Introduction to Matrix Product State: Detecting Symmetry Protected Topological Phase

Department of Physics, **Tsuyoshi Okubo**

理学系研究科　物理学専攻　大久保　毅

Ref: Lecture of F. Pollmann at TNQMP2016

You can find contents (slides and movies) of the lectures at

http://www.issp.u-tokyo.ac.jp/public/tnqmp2016/

Outline

- Symmetry of MPS
    - Effect of "local" symmetry to iMPS
    - Possibility of non-trivial phase
    - Important symmetry of S=1 spin chain
    - Expected behavior in Haldane phase and Large-D phase
    - Detection of SPT phase from iMPS
- **Exercise 5: Observe degeneracy in entanglement spectrum**
- **Exercise 6: Observe non-trivial phases**

Symmetry

Symmetry operation:

$\hat{O}_g$:symmetry operator

$$|\tilde{\Psi}\rangle = \hat{O}_g|\Psi\rangle$$

$$g \in G$$

When a quantum state is invariant under a symmetry

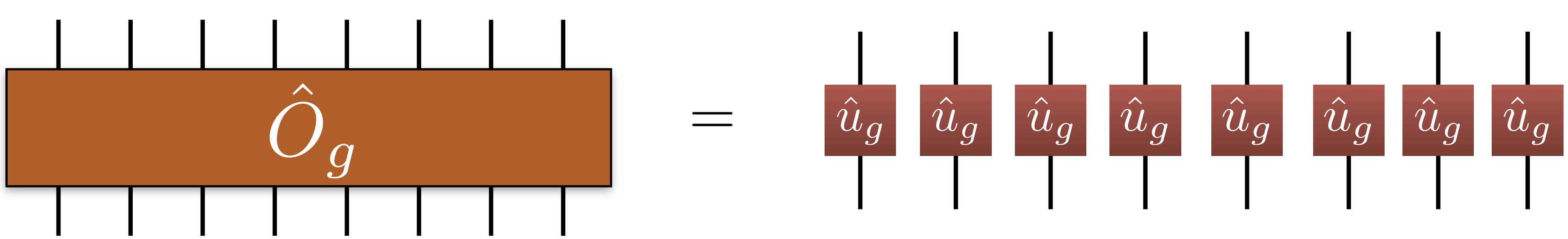
The state after symmetry operation is invariant up to a phase.

$$|\tilde{\Psi}\rangle = e^{i\theta_g}|\Psi\rangle$$

"Local" symmetry:

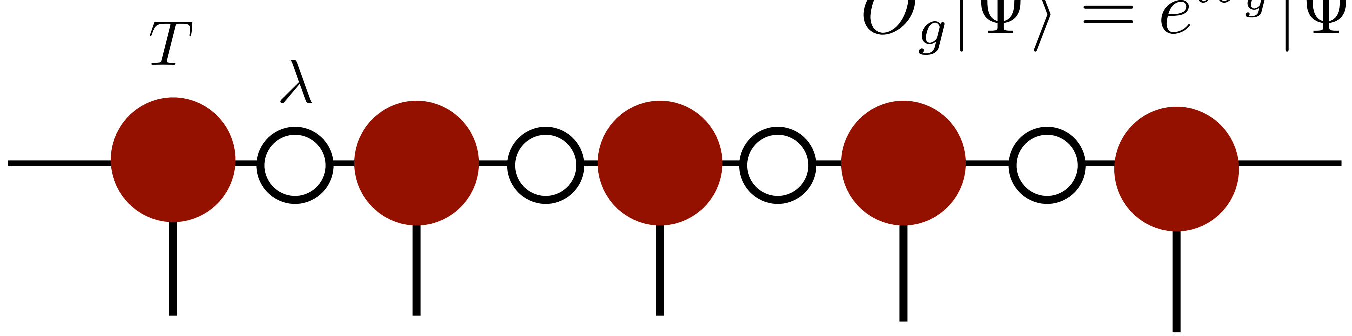
The symmetry operator can be represented as a product of local operators:

$$\hat{O}_g = \cdots \hat{u}(g)_1 \otimes \hat{u}(g)_2 \otimes \cdots \hat{u}(g)_i \otimes \cdots$$

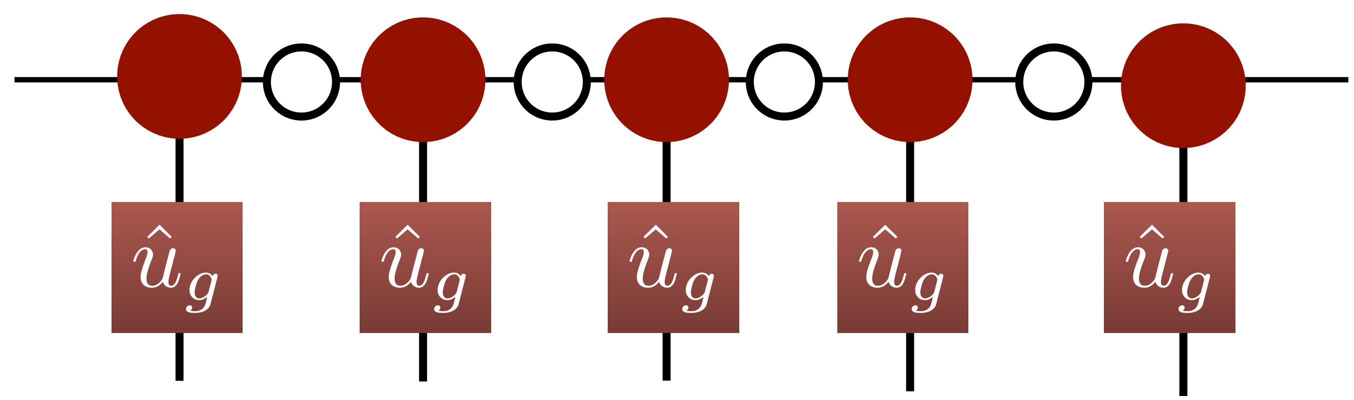


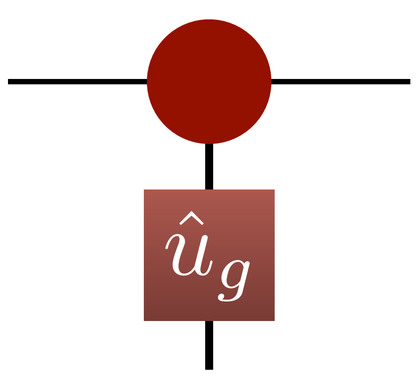
Effect of local symmetry to iMPS

$$\hat{O}_g|\Psi\rangle = e^{i\theta_g}|\Psi\rangle$$

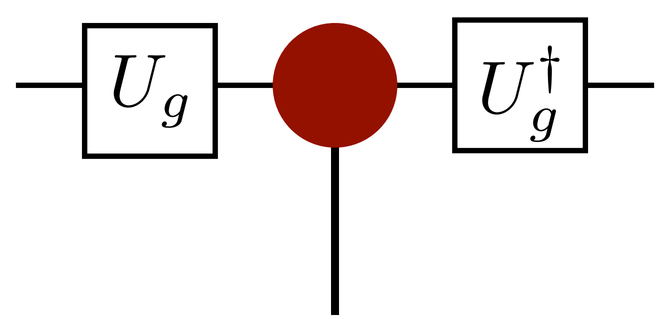


$= \quad e^{i\theta_g}$





$= \quad e^{i\phi_g}$



U_g :Unitary matrix

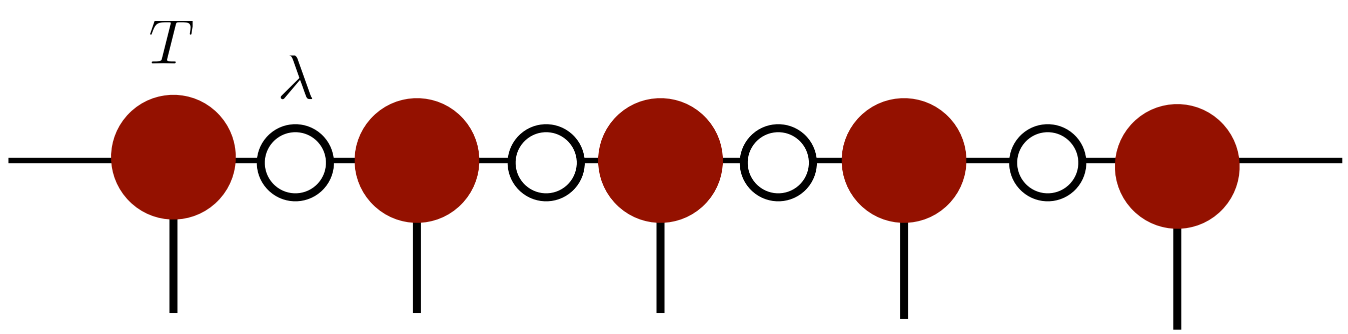
Commute with λ

$$U_g\lambda = \lambda U_g$$

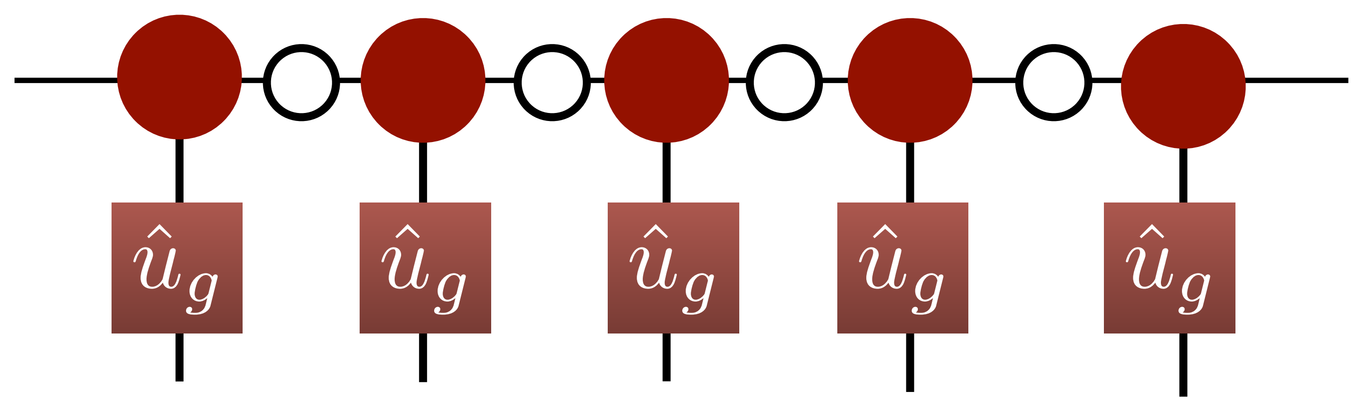
*Note: $U_g' = e^{ix}U_g$ also satisfies the same relation.

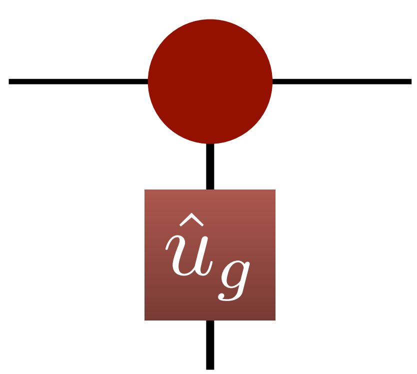
Effect of local symmetry to iMPS

$$\hat{O}_g|\Psi\rangle = e^{i\theta_g}|\Psi\rangle$$

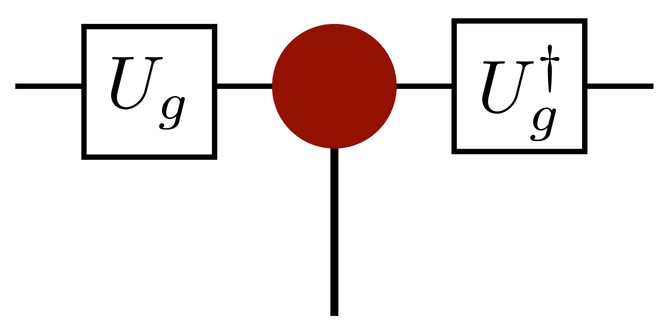


$$= \quad e^{i\theta_g}$$





$$= \quad e^{i\phi_g}$$



U_g :Unitary matrix

Commute with λ

$$U_g\lambda = \lambda U_g$$

(D. Pérez-García, et al,Phys. Rev. Lett. **100**, 167202 (2008))

Representation and projective representation

$g, g' \in G$

Symmetry operator is a representation of the symmetry group G:

$$\hat{u}_g \hat{u}_{g'} = \hat{u}_{gg'}$$

Matrix U_g satisfies a similar relation with a phase:

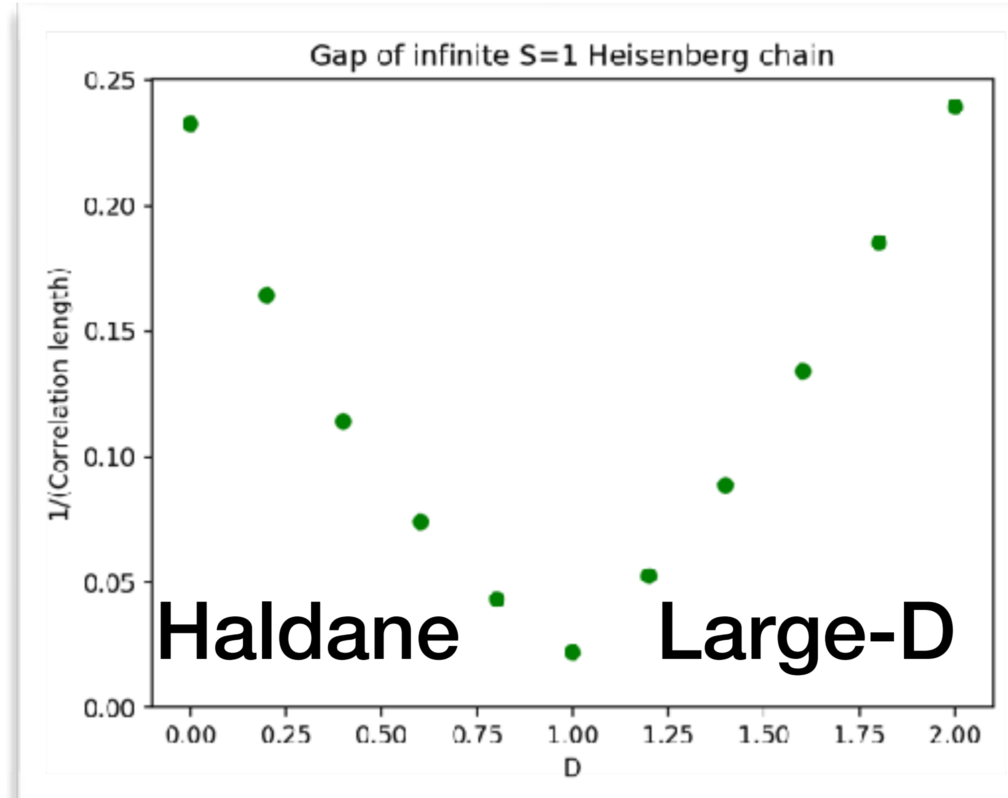
$$U_g U_{g'} = e^{i\psi_{gg'}} U_{gg'}$$

"Projective representation"

If the phase ψ cannot be absorbed by a rescaling,
the state is distinguished from a trivial state.

Important symmetry of the S=1 spin chain

$$\mathcal{H} = \sum_i \vec{S}_i \cdot \vec{S}_{i+1} + D \sum_i S_{z,i}^2$$



Spin rotation:

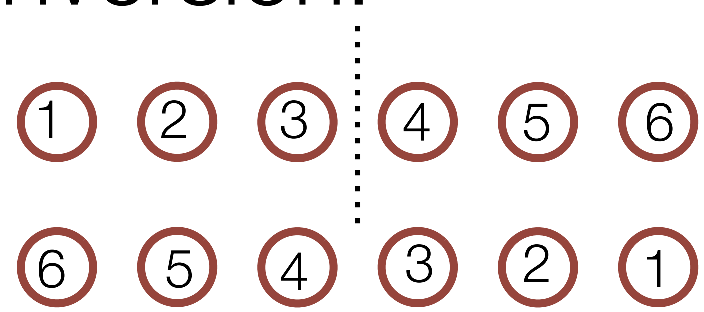
eg. π rotation

$$R_x = e^{i\pi S_x}, R_y = e^{i\pi S_y}, R_z = e^{i\pi S_z}$$

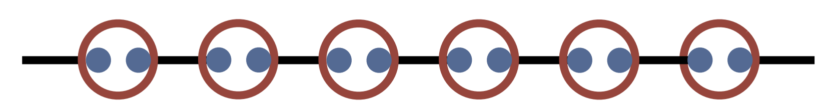
Time reversal:

$$\mathcal{T}|\Psi\rangle = R_y|\Psi^*\rangle$$

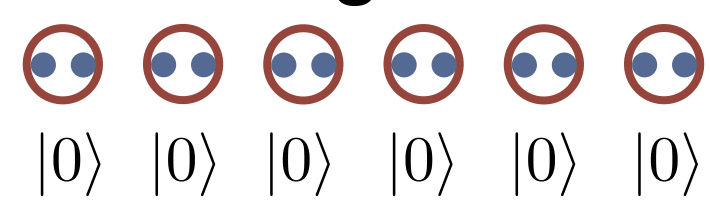
Space Inversion:



Haldane

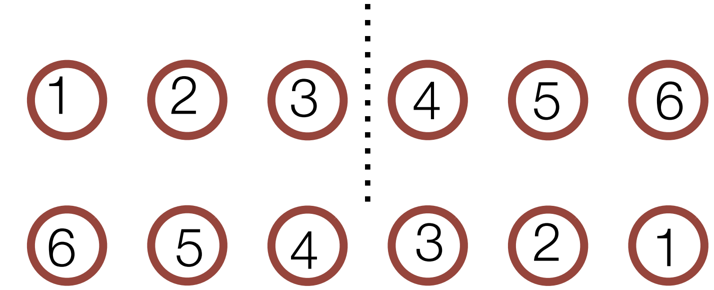


Large-D



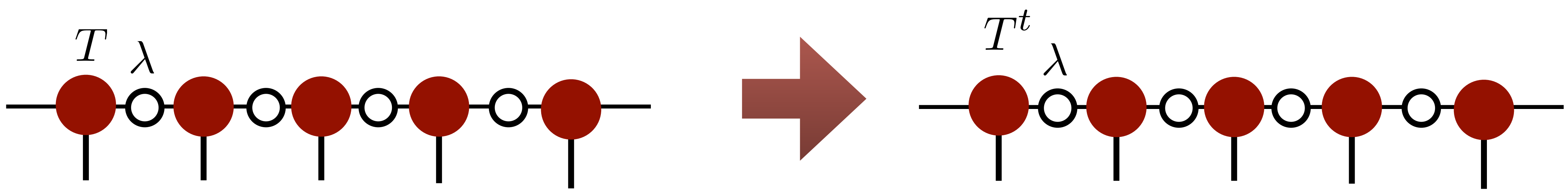
Both of Haldane and Large-D phases keep these symmetries.

Space inversion in iMPS

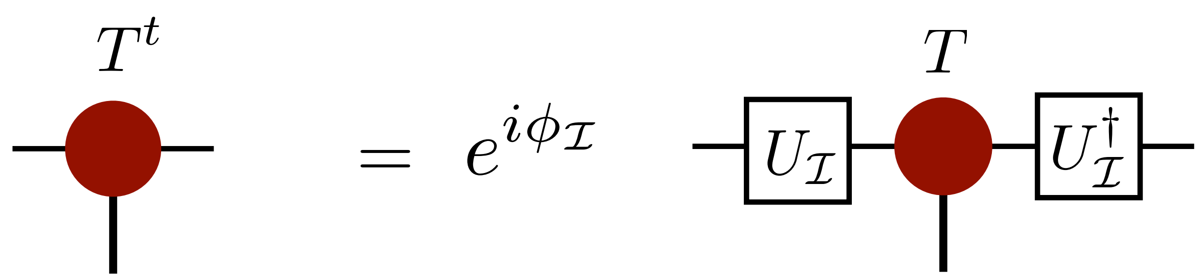


(F. Pollmann, et al, Phys. Rev. B **81**, 064439 (2010))
(F. Pollmann, et al, Phys. Rev. B **86**, 125441 (2012))

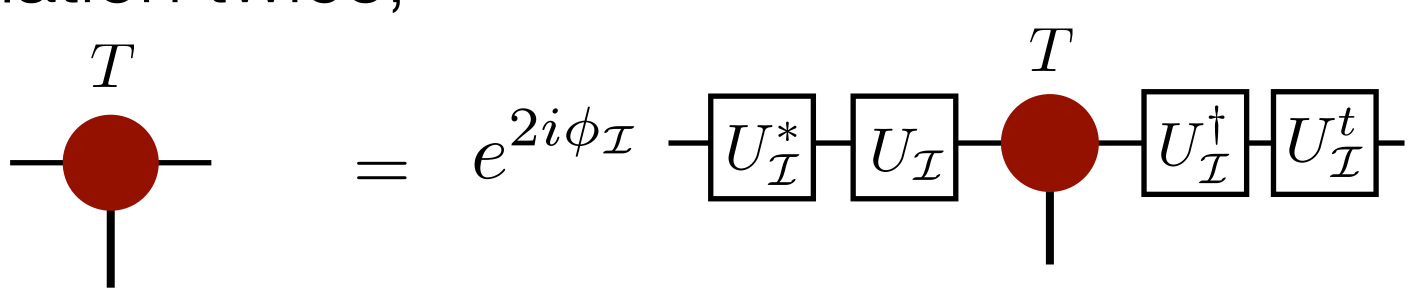
Space inversion:



Take transpose of the matrix T



Apply this relation twice,



From the condition of the canonical form,

$$e^{2i\phi_{\mathcal{I}}} = 1\,, U_{\mathcal{I}} U_{\mathcal{I}}^* = e^{i\psi_{\mathcal{I}}} \Rightarrow \boxed{\psi_{\mathcal{I}} = 0, \pi}$$

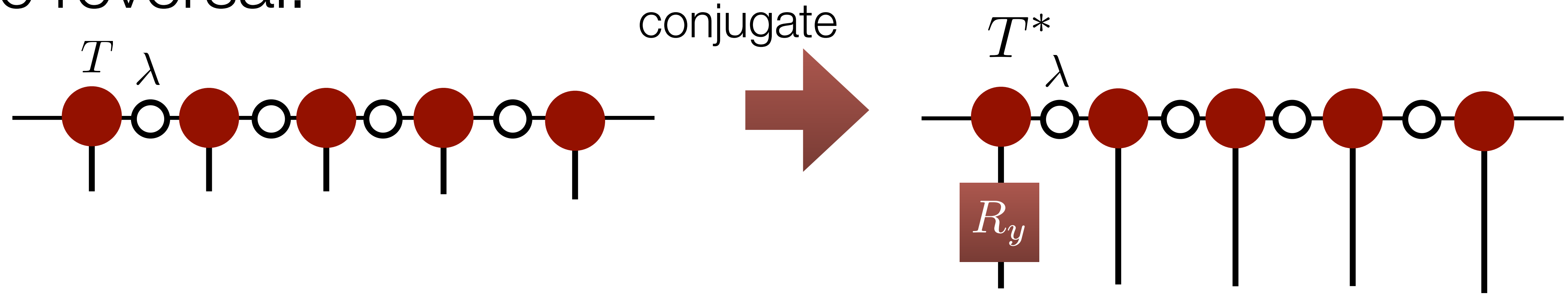
Time reversal in iMPS

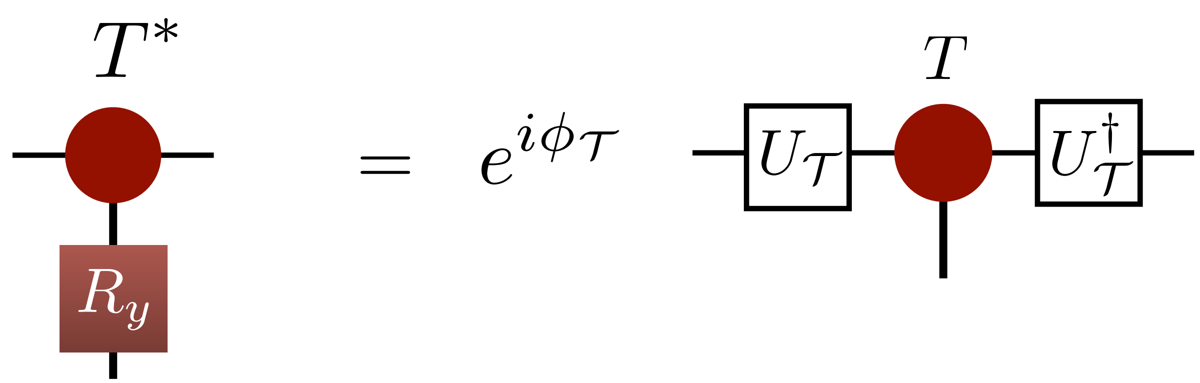
$$\mathcal{T}|\Psi\rangle = R_y|\Psi^*\rangle$$

(F. Pollmann, et al, Phys. Rev. B **81**, 064439 (2010))
(F. Pollmann, et al, Phys. Rev. B **86**, 125441 (2012))

Time reversal:

Take complex conjugate





From the same logic with the inversion,

$$U_{\mathcal{T}}U_{\mathcal{T}}^* = e^{i\psi_{\mathcal{T}}}$$

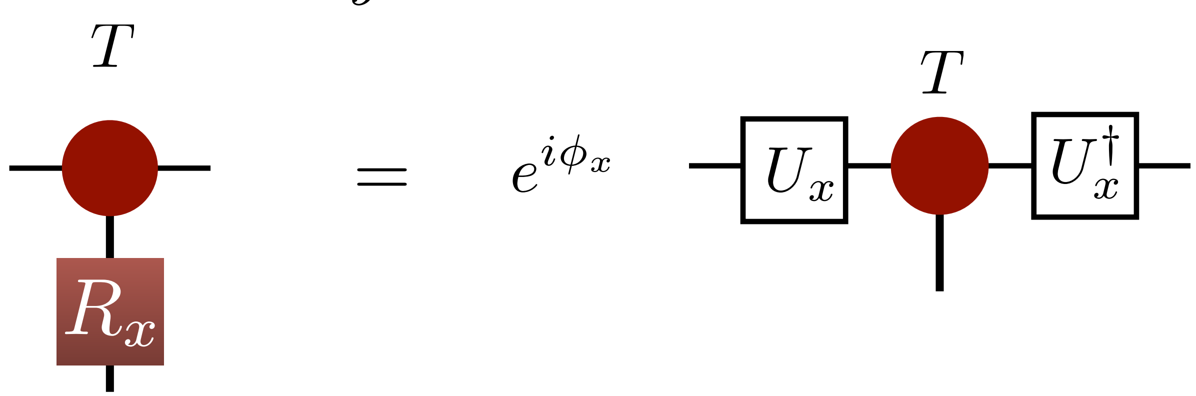
$$\boxed{\psi_{\mathcal{T}} = 0, \pi}$$

Set of rotations $R_x R_y |\Phi\rangle$

(F. Pollmann, et al, Phys. Rev. B **81**, 064439 (2010))
(F. Pollmann, et al, Phys. Rev. B **86**, 125441 (2012))

Single rotation always gives a trivial phase.
However, set of rotations can be non-trivial.

$$R_x = e^{i\pi S_x}, R_y = e^{i\pi S_y}, R_z = e^{i\pi S_z}$$



(Similar relations for R_y and R_z)

From the relation $R_x R_z = R_z R_x = R_y$

$$U_x U_z = e^{i\psi_{\mathcal{D}_2}} U_z U_x$$

$$U_x U_z U_x^\dagger U_z^\dagger = e^{i\psi_{\mathcal{D}_2}}$$

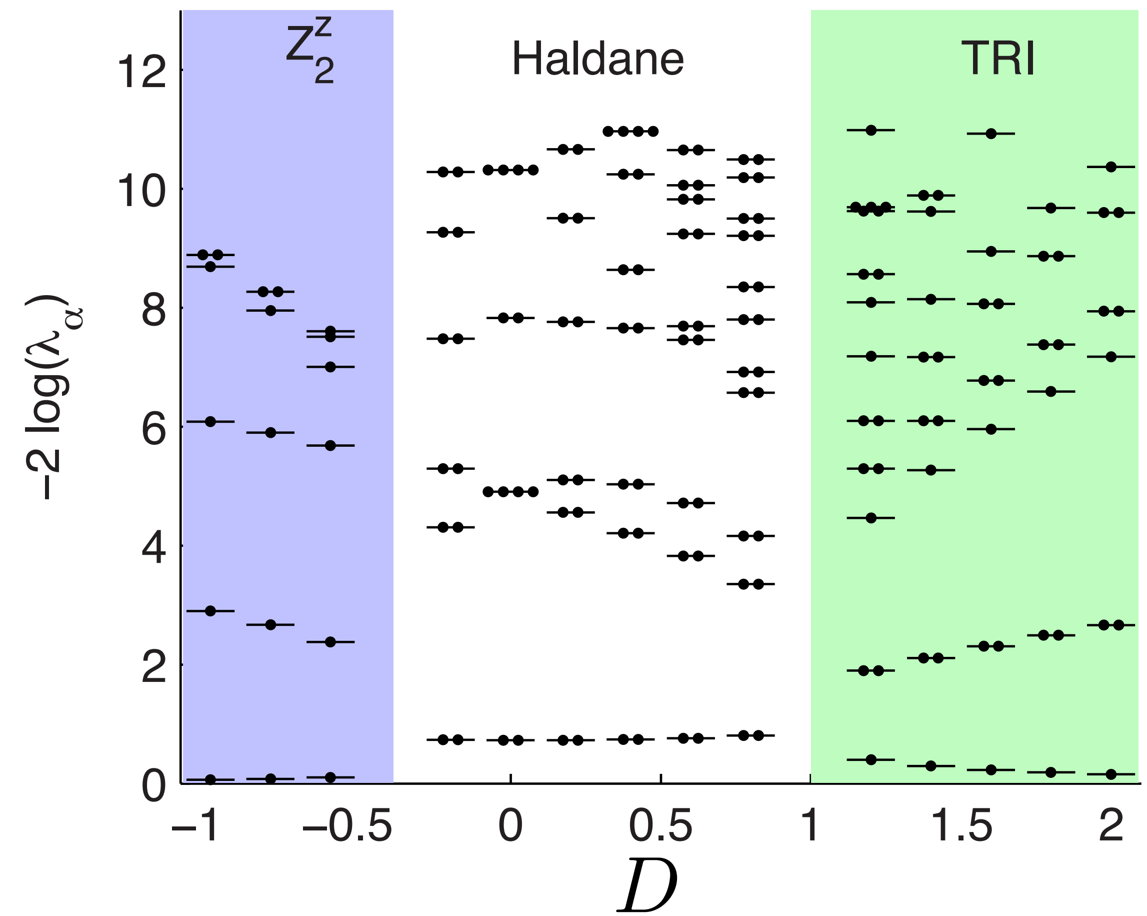
$$\boxed{\phi_{\mathcal{D}_2} = 0, \pi}$$

Expected behavior in Haldane and large-D phases

(F. Pollmann, et al, Phys. Rev. B **81**, 064439 (2010))

Phase	Time reversal	Inversion	RxRz
Haldane	π	π	π
Large-D	0	0	0

Spectrum of Schmidt coefficient



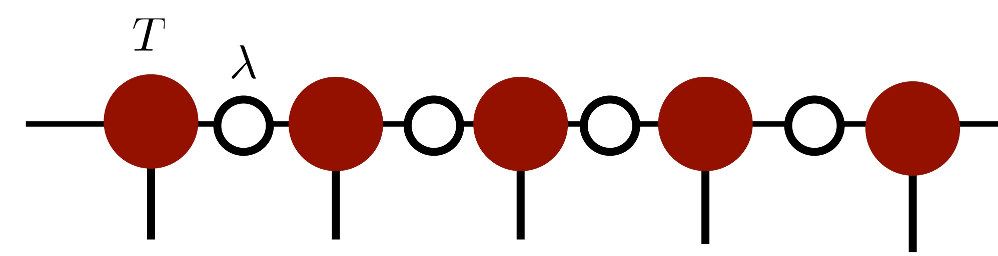
If the phase is π, all Schmidt coefficients are degenerated in even number.

By observing Schmidt coefficients, we can distinguish Haldane and large-D phases.

Detection of SPT from iMPS

(F. Pollmann, et al, Phys. Rev. B **81**, 064439 (2010))
(F. Pollmann, et al, Phys. Rev. B **86**, 125441 (2012))

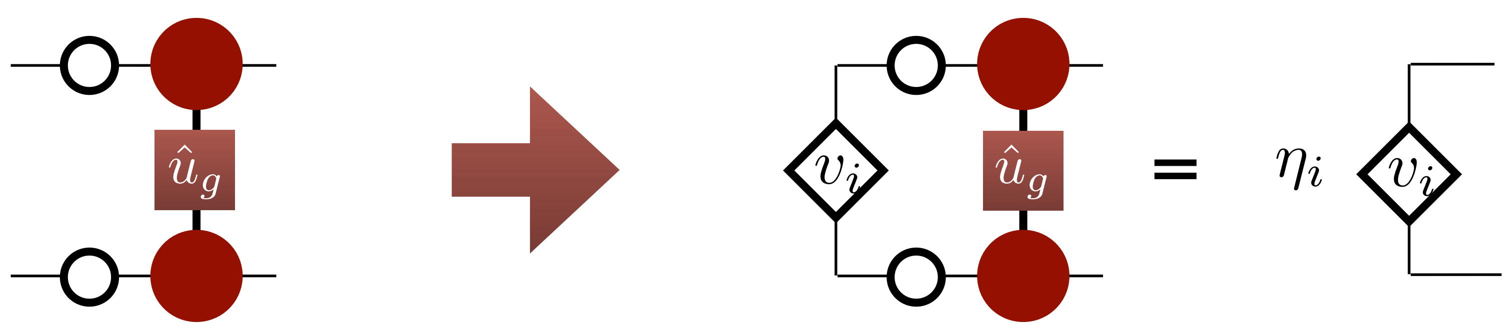
1. Just observe λ and check the degeneracy

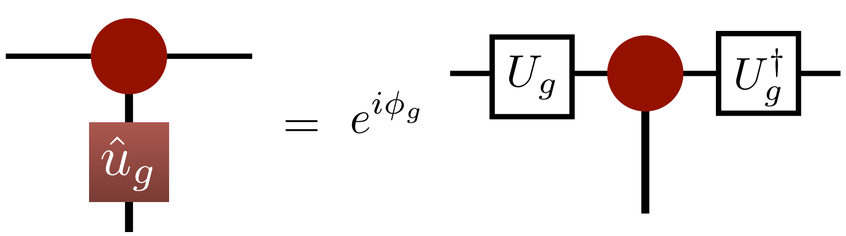


2. Directly calculate a non-trivial phase

We can calculate the matrix U_g from a transfer matrix

Transfer matrix **eigen value problem**





Dominant eigen vector is U_g

$$\eta = e^{-i\phi_g}, v_i = U_g^\dagger$$

Note: if the state does not have symmetry,

$$|\eta| < 1$$

Exercise 5: Observe entanglement spectrum

Simulate infinite system and output singular value λ.

Sample code: Ex5.py

python Ex5.py

- Compare degeneracies varying parameter "D"
    - In Haldane phase, all λs are expected to degenerated in even number
- By using TEBD (or ED) instead of iTEBD, check the degeneracy in a finite system

Exercise 6: Calculate non-trivial phases

Simulate infinite system and output $e^{i\psi}$ for

- Time reversal
- Inversion
- $R_x R_y$

Sample code: Ex6.py

python Ex6.py

- Compare phases varying parameter "D"
  - In Haldane phase, all phases are expected to be π, while it is 0 in large D phase.
  - Plot $e^{i\psi}$ as a function of D. (By combining Ex4.py and Ex6.py, you might make a code for this purpose.)
- For large negative D, the GS is an ordered state which breaks symmetries.
  - Check that the dominant eigenvalue of the transfer matrix is less than 1

Ex6.py

```python
## time reversal
Ry = linalg.expm(0.5*np.pi * (Sp -Sm))
eig, UTR = iTEBD.Transfer_Matrix_SPT_TR(Tn,lam,Ry)

UTR= UTR.reshape(chi,chi).T

UTR /= np.sqrt(np.trace(np.dot(UTR,UTR.T.conj()))/chi)
print "##Time reversal",eig, np.trace(np.dot(UTR,UTR.conj()))/chi

## Inversion
eig_I, UI = iTEBD.Transfer_Matrix_Inv_bond(Tn,lam)
UI = UI.reshape(chi,chi).T
UI /= np.sqrt(np.trace(np.dot(UI,UI.T.conj()))/chi)

print "##Inversion",eig_I, np.trace(np.dot(UI,UI.conj()))/chi

## set of rotation (D2)

Rx = linalg.expm(0.5j * np.pi * (Sp+Sm))
Rz = linalg.expm(1.0j * np.pi * Sz)
eigx, Ux = iTEBD.Transfer_Matrix_SPT(Tn,lam,Rx)
eigz, Uz = iTEBD.Transfer_Matrix_SPT(Tn,lam,Rz)

Ux = Ux.reshape(chi,chi).T
Ux /= np.sqrt(np.trace(np.dot(Ux,Ux.T.conj()))/chi)

Uz = Uz.reshape(chi,chi).T
Uz /= np.sqrt(np.trace(np.dot(Uz,Uz.T.conj()))/chi)

print "##D2",eigx,eigz, np.trace(np.dot(np.dot(np.dot(Ux,Uz),Ux.T.conj()),Uz.T.conj()))/chi
```

Exponential of a matrix is easily calculated by

scipy.linalg.expm(Mat)

iTEBD.Transfer_Matrix_SPT(Tn,lam,op) returns dominant eigenvalue and eigenvectors for a symmetry operator "op".

For time reversal and space inversion, we need special routines: **iTEBD.Transfer_Matrix_SPT_TR**, **iTEBD.Transfer_Matrix_Inv_bond**, to treat a "complex conjugate" and "transpose" operations.

$$U_{\mathcal{T}} U_{\mathcal{T}}^* = e^{i\psi_{\mathcal{T}}} \quad \Rightarrow \quad \mathrm{Tr}\, U_{\mathcal{T}} U_{\mathcal{T}}^* = \chi e^{i\phi_{\mathcal{T}}}$$

$$U_{\mathcal{I}} U_{\mathcal{I}}^* = e^{i\psi_{\mathcal{I}}} \quad \Rightarrow \quad \mathrm{Tr}\, U_{\mathcal{I}} U_{\mathcal{I}}^* = \chi e^{i\phi_{\mathcal{I}}}$$

$$U_x U_z U_x^\dagger U_z^\dagger = e^{i\psi_{\mathcal{D}_2}} \quad \Rightarrow \quad \mathrm{Tr}\, U_x U_z U_x^\dagger U_z^\dagger = \chi e^{i\phi_{\mathcal{D}_2}}$$