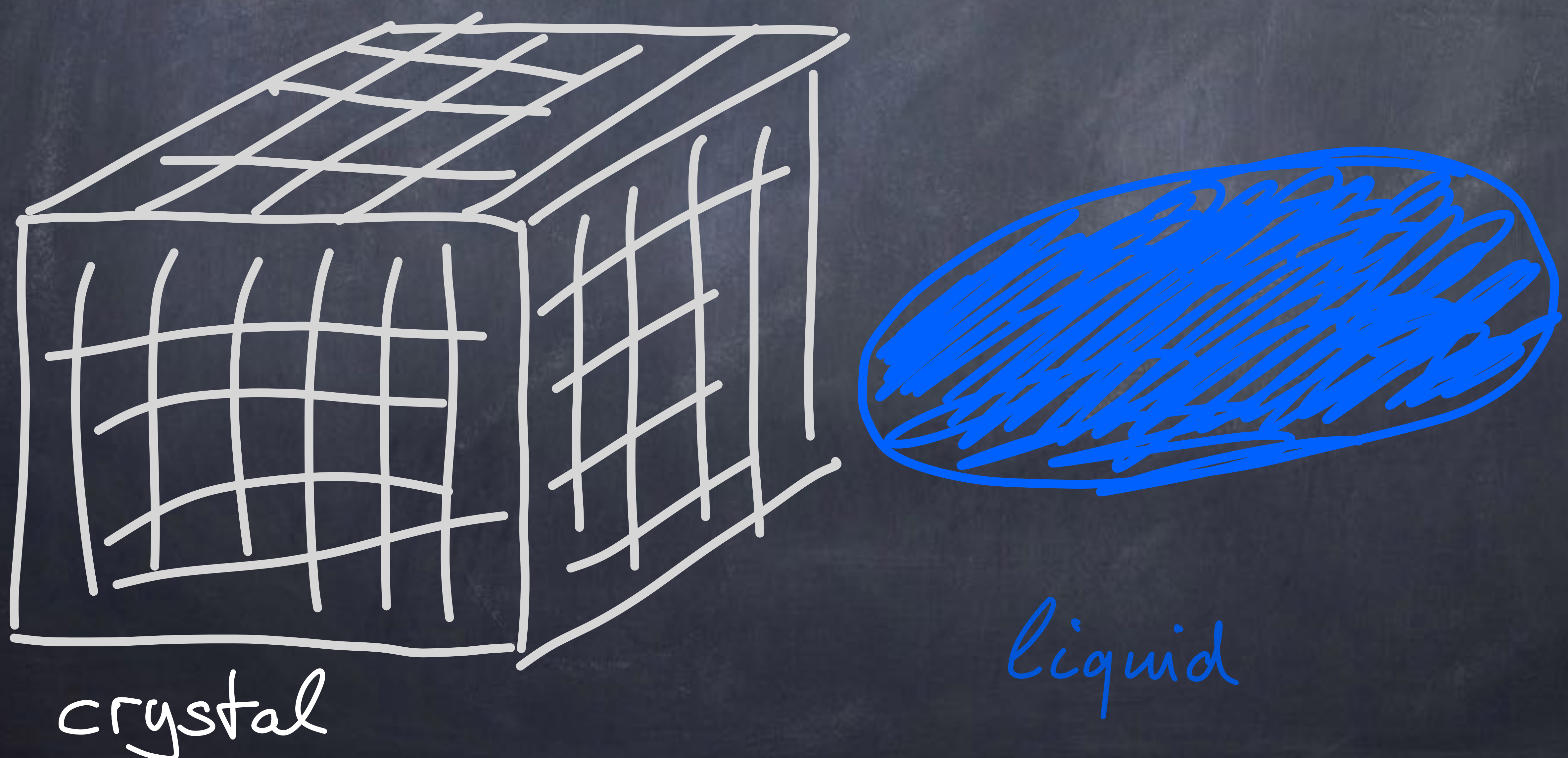
Machine learning techniques for phase detection in stat. phys. problems

What is a phase?

What is a phase ?



How do we distinguish?

order parameter

$$\rho(\vec{r}) = \rho_0 \sum_{\{\vec{R}\}} \delta(\vec{r}-\vec{R})$$

How do we distinguish?

order parameter

$$\rho(\vec{r}) = \rho_0 \sum_{\{\vec{R}\}} \delta(\vec{r}-\vec{R})$$

correlation function

$$\langle \rho(\vec{r})\rho(\vec{r}\,') \rangle \sim \rho_0^2 \qquad \forall\ \vec{r},\vec{r}\,' \in \{\vec{R}\}$$

Possible problems

- no order parameter
  - KT transition
  - topological transition
  - gauge theory
  - ...

- order parameter unknown

Possible problems

- no order parameter
  - KT transition
  - topological transition
  - gauge theory
  - ...

- order parameter unknown

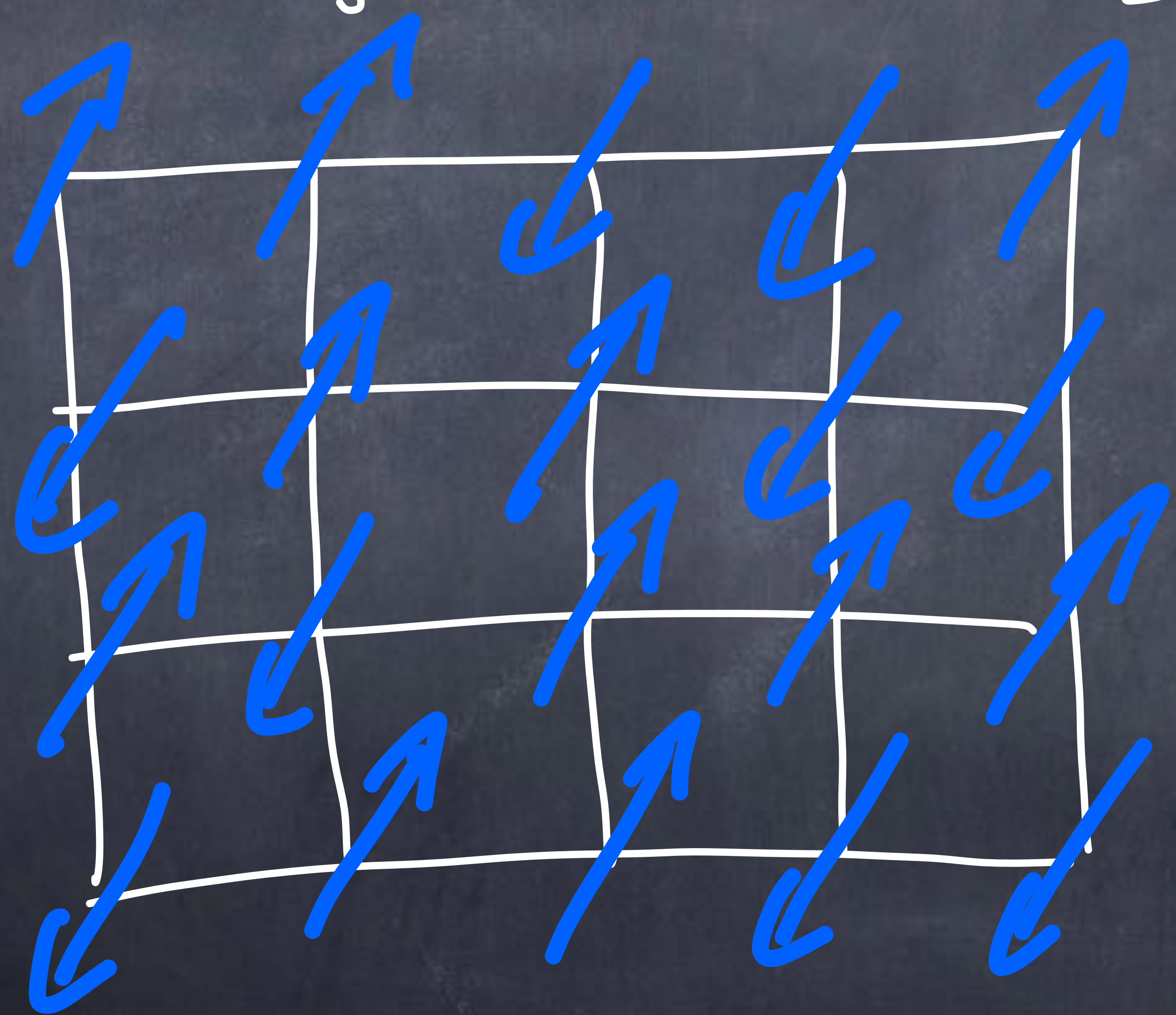
Can AI/ML help us?

Simple example: Ising model

$$H = -J \sum_{\langle ij \rangle} S_i S_j$$

$S_i = \pm 1$ $\quad \langle ij \rangle$ nearest neighbor

$$J \begin{cases} > 0 & \text{ferromag} \\ < 0 & \text{antiferromag} \end{cases}$$



best studied model in Stat. Mech.

Some properties

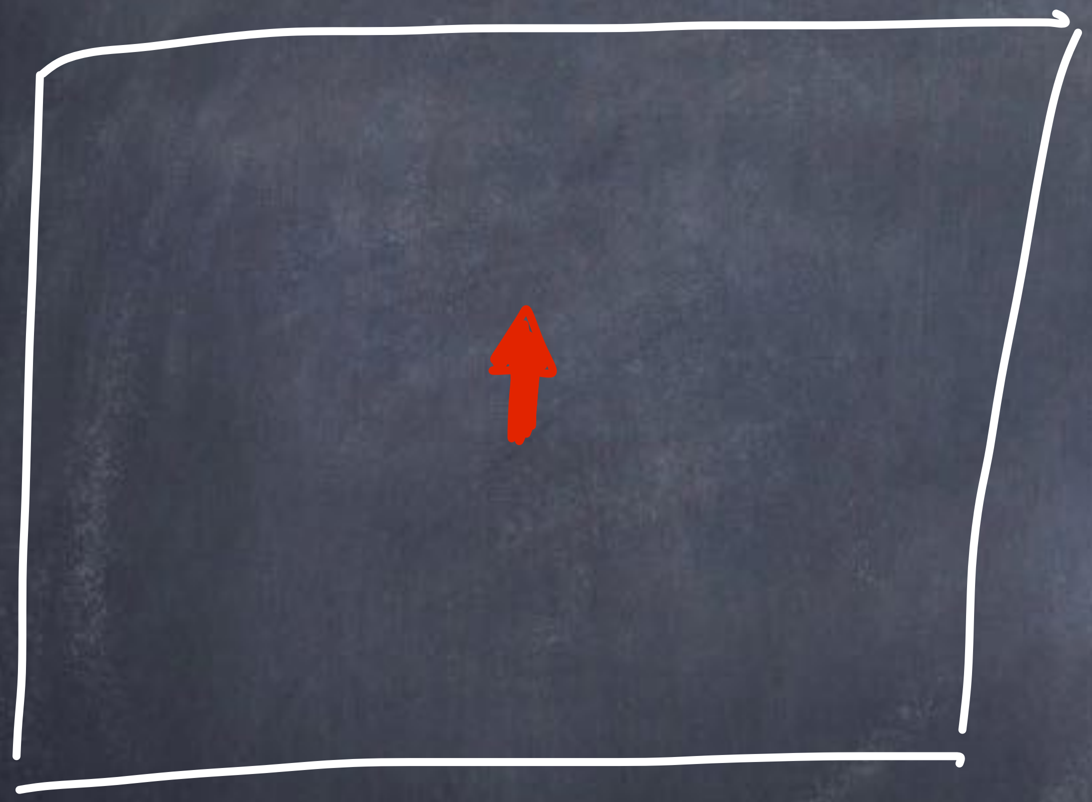
order parameter

$$m = \begin{cases} 0 & T > T_c \\ \text{finite} & T < T_c \end{cases}$$

correlation function

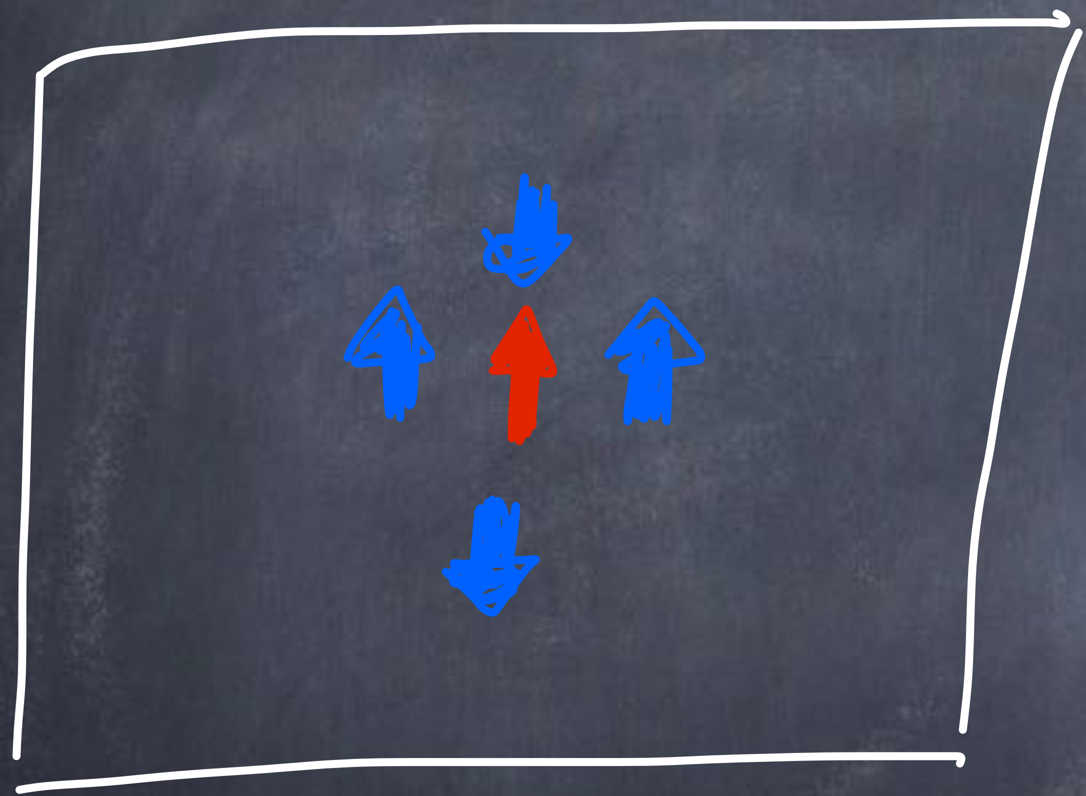
$$\langle S_i S_j \rangle = \begin{cases} \dfrac{1}{|r_i - r_j|^x} e^{-\frac{|r_i - r_j|}{\xi}} & T > T_c \\ \dfrac{1}{|r_i - r_j|^x} & T = T_c \\ \text{constant} & T < T_c \end{cases}$$

Monte Carlo Simulation



I pick at random spin

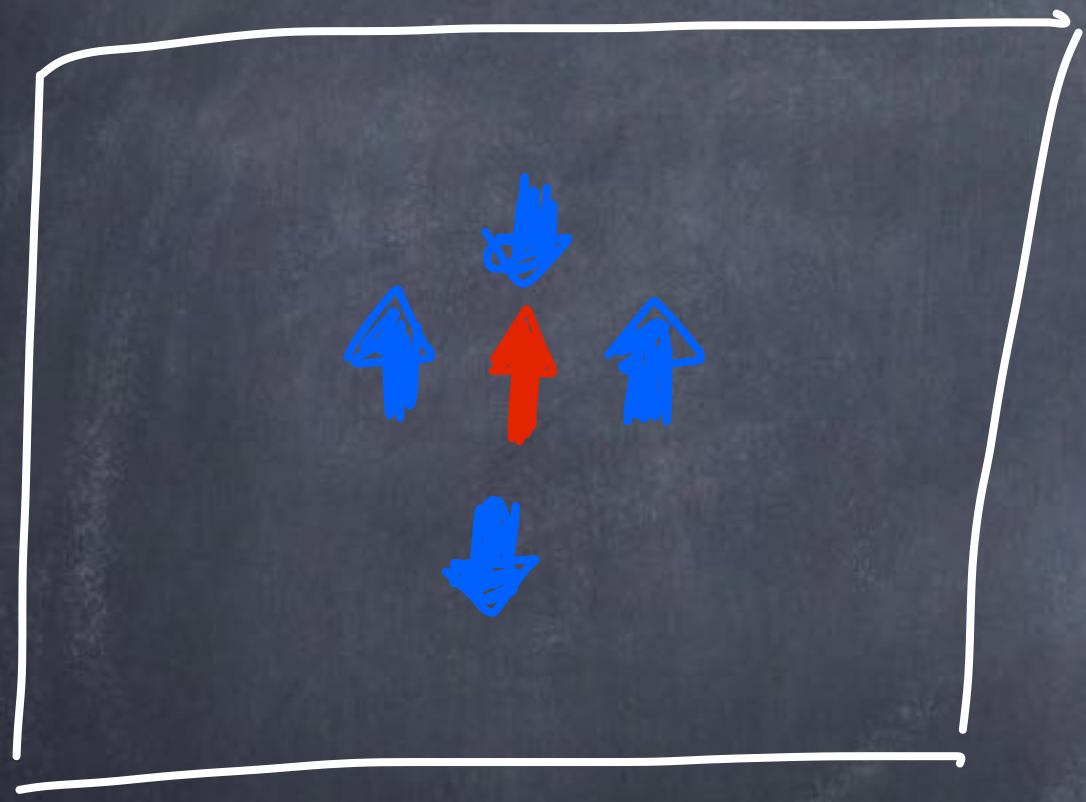
Monte Carlo Simulation



I pick at random spin

II determine neighbors ($\langle ij \rangle$)

Monte Carlo Simulation

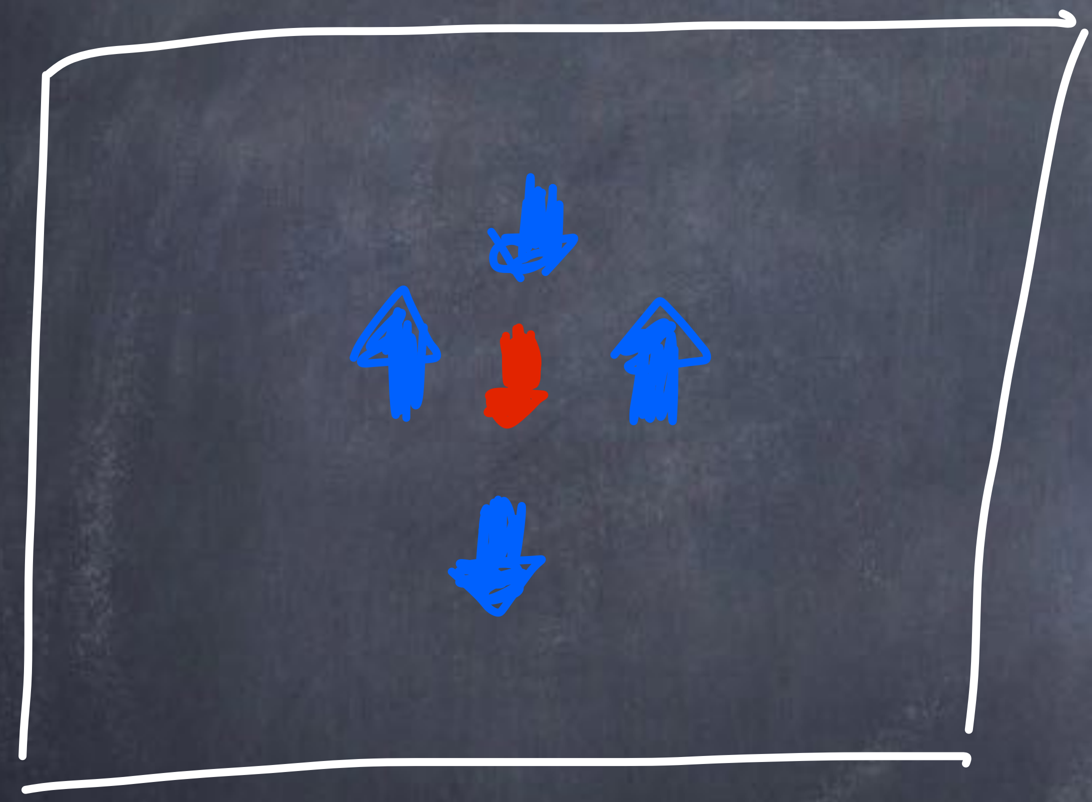


I pick at random spin

II determine neighbors ($\langle ij \rangle$)

III E_{config}

Monte Carlo Simulation



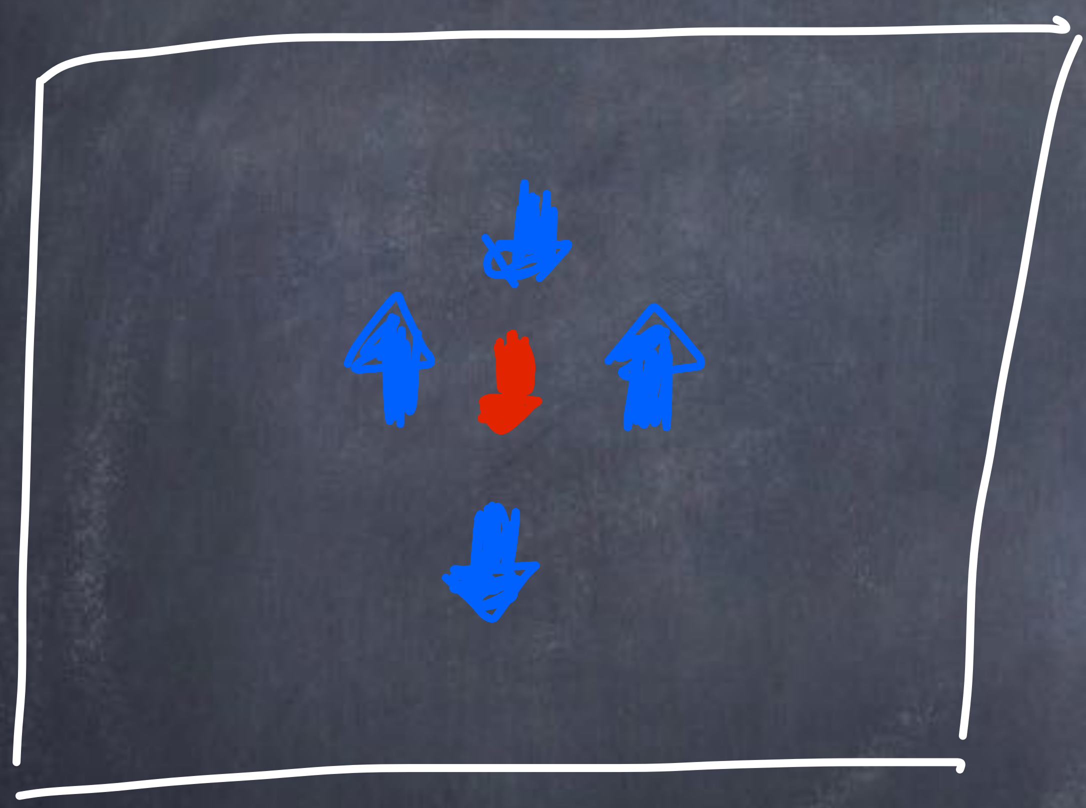
I pick at random spin

II determine neighbors ($\langle ij \rangle$)

III E_{config}

IV. reverse spin, E_{new}

Monte Carlo Simulation



I pick at random spin

II determine neighbors ($\langle ij \rangle$)

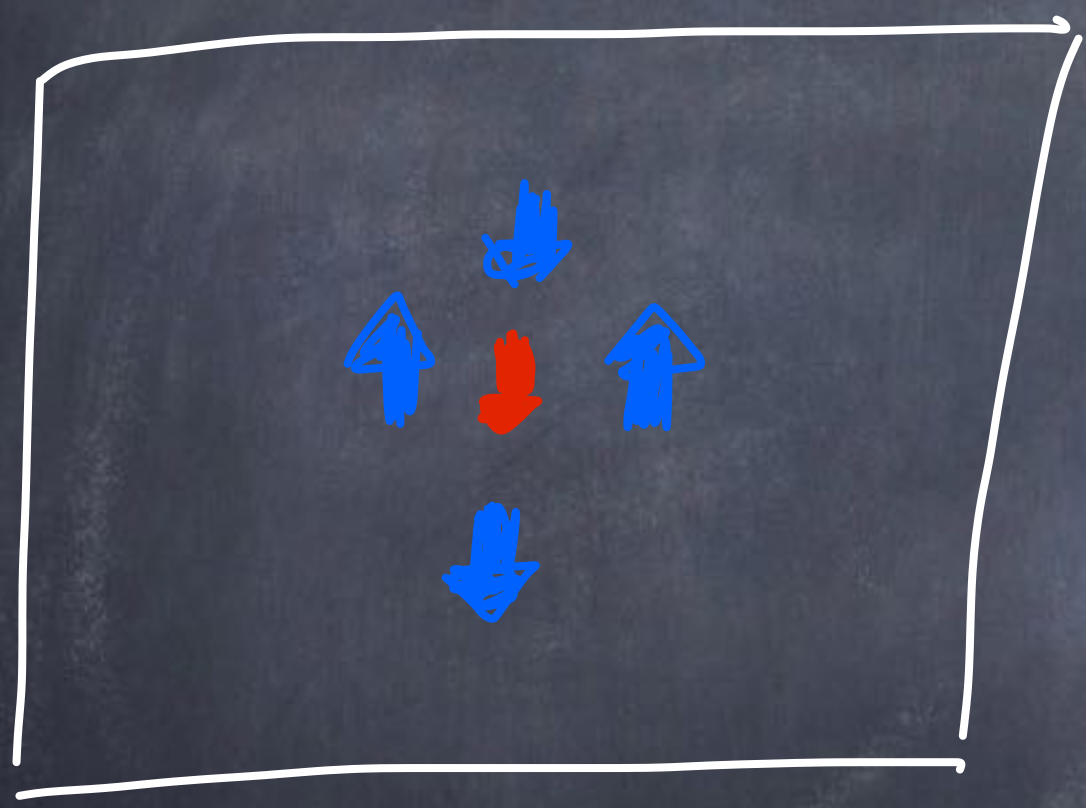
III E_{config}

IV reverse spin, E_{new}

V accept with probability

$$p = e^{-\frac{E_{new} - E_{config}}{T}}$$

Monte Carlo Simulation



I pick at random spin

II determine neighbors ($\langle ij \rangle$)

III E_{config}

IV reverse spin, E_{new}

V accept with probability

$$p = e^{-\frac{E_{new} - E_{config}}{T}}$$

..... many times .....

Supervised vs. Unsupervised

supervised: labeled sample

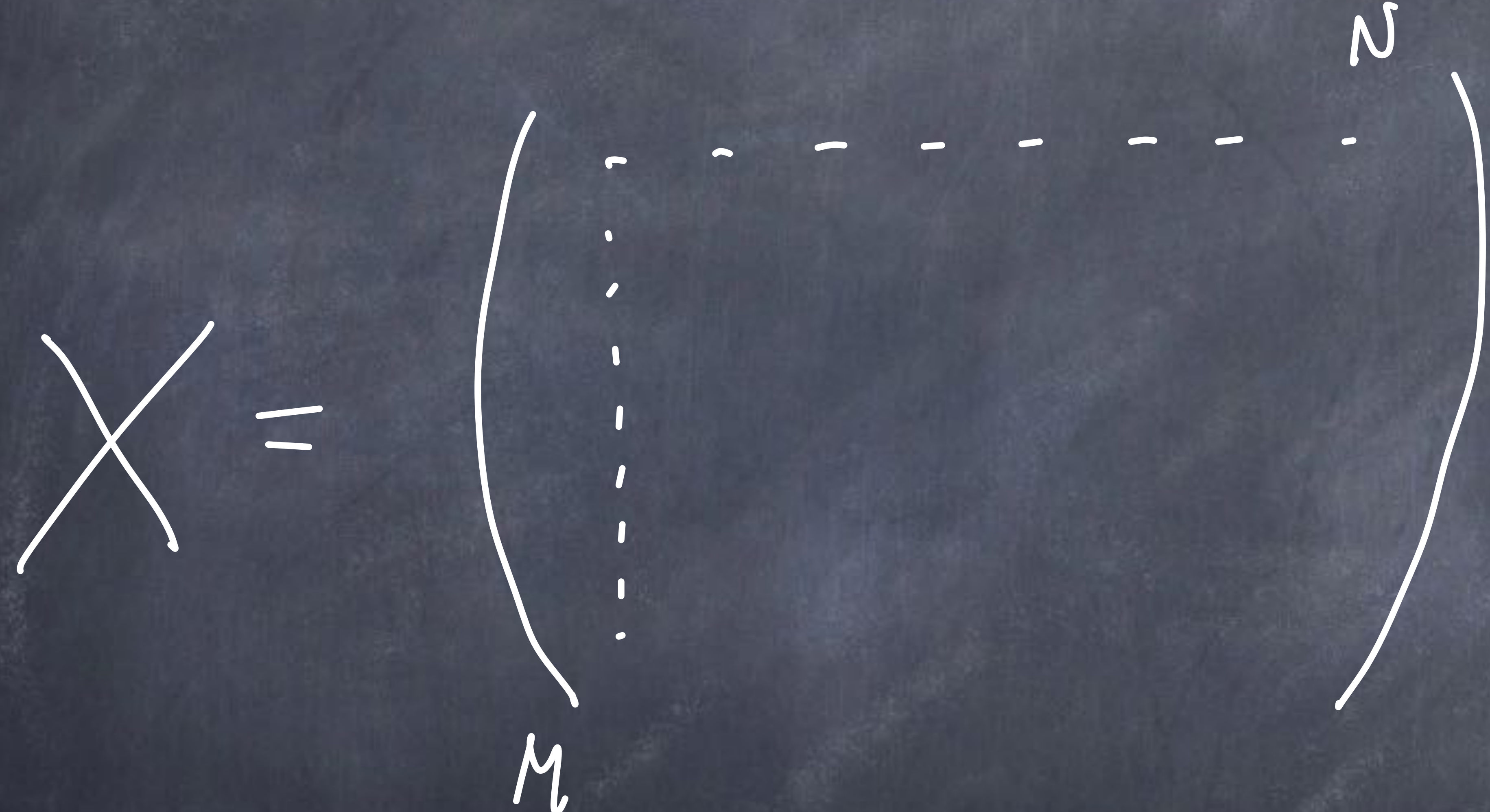
unsupervised: unlabeled sample

Supervised vs. Unsupervised

supervised: labeled sample

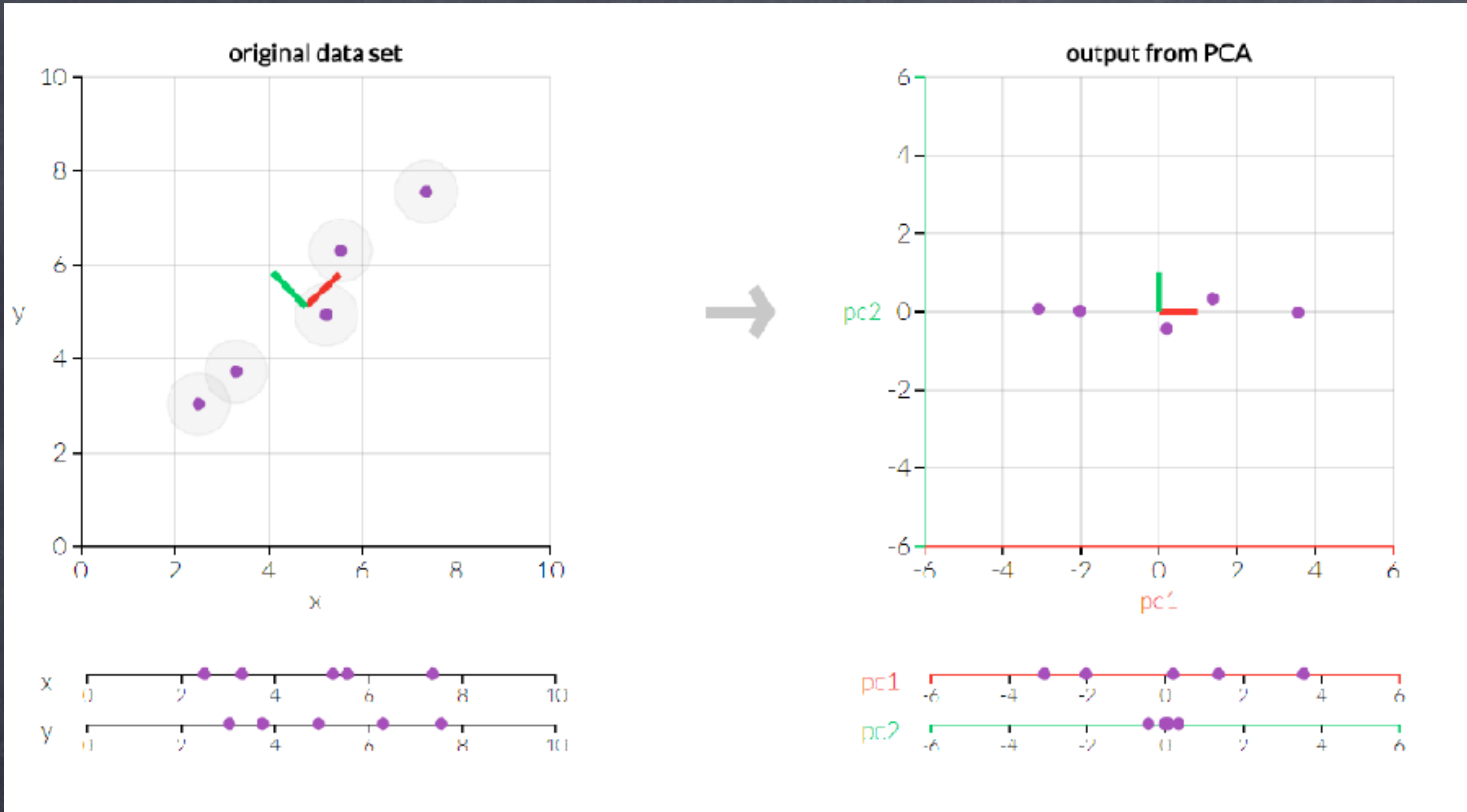
unsupervised: unlabeled sample

P(rincipal) C(omponent) A(nalysis)



idea: find most efficient basis to represent data

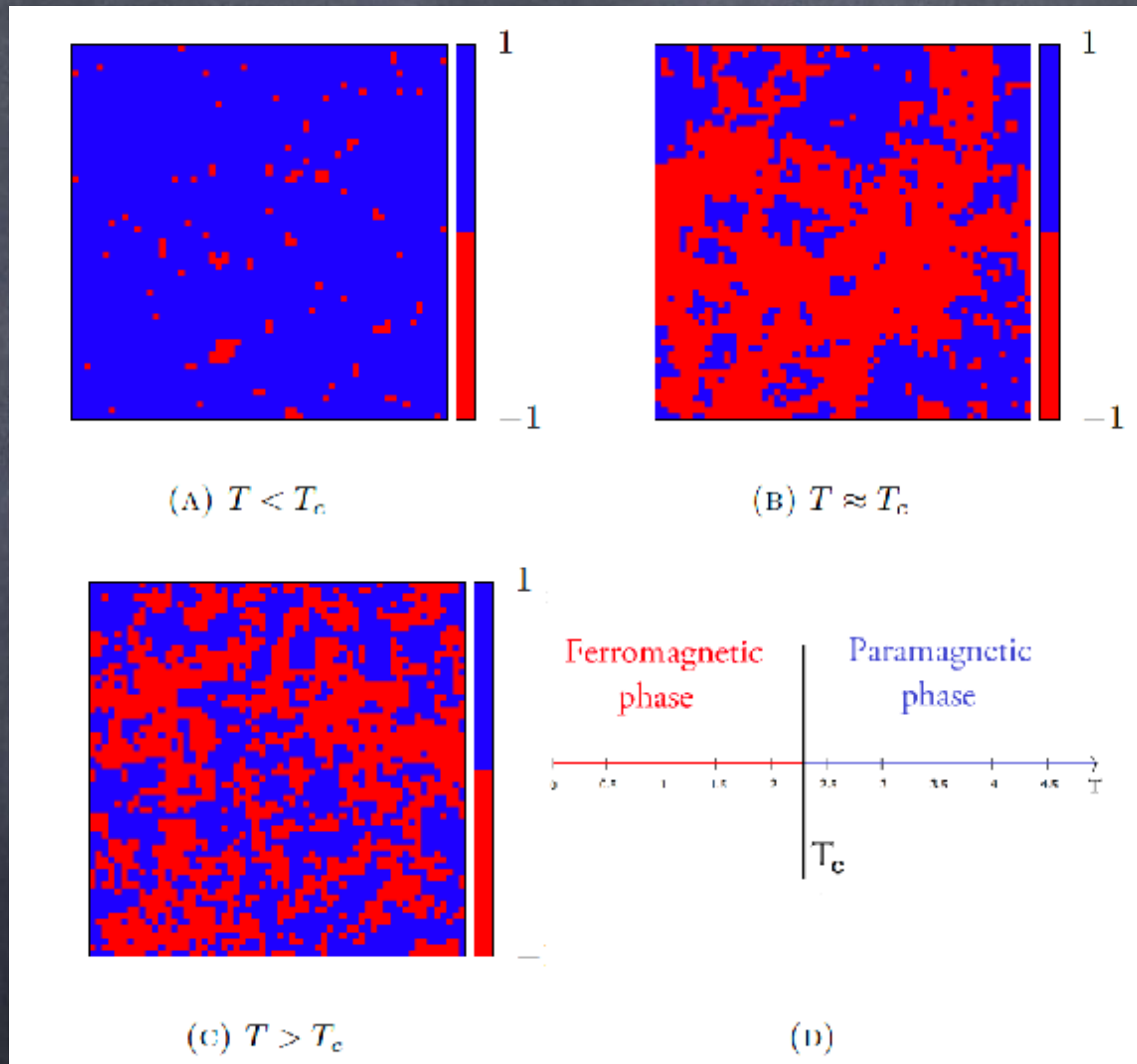
(N: # of lattice sites ; M: # of snapshots)

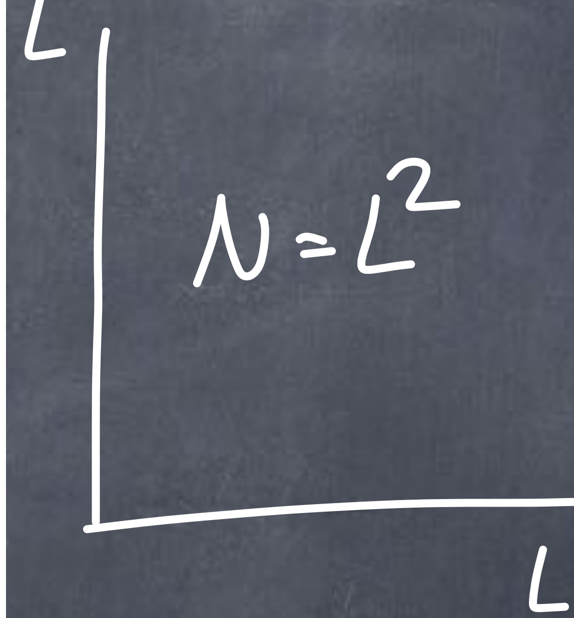


$$X^T \cdot X \vec{w}_i = \lambda_i \vec{w}_i$$

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \ldots \geq \lambda$$

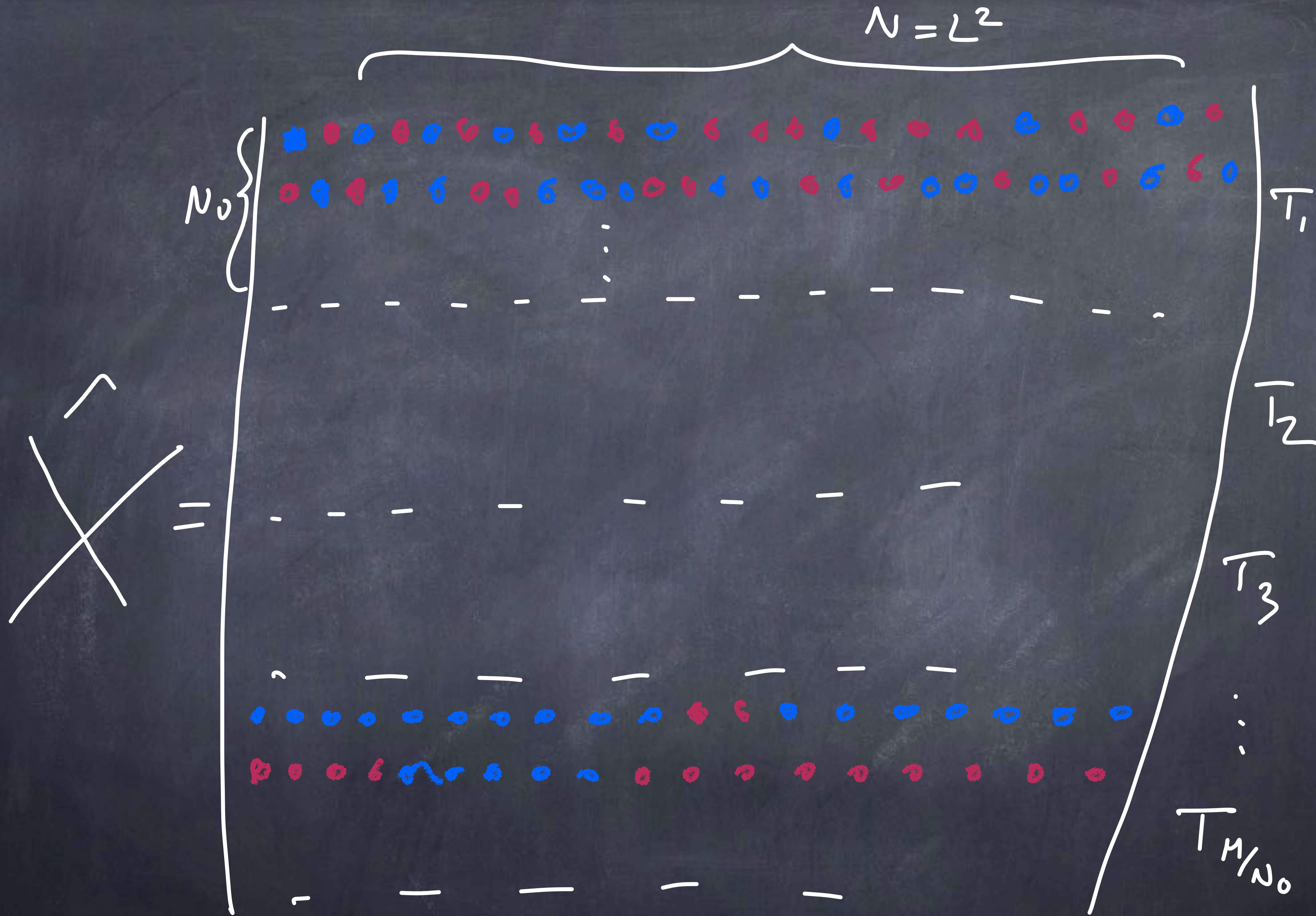
Back to Ising



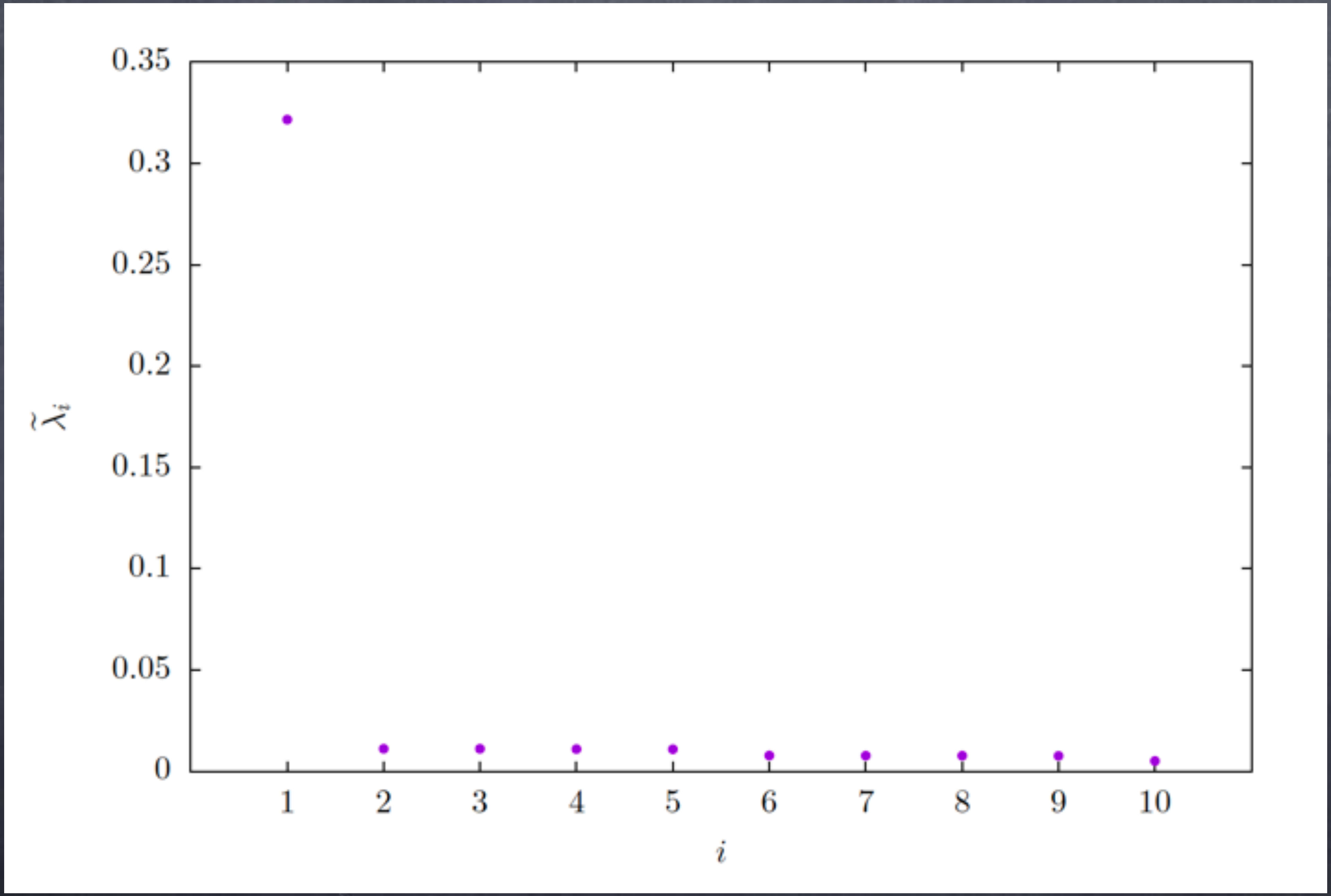


$N = L^2$

https://mattbierbaum.github.io/ising.js/


$N = L^2$
N_0
$\hat{X} =$
T_1
T_2
T_3
T_{M/N_0}

Eigenvalues λ

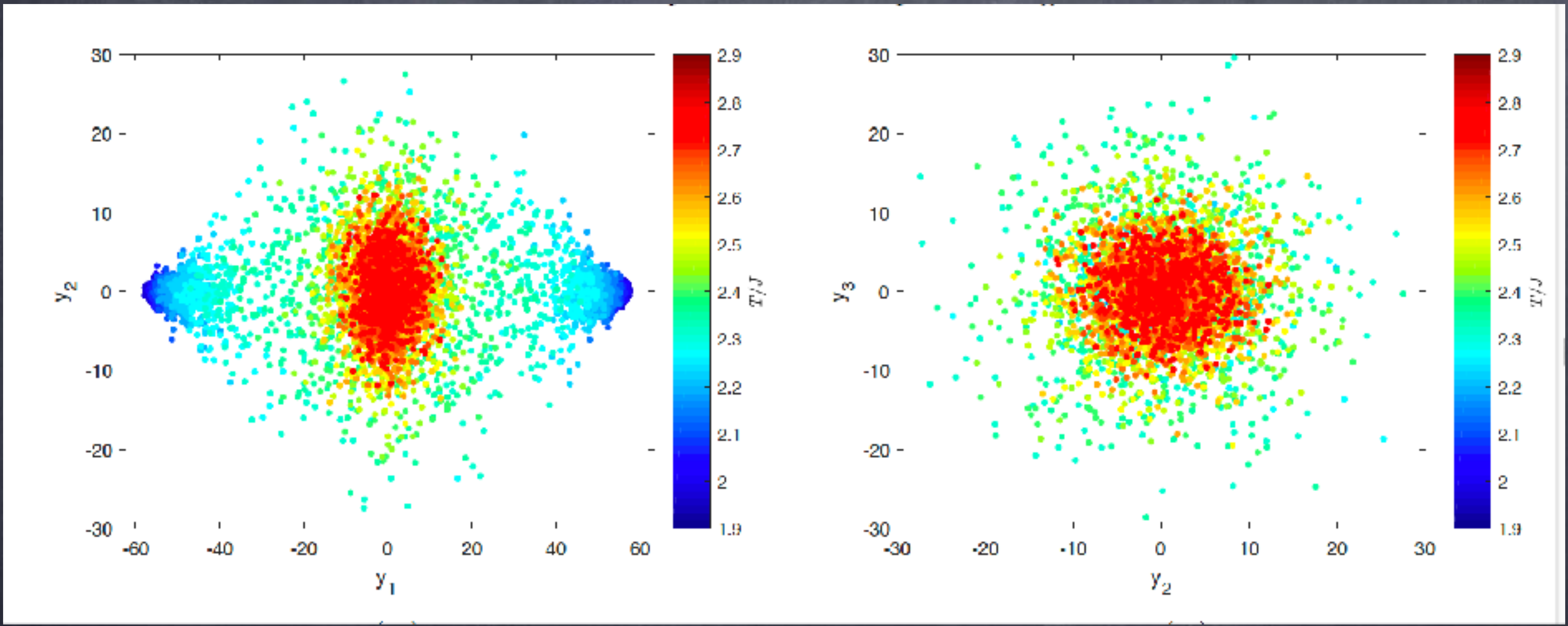


$$\vec{y}_1 = \hat{X}\vec{\omega}_1$$

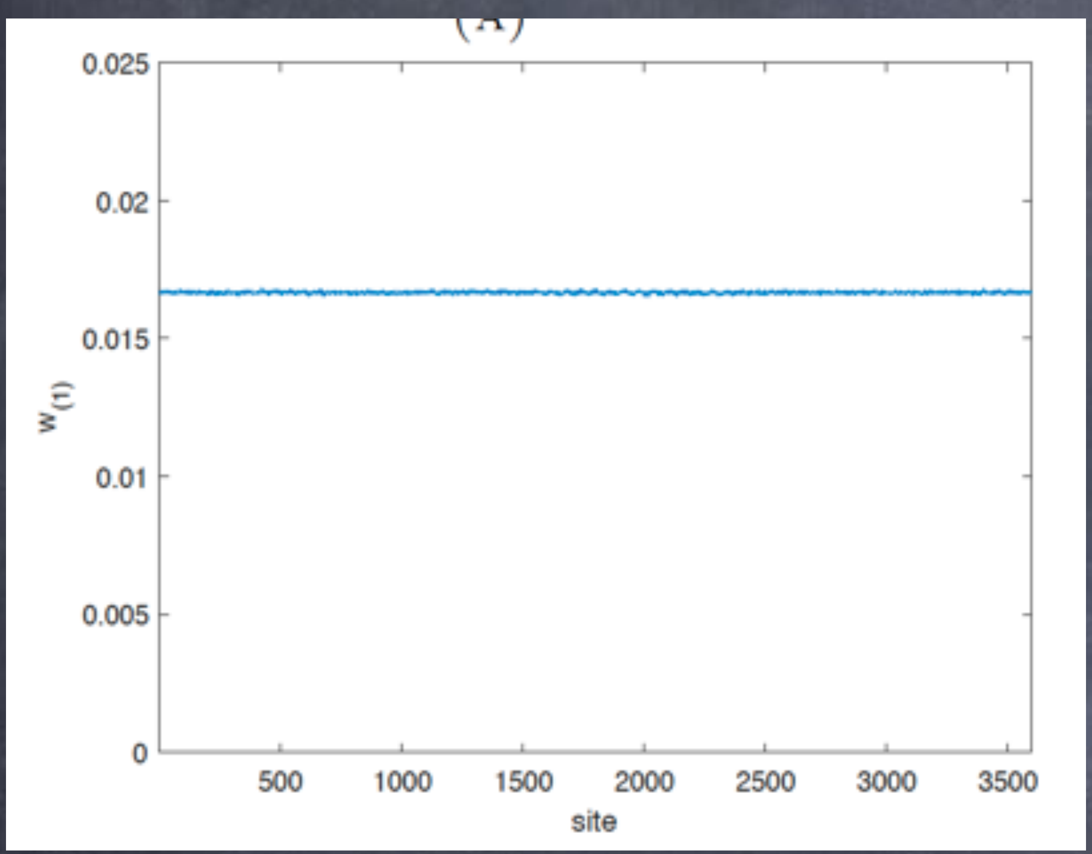
we plot y_1^i $\quad i = 1, \dots, M$

projection of config on principal axis

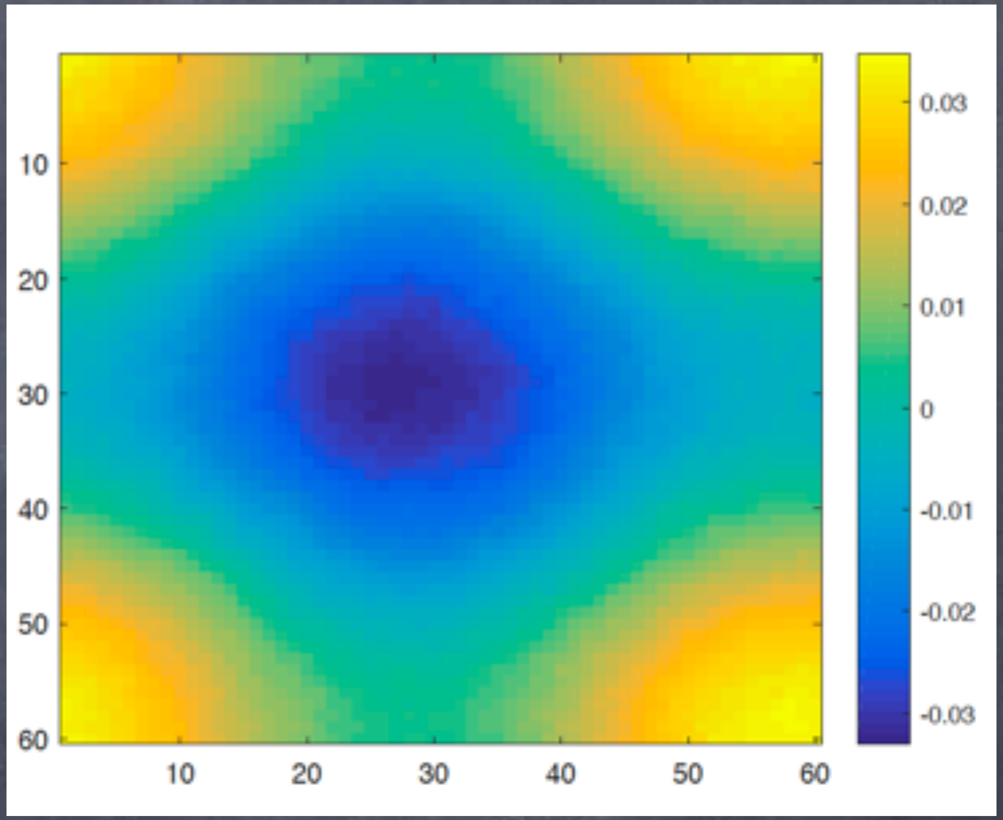
$$\vec{y}_2 = \hat{X}\vec{\omega}_2$$



30
20
10
0
-10
-20
-30
-60
-40
-20
0
20
40
60
y_1
y_2
2.9
2.8
2.7
2.6
2.5
2.4
2.3
2.2
2.1
2
1.9
T/J
30
20
10
0
-10
-20
-30
-30
-20
-10
0
10
20
30
y_2
y_3
2.9
2.8
2.7
2.6
2.5
2.4
2.3
2.2
2.1
2
1.9
T/J

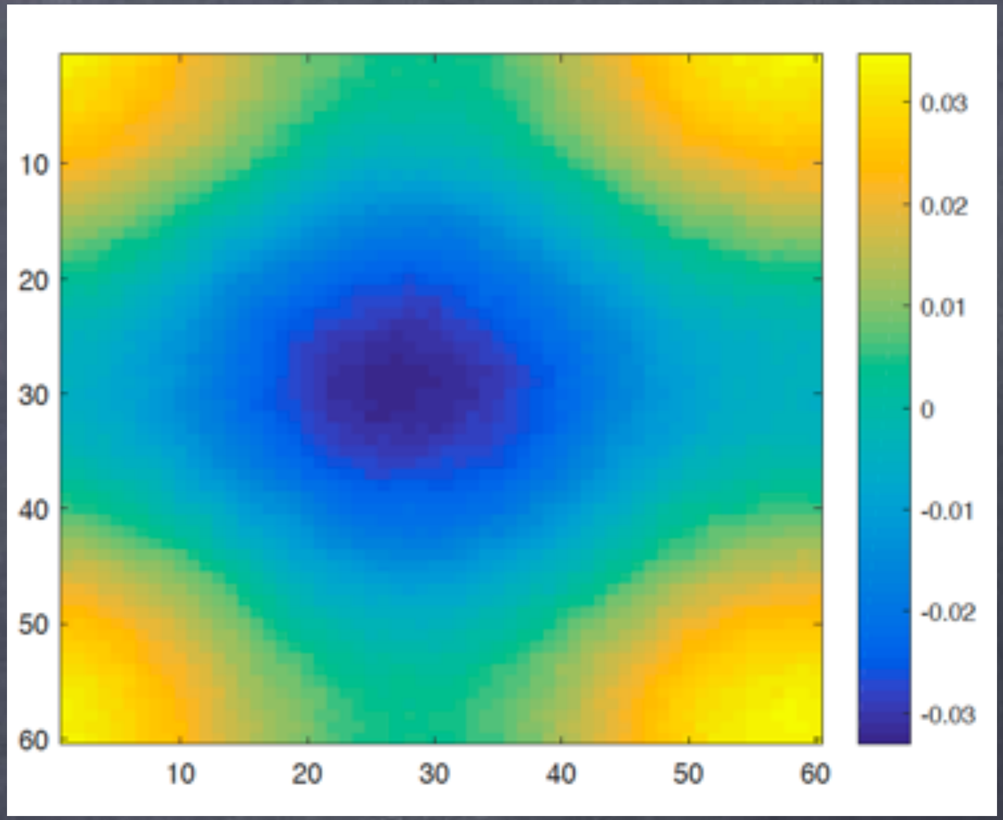


(A)
0.025
0.02
0.015
0.01
0.005
0
$w_{(1)}$
500
1000
1500
2000
2500
3000
3500
site



10
20
30
40
50
60
10
20
30
40
50
60
0.03
0.02
0.01
0
-0.01
-0.02
-0.03

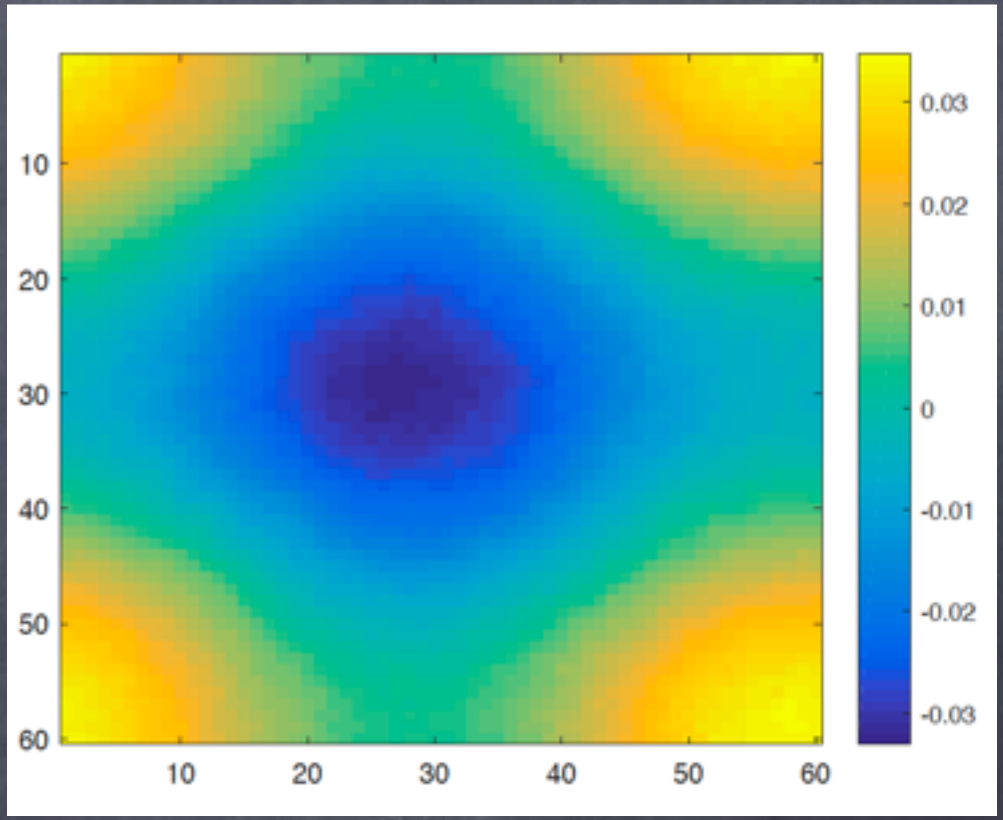
$\lvert c(\mathbf{k})\rvert$	λ_i
206	151.0
43	7.7
43	7.7
43	7.7
43	7.7
35	4.8
35	4.8



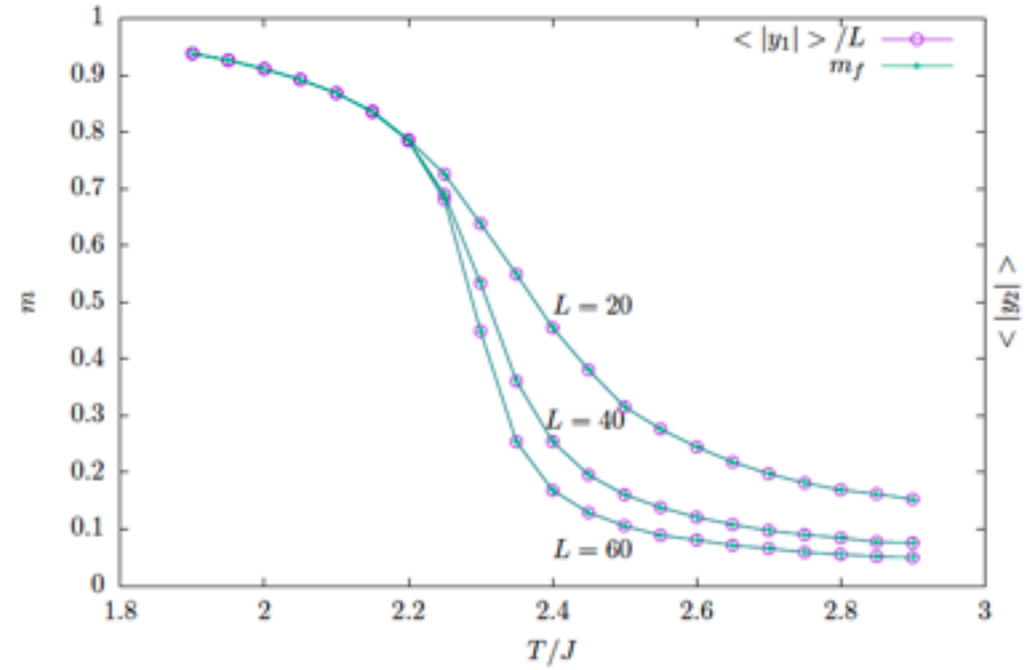
Fourier mode

$\|c(\mathbf{k})\|$	λ_i
206	151.0
43	7.7
43	7.7
43	7.7
43	7.7
35	4.8
35	4.8

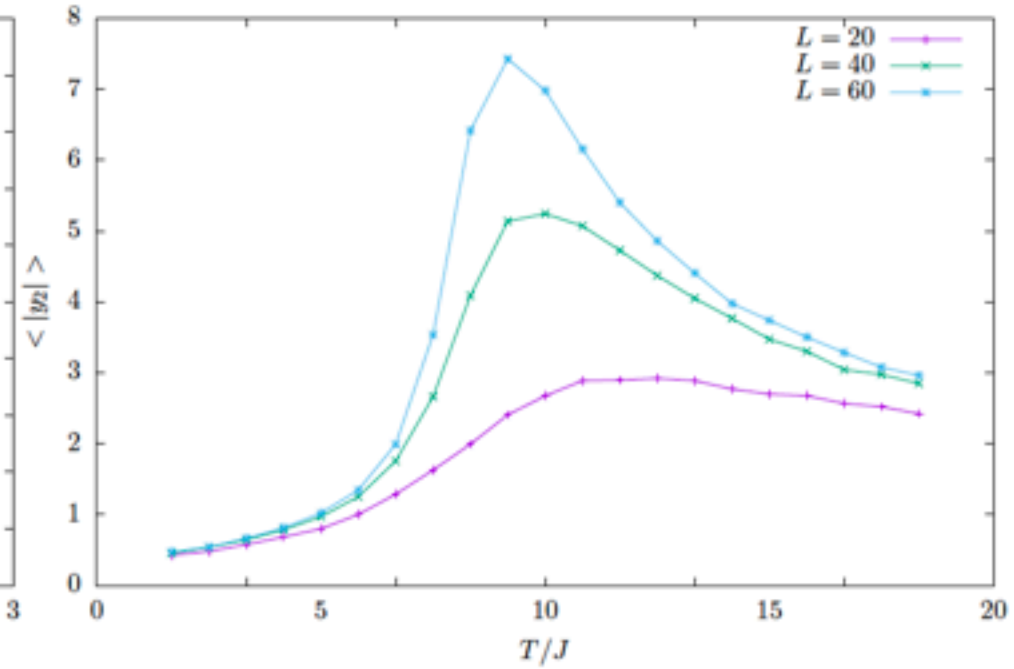
i/j	c_i^2/c_j^2	λ_i/λ_j	$\delta(\%)$
1/2	22.5	19.6	13
1/3	34.4	31.3	9
1/4	55.6	52.4	6
2/3	1.5	1.6	4
2/4	2.5	2.7	8



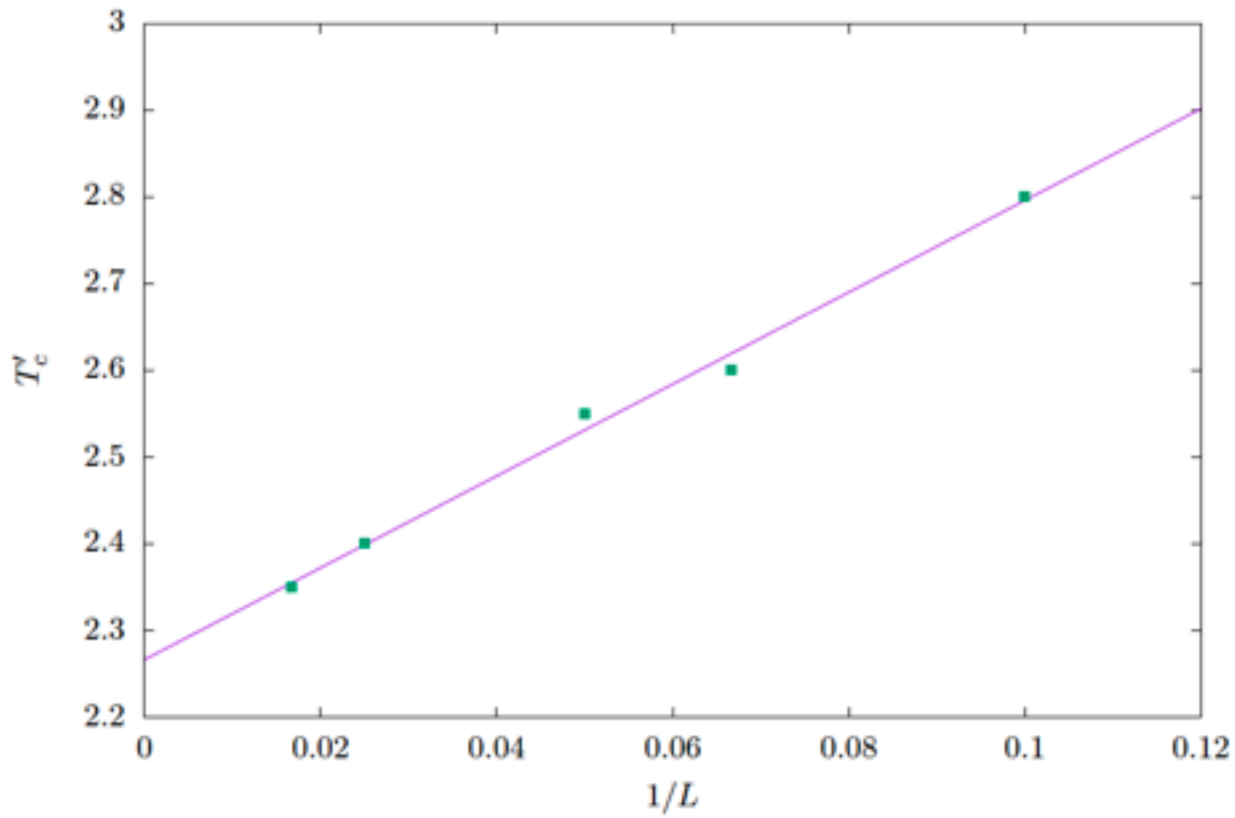
Fourier mode



(A)



(B)



Conclusion

we tested unsupervised learning to both 'detect' the phase transition and 'learn' the order parameter !