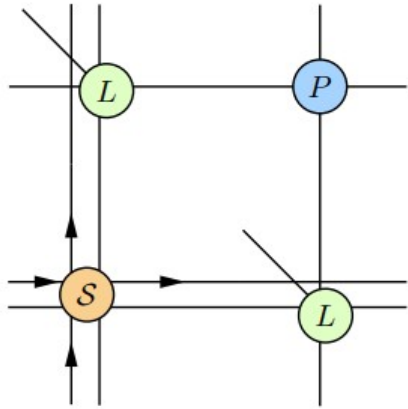


10 bosonic legs & 4 fermionic legs  
=  $K^{10} 16^4$  components

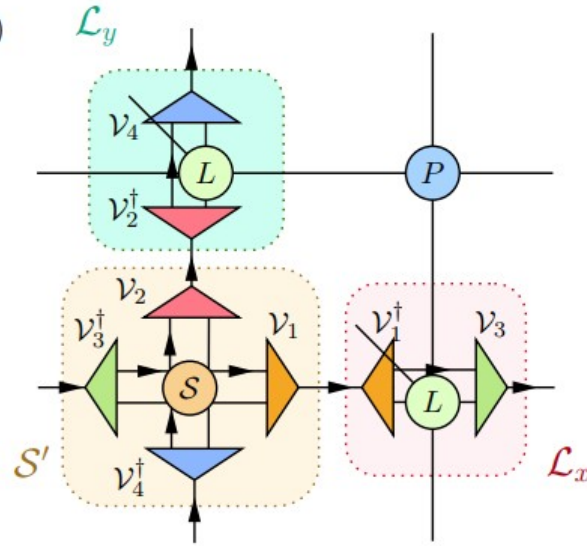
Some compression is needed  
to reduce the tensor size first

# Tensor compression

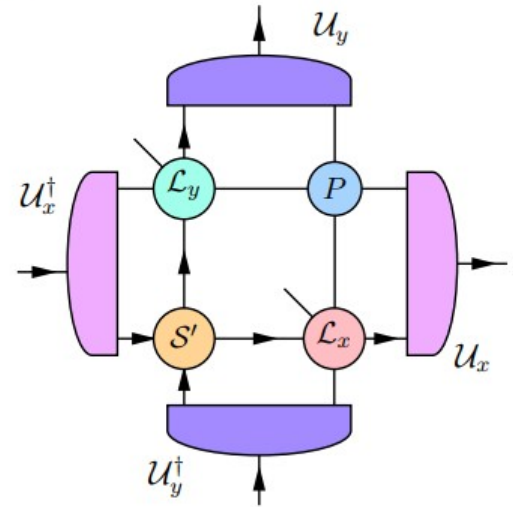
a)



b)



c)



Isometries are first applied around the Grassmann tensor  $S$ : a  $\rightarrow$  b

Then another set is applied around the whole tensor: b  $\rightarrow$  c

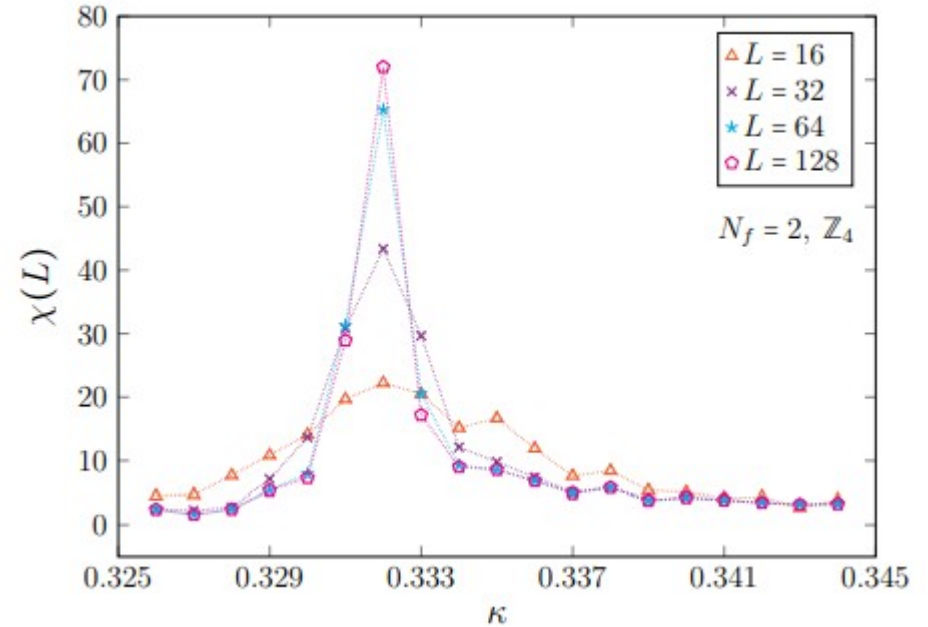
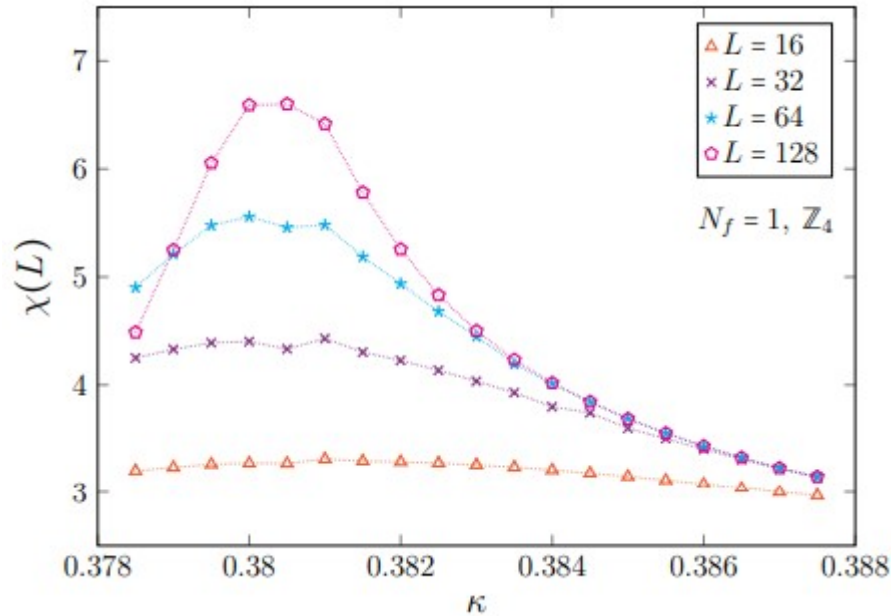
# Tensor compression

$\beta$	$\tilde{\mu}$	$N_f$	$K$	original size	compressed size	compression ratio	$D_x$	$D_y$
0.0	0.0	1	2	67108864	1024	$1.53 \times 10^{-5}$	4	4
0.0	0.0	1	3	3869835264	2304	$5.95 \times 10^{-7}$	4	4
0.0	0.0	1	4	68719476736	4096	$5.96 \times 10^{-8}$	4	4
0.0	0.0	1	5	640000000000	6400	$1.00 \times 10^{-9}$	4	4
2.0	0.0	1	2	67108864	16384	$2.44 \times 10^{-4}$	8	8
2.0	0.0	2	2	67108864	16384	$2.44 \times 10^{-4}$	8	8
2.0	3.0	1	2	67108864	16384	$2.44 \times 10^{-4}$	8	8
2.0	3.0	2	2	67108864	16384	$2.44 \times 10^{-4}$	8	8

Performance of the compression

# Results

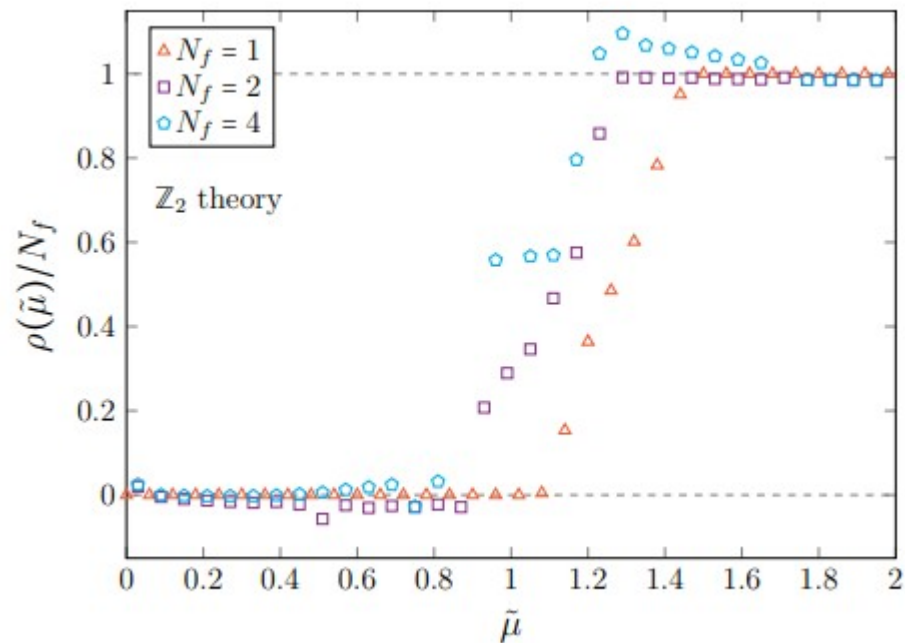
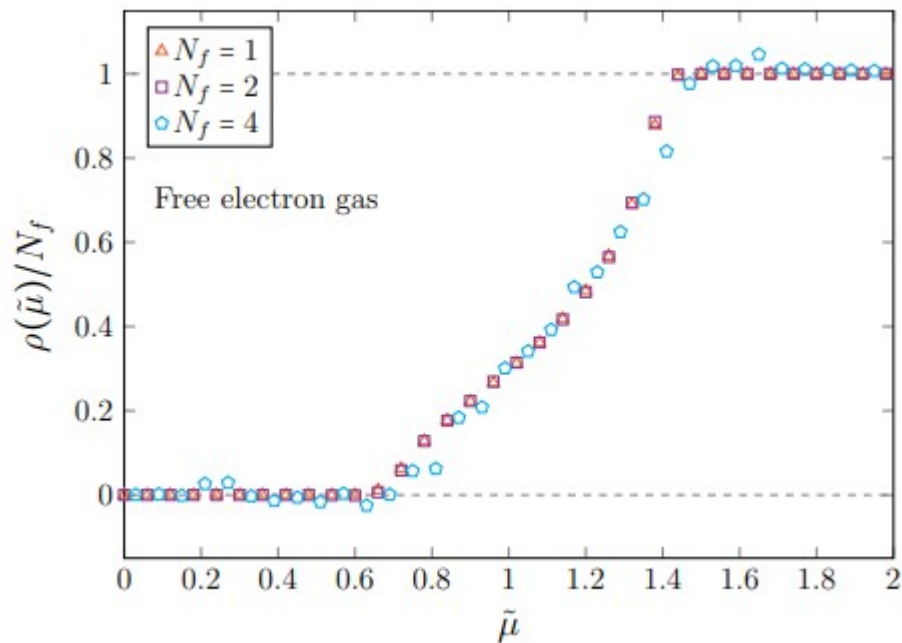
- Chiral symmetry breaking



Critical behavior is observed near the chiral sym breaking point.  
The critical point is in agreement with the literature.

# Results

- Finite density and Silver Blaze phenomena



This cannot be observed in traditional Monte Carlo simulation because of the **sign problem!**



# Summary and Outlooks

- We propose a way to incorporate multiple flavors for gauge theory
- Compression on the initial tensor is essential
- Chiral symmetry breaking and Silver Blaze phenomenon are demonstrated

## Future works:

- Increase  $D_{\text{cut}}$  (new workstation)
- Non-abelian theory
- Domain-wall fermion

# GrassmannTN:

a python package for Grassmann tensor networks

The screenshot shows the GitHub repository page for 'grassmanntn'. At the top, the repository name 'grassmanntn' is displayed with a 'Public' badge. To the right are buttons for 'Pin', 'Unwatch' (3), 'Fork' (0), and 'Star' (6). Below this, there are navigation buttons for 'main', 'Go to file', 'Add file', and 'Code'. The repository description is 'A python package for Grassmann tensor network computation'. The file list shows a commit by 'ayosprakob' 5 days ago with 111 changes. The files listed are 'docs', 'LICENSE', 'README.md', '\_\_init\_\_.py', 'example.py', 'gauge2d.py', and 'param.py'. The right sidebar shows 'About' information: 'Readme', 'Apache-2.0 license', 'Activity', '6 stars', '3 watching', and '0 forks'. Below that, the 'Releases' section shows 17 releases, with the latest being 'v 1.2.3' from 3 weeks ago.

grassmanntn Public

Pin Unwatch 3 Fork 0 Star 6

main Go to file Add file Code

Branches Tags

ayosprakob Update the arxiv link ... 5 days ago 111

docs	Update the arxiv link	5 days ago
LICENSE	Initial commit	4 months ago
README.md	Update README.md	5 days ago
__init__.py	update gauge2d.trg with more o...	2 months ago
example.py	Update the quadrature function	3 weeks ago
gauge2d.py	Update the quadrature function	3 weeks ago
param.py	add trg function (incomplete) & ...	4 months ago

About

A python package for Grassmann tensor network computation

Readme Apache-2.0 license Activity 6 stars 3 watching 0 forks

Releases 17

v 1.2.3 Latest 3 weeks ago

<https://github.com/ayosprakob/grassmanntn>