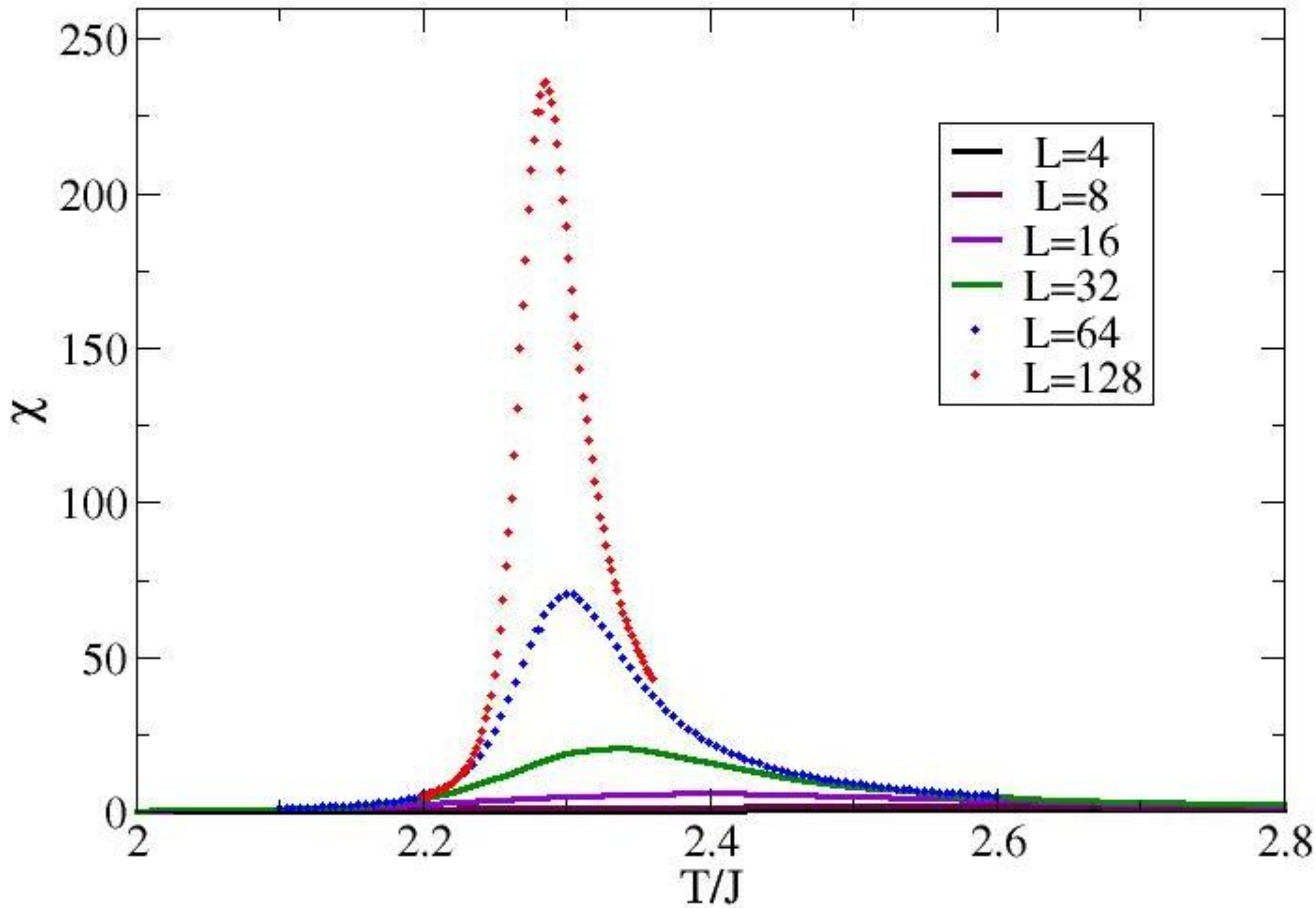
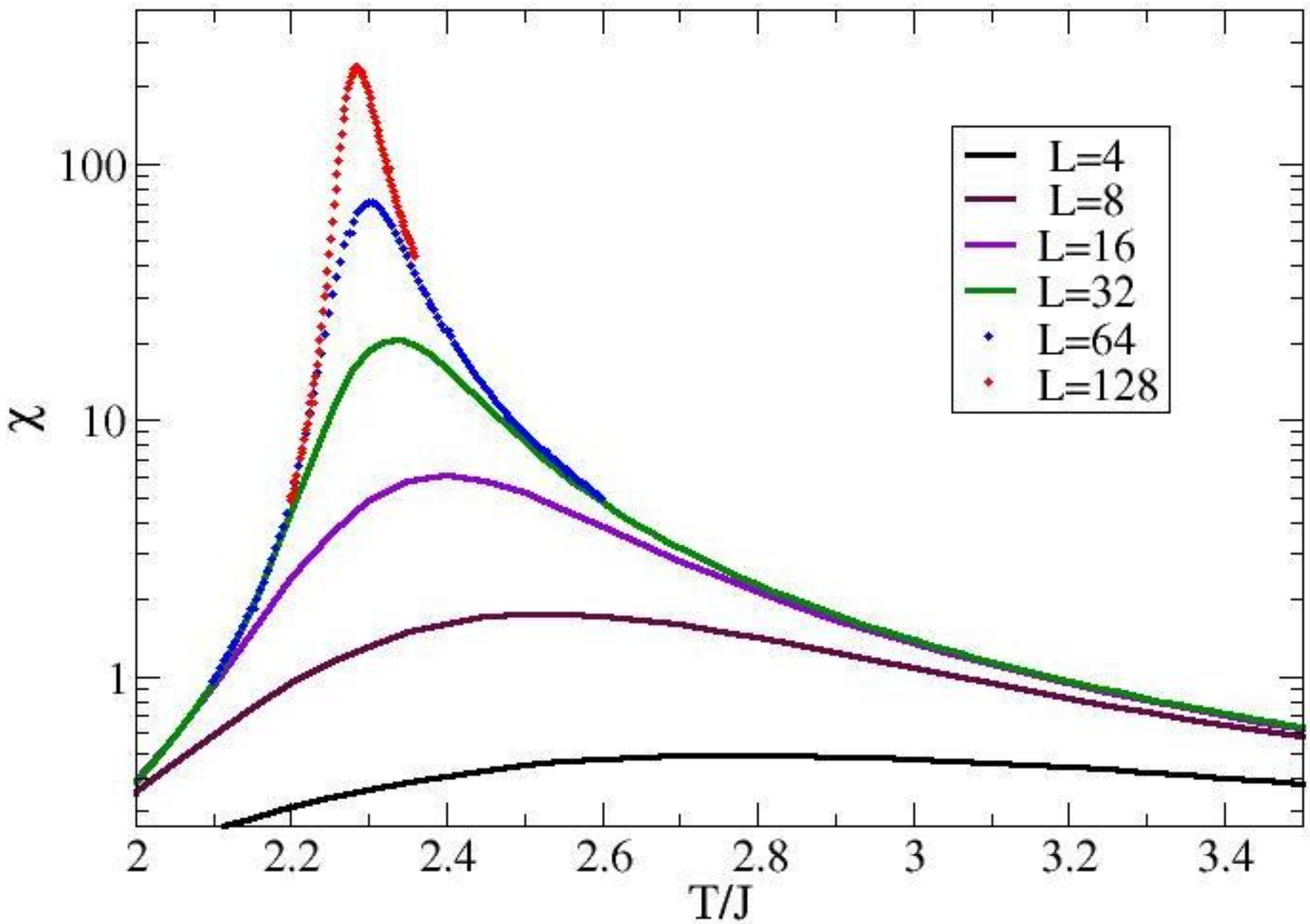
Susceptibility: $\chi = \frac{1}{N}\frac{1}{T}\left(\langle M^2\rangle - \langle |M|\rangle^2\right)$

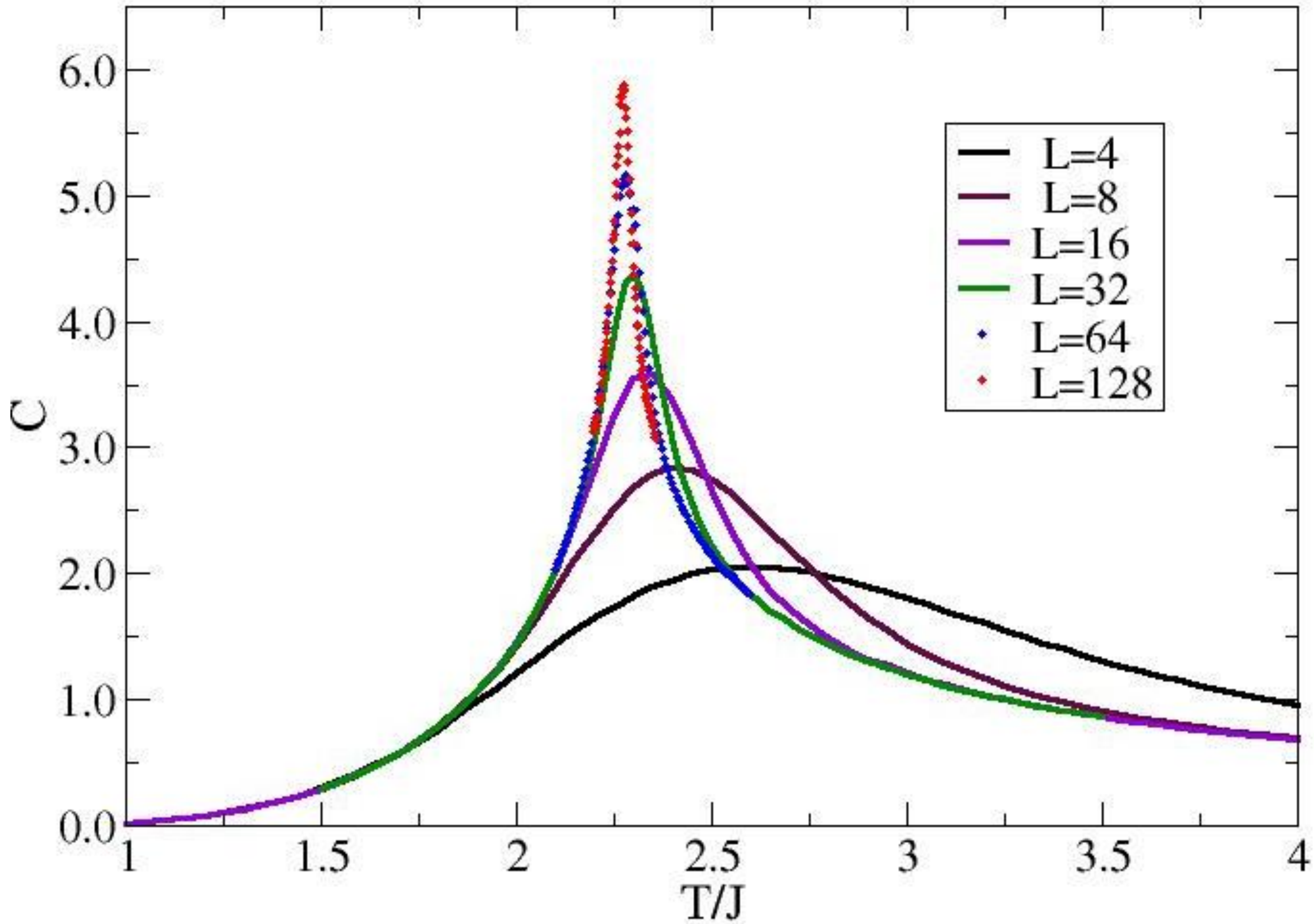


Diverges at the transition: $\chi \sim |T - T_c|^{-\gamma}$

On a logarithmic scale



Specific heat



$$C \sim |T - T_c|^{-\alpha}$$

(actually α=0 and log divergence for 2D Ising)

General finite-size scaling hypothesis

The ratio $\xi/L = t^{-\nu}L^{-1}$ should control the behavior of finite-size data also close to Tc

Test this finite-size scaling form

$$\chi(t) = L^{\sigma} f(\xi/L) = L^{\sigma} f(t^{-\nu}L^{-1}) = L^{\sigma} g(tL^{1/\nu})$$

What is the exponent σ?

We know that for fixed (small) t, the infinite L form should be

$$\chi(t) \sim t^{-\gamma}, \quad (L \to \infty)$$

To reproduce this, the scaling function $g(x)$ must have the limit

$$g(x) \to x^{b}, \qquad (x \to \infty)$$

We can determine the exponents as follows

$$\chi(t) \sim L^{\sigma} g(tL^{1/\nu}) = L^{\sigma}(tL^{1/\nu})^{b} = t^{b} L^{\sigma + b/\nu}$$

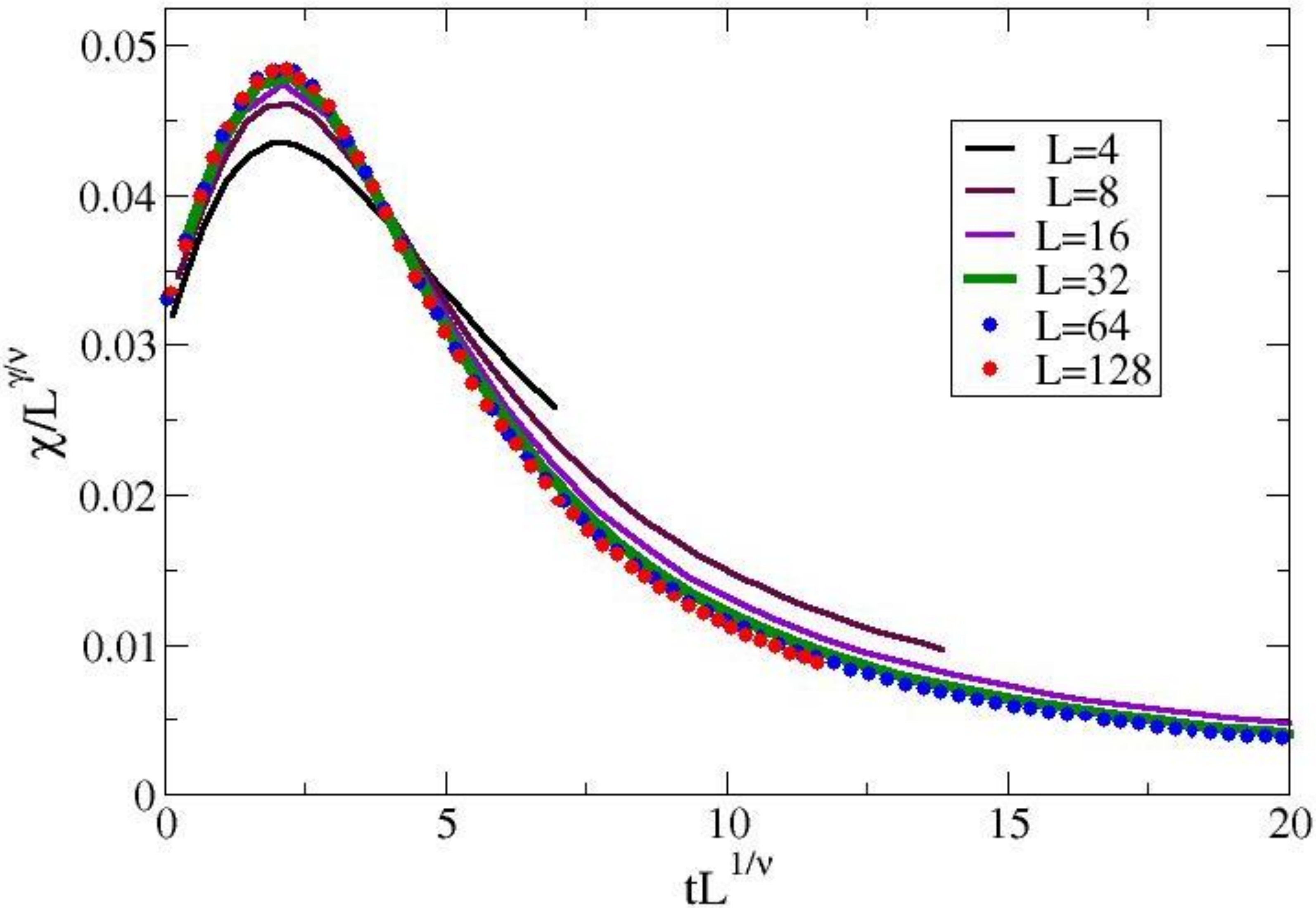
Hence $b = -\gamma,\ \sigma = \gamma/\nu$

$$\chi(t) = L^{\gamma/\nu} g(tL^{1/\nu})$$

Find g by graphing $\chi(t)/L^{\gamma/\nu}$ versus $tL^{1/\nu}$

2D Ising model; $\gamma = 7/4, \quad \nu = 1$

$$T_c = 2/\ln(1+\sqrt{2}) \approx 2.2692$$



In general; find Tc and exponents so that large-L curves scale

Binder ratio $Q = \frac{\langle m^2 \rangle}{\langle |m| \rangle^2}$ $\left(Q_{2n} = \frac{\langle m^{2n} \rangle}{\langle m^n \rangle^2}, \quad n = 1, 2, \ldots \right)$

Useful dimensionless quantity for accurately locating Tc

Infinite-size behavior:

$$\langle |m| \rangle \sim t^{\beta}$$

$$\langle m^2 \rangle \sim t^{2\beta}$$

Implies finite-size scaling forms

$$\langle m^2 \rangle \sim L^{-2\beta/\nu}$$

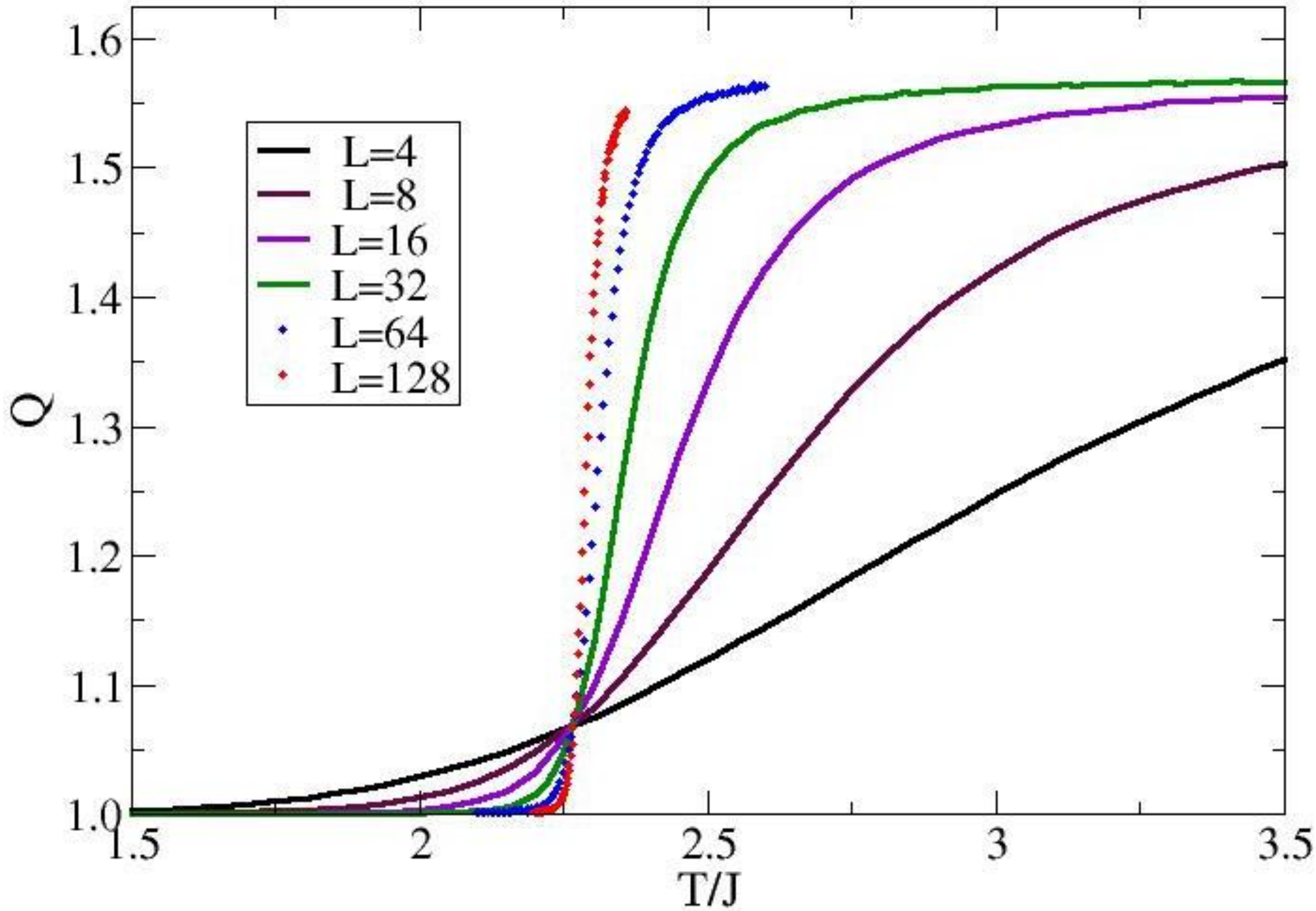
$$\langle |m| \rangle \sim L^{-\beta/\nu}$$

Hence Q should be size-independent at the critical point

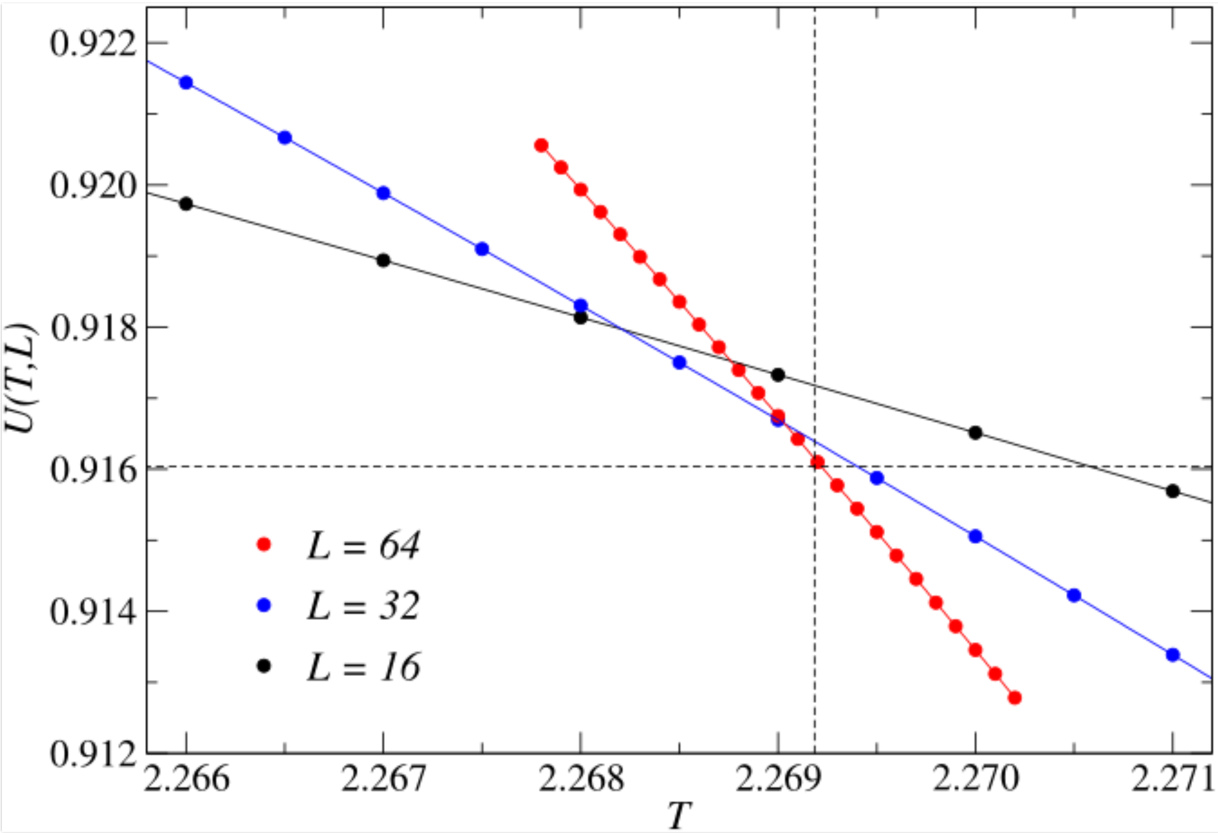
$$Q \to 1 \ \text{for} \ T \to 0, \qquad Q \to \ \text{constant for} \ T \to \infty$$

Q(L) curves for different L cross at Tc; often small corrections

Binder ratio: $Q = \frac{\langle m^2 \rangle}{\langle |m| \rangle^2}$



Q is size independent at Tc (useful for locating Tc)

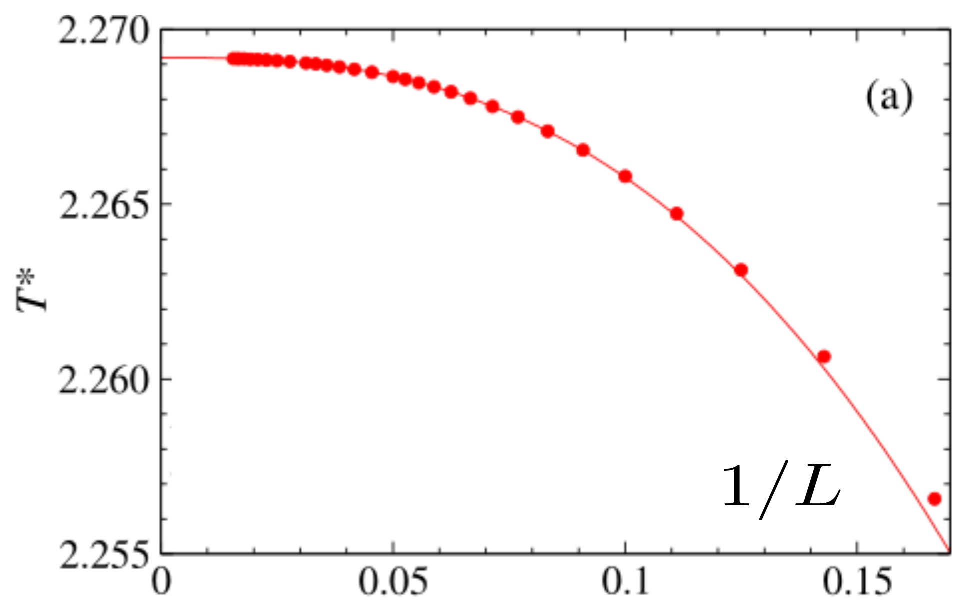


Scaling theory with corrections predicts:

$$T^*(L,2L) = T_c + aL^{-(1/\nu+\omega)}$$

ω is an exponent governing scaling corrections, ω=2 for 2D Ising

Crossing points for, e.g., sizes L, 2L can be extrapolated to infinite L to give an accurate value for Tc



Autocorrelation functions

Value of some quantity at Monte Carlo step i: Q_i

The autocorrelation function measures how a quantity becomes statistically independent from its value at previous steps

$$A_Q(\tau) = \frac{\langle Q_{i+\tau} Q_i \rangle - \langle Q_i \rangle^2}{\langle Q_i^2 \rangle - \langle Q_i \rangle^2}$$ (averaged over time i)

Asymptotical decay

$$A_Q(\tau) \sim \mathrm{e}^{-\tau/\Theta}, \quad \Theta = \text{ autocorrelation time}$$

Integrated autocorrelation time

$$\Theta_{\mathrm{int}} = \frac{1}{2} + \sum_{\tau=1}^{\infty} A_Q(\tau)$$

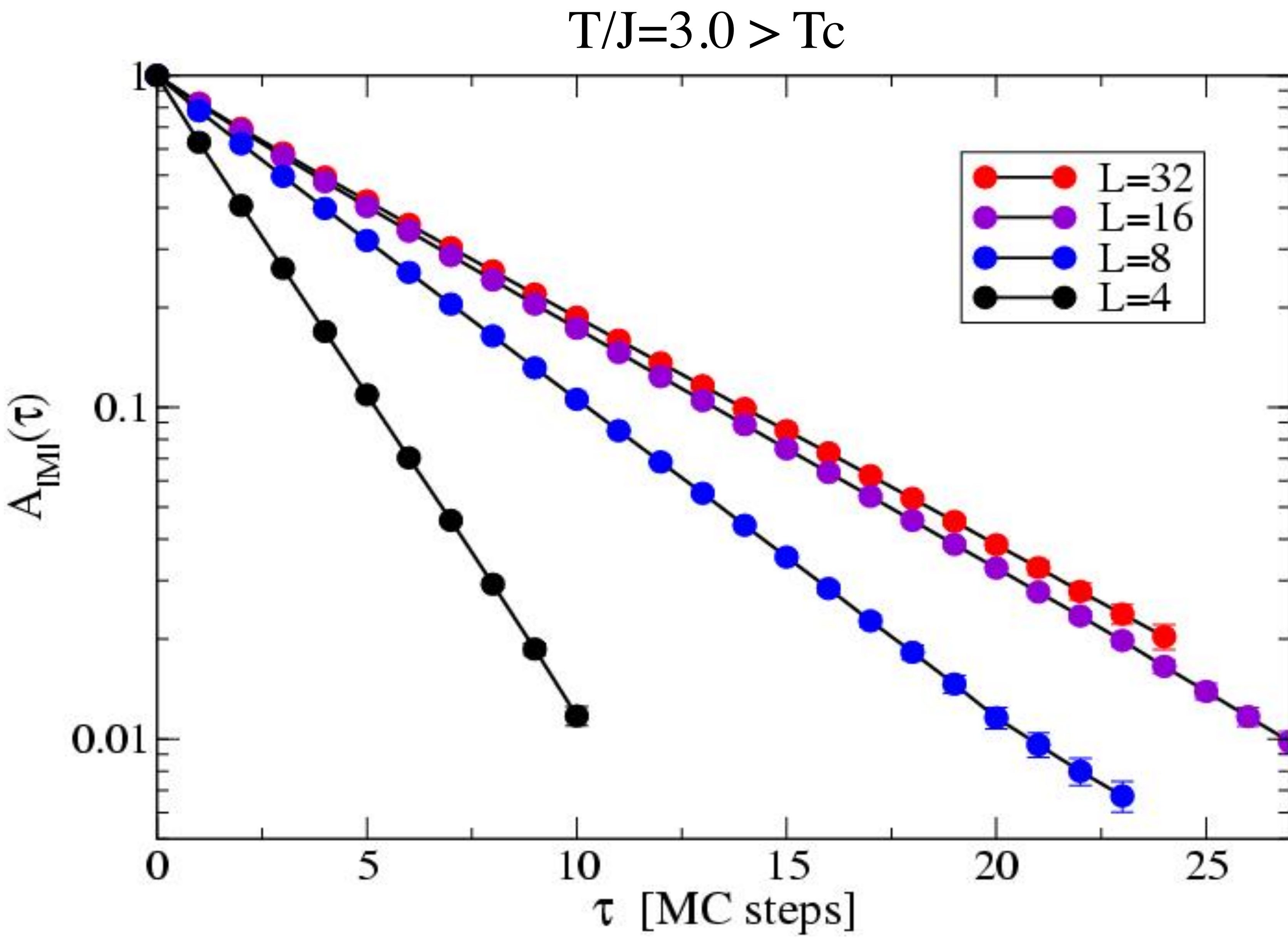
Critical slowing down

$$\Theta \to \infty \text{ as } T \to T_c$$

At a critical point for system of length L; Q=order parameter

$$\Theta \sim L^z, \quad z = \text{ dynamic exponent}$$

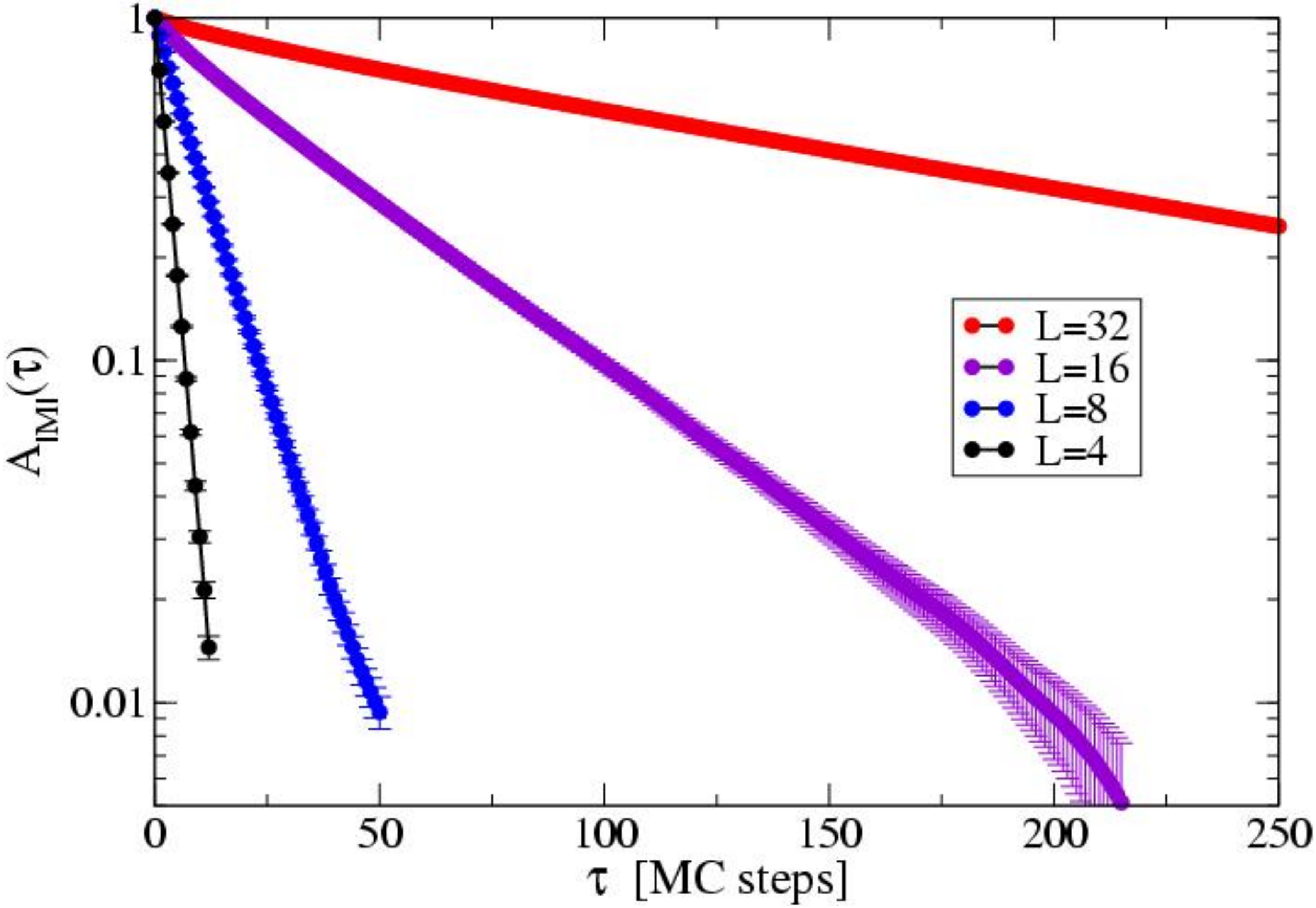
2D Ising autocorrelation functions for |M|



Exponentially decaying autocorrelation function

- convergent autocorrelation time as L increases

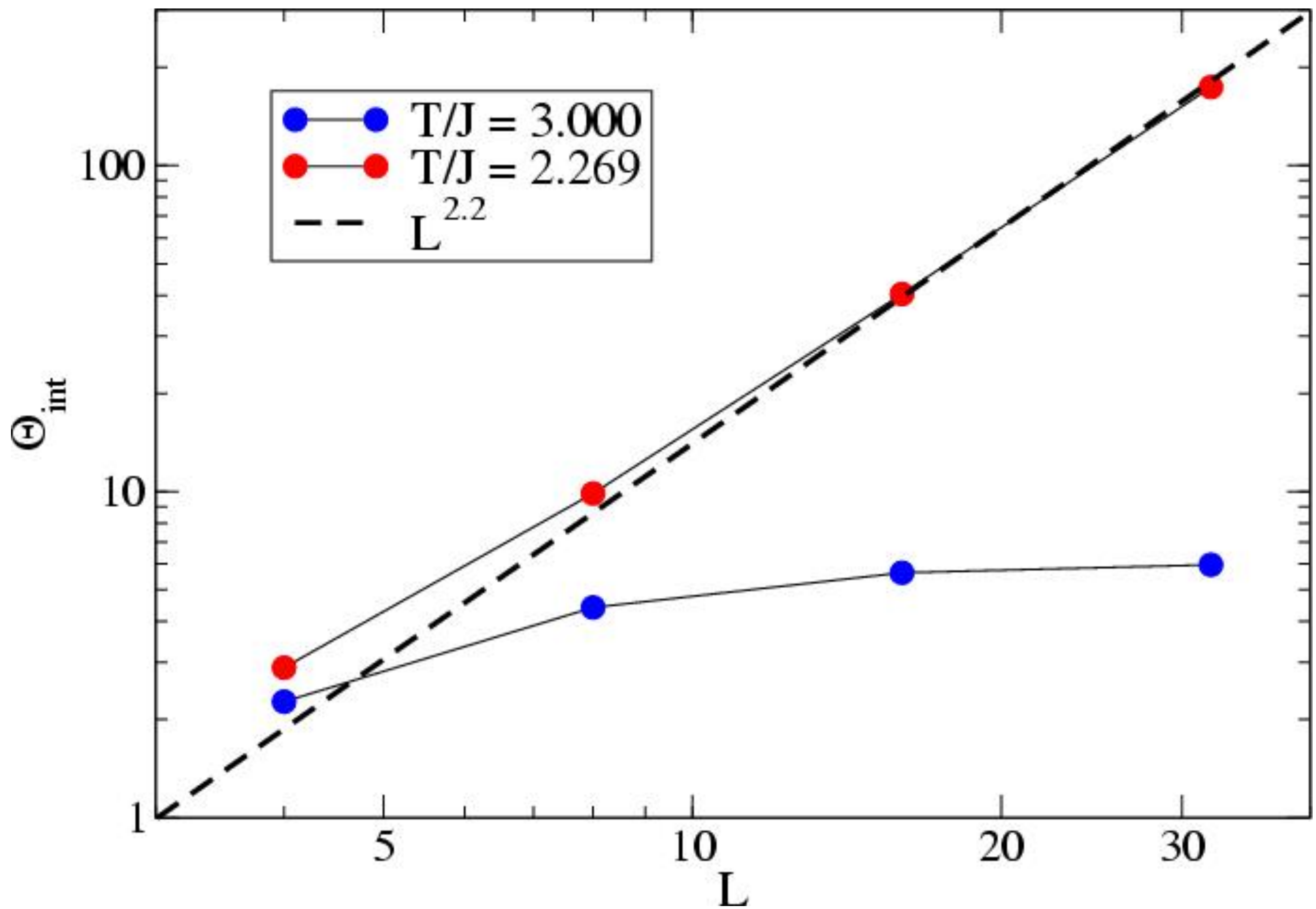
T/J = 2.269 = Tc



Autocorrelation time diverges with L

Critical slowing down

Dynamic exponent Z: $\Theta, \Theta_{\mathrm{int}} \sim L^Z$



For the Metropolis algorithm (Metropolis dynamics) $Z \approx 2.2$

How to calculate autocorrelation functions

If we want autocorrelations for up to K MC step separations, we need to store K successive measurements of quantity Q

Store values in vector `tobs[1:K]`; first k steps to fill the vector. Then, shift values after each step, add latest measurement:

vector contents after MC step n

Q_n	Q_{n-1}	Q_{n-2}	...		...	Q_{n-K+1}

vector contents after MC step n+1

Q_{n+1}	Q_n	Q_{n-1}	...		...	Q_{n-K+2}

Accumulate time-averaged correlation functions of Q (variable q)

```
for t=2:k
   tobs[t]=tobs[t-1]
end
tobs[1]=q
for t=0:k-1
   acorr[t]=acorr[t]+tobs[1]*tobs[1+t]
end
```