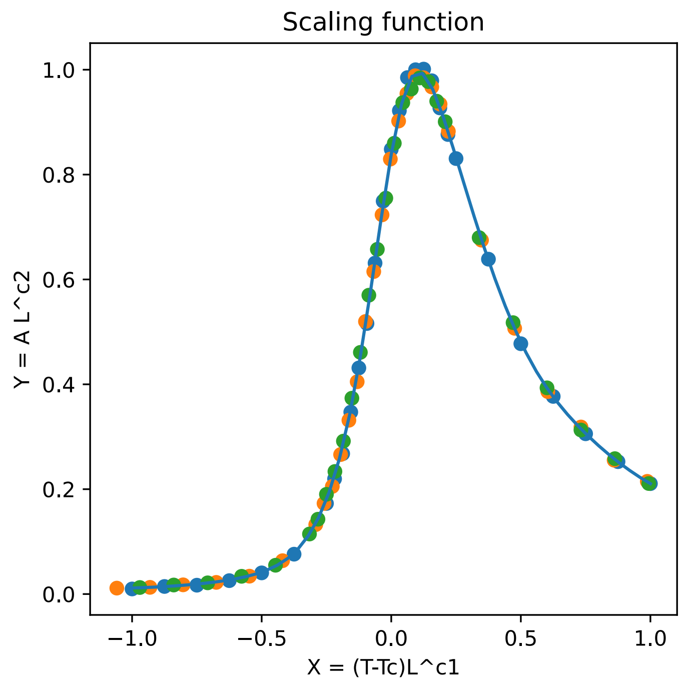
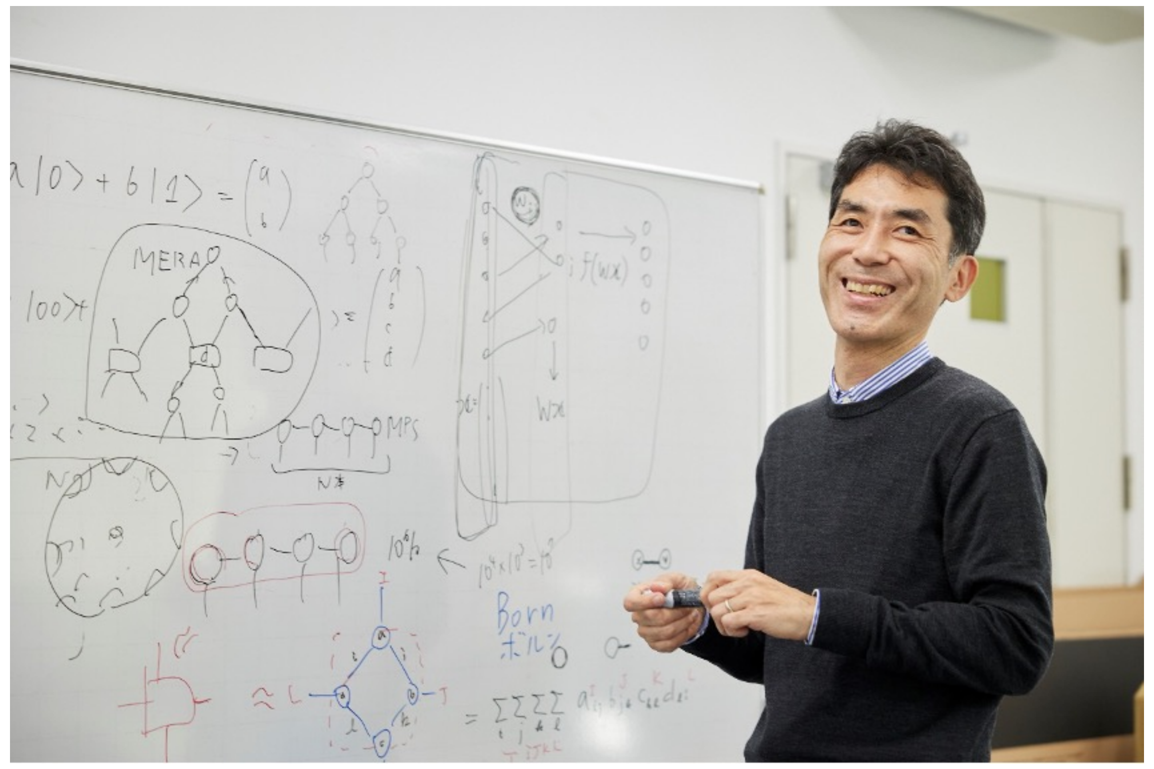
15aB12-9, JPS Annual meeting 2022

ニューラルネットワークを用いた スケーリング解析手法



2022年3月15日
米田亮介，**原田健自**
京都大学大学院情報学研究科

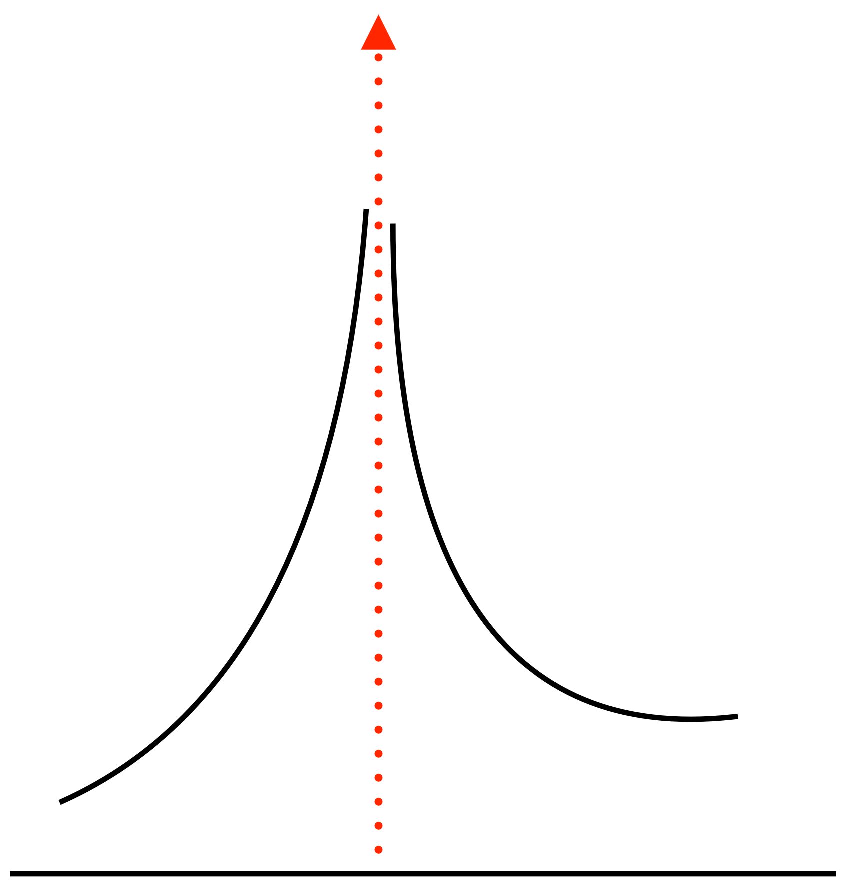


Critical phenomena

Critical behaviors of observables at a critical point. A scaling invariance is a key.

Equilibrium systems

- Classical critical phase transition at a finite temperature
  - Ising model in two and three dimensions (Ising universality class)
  - XY model in two dimensions (BKT universality class)
  - Heisenberg model in three dimensions (O(3) universality class)
- Quantum critical phase transition at a zero temperature
  - Ising model with a transverse field
  - SU(N) JQ model in two dimensions (deconfined quantum criticality?)



None-equilibrium systems

- Directed percolation in one dimension (directed percolation universality class)

  Related talk : **K.H.**, 21aL3-3, JPS 2021/09.
- Coupled oscillator models (Kuramoto model)

  Related talk : R.Yoneda, Y.Yamaguchi, **K.H.**, 9pL1-9, JPS 2020/09.
- Vicsek model in two dimensions

Various types of critical phenomena in the world! → **We need to identify the universality class!**

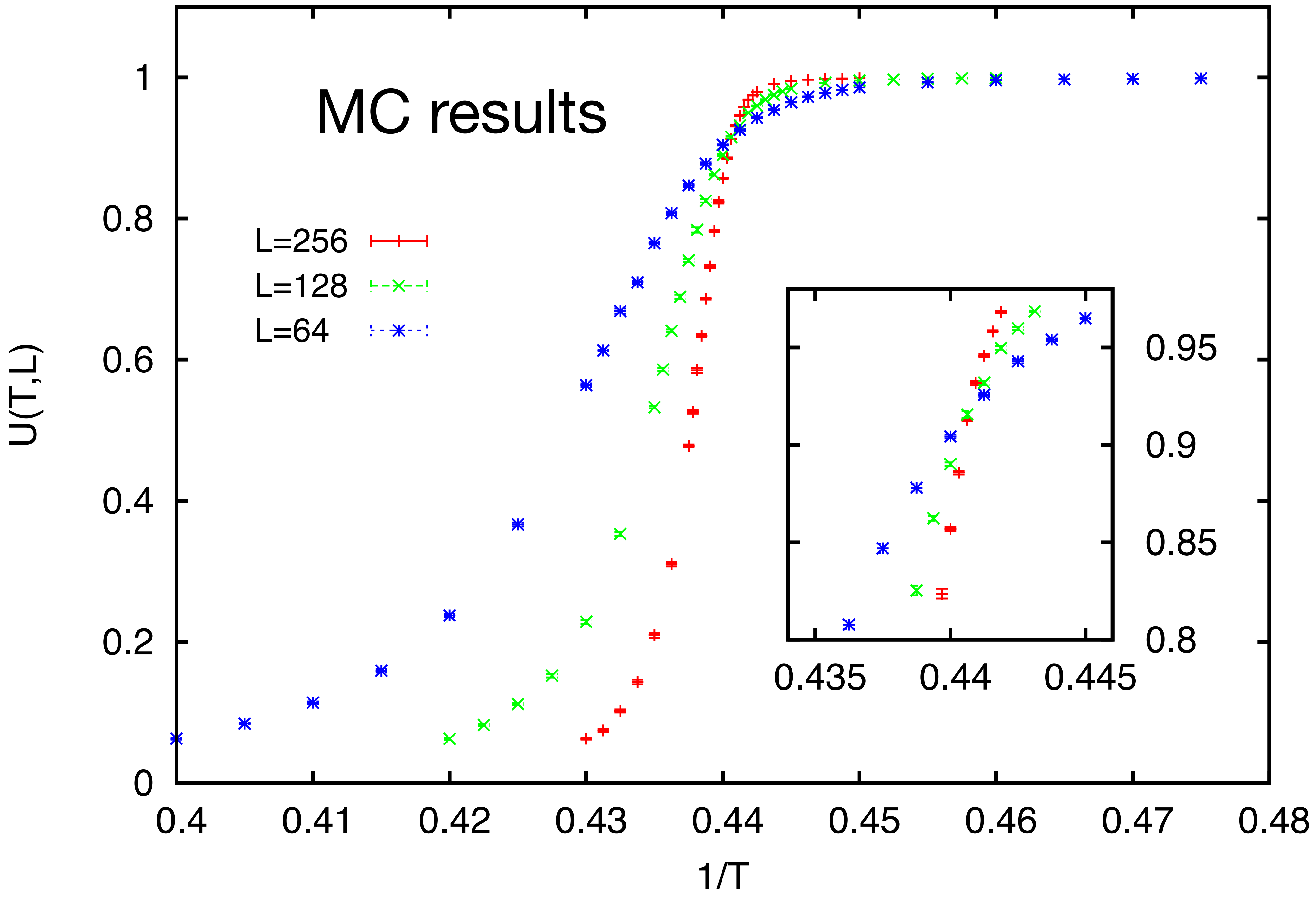
Scaling analysis of critical phenomena

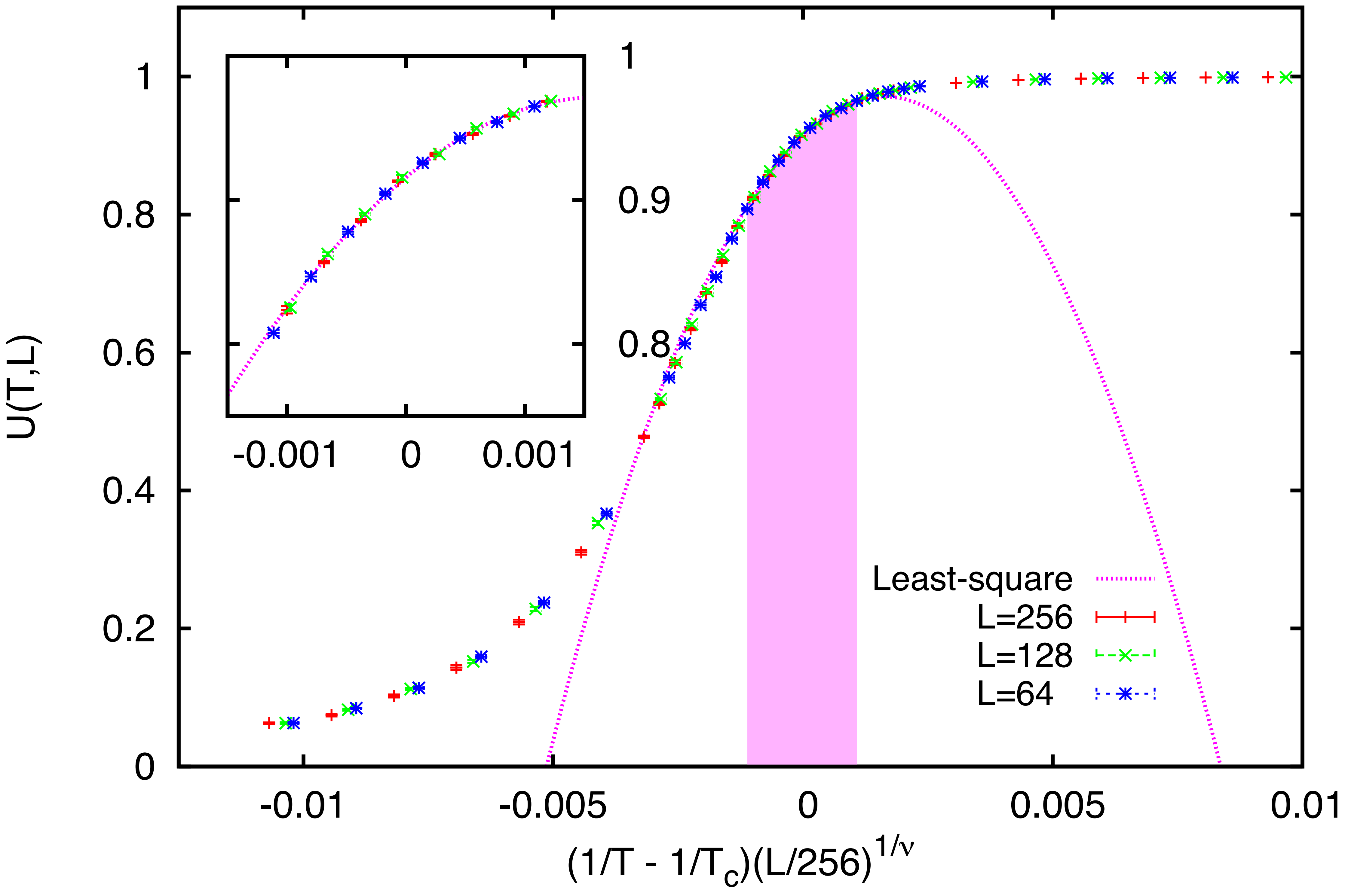
Finite-size scaling law

$$A(T, L) = L^{-c_2} f[(T - T_c)L^{c_1}]$$

L: system size
T: temperature
T_c: critical temperature
c_1, c_2: critical exponents

Ex. Binder ratio of 2D Ising





$$U = \frac{1}{2}\left(3 - \frac{\langle M^4 \rangle}{\langle M^2 \rangle^2}\right) = f[(T - T_c)L^{1/\nu}]$$

$$T_c = 0.440683(7), 1/\nu = 0.996(2)$$

$$\text{Exact: } T_c = 0.440687, 1/\nu = 1$$

Scaling analysis of critical phenomena

Finite-size scaling law

$$A(T, L) = L^{-c_2} f[(T - T_c)L^{c_1}]$$

L: system size

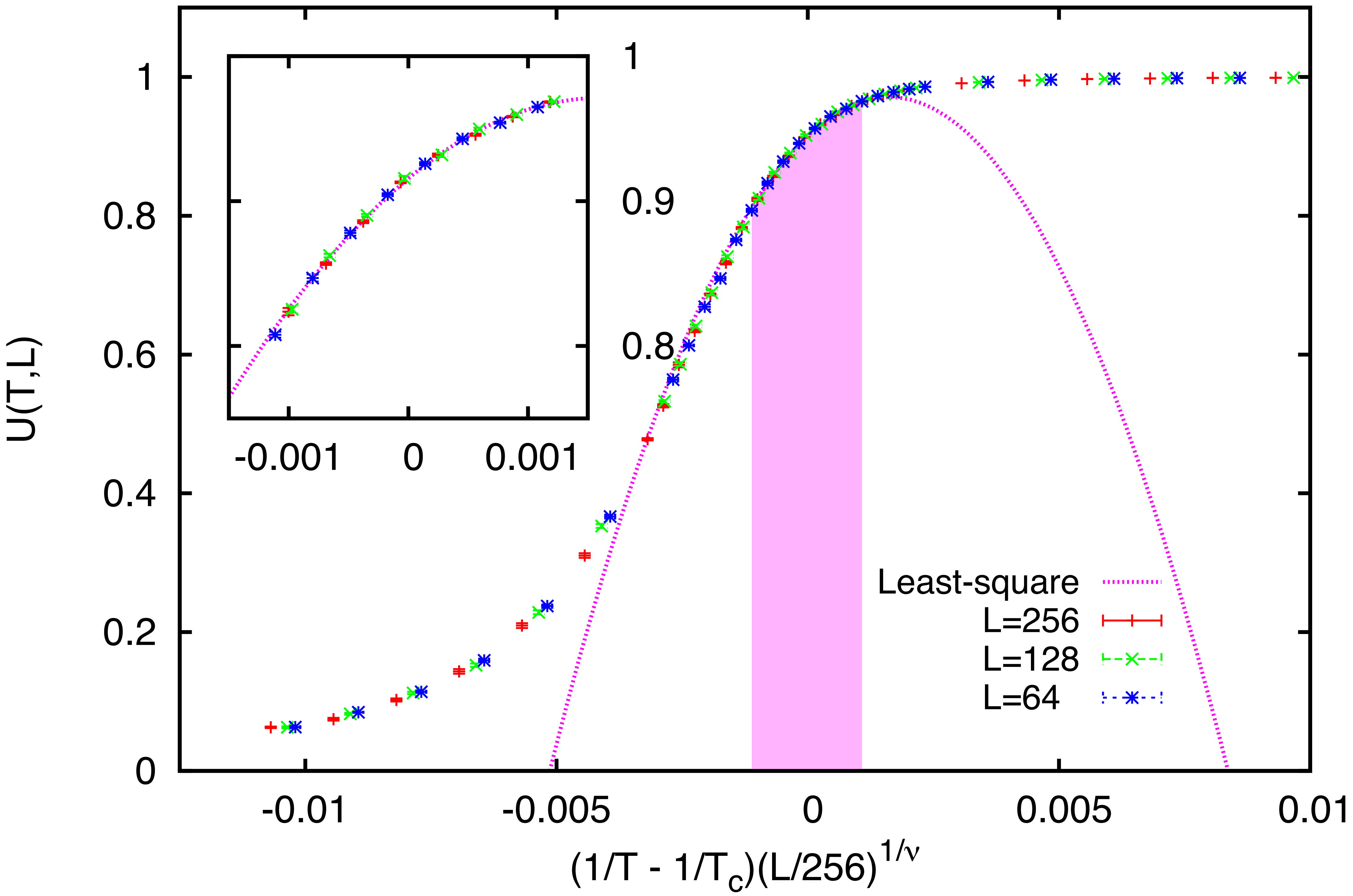
T: temperature

T_c: critical temperature

c_1, c_2: critical exponents

$$X = (T - T_c)L^{c_1},$$
$$Y = A/L^{-c_2},$$
$$E = \delta A/L^{-c_2}$$

All data points collapse on a scaling function



$$T_c = 0.440683(7), 1/\nu = 0.996(2)$$
$$\text{Exact: } T_c = 0.440687, 1/\nu = 1$$

Least-square method

Parametrized scaling function

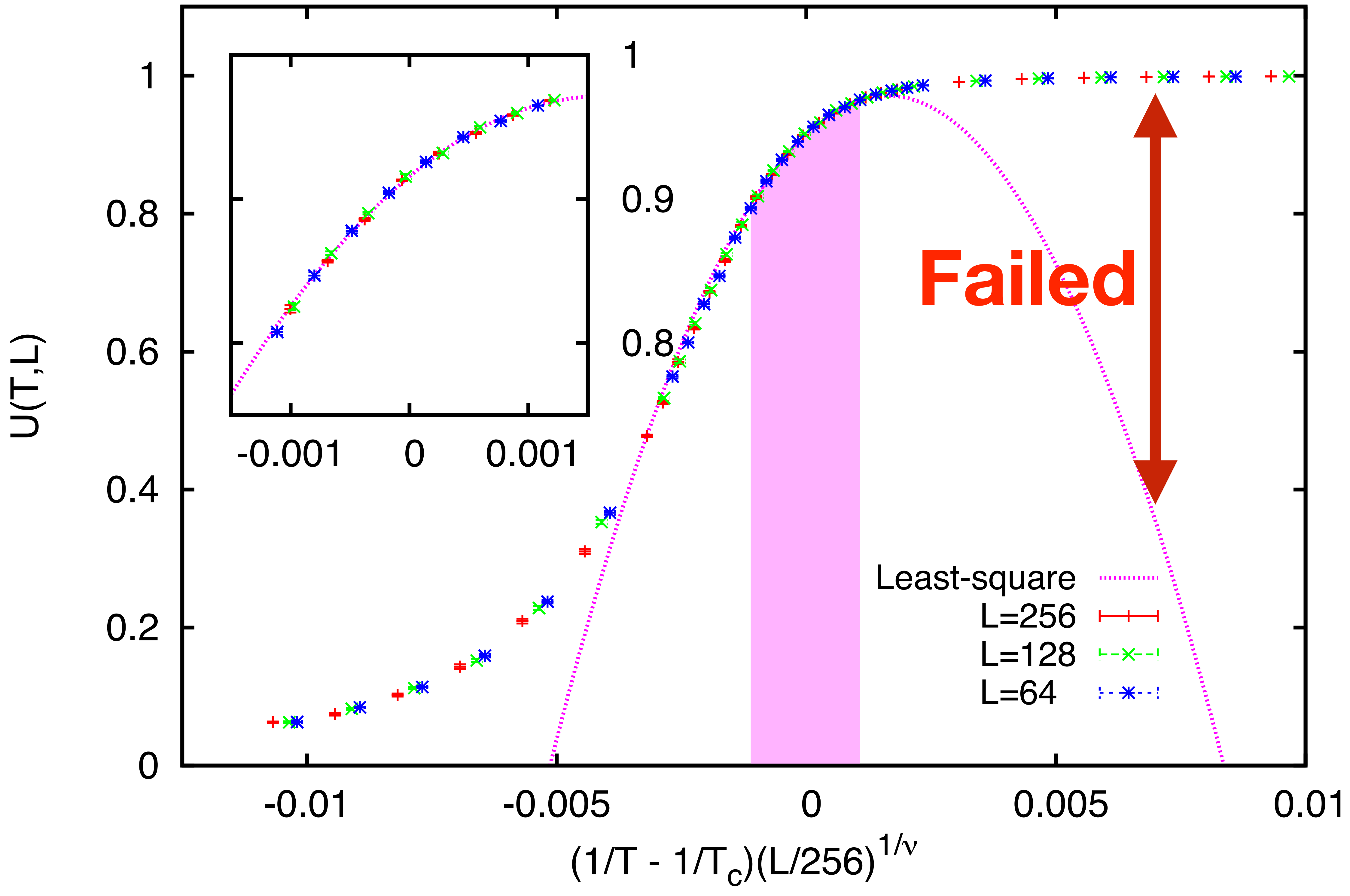
Ex. Polinomial

$$F(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots$$

Least-square

$$\arg\min_{\Theta} \left| \frac{\mathbf{Y} - F(\mathbf{X})}{\mathbf{E}} \right|^2$$

Failed



What is a more powerful function?

The least-square method is based on the maximum Gaussian log-likelihood in the Bayesian inference

Bayesian Scaling Analysis with Gaussian Process

KH, PRE 2011, 2015

Bayesian inference from a stochastic model

$$P(\{A\}, T_c, \nu, \cdots) \Rightarrow P(T_c, \nu, \cdots | \{A\}) \Rightarrow \arg\max_{T_c, \nu, \cdots} [\log(P(T_c, \nu, \cdots | \{A\}))]$$

Gaussian process model

$$A(T, L), \delta A(T, L) \;\Rightarrow\; \mathbf{T},\, \mathbf{L},\, \mathbf{A},\, \delta\mathbf{A} \;\Rightarrow\; \mathbf{X} = (\mathbf{T} - T_c)\mathbf{L}^{c_1}, \mathbf{Y} = \mathbf{A}/\mathbf{L}^{-c_2}, \mathbf{E} = \delta\mathbf{A}/\mathbf{L}^{-c_2}$$

parameters $\Theta = (T_c, c_1, c_2, \theta_0, \theta_1, \theta_2)$

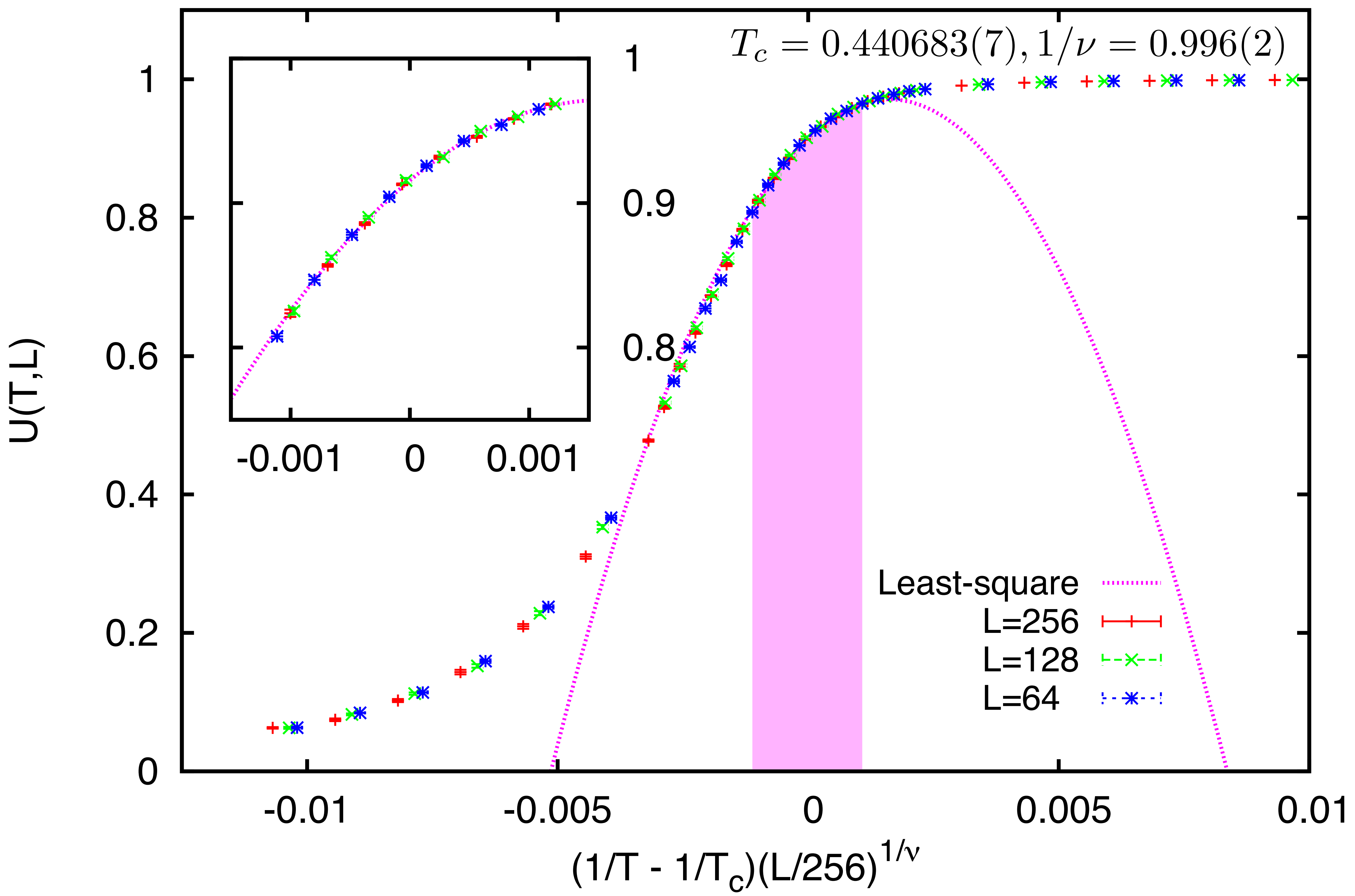
$$P(\{A\}, \Theta) = P(\mathbf{X}, \mathbf{Y}, \mathbf{E}, \Theta) = \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left[-\frac{1}{2}\mathbf{Y}^t \Sigma^{-1} \mathbf{Y}\right]$$

Gaussian process

Gram matrix

$$[\Sigma]_{ij} = \theta_0^2 \exp\left[-\frac{(X_i - X_j)^2}{2\theta_1^2}\right] + (\theta_2^2 + E_i^2)\delta_{ij}$$

Gaussian kernel



The performance is very nice!
However, the diagonalization of a gram matrix is heavy!

$$\text{Cost: } O(N^3)$$

Least-square method with Neural Network

Parametrized scaling function

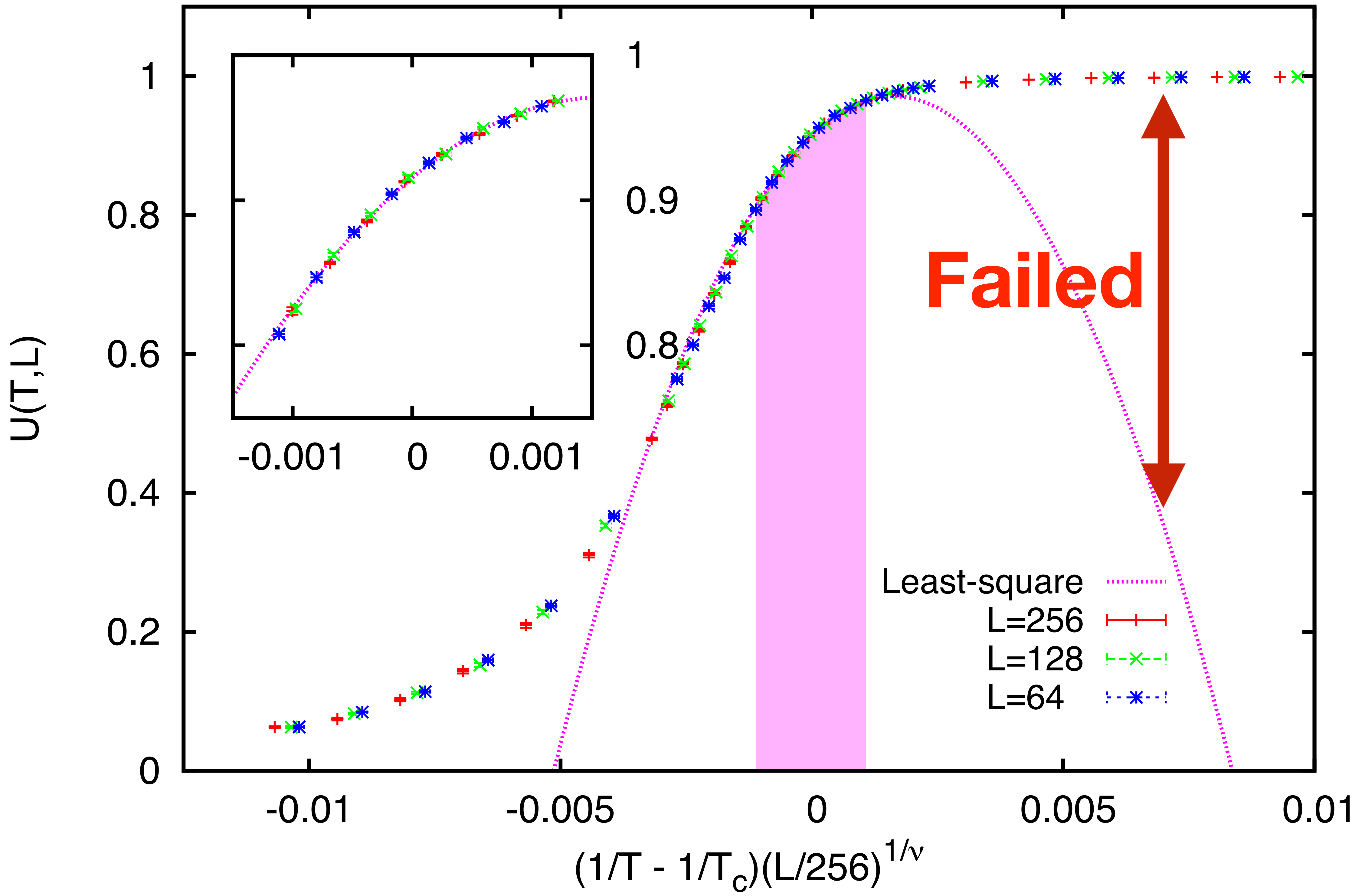
Ex. Polinomial

$$F(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots$$

Least-square

$$\arg\min_{\Theta} \left| \frac{\mathbf{Y} - F(\mathbf{X})}{\mathbf{E}} \right|^2$$

Failed



What is a more powerful function?

Neural network scaling function

Feed-forward

$$F_1(x) = a(W_1 x + b_1)$$
$$F_2(x) = a(W_2 F_1(x) + b_2)$$
$$\cdots$$
$$F(x) = F_N(x)$$

Ex. FC2 $1 \to 25 \to 25 \to 1$

$$W_1 : 25 \times 1, W_2 : 25 \times 25, W_3 : 1 \times 25$$

Num. of parameters ~1000

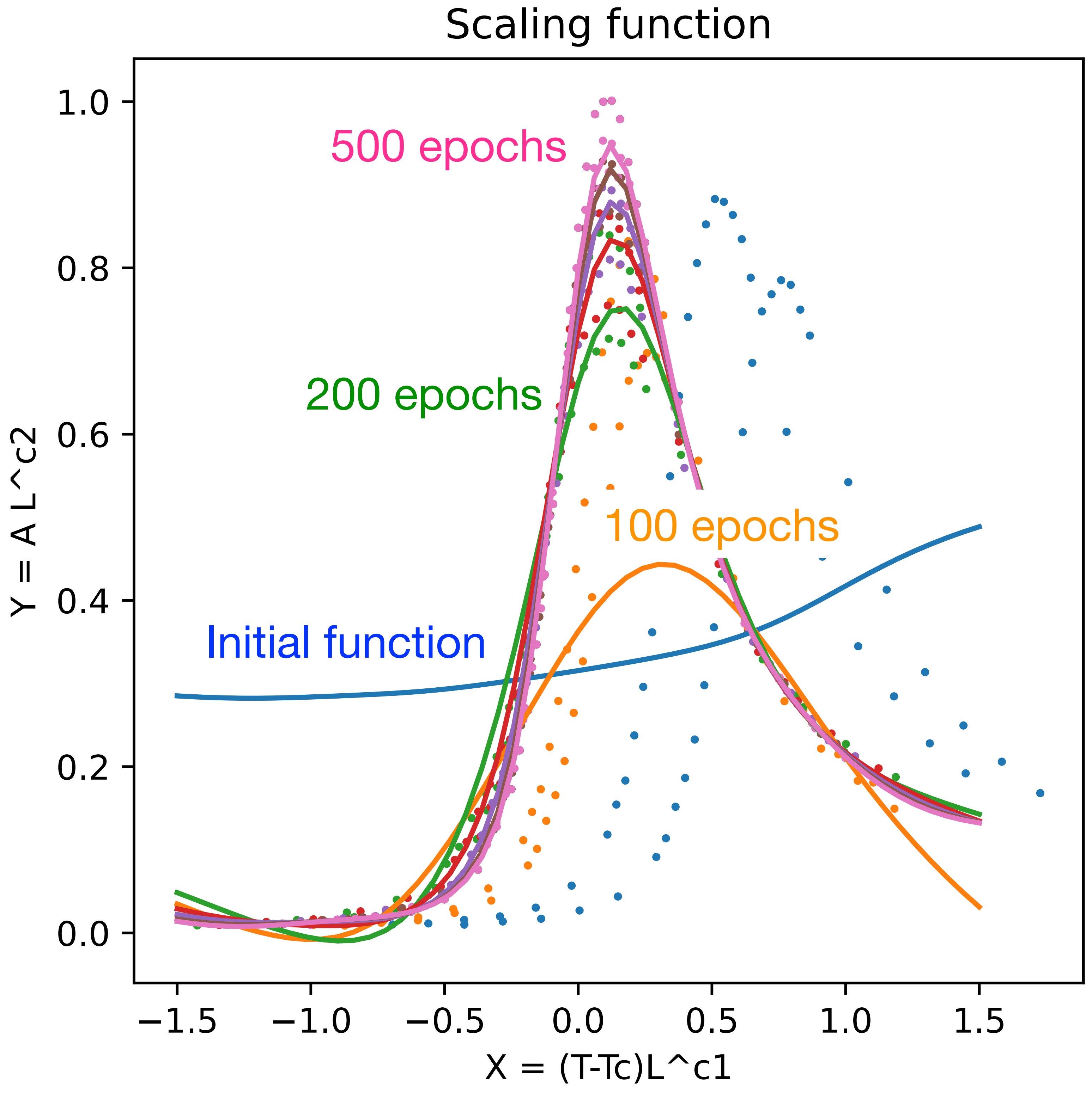
Optimization of NN and scaling parameters

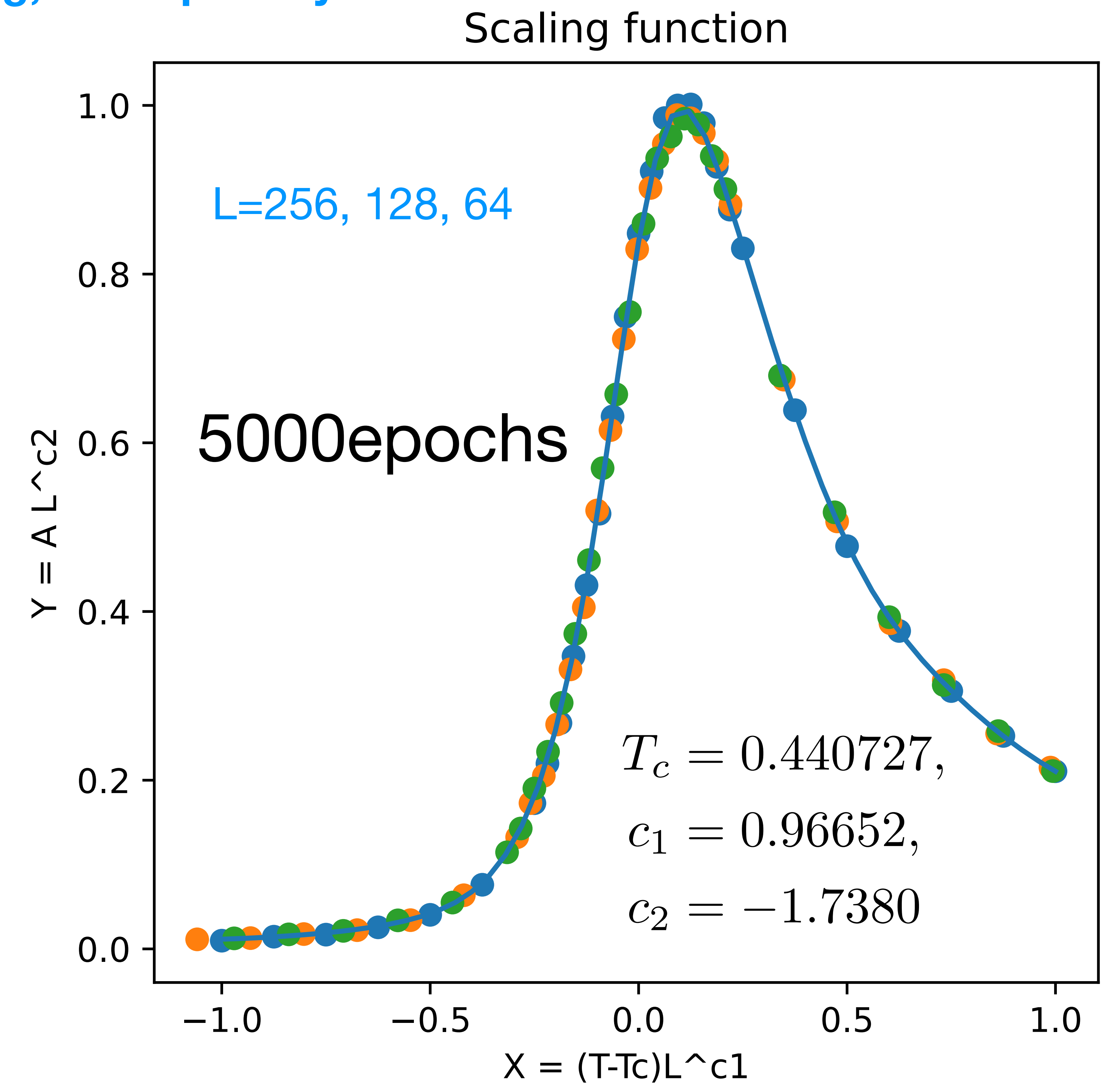
$$\text{loss} = \frac{|\mathbf{Y} - f[\mathbf{X}]|^2}{\mathbf{E}^2} + \log(\mathbf{E}^2)$$

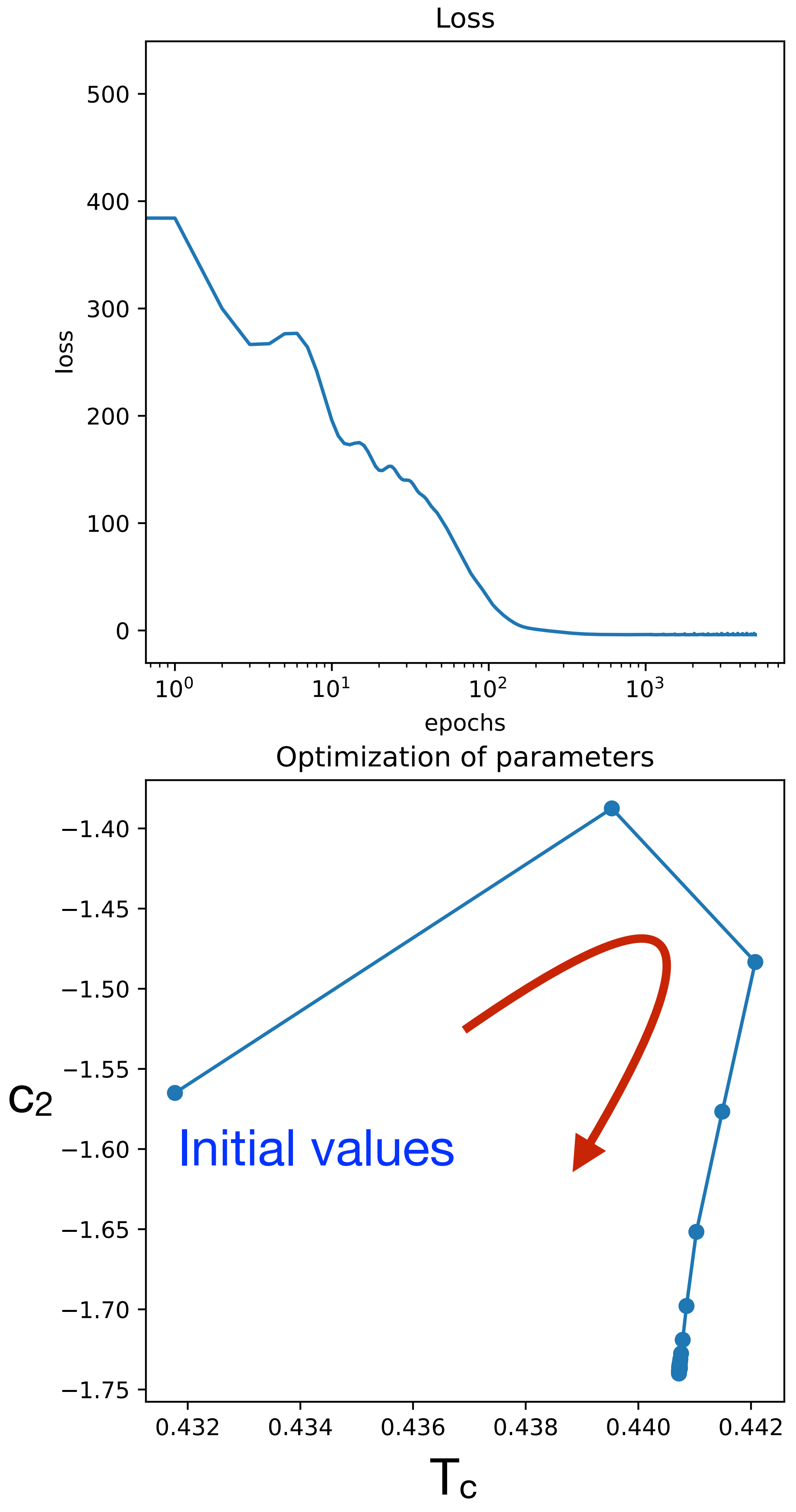
Optimizer

Adam (lr = 0.001) for NN(50, 50), but **Adam** (lr = 0.01) for scaling parameters

Ex. 2D Ising, susceptibility







We need a tuning in an optimization

Results of FSS analysis by NN

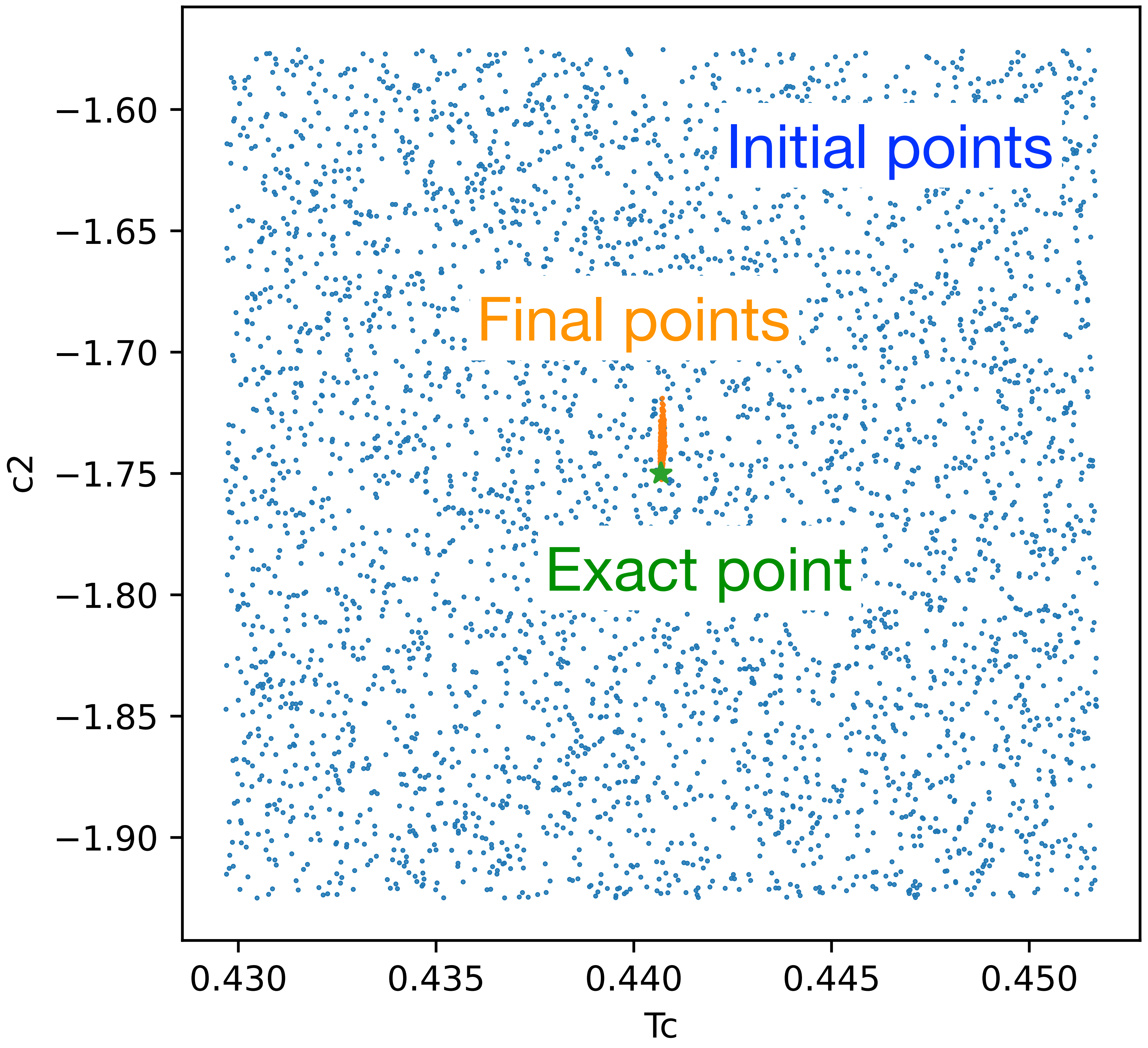
Bootstrap-like estimation of scaling parameters

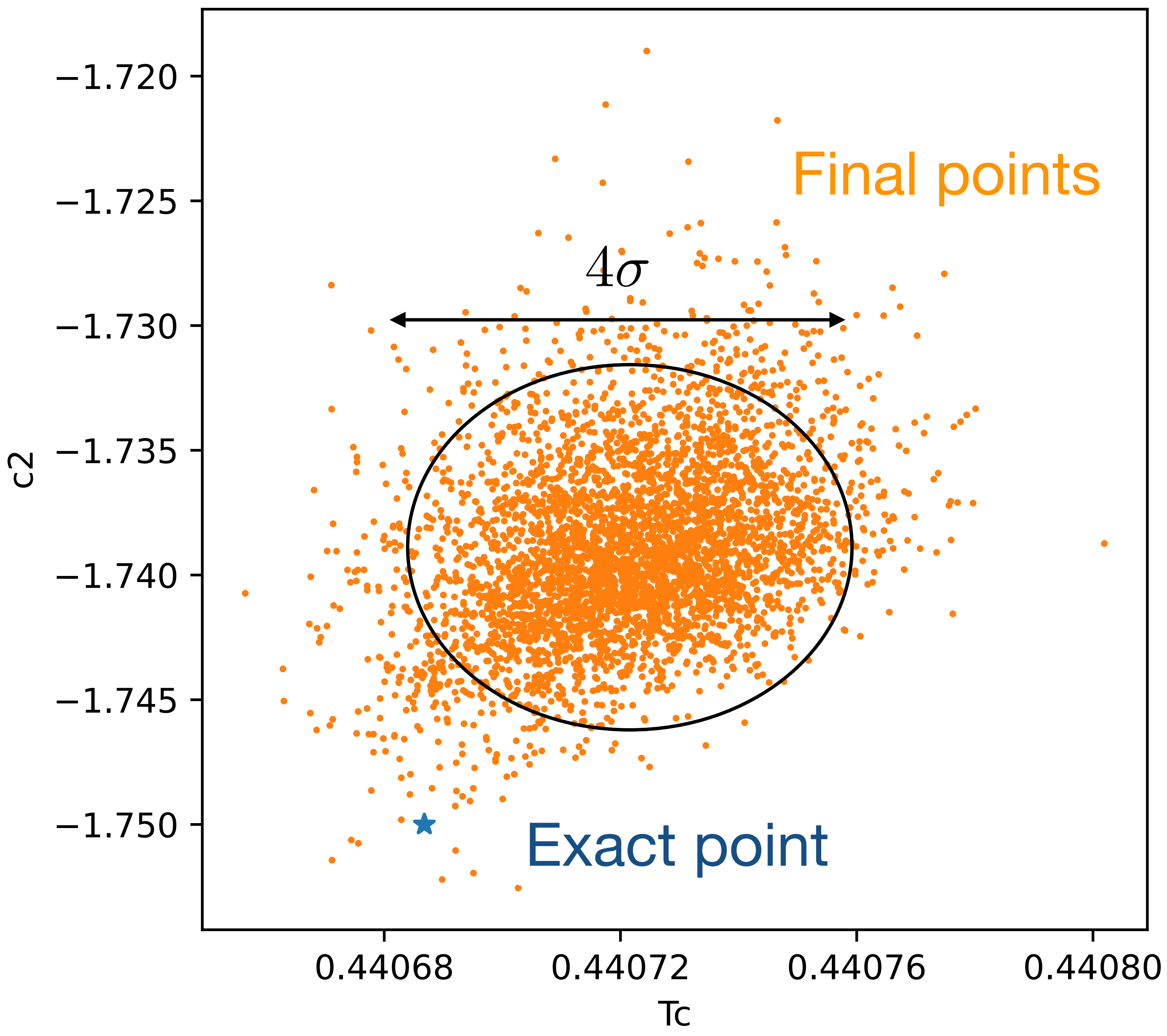
(1) make a dataset

80% resampling of data

(2) set random initial values

(3) optimize a NN and scaling parameters for a given dataset

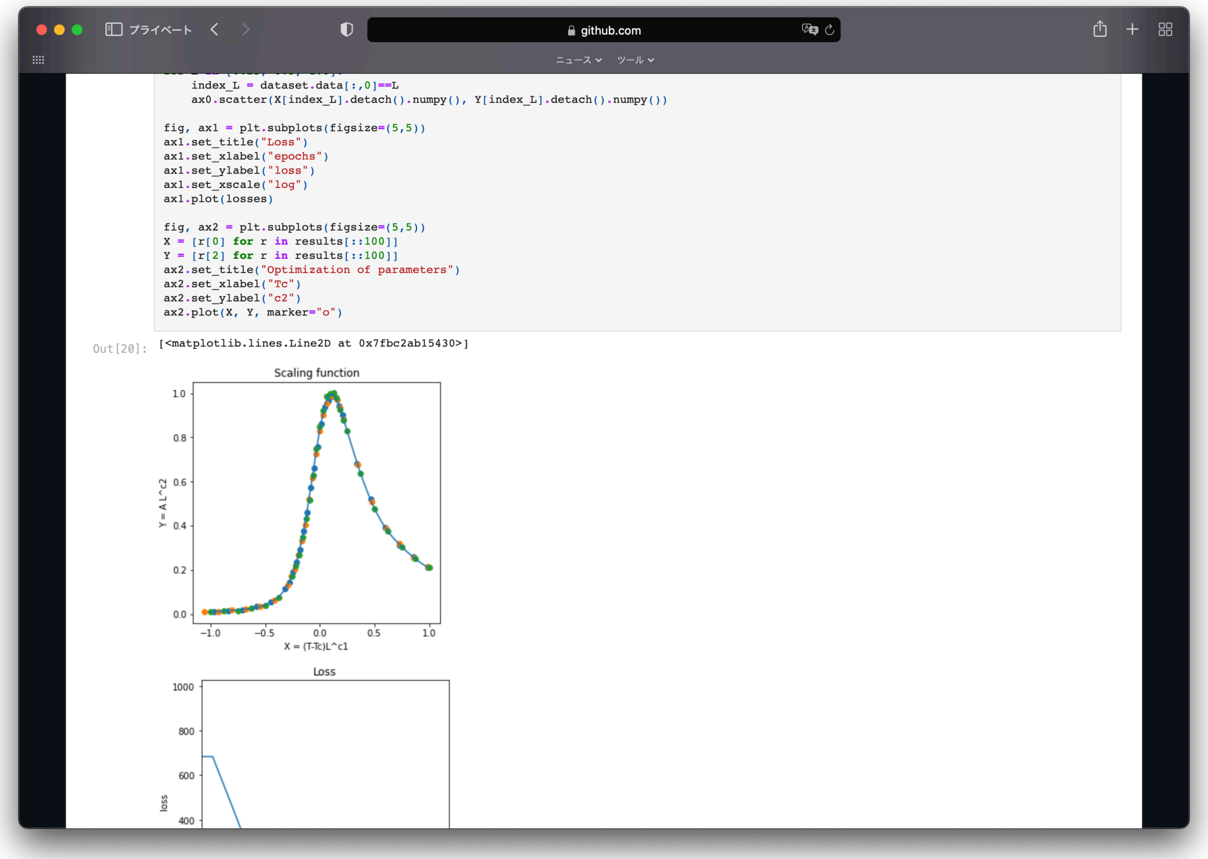




Summary and future issue

Summary

- Least-square method $$\text{loss} = \frac{|\mathbf{Y} - f[\mathbf{X}]|^2}{\mathbf{E}^2} + \log(\mathbf{E}^2)$$
  - New modeling of scaling function by neural network (paper in preparation)
- Demonstration
  - Susceptibility of 2D Ising Model → nice convergence!
- Implementation
  - Python
    - FSS-tools in GitHub https://github.com/KenjiHarada/FSS-tools
    - jaxfss in GitHub https://github.com/yonesuke/jaxfss



Future issue

- Optimization method
  - Tuning is crucial. What is more better optimizer?