Irreversible worm dynamics

On critical speeding-up for Ising models in high dimensions

Eren M. Elçi <elci@posteo.de>, School of Mathematical Sciences (Monash University) & Bayer AG

Collaboration

- Jens Grimm (Monash, 🇦🇺)
- Lijie Ding (USTC, 🇨🇳)
- Abrahim Nasrawi (Monash, 🇦🇺)
- Timothy Garoni (Monash, 🇦🇺)
- Youjin Deng (USTC, 🇨🇳)

Lifted Worm Algorithm for the Ising Model

Eren Metin Elçi[1], Jens Grimm[2], Lijie Ding[3], Abrahim Nasrawi[2], Timothy M. Garoni[2], and Youjin Deng[3]
[1]*School of Mathematical Sciences, Monash University, Clayton, Victoria 3800, Australia*
[2]*ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS), School of Mathematical Sciences, Monash University, Clayton, VIC 3800, Australia*
[3]*Department of Physics, University of Science and Technology of China, Hefei, Anhui 230026, China and*
[4]*National Laboratory for Physical Sciences at Microscale and Department of Modern Physics, University of Science and Technology of China, Hefei, Anhui 230026, China*

We design an irreversible worm algorithm for the zero-field ferromagnetic Ising model by using the lifting technique. We study the dynamic critical behavior of an energy estimator on both the complete graph and toroidal grids, and compare our findings with reversible algorithms such as the Prokof'ev-Svistunov worm algorithm. Our results show that the lifted worm algorithm improves the dynamic exponent of the energy estimator on the complete graph, and leads to a significant constant improvement on toroidal grids.

I. INTRODUCTION

Markov-chain Monte Carlo (MCMC) algorithms are a powerful and widely-used tool in various areas of physics and other disciplines, such as in machine learning [1] and statistics [2]. In many practical applications MCMC algorithms are constructed via the Metropolis [3] or heat bath update scheme [4]. Such algorithms are necessarily *reversible*.

One important example of a Metropolis algorithm is the Prokof'ev-Svistunov Worm Algorithm (P-S worm algorithm) which has widespread application for both classical and quantum systems [5, 6]. As opposed to cluster algorithms like the Wolff [7] or Swendsen-Wang algorithm [8], the updates of the worm algorithm are purely *local*. On the simple-cubic lattice with periodic bound-

For the Ising model on the complete graph, it was numerically observed that the lifted single-spin flip Metropolis algorithm improves the scaling (with volume) of the rate of decay of the autocorrelation function of the magnetization [14]. Another study [13] proved that a lifted MCMC algorithm for uniformly sampling leaves from a given tree reduces the mixing time. In other examples [16, 20, 22] it was numerically observed that lifting speeds up reversible MCMC algorithms by a possibly large constant factor but does not asymptotically affect the scaling with the system size.

In this work we investigate how lifting affects worm algorithms. More precisely, we design a lifted worm algorithm for the zero-field ferromagnetic Ising model, and numerically study the dynamic critical behavior of an estimator of the energy. Our simulations were performed on both the complete graph and toroidal grids in dimen-

https://arxiv.org/abs/1711.05346

Table of contents

Table of contents

Table of contents

Table of contents

1. From reversible to irreversible worm through Berretti-Sokal & lifting.

2. Non-extensive (*but impressive*) efficiency boosts on high-dim. tori.

3. Critical speeding-up on the complete graph.

Table of contents

1. From reversible to irreversible worm through Berretti-Sokal & lifting.
2. Non-extensive (*but impressive*) efficiency boosts on high-dim. tori.
3. Critical speeding-up on the complete graph.
4. An induced irreversible cluster algorithm for the Ising model.

P-S reversible worm

$$Z = 2^{|V|} \cosh^{|E|}(\beta) \sum_{\omega \in \mathcal{C}_0} \tanh^{|\omega|}(\beta)$$

Algorithm 1 P-S Worm Algorithm

if $\omega \in \mathcal{C}_0$ **then**
  Choose a uniformly random vertex x
else
  Choose a uniformly random odd vertex x
end if
Choose a uniformly random edge xx' among the set of edges incident to x. With probability $a_{\text{P-S}}(\omega, \omega\Delta xx')$, let $\omega \to \omega\Delta xx'$. Otherwise $\omega \to \omega$

P-S := Prokof'ev-Svistunov

P-S reversible worm

- Applications in classical and quantum systems

$$Z = 2^{|V|} \cosh^{|E|}(\beta) \sum_{\omega \in \mathcal{C}_0} \tanh^{|\omega|}(\beta)$$

Algorithm 1 P-S Worm Algorithm

if $\omega \in \mathcal{C}_0$ **then**
  Choose a uniformly random vertex x
else
  Choose a uniformly random odd vertex x
end if
Choose a uniformly random edge xx' among the set of edges incident to x. With probability $a_{\text{P-S}}(\omega, \omega\Delta xx')$, let $\omega \to \omega\Delta xx'$. Otherwise $\omega \to \omega$

P-S := Prokof'ev-Svistunov

P-S reversible worm

- Applications in classical and quantum systems
  - <u>Here classical Ising only</u>

$$Z = 2^{|V|} \cosh^{|E|}(\beta) \sum_{\omega \in \mathcal{C}_0} \tanh^{|\omega|}(\beta)$$

Algorithm 1 P-S Worm Algorithm

if $\omega \in \mathcal{C}_0$ **then**
    Choose a uniformly random vertex x
else
    Choose a uniformly random odd vertex x
end if
Choose a uniformly random edge xx' among the set of edges incident to x. With probability $a_{\text{P-S}}(\omega, \omega\Delta xx')$, let $\omega \to \omega\Delta xx'$. Otherwise $\omega \to \omega$

P-S := Prokof'ev-Svistunov

P-S reversible worm

- Applications in classical and quantum systems
  - Here classical Ising only
- Updates completely local.

$$Z = 2^{|V|} \cosh^{|E|}(\beta) \sum_{\omega \in \mathcal{C}_0} \tanh^{|\omega|}(\beta)$$

Algorithm 1 P-S Worm Algorithm

if $\omega \in \mathcal{C}_0$ **then**
    Choose a uniformly random vertex x
else
    Choose a uniformly random odd vertex x
end if
Choose a uniformly random edge xx' among the set of edges incident to x. With probability $a_{\text{P-S}}(\omega, \omega \Delta xx')$, let $\omega \to \omega \Delta xx'$. Otherwise $\omega \to \omega$

P-S := Prokof'ev-Svistunov

P-S reversible worm

- Applications in classical and quantum systems
  - Here classical Ising only
- Updates completely local.
- Rapidly mixing on any graph and any temperature.

$$Z = 2^{|V|} \cosh^{|E|}(\beta) \sum_{\omega \in \mathcal{C}_0} \tanh^{|\omega|}(\beta)$$

Algorithm 1 P-S Worm Algorithm

if $\omega \in \mathcal{C}_0$ **then**
    Choose a uniformly random vertex x
else
    Choose a uniformly random odd vertex x
end if
Choose a uniformly random edge xx' among the set of edges incident to x. With probability $a_{\text{P-S}}(\omega, \omega\Delta xx')$, let $\omega \to \omega\Delta xx'$. Otherwise $\omega \to \omega$

P-S := Prokof'ev-Svistunov

P-S reversible worm

- Applications in classical and quantum systems
  - Here classical Ising only
- Updates completely local.
- Rapidly mixing on any graph and any temperature.
- On Z^3_L more efficient than SW for measuring susceptibility & 2nd-moment correlation length.

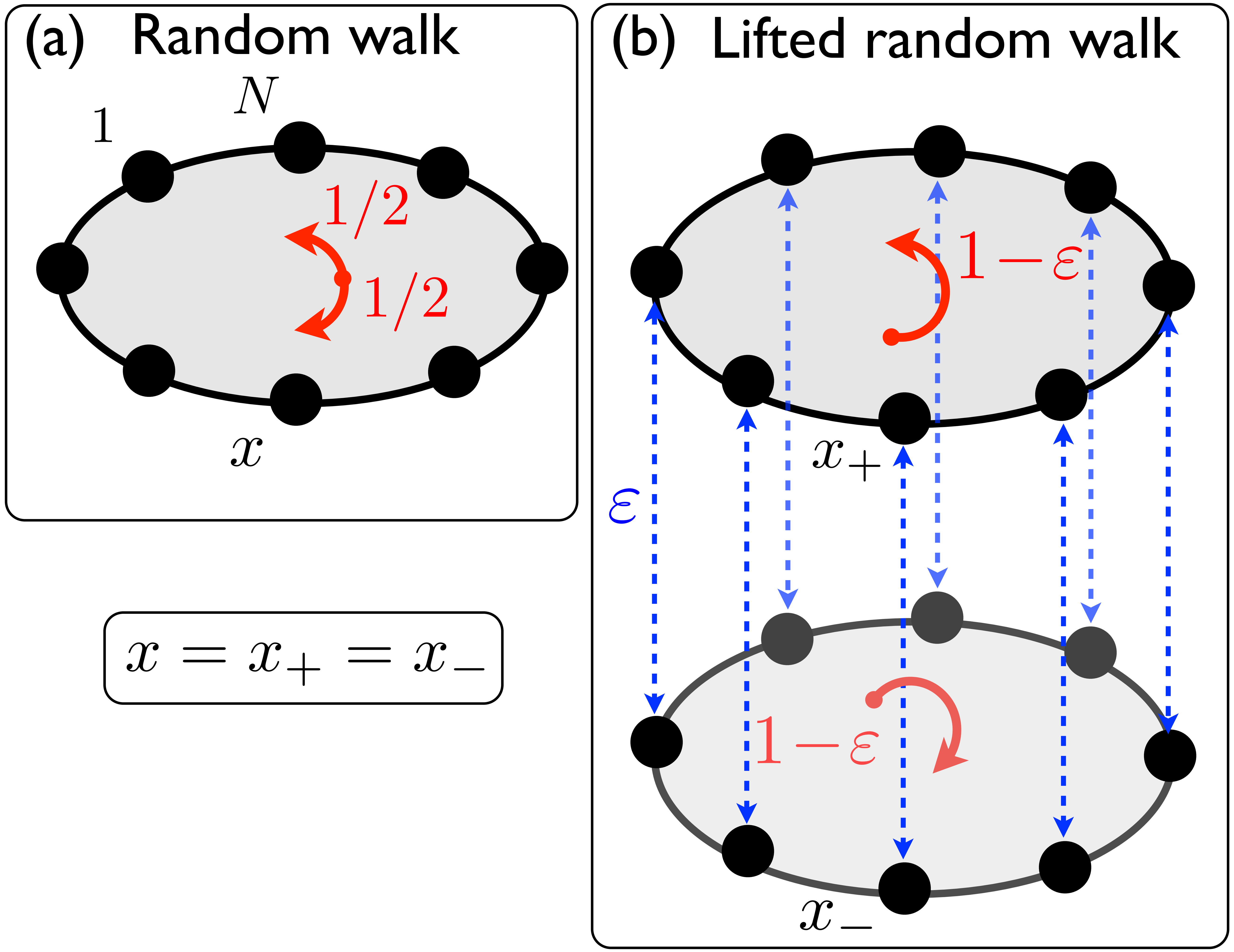
$$Z = 2^{|V|} \cosh^{|E|}(\beta) \sum_{\omega \in \mathcal{C}_0} \tanh^{|\omega|}(\beta)$$

Algorithm 1 P-S Worm Algorithm

if $\omega \in \mathcal{C}_0$ **then**
  Choose a uniformly random vertex x
else
  Choose a uniformly random odd vertex x
end if
Choose a uniformly random edge xx' among the set of edges incident to x. With probability $a_{\text{P-S}}(\omega, \omega\Delta xx')$, let $\omega \to \omega\Delta xx'$. Otherwise $\omega \to \omega$

P-S := Prokof'ev-Svistunov

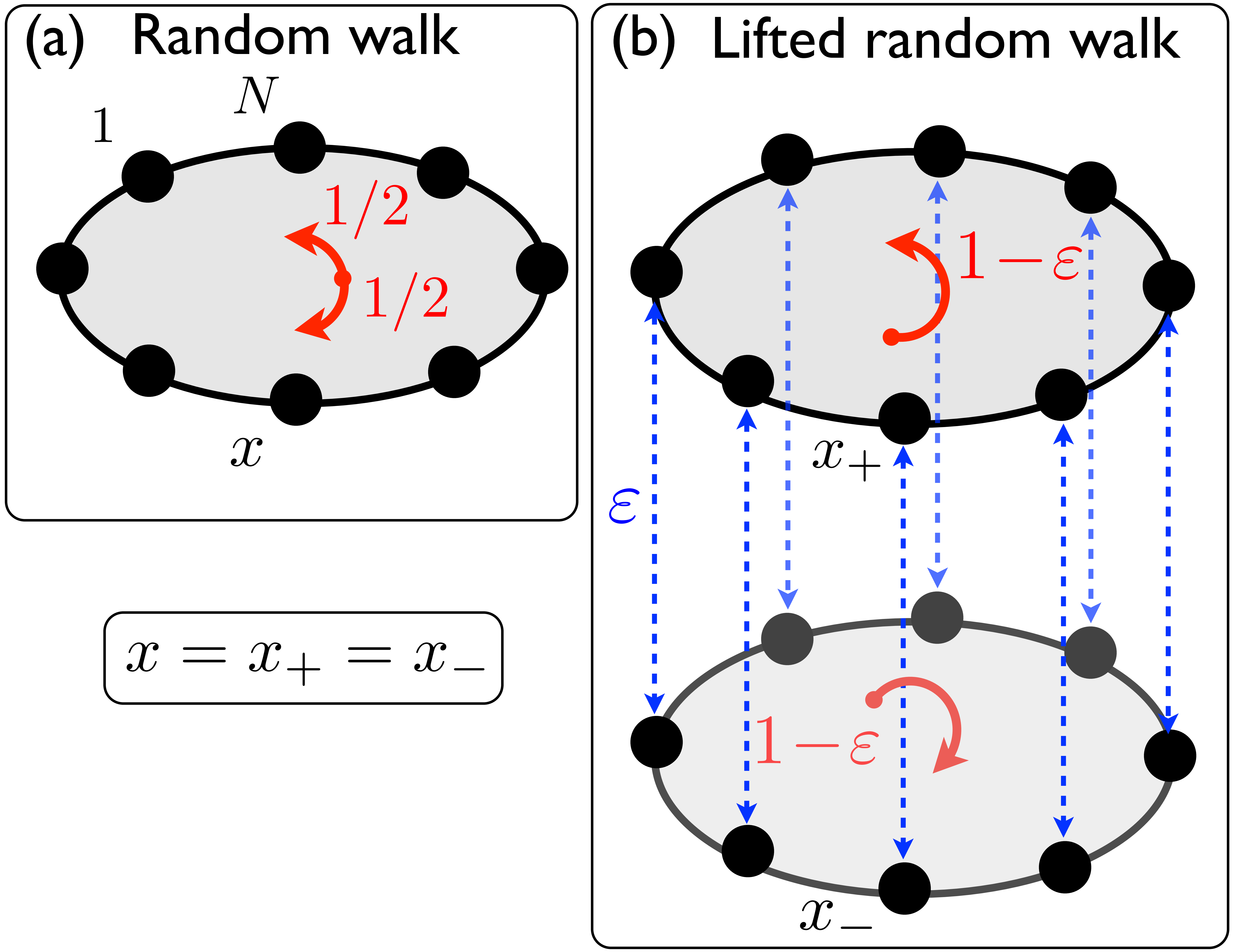
The lifting technique



$$x = x_+ = x_-$$

Figure taken from "Lifting — A nonreversible Markov chain Monte Carlo algorithm" by M. Vucelja, arXiv:1412.8762

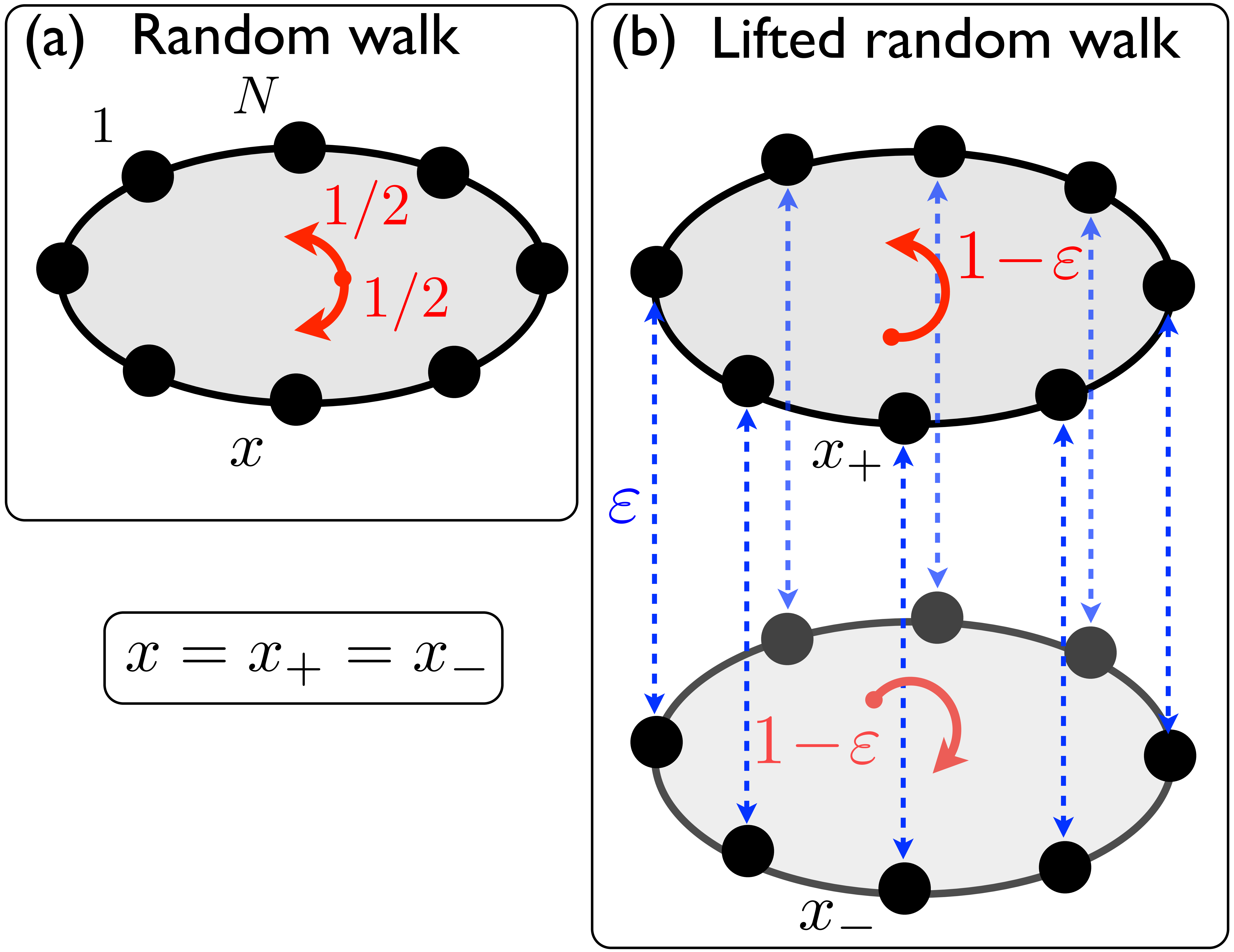
The lifting technique



- For worm dynamics one expects energy (edge) diffusion to be the slowest "mode" (cause of slowing-down)

Figure taken from "Lifting — A nonreversible Markov chain Monte Carlo algorithm" by M. Vucelja, arXiv:1412.8762

The lifting technique



- For worm dynamics one expects energy (edge) diffusion to be the slowest “mode” (cause of slowing-down)
- Idea: Apply lifting along “edge”-direction, but how?

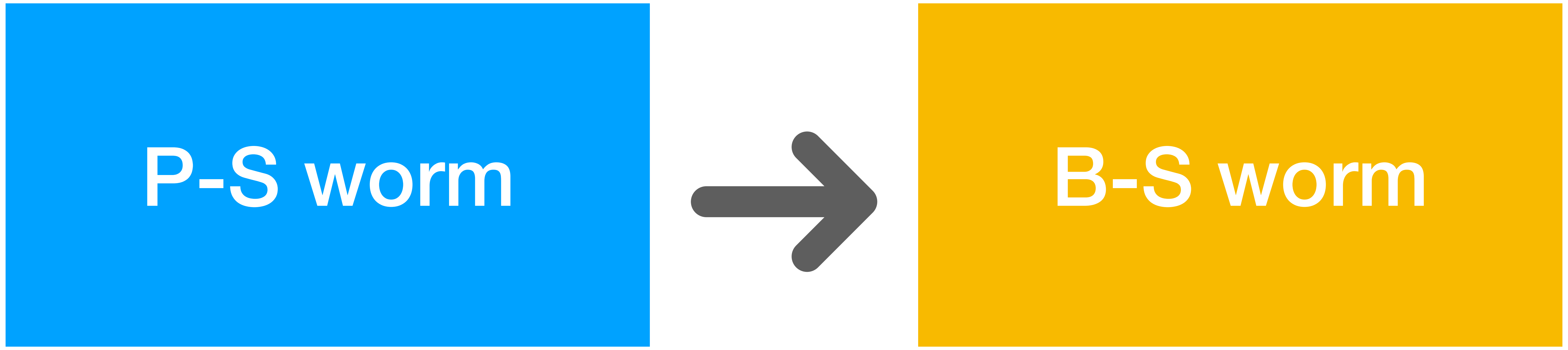
Figure taken from “Lifting — A nonreversible Markov chain Monte Carlo algorithm” by M. Vucelja, arXiv:1412.8762

Our construction

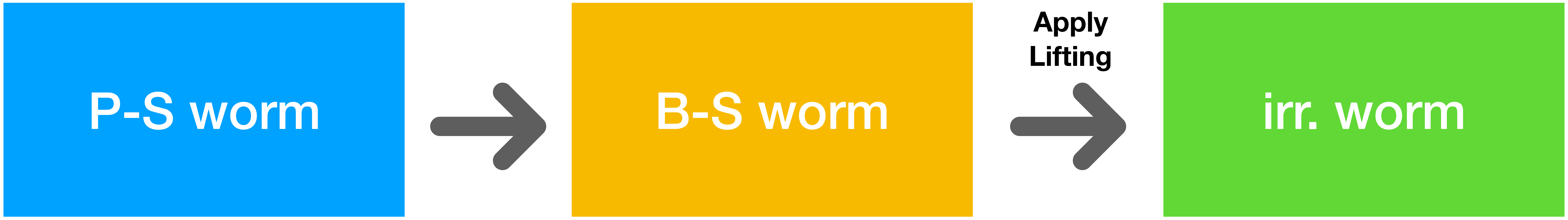
Our construction

P-S worm

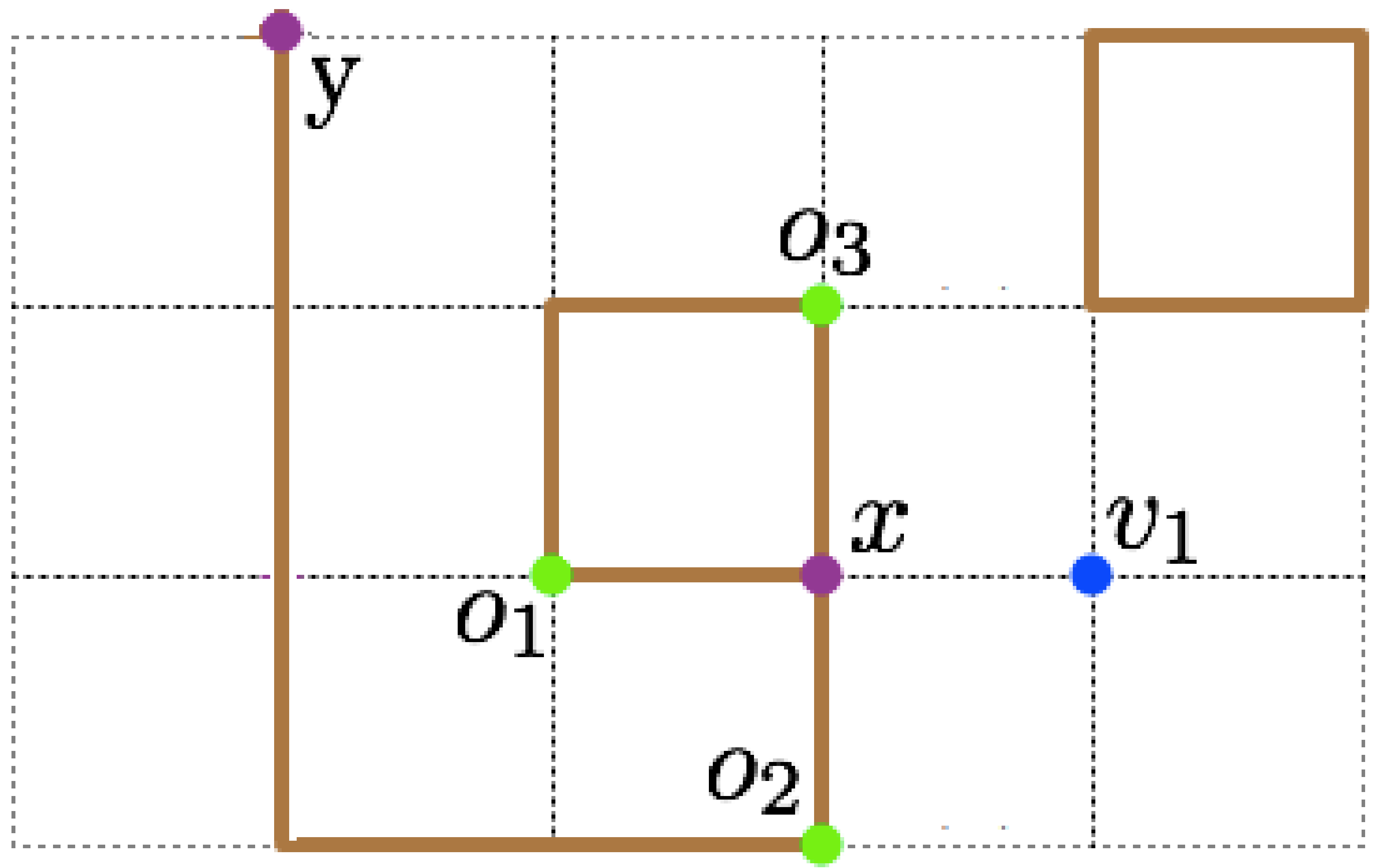
Our construction



Our construction



The Berretti-Sokal worm



Algorithm 2 B-S type Worm Algorithm

Choose $\lambda = \{+, -\}$ uniformly at random
if $\omega \in \mathcal{C}_0$ **then**
    Choose a uniformly random vertex x
else
    Choose a uniformly random odd vertex x
end if

if $N_\omega(x, \lambda) = \emptyset$ **then**
    Set $\omega \to \omega$ and skip all following steps
else
    Choose a uniformly random edge $xx' \in N_\omega(x, \lambda)$. With probability $a_{\text{B-S}}(\omega, \omega\Delta xx')$, let $\omega \to \omega\Delta xx'$. Otherwise $\omega \to \omega$.
end if

Lifting the B-S worm

Algorithm 2 B-S type Worm Algorithm

Choose $\lambda = \{+, -\}$ uniformly at random
if $\omega \in \mathcal{C}_0$ **then**
    Choose a uniformly random vertex x
else
    Choose a uniformly random odd vertex x
end if

if $N_\omega(x, \lambda) = \emptyset$ **then**
    Set $\omega \to \omega$ and skip all following steps
else
    Choose a uniformly random edge $xx' \in N_\omega(x, \lambda)$. With probability $a_{\text{B-S}}(\omega, \omega\Delta xx')$, let $\omega \to \omega\Delta xx'$. Otherwise $\omega \to \omega$.
end if

Lifting the B-S worm

Algorithm 2 B-S type Worm Algorithm

Choose $\lambda = \{+, -\}$ uniformly at random
if $\omega \in \mathcal{C}_0$ **then**
  Choose a uniformly random vertex x
else
  Choose a uniformly random odd vertex x
end if

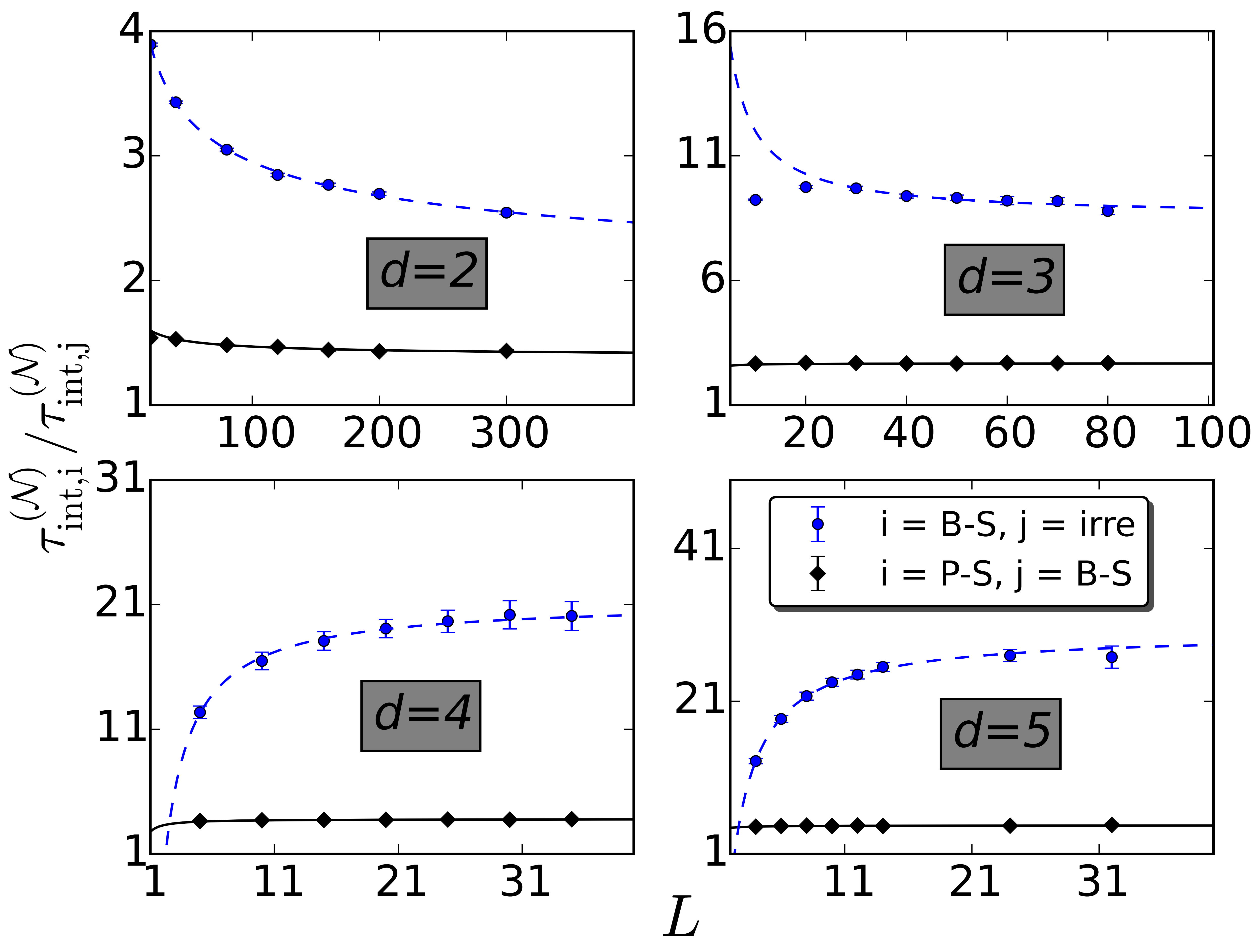
if $N_\omega(x, \lambda) = \emptyset$ **then**
  Set $\omega \to \omega$ and skip all following steps
else
  Choose a uniformly random edge $xx' \in N_\omega(x, \lambda)$. With probability $a_{\text{B-S}}(\omega, \omega\Delta xx')$, let $\omega \to \omega\Delta xx'$. Otherwise $\omega \to \omega$.
end if

Algorithm 3 Irreversible Worm Algorithm

if $\tilde{\omega} = (\omega, \lambda)$ where $\omega \in \mathcal{C}_0$ **then**
  Choose a uniformly random vertex x
else
  Choose a uniformly random odd vertex x
end if

if $N_\omega(x, \lambda) = \emptyset$ **then**
  Set $(\omega, \lambda) \to (\omega, -\lambda)$ and skip all following steps
else
  Choose a uniform random edge $xx' \in N_\omega(x, \lambda)$. With probability $a_{\text{B-S}}(\omega, \omega\Delta xx')$, let $(\omega, \lambda) \to (\omega\Delta xx', \lambda)$. Otherwise $(\omega, \lambda) \to (\omega, -\lambda)$
end if

Boosts on tori

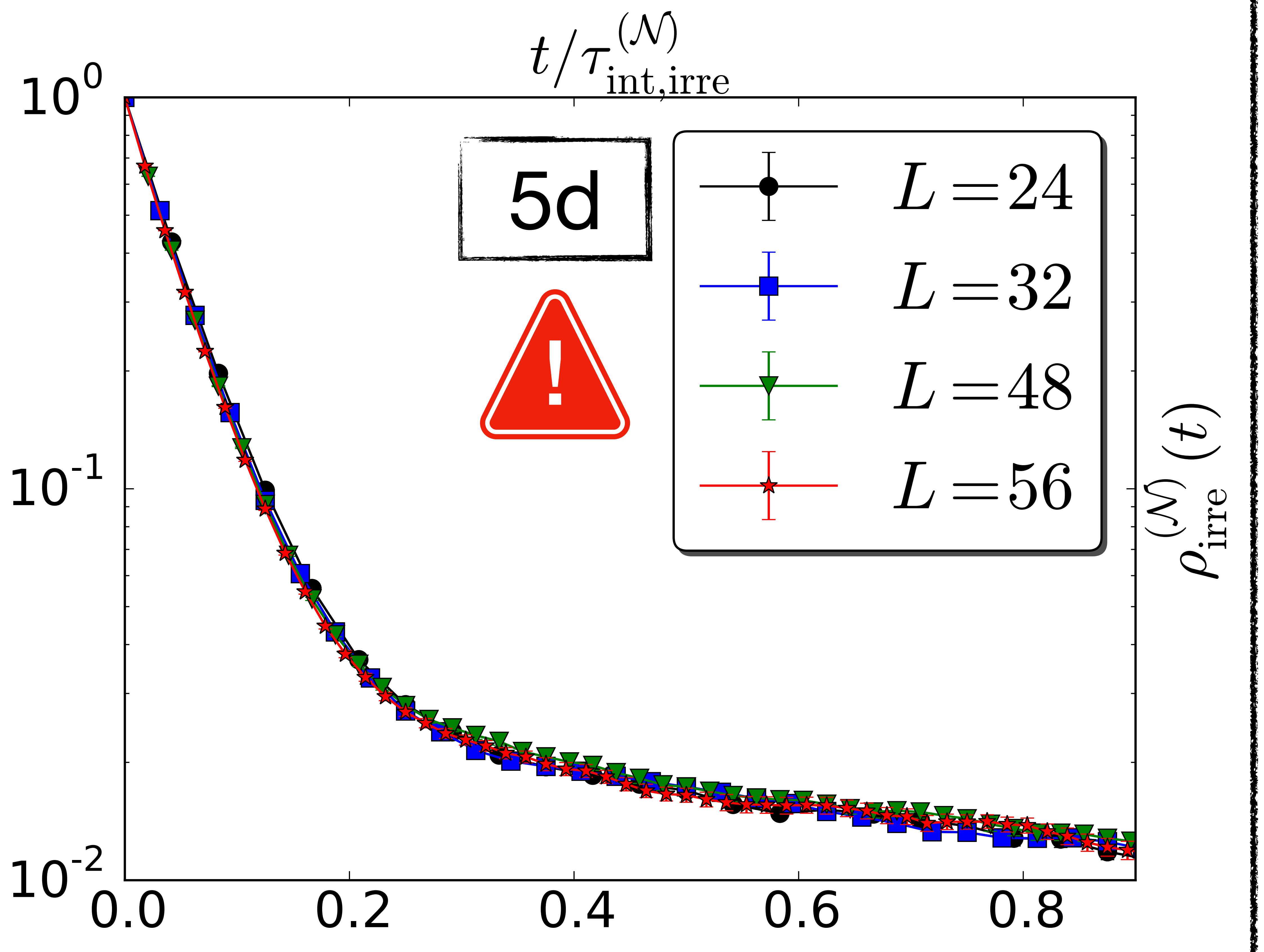


$$\mathrm{Var}(\overline{\mathcal{N}}) \sim 2\tau^{(\mathcal{N})}_{\mathrm{int}} \frac{\mathrm{Var}(\mathcal{N}_0)}{M} \qquad M \to \infty$$

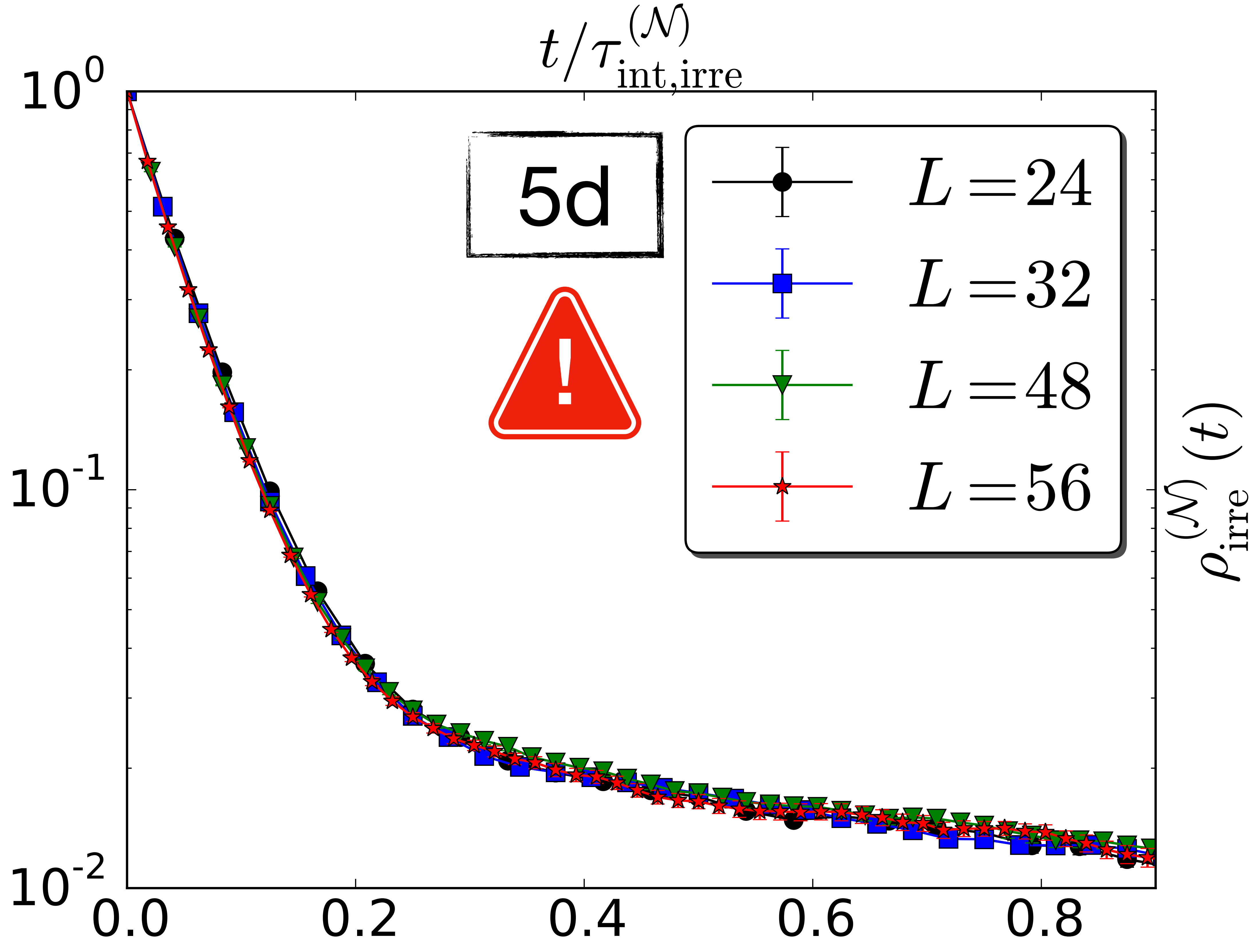
$$\tau^{(\mathcal{N})}_{\mathrm{int}} := \frac{1}{2} + \sum_{t=1}^{\infty} \rho^{(\mathcal{N})}(t).$$

	P-S $\to$ B-S	B-S $\to$ irre	P-S $\to$ irre
$d=2$	1.4(1)	1.7(2)	2.4(4)
$d=3$	2.68(9)	8.2(4)	22(2)
$d=4$	3.79(3)	24(1)	91(4)
$d=5$	4.7(1)	30(1)	141(6)

Slow suppressed modes

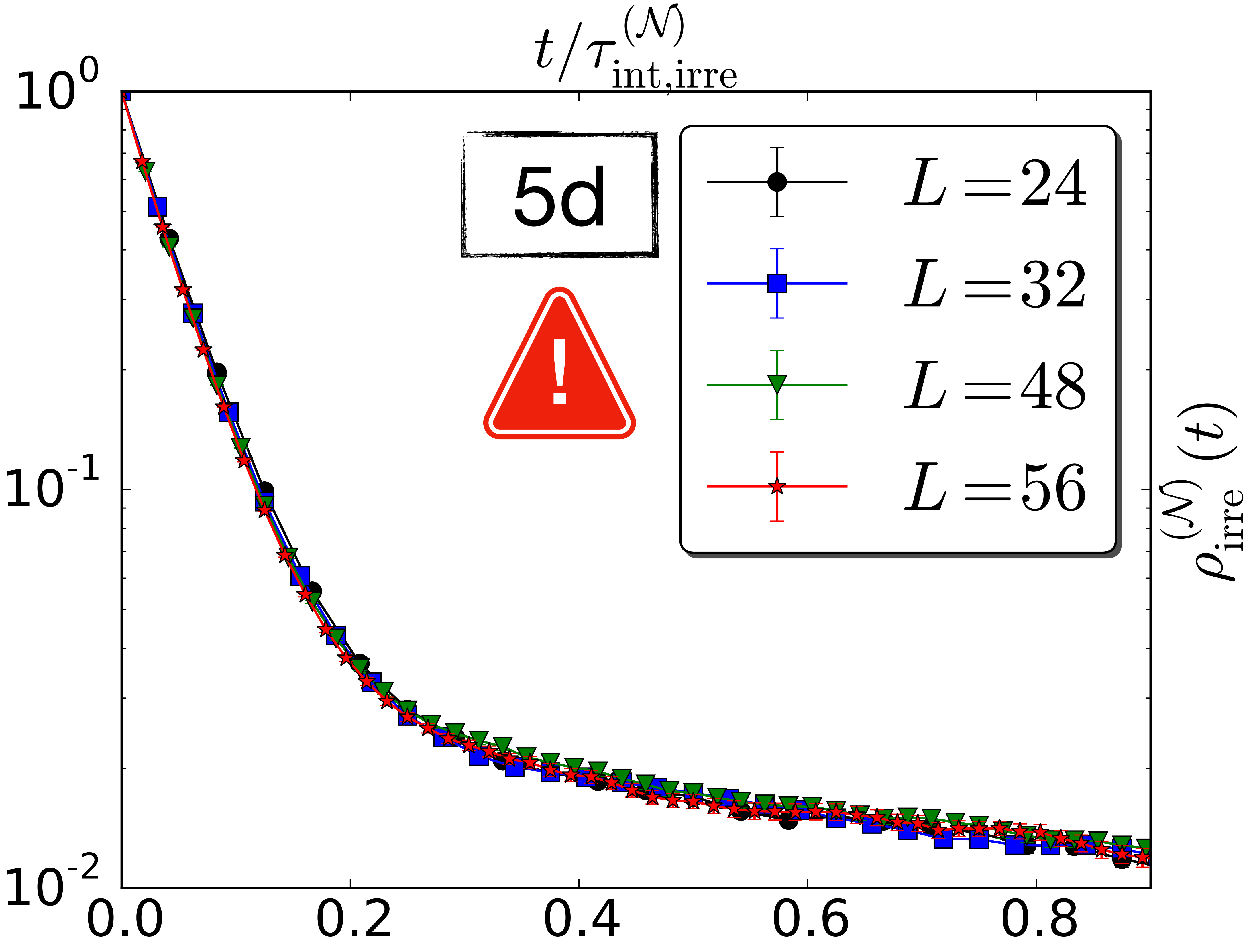


Slow suppressed modes



Two-time-scale ansatz

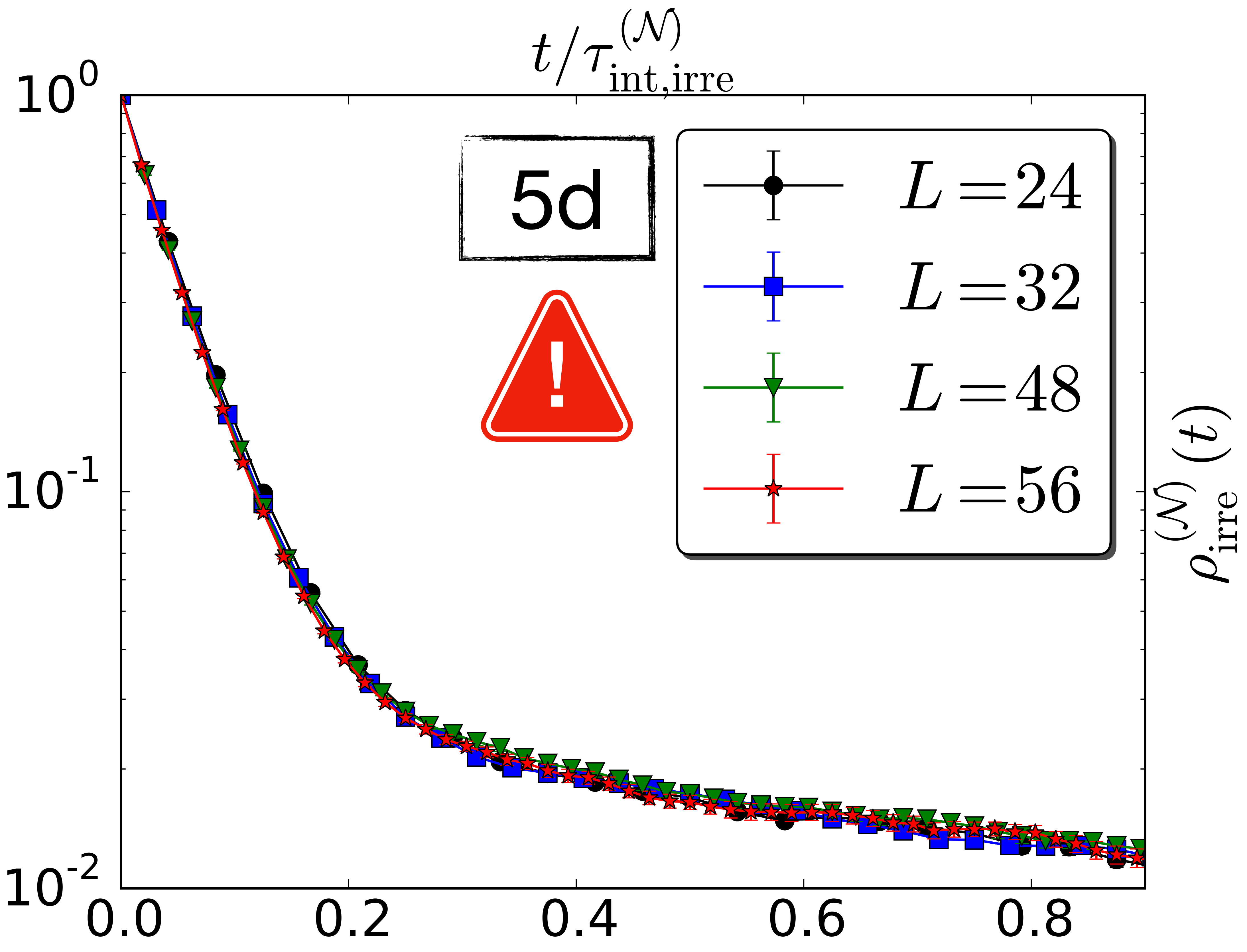
Slow suppressed modes



Two-time-scale ansatz

$$\rho^{(\mathcal{N})}_{\mathrm{irre}}(t) = \alpha_1 \exp(-t/\tau_1) + \alpha_2 \exp(-t/\tau_2)$$

Slow suppressed modes

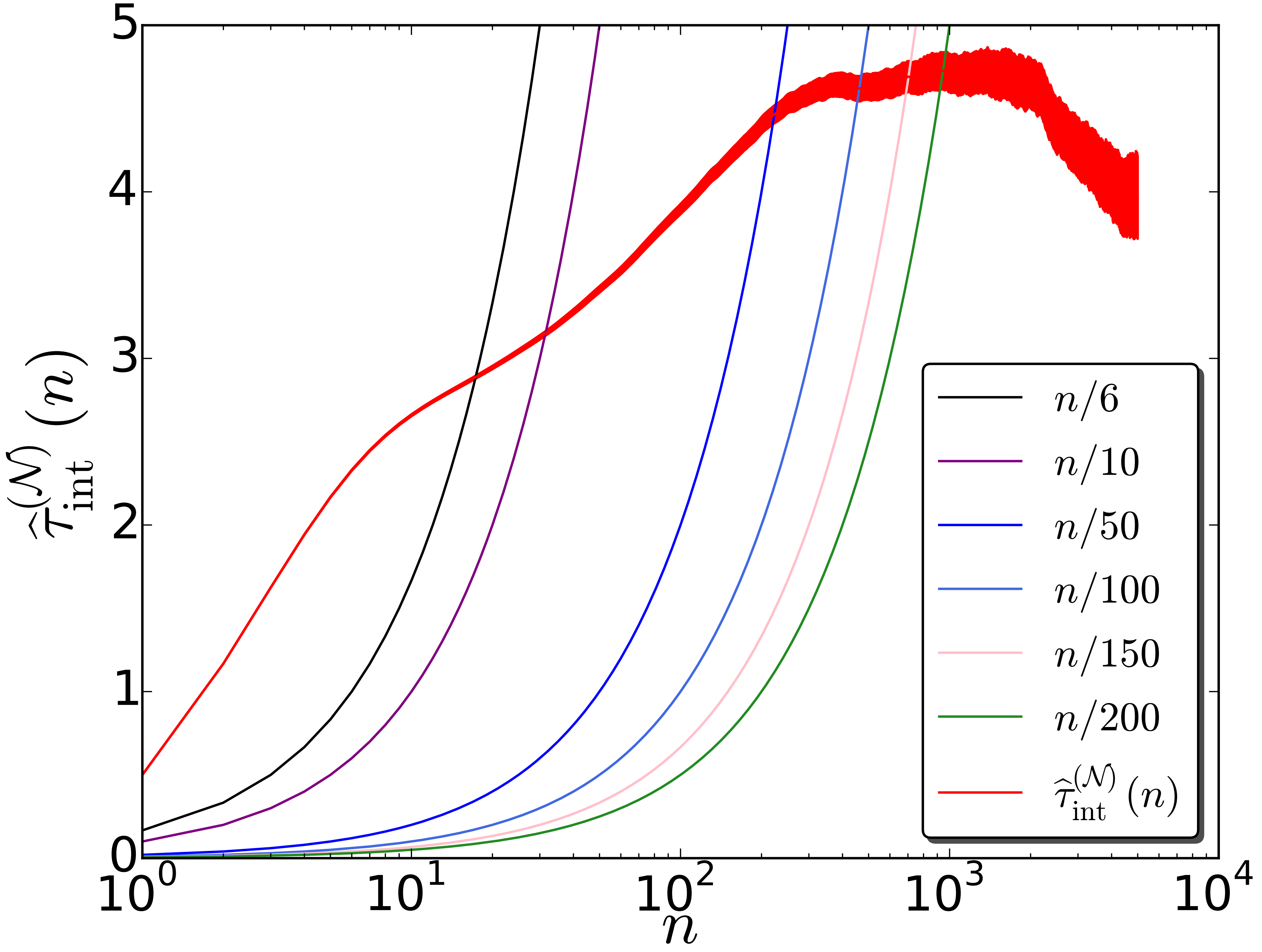


Two-time-scale ansatz

$$\rho^{(\mathcal{N})}_{\mathrm{irre}}(t) = \alpha_1 \exp(-t/\tau_1) + \alpha_2 \exp(-t/\tau_2)$$

$$\frac{\alpha_2}{\alpha_1} \overset{L\to\infty}{\sim} 40.6(4) \quad \& \quad \frac{\tau_1}{\tau_2} \overset{L\to\infty}{\sim} 32.4(6)$$

Slow suppressed modes



$$\widehat{\tau}_{\mathrm{int}}^{(\mathcal{N})}(n) := \frac{1}{2} + \sum_{t=1}^{n} \rho^{(\mathcal{N})}(t).$$

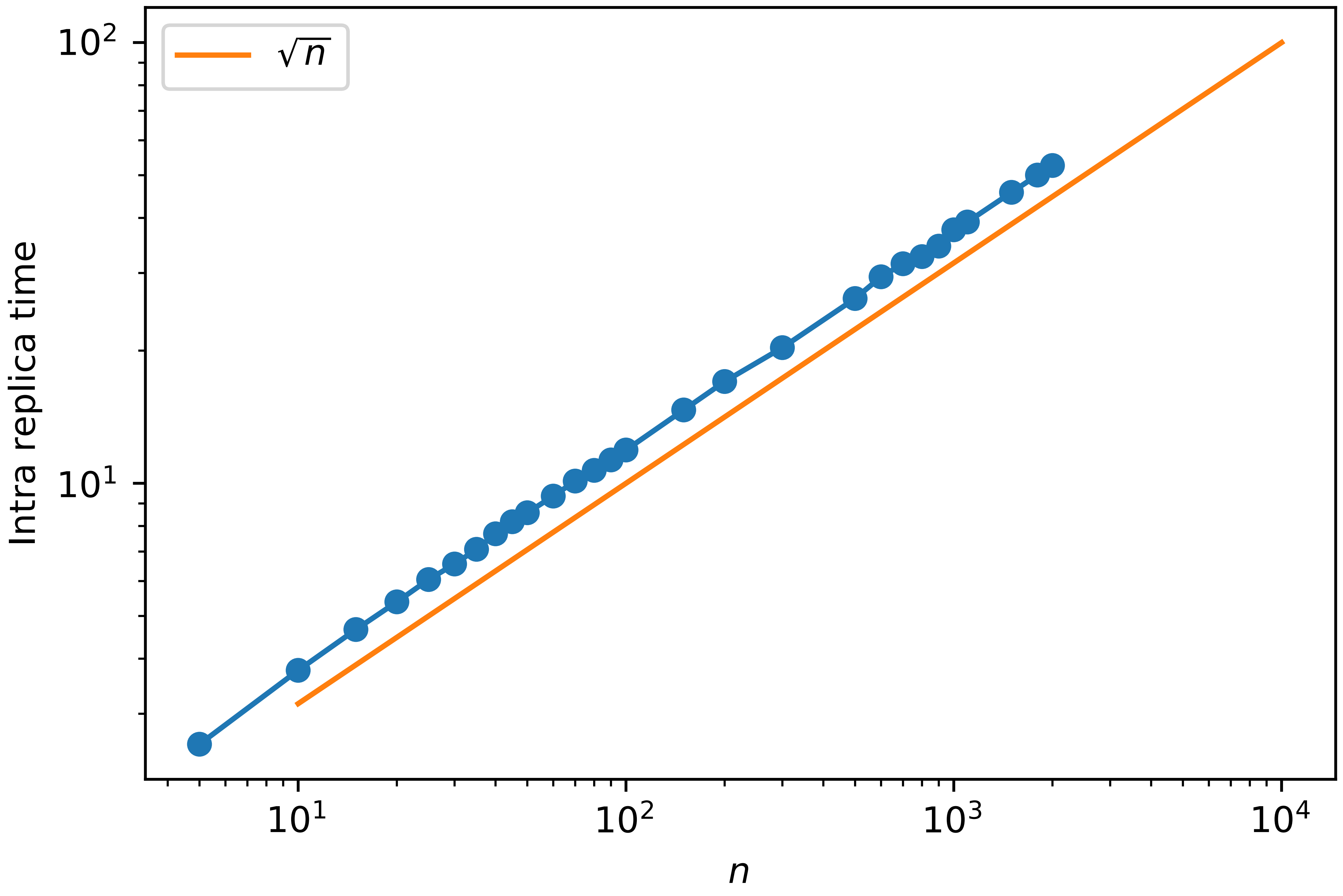
<u>Two-time-scale ansatz</u>

$$\rho_{\mathrm{irre}}^{(\mathcal{N})}(t) = \alpha_1 \exp(-t/\tau_1) + \alpha_2 \exp(-t/\tau_2)$$

$$\frac{\alpha_2}{\alpha_1} \overset{L\to\infty}{\sim} 40.6(4) \quad \& \quad \frac{\tau_1}{\tau_2} \overset{L\to\infty}{\sim} 32.4(6)$$

Critical speeding-up in the mean-field limit

Extensive ballistic drift



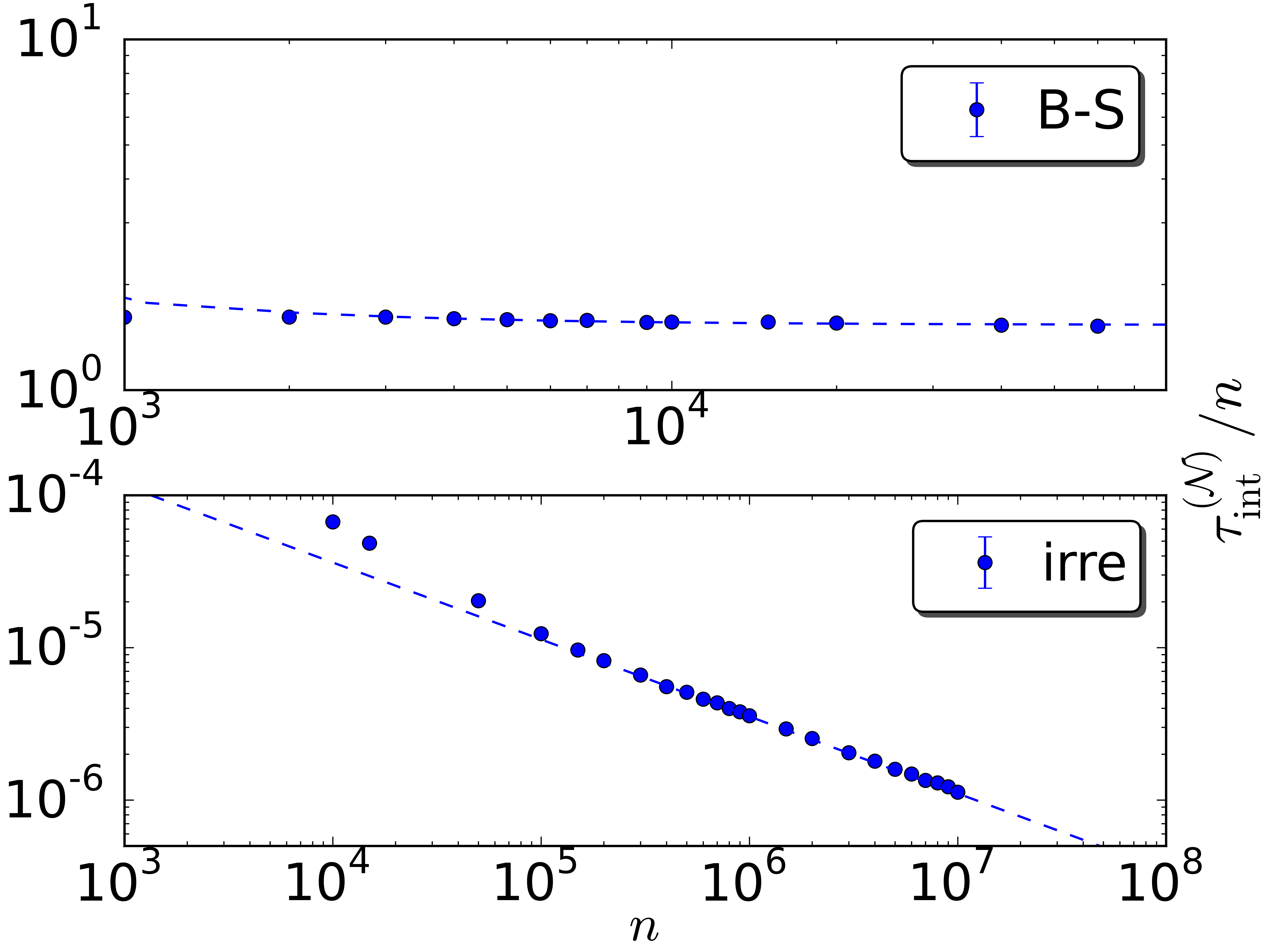
Some scaling results

$$\beta = \frac{1}{n} \Rightarrow x \sim \frac{1}{n}$$

$$\mu_N^{(\mathscr{C}_0)} \sim \frac{\Gamma(7/4)}{\sqrt{3}\Gamma(5/4)} n^{\frac{1}{2}} \approx 0.5854143 \; n^{\frac{1}{2}}$$

$$\sigma_N \sim \sqrt{\frac{9}{4} - \frac{24\Gamma(5/4)^4}{\pi^2}} n^{\frac{1}{2}} \approx 0.7801732 \; n^{\frac{1}{2}}$$

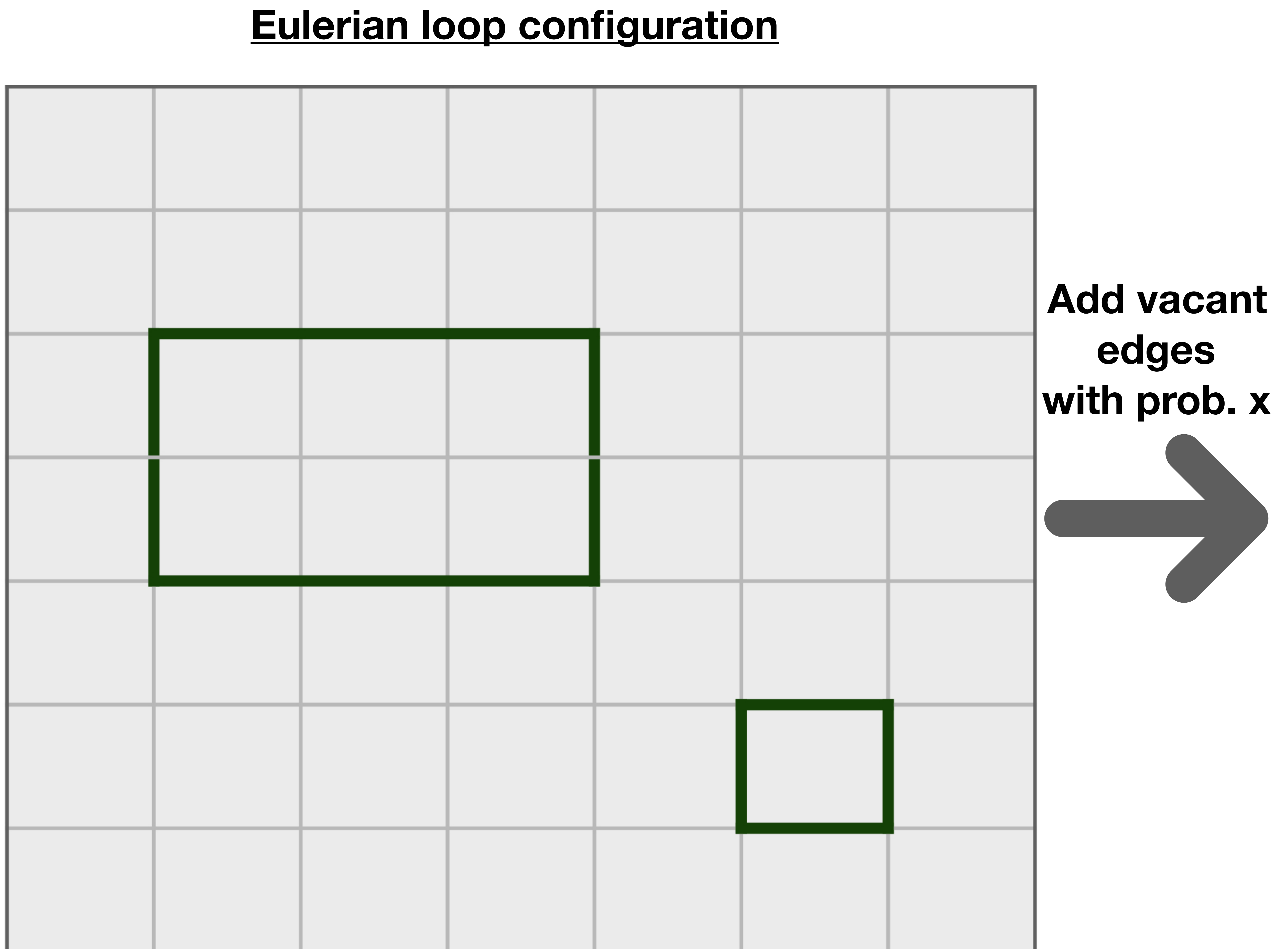
Critical speeding-up



$$\tau_{\mathrm{int},\mathcal{N}}^{(\mathrm{BS})} \approx n$$

$$\tau_{\mathrm{int},\mathcal{N}}^{(\mathrm{irr})} \approx n^{\frac{1}{2}}$$

An irreversible cluster algorithm

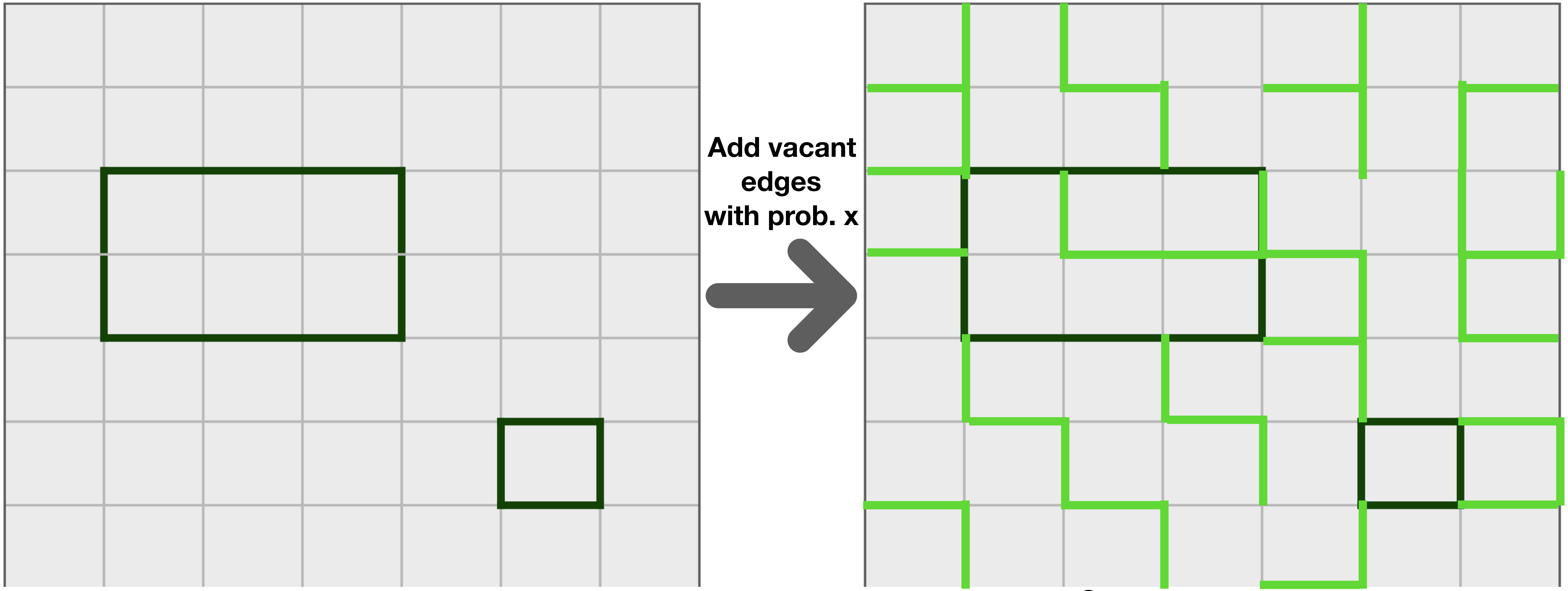


$x \in [0,1]$

Grimmett, Geoffrey, and Svante Janson. "Random even graphs." *the electronic journal of combinatorics* 16.1 (2009): R46.

An irreversible cluster algorithm

Eulerian loop configuration | **FK Ising configuration**



$x \in [0,1]$

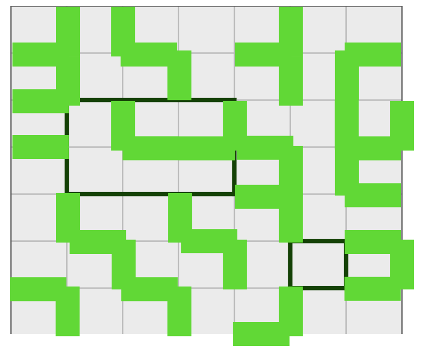
$p = \dfrac{2x}{1+x}$ $\qquad q = 2$

Grimmett, Geoffrey, and Svante Janson. "Random even graphs." *the electronic journal of combinatorics* 16.1 (2009): R46.

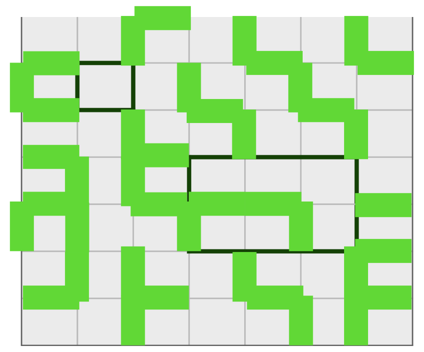
An irreversible cluster algorithm

$$\omega_1 \in \mathscr{C}_0 \to \omega_2 \in \mathscr{C}_2 \to \omega_3 \in \mathscr{C}_2 \to \cdots \to \omega_T \in \mathscr{C}_2 \to \omega_{T+1} \in \mathscr{C}_0$$

Random **running time T**



$$\mu_T \approx \chi$$

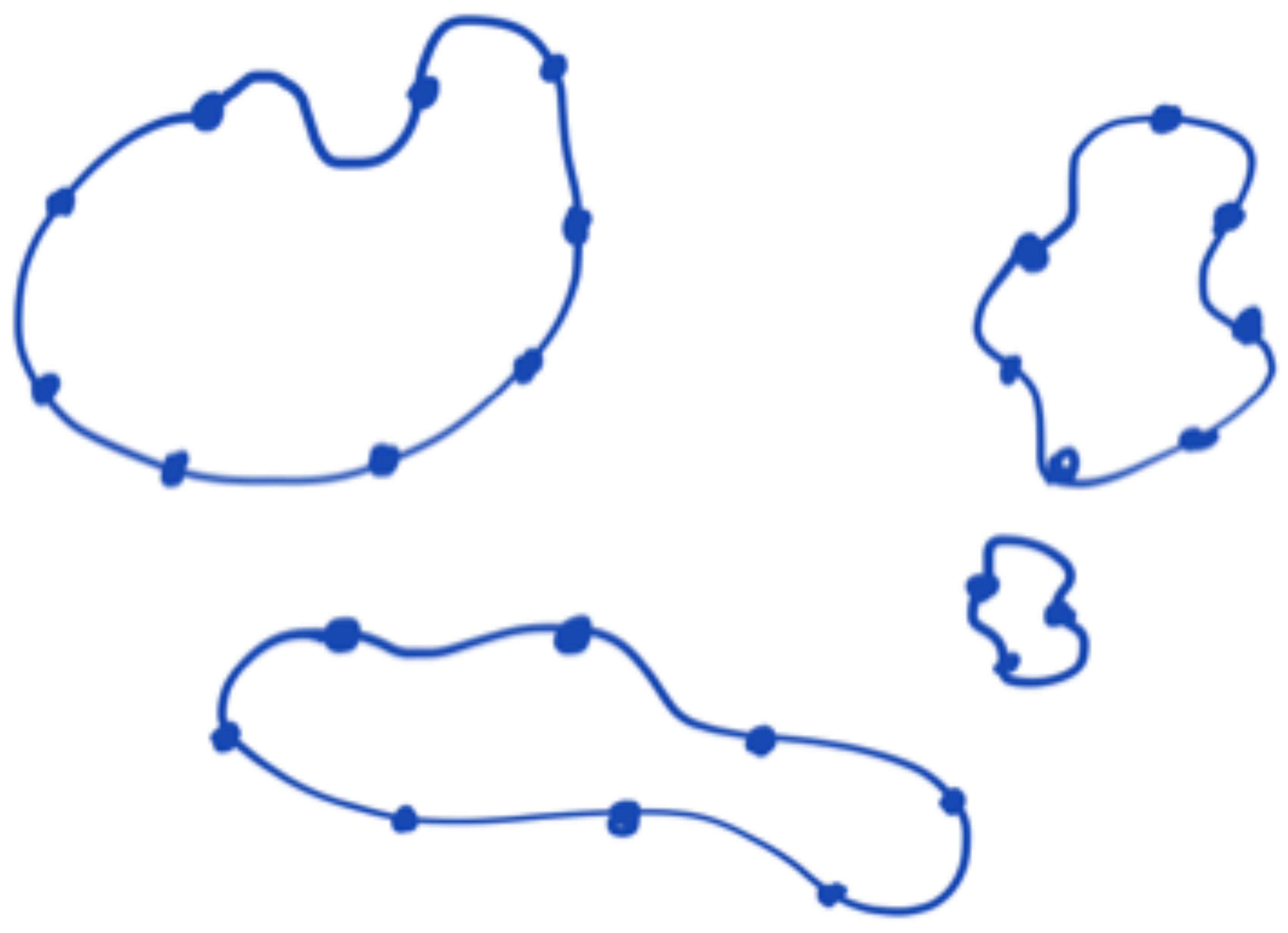


<u>Preliminary analysis / picture on complete graph</u>

- Scaling $\chi \approx n^{\frac{1}{2}}$
- Mean of T is dominant time scale (deviations that dominate mean asymptotically are at least exp. unlikely)
- Induced FK cluster algorithm has integrated autocorrelation for edges (energy) that is bounded as n becomes large.
- Compare: SW is expected to have mixing time $n^{\frac{1}{4}}$

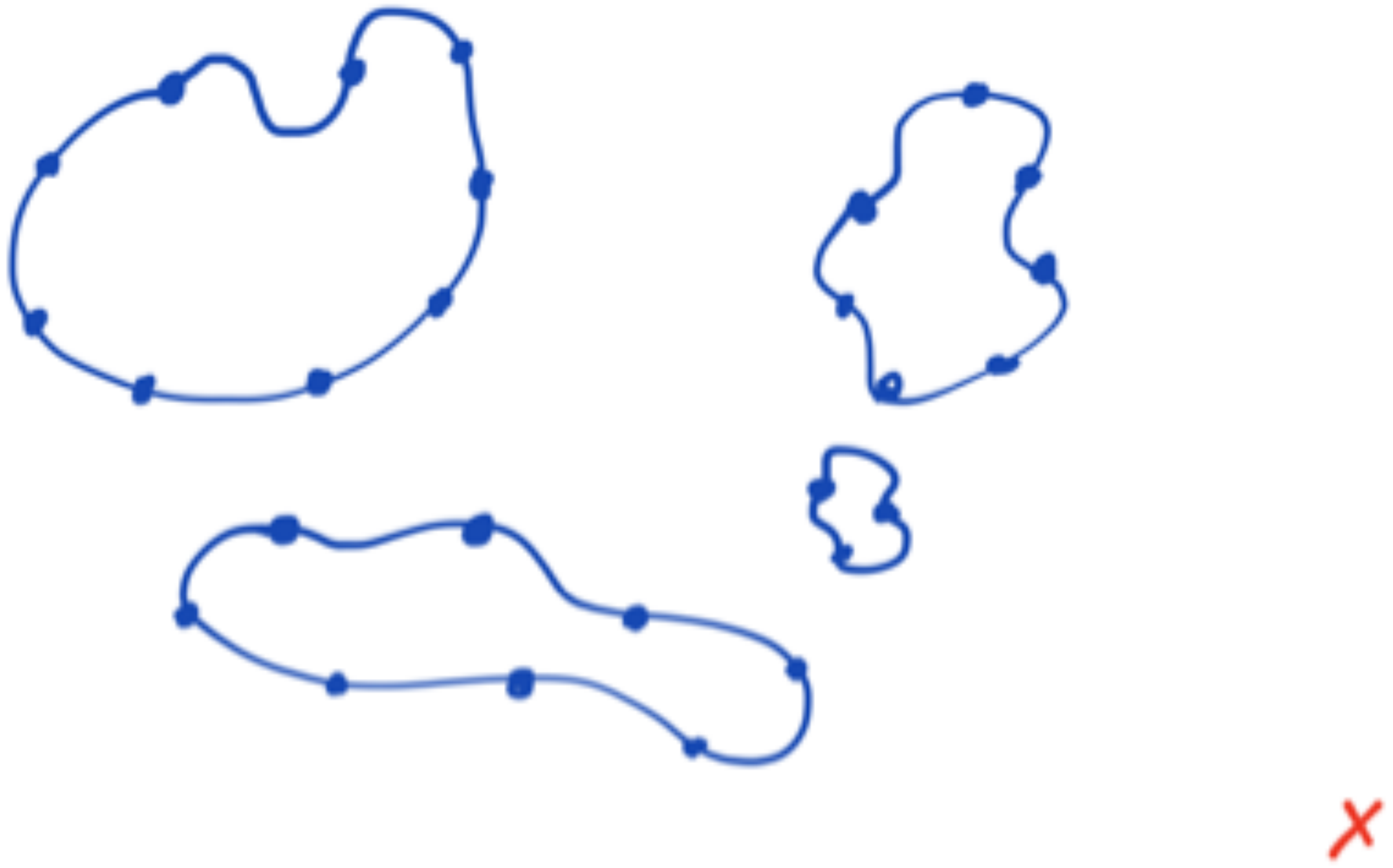
Ballistic path creation

- Some text



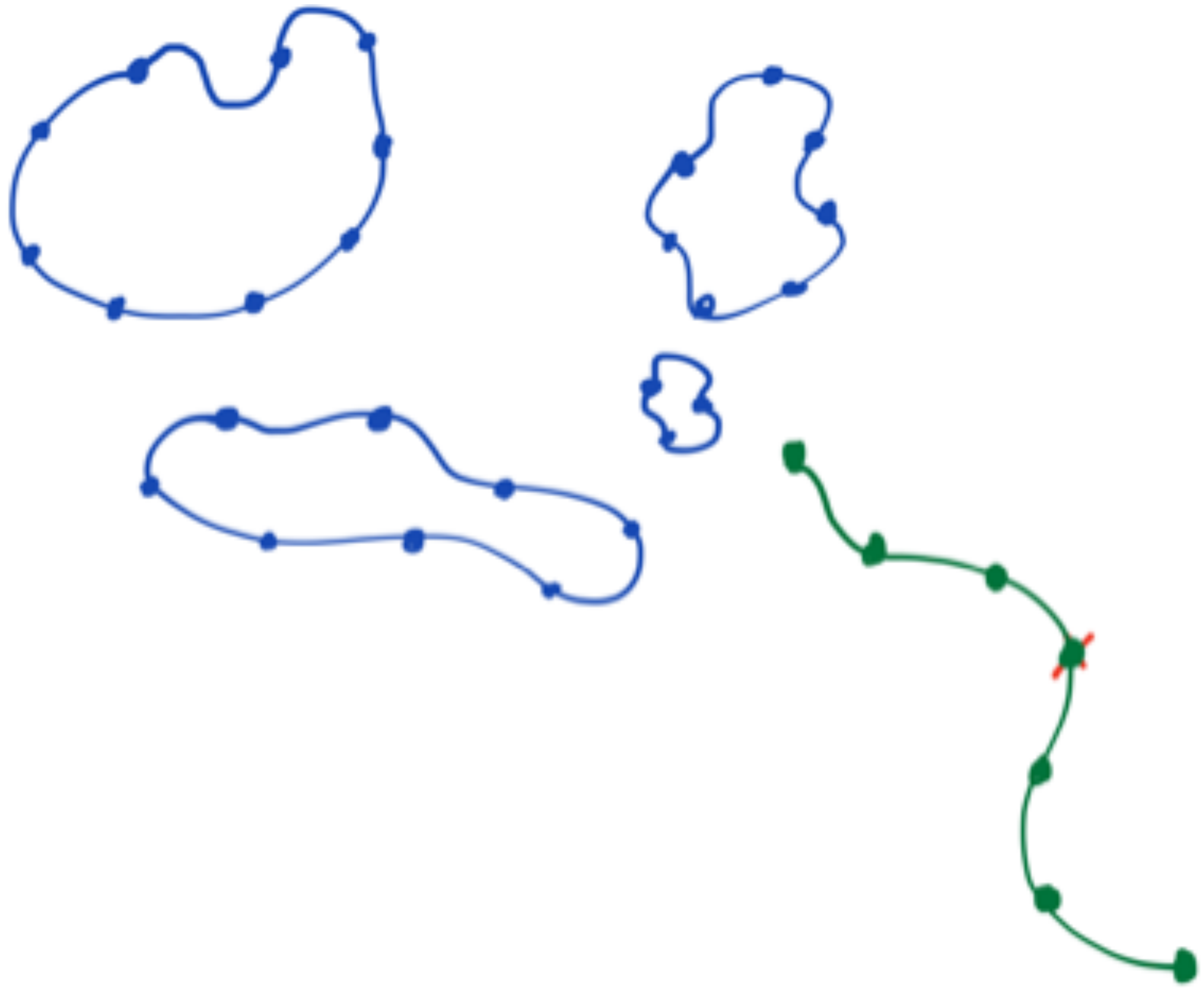
Ballistic path creation

- Some text



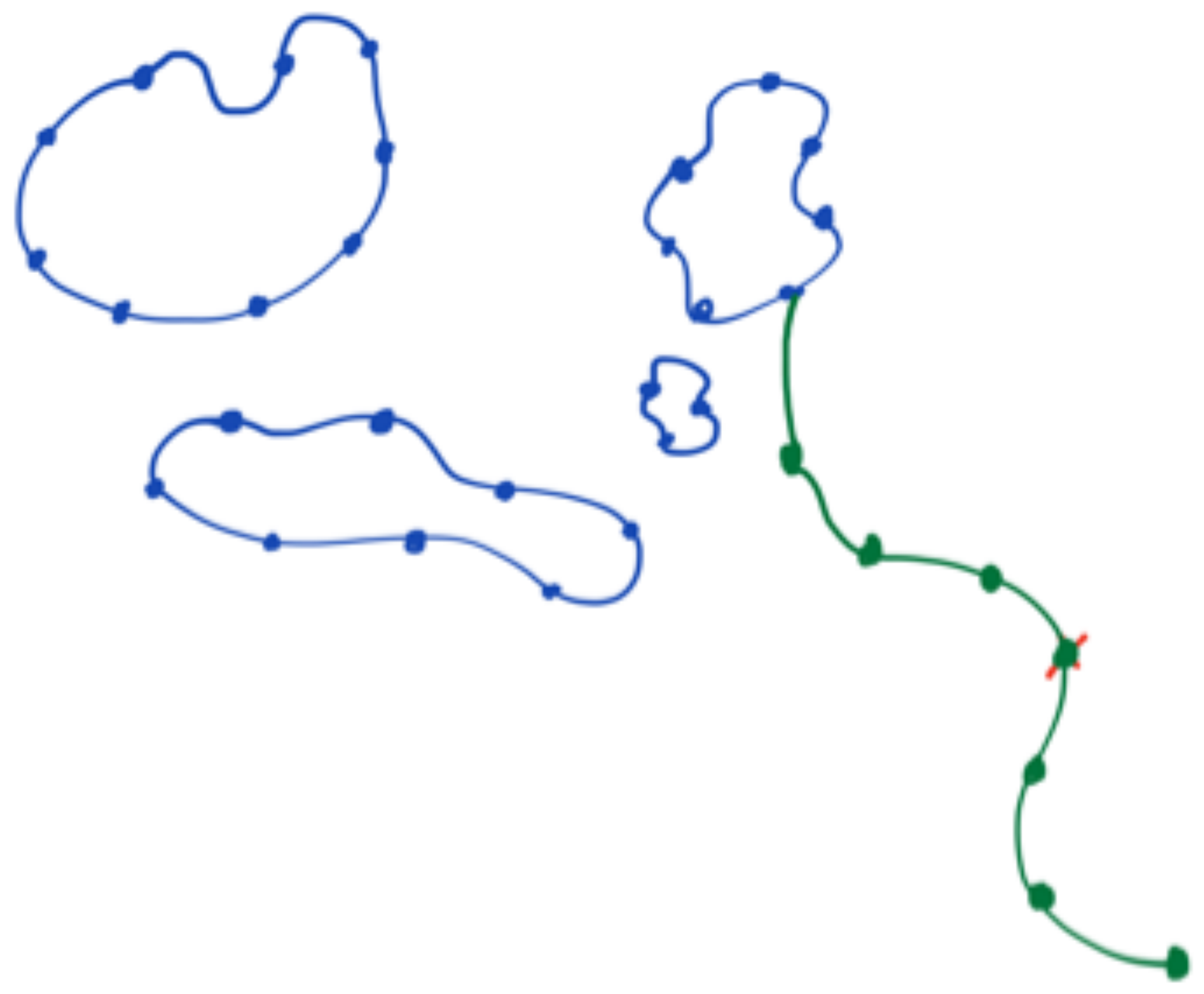
Ballistic path creation

- Some text

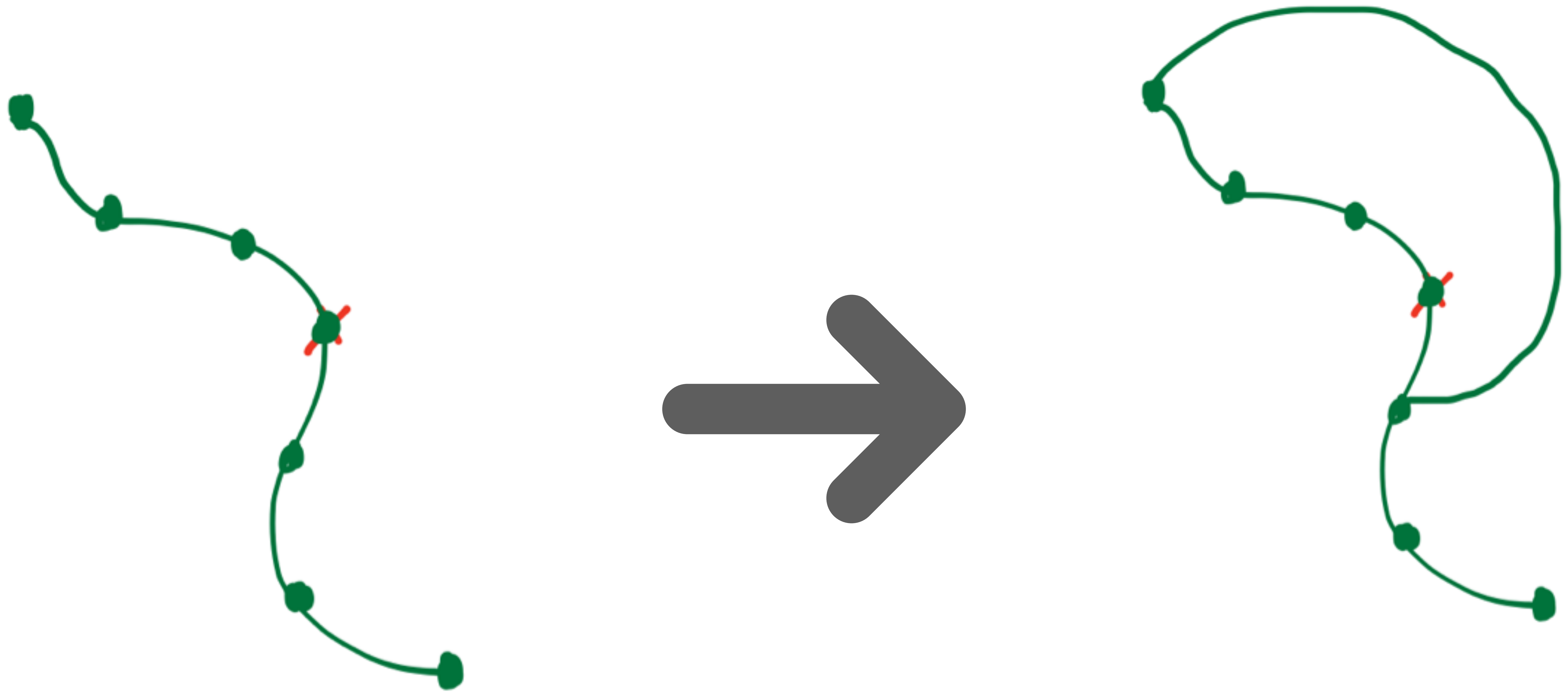


Ballistic path creation

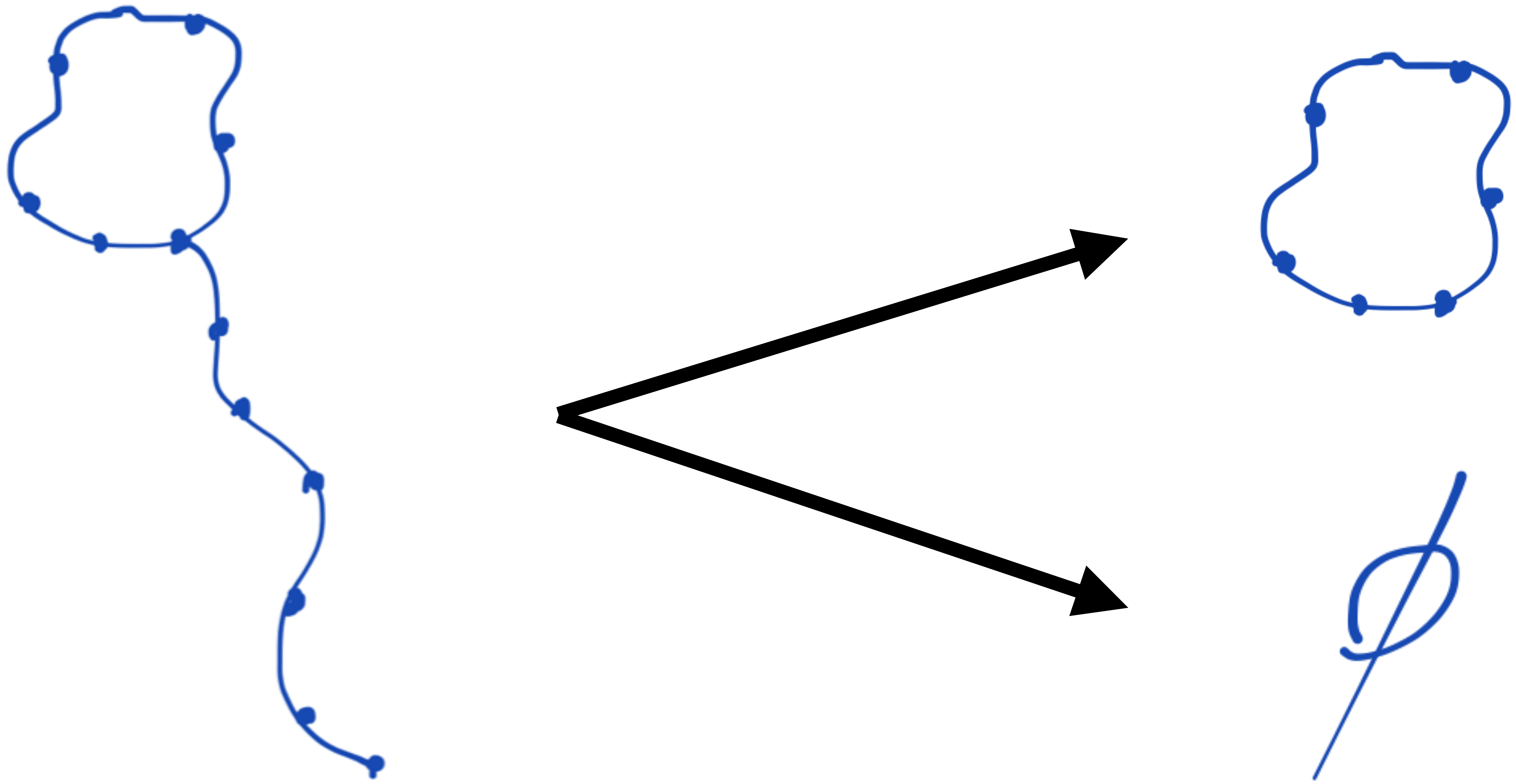
- Some text



Ballistic cycle creation



Ballistic path / cycle erasure



Facts

$$\sigma_N \sim \sqrt{\frac{9}{4} - \frac{24\Gamma(5/4)^4}{\pi^2}} n^{\frac{1}{2}} \approx 0.7801732\ n^{\frac{1}{2}}$$

$$\sigma_N^{(\mathscr{C}_0)} \sim \sqrt{\frac{3}{4} - \frac{6\pi^2}{\Gamma(1/4)^4}} n^{\frac{1}{2}} \approx 0.4072901\ n^{\frac{1}{2}}$$

$$\mu_N^{(\mathscr{C}_0)} \sim \frac{\Gamma(7/4)}{\sqrt{3}\Gamma(5/4)} n^{\frac{1}{2}} \approx 0.5854143\ n^{\frac{1}{2}}$$

$$\mu_N \sim \frac{2\sqrt{6}\Gamma(5/4)^2}{\pi} n^{\frac{1}{2}} \approx 1.2811439\ n^{\frac{1}{2}}$$

$$\beta = \frac{1}{n} \Rightarrow x \sim \frac{1}{n}$$

$$\chi \approx n^{\frac{1}{2}}$$

Figure pool