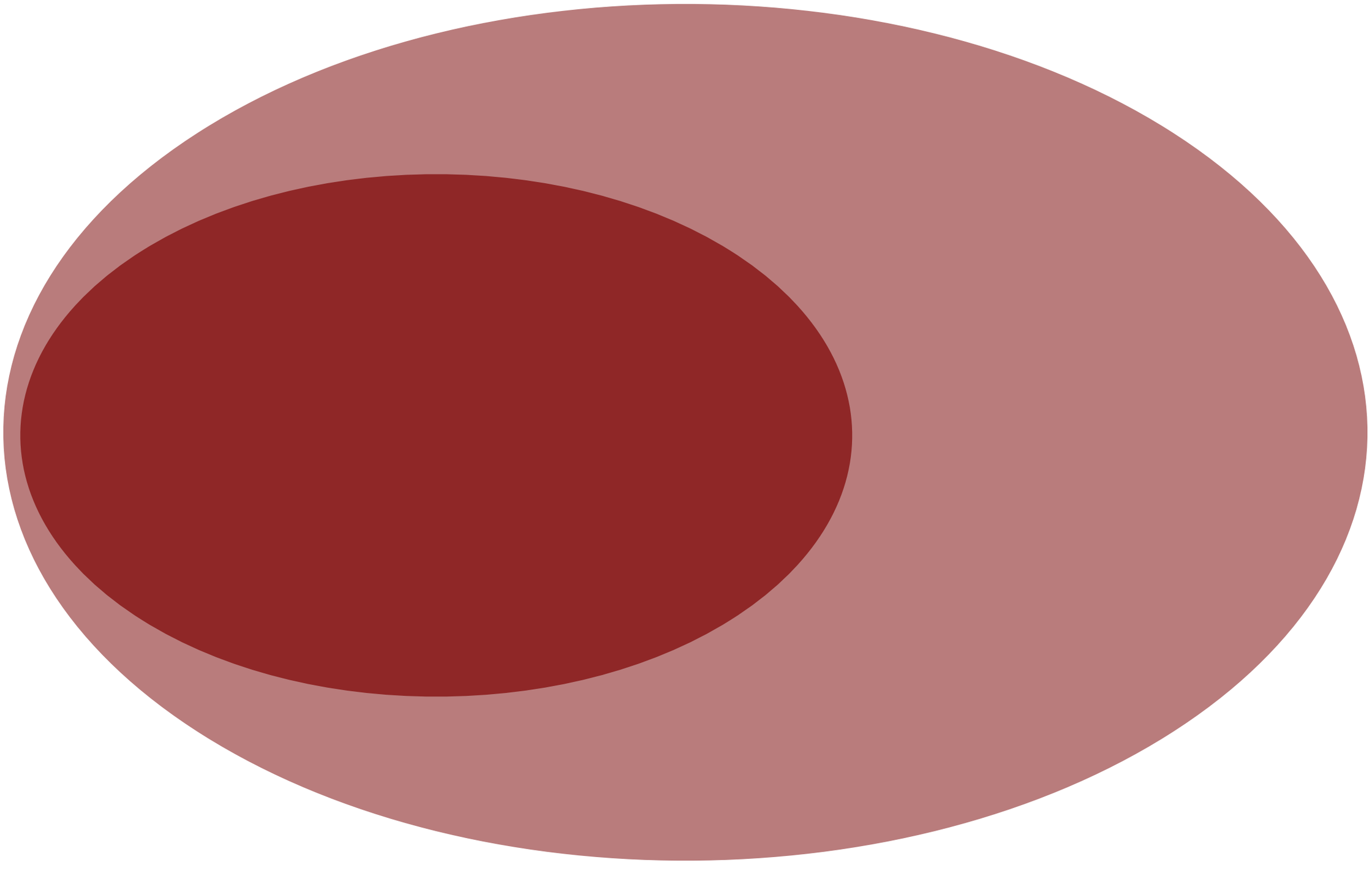
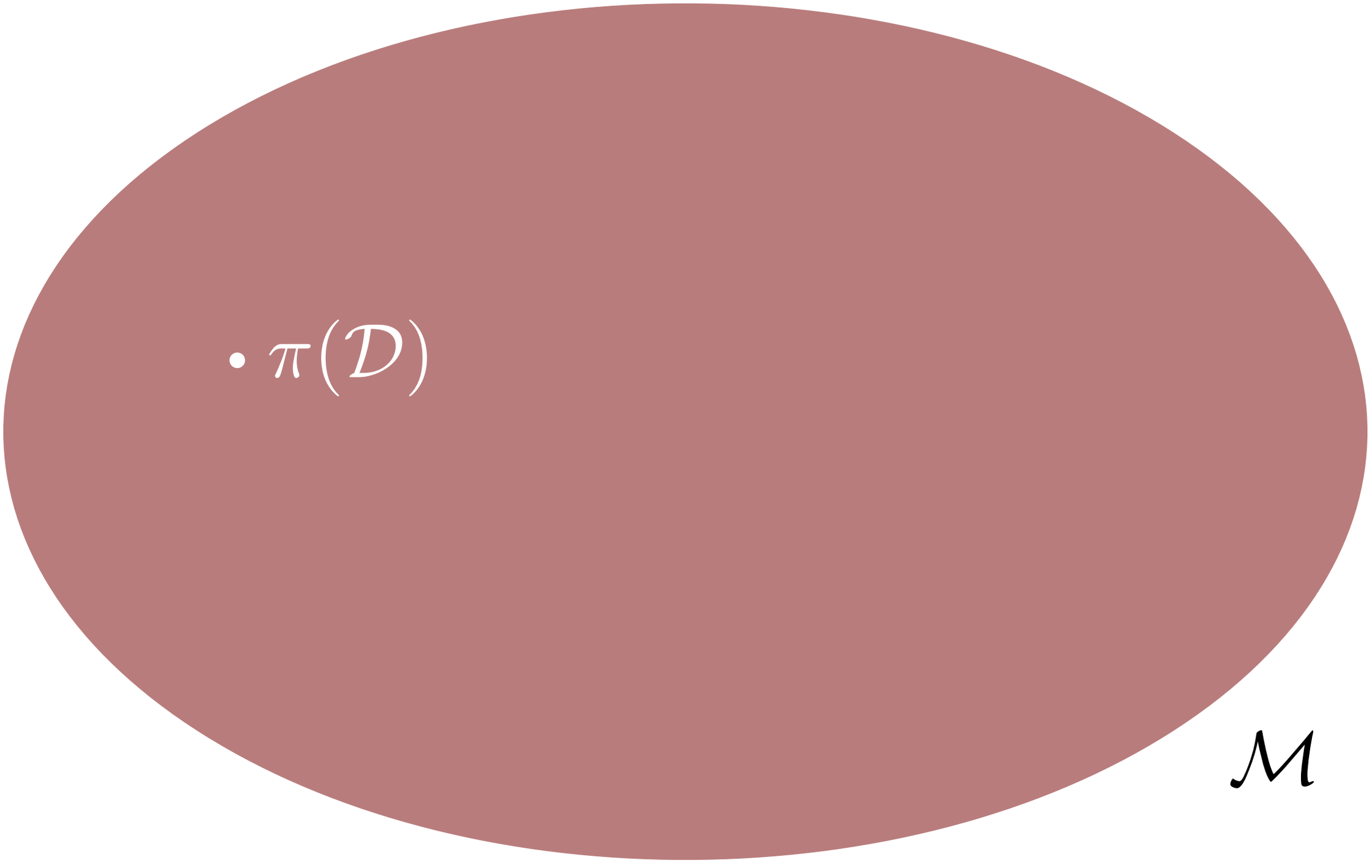
Let's review…



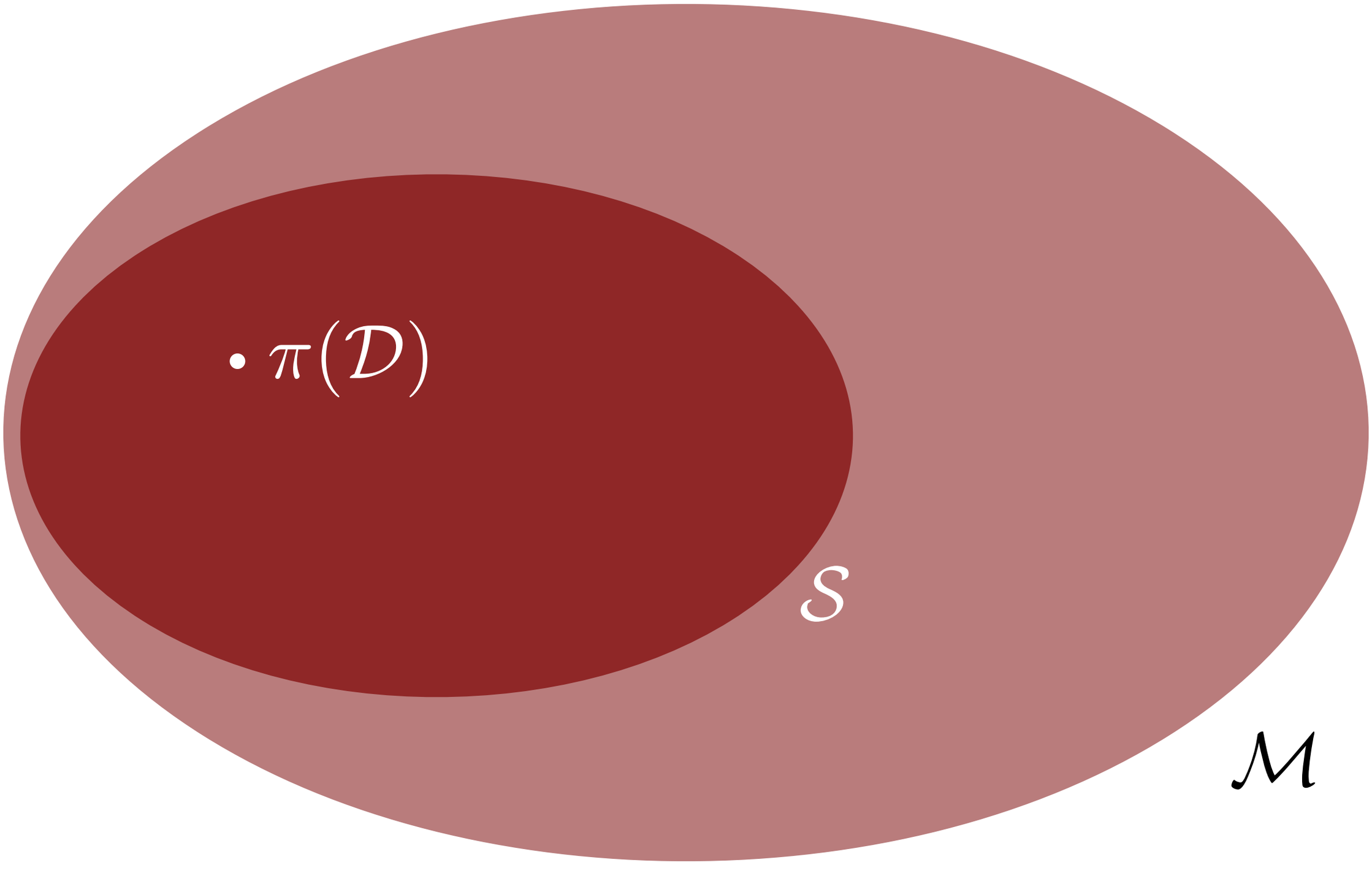
We want to infer some data generating process.

$$\pi(\mathcal{D})$$

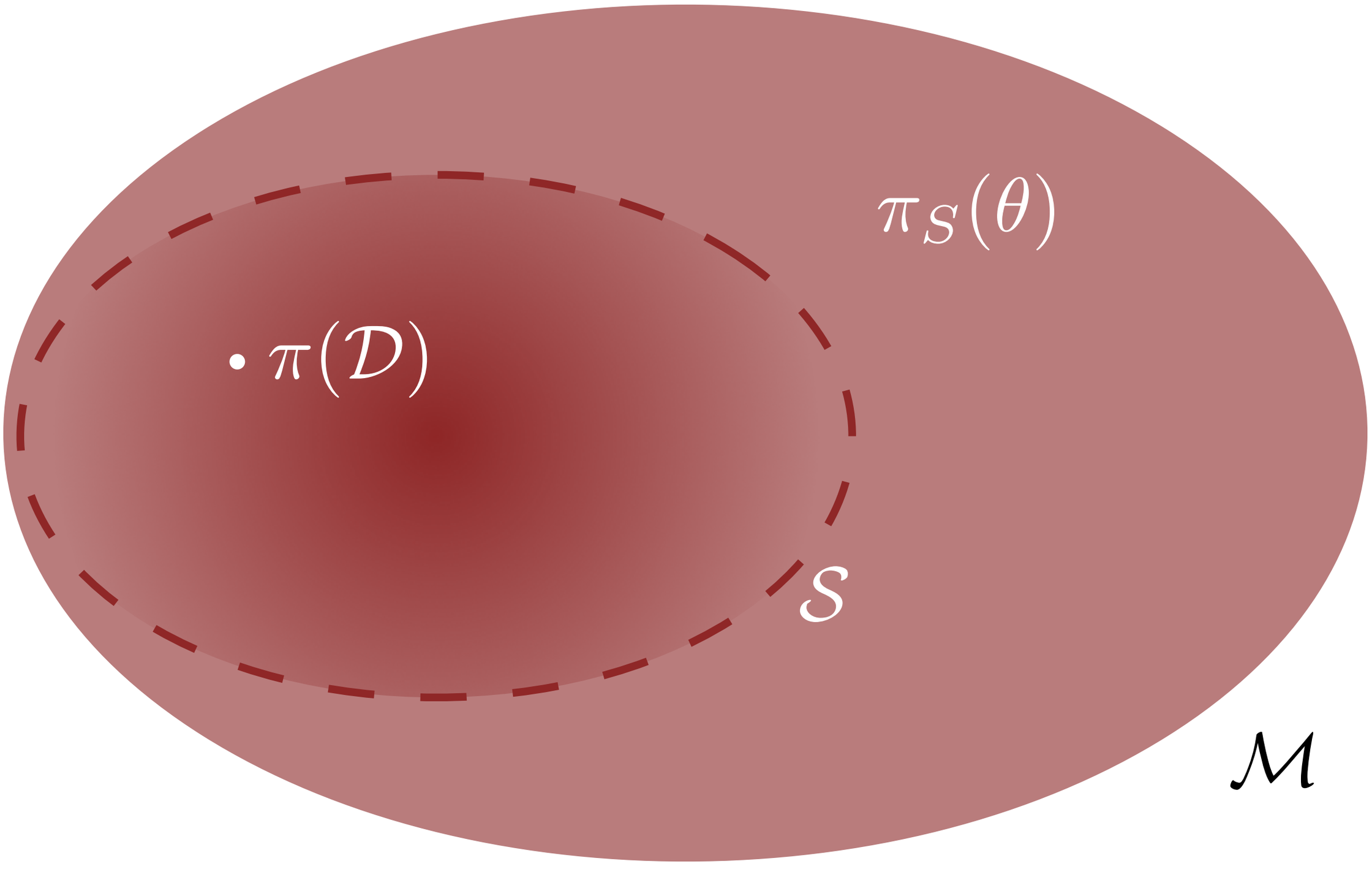
Inference is an identification of this process from the space of all possible data generating processes.



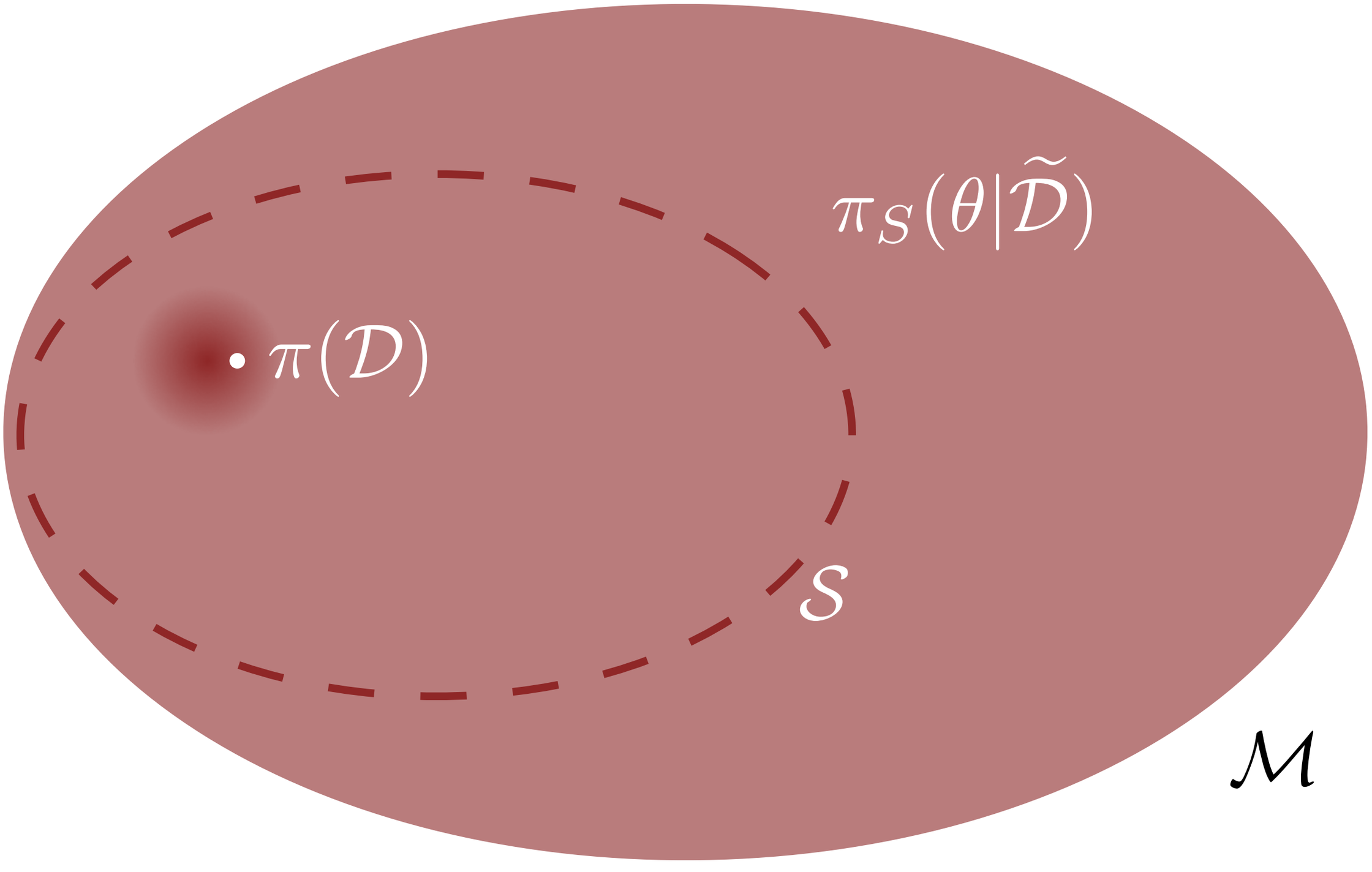
The space of all data generating processes is too big so we restrict our analysis to a *small world*.



In Bayesian inference we use probability distributions to quantify our information about the *small world*.



In Bayesian inference we use probability distributions to quantify our information about the *small world*.

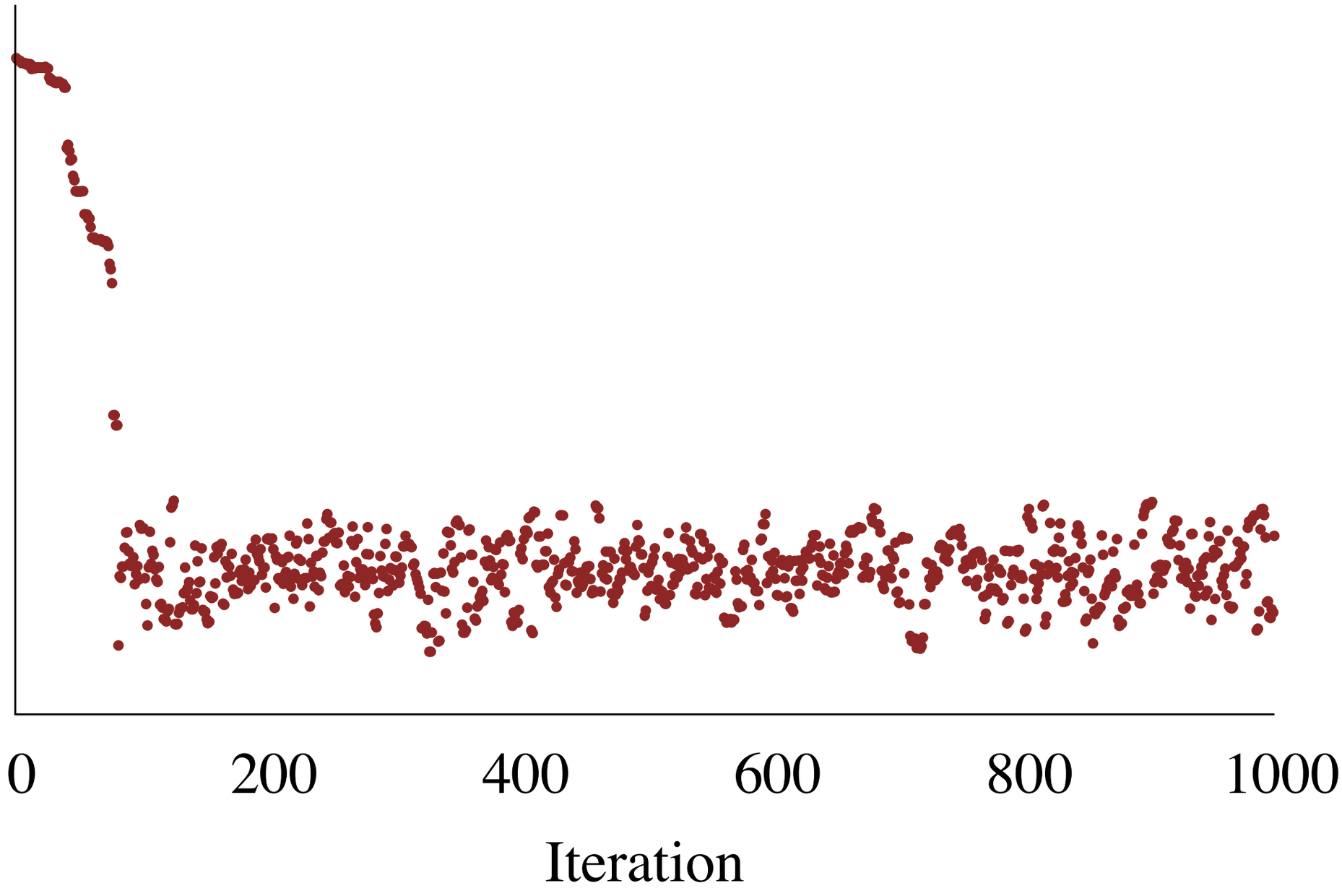


All Bayesian computations reduce to expectations with respect to the the posterior distribution.

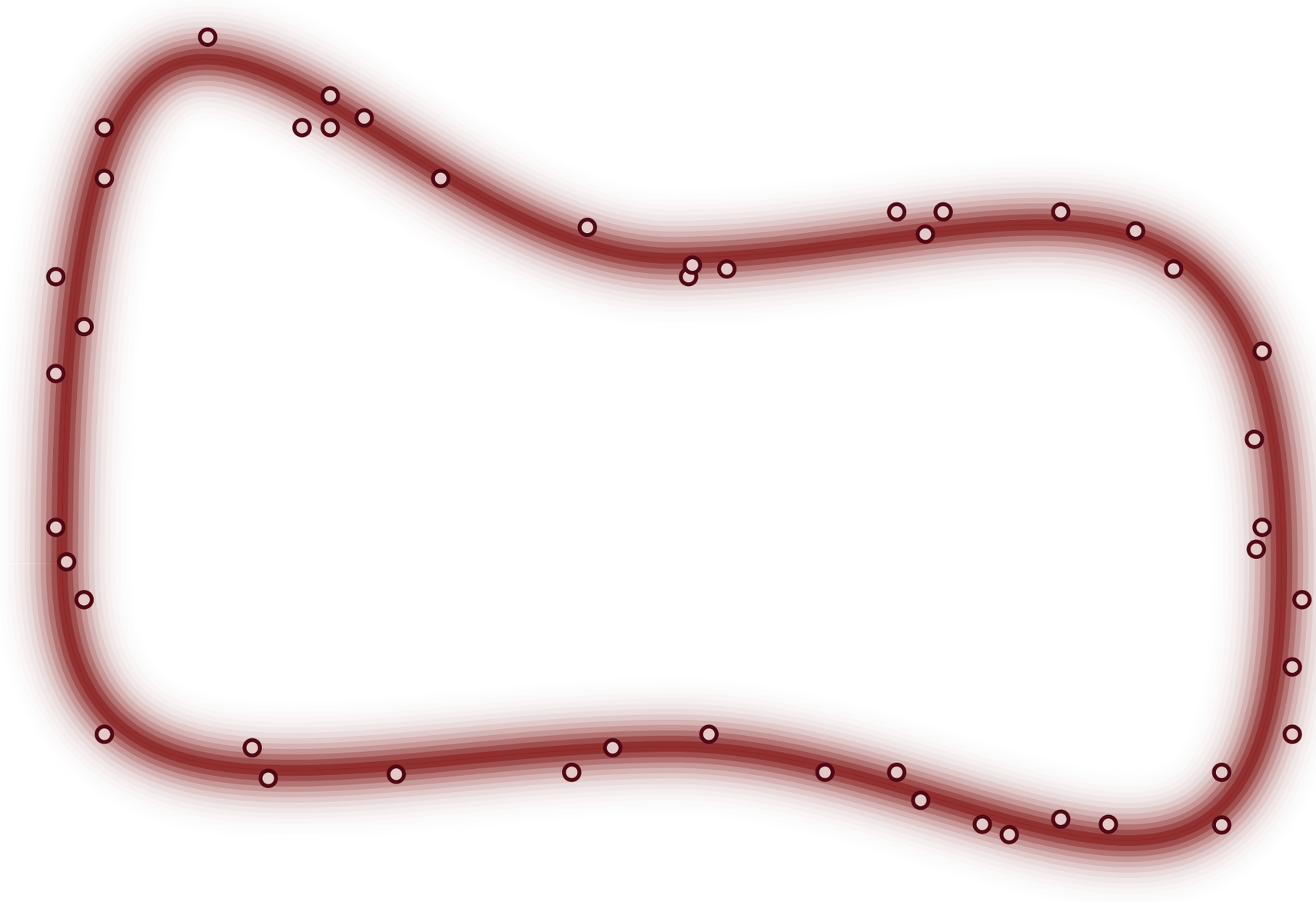
$$\mathbb{E}[f] = \int \mathrm{d}\theta \, \pi_S(\theta|\widetilde{\mathcal{D}}) f(\theta)$$

To accurately estimate these expectations we need to be able to find and explore the typical set.

Markov Chain Monte Carlo



Ultimately, there are many of methods for quantifying the typical set and, consequently, expectations.



Markov chains provide a generic and practical scheme for finding and then exploring this typical set.

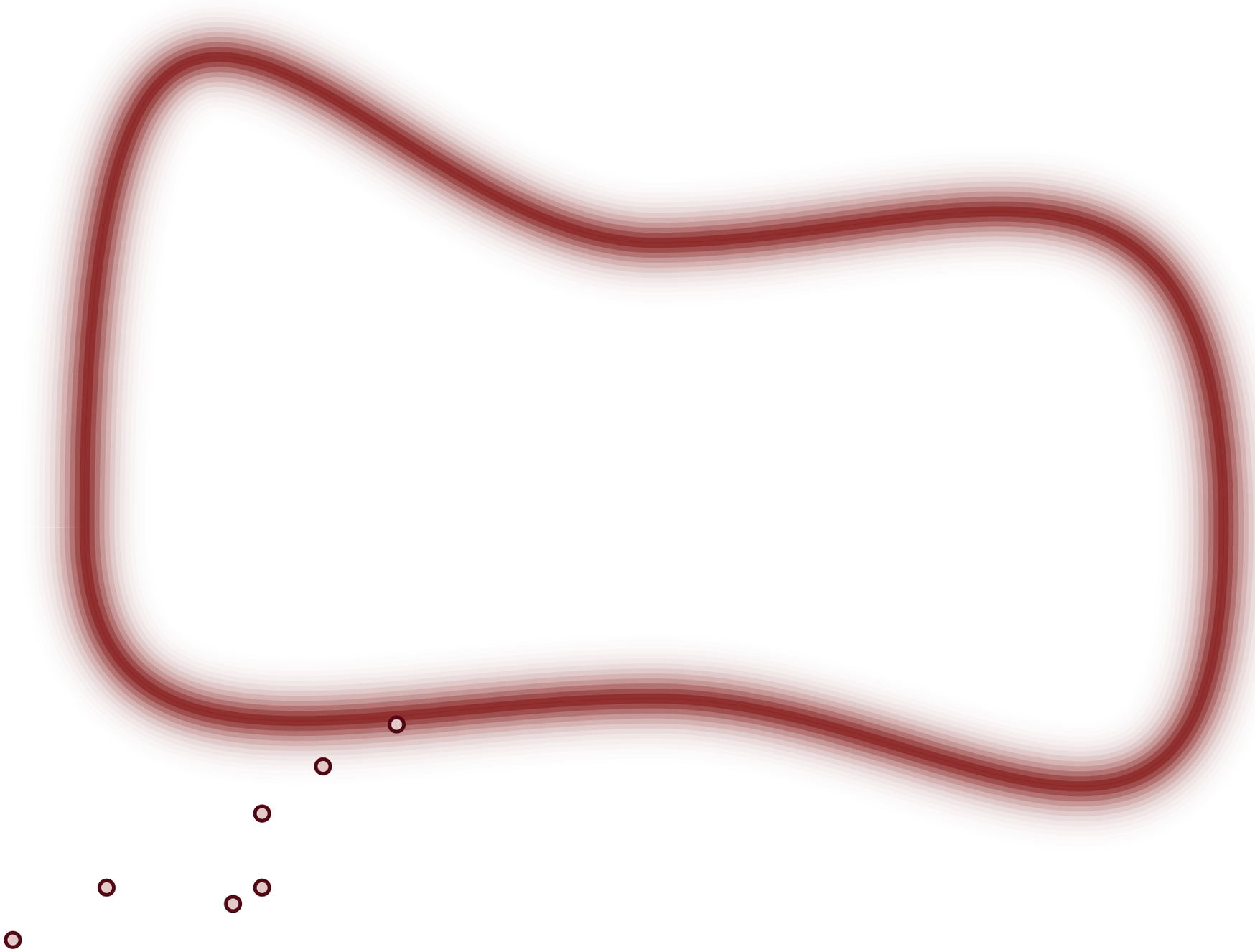
$$T(\theta, \theta')$$

Markov chains provide a generic and practical scheme for finding and then exploring this typical set.

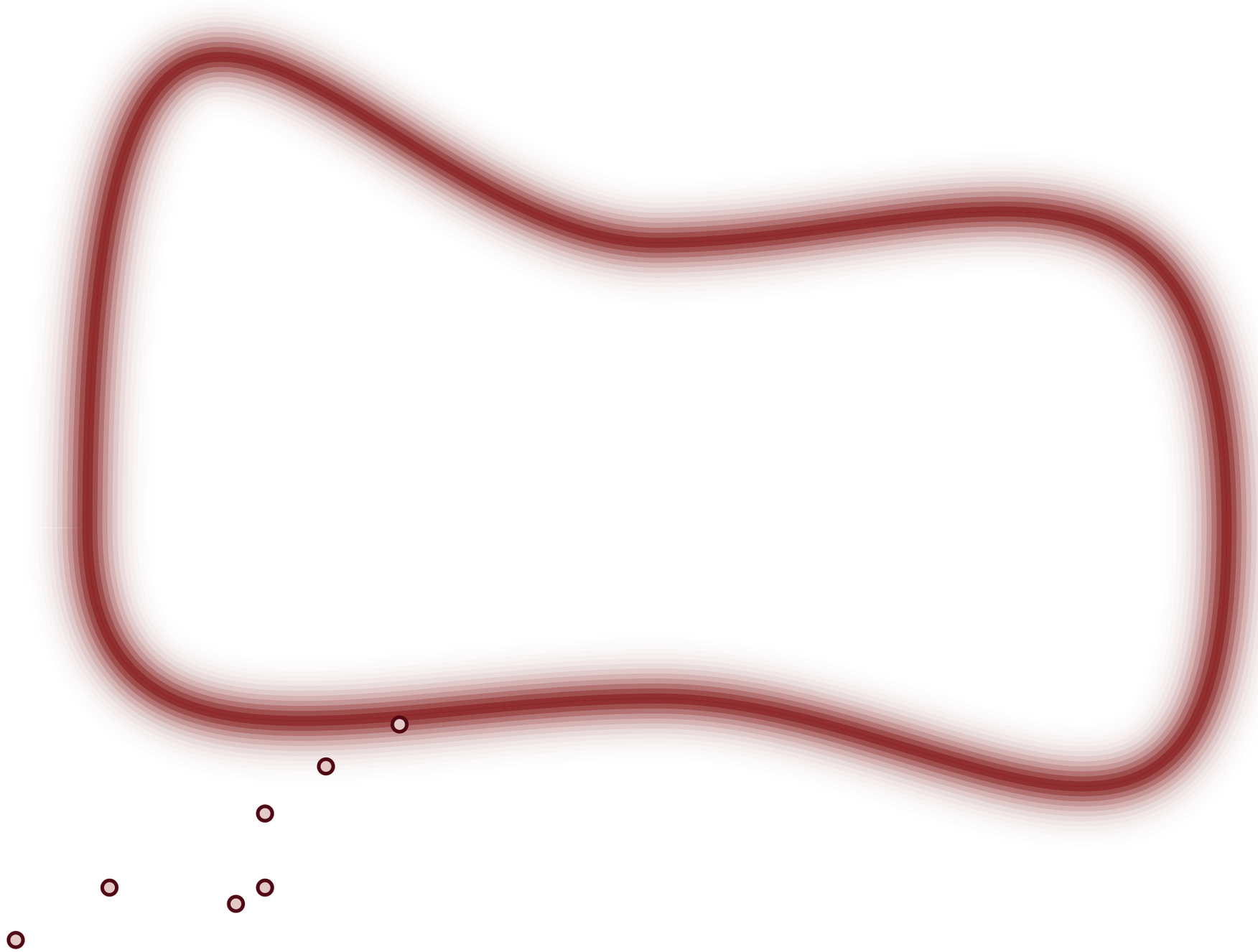
$$\pi_S(\theta|\widetilde{\mathcal{D}}) = \int \mathrm{d}\theta'\, T(\theta, \theta')\pi_S(\theta'|\widetilde{\mathcal{D}})$$

Intuitively, a Markov chain first finds the typical set, or *converges*.

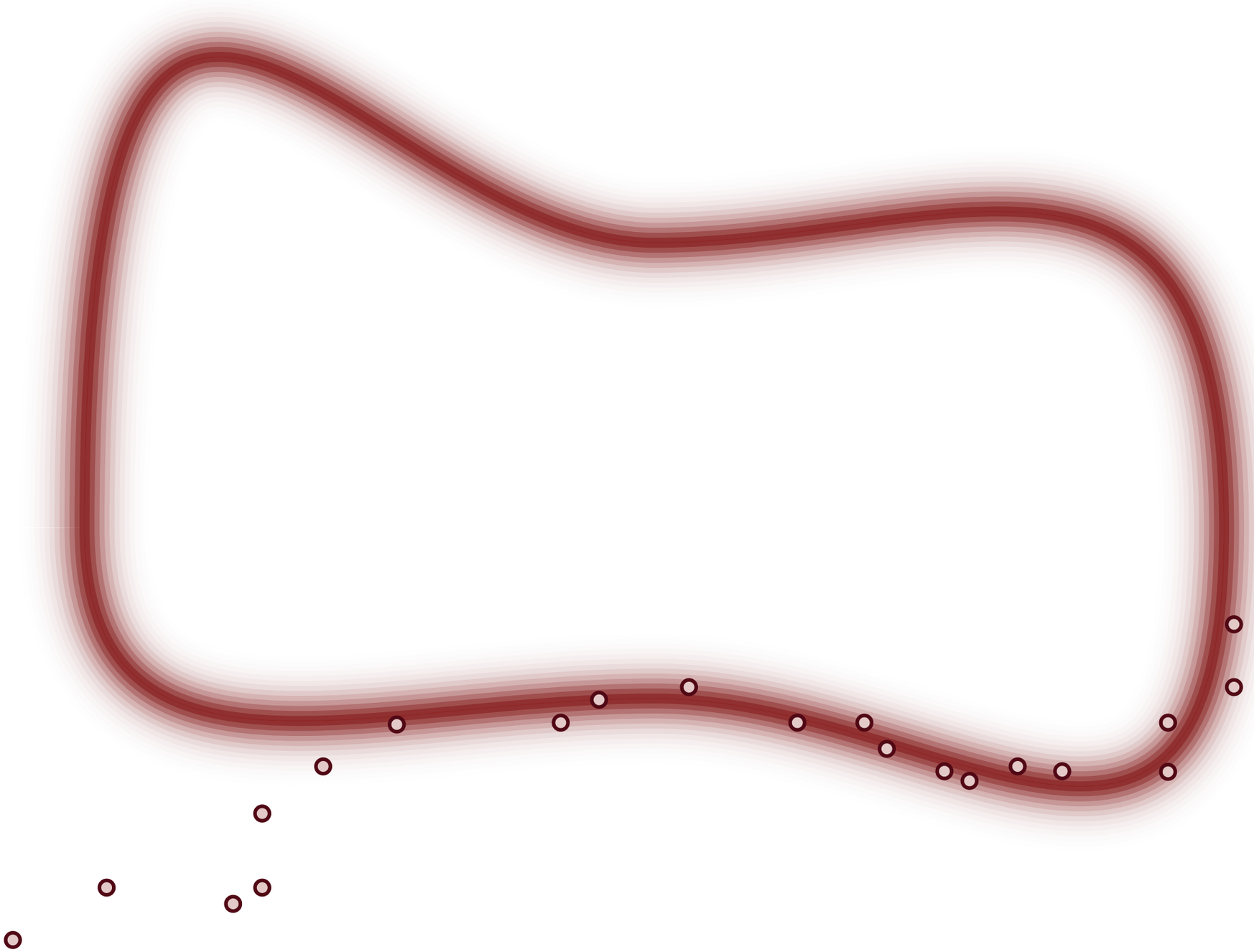
Intuitively, a Markov chain first finds the typical set, or *converges*.



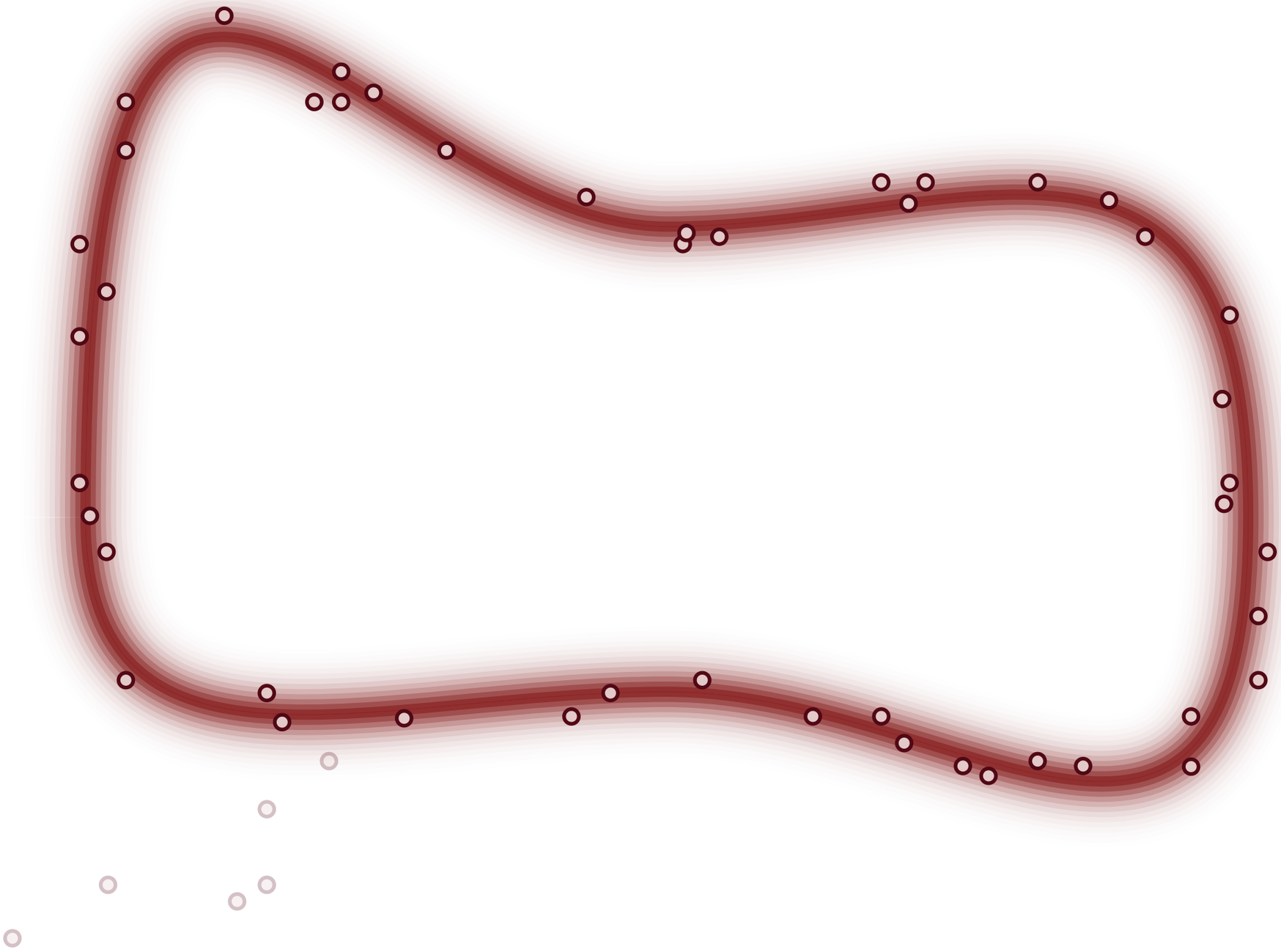
Once it has found the typical set the Markov chain then explores it, or *mixes*.



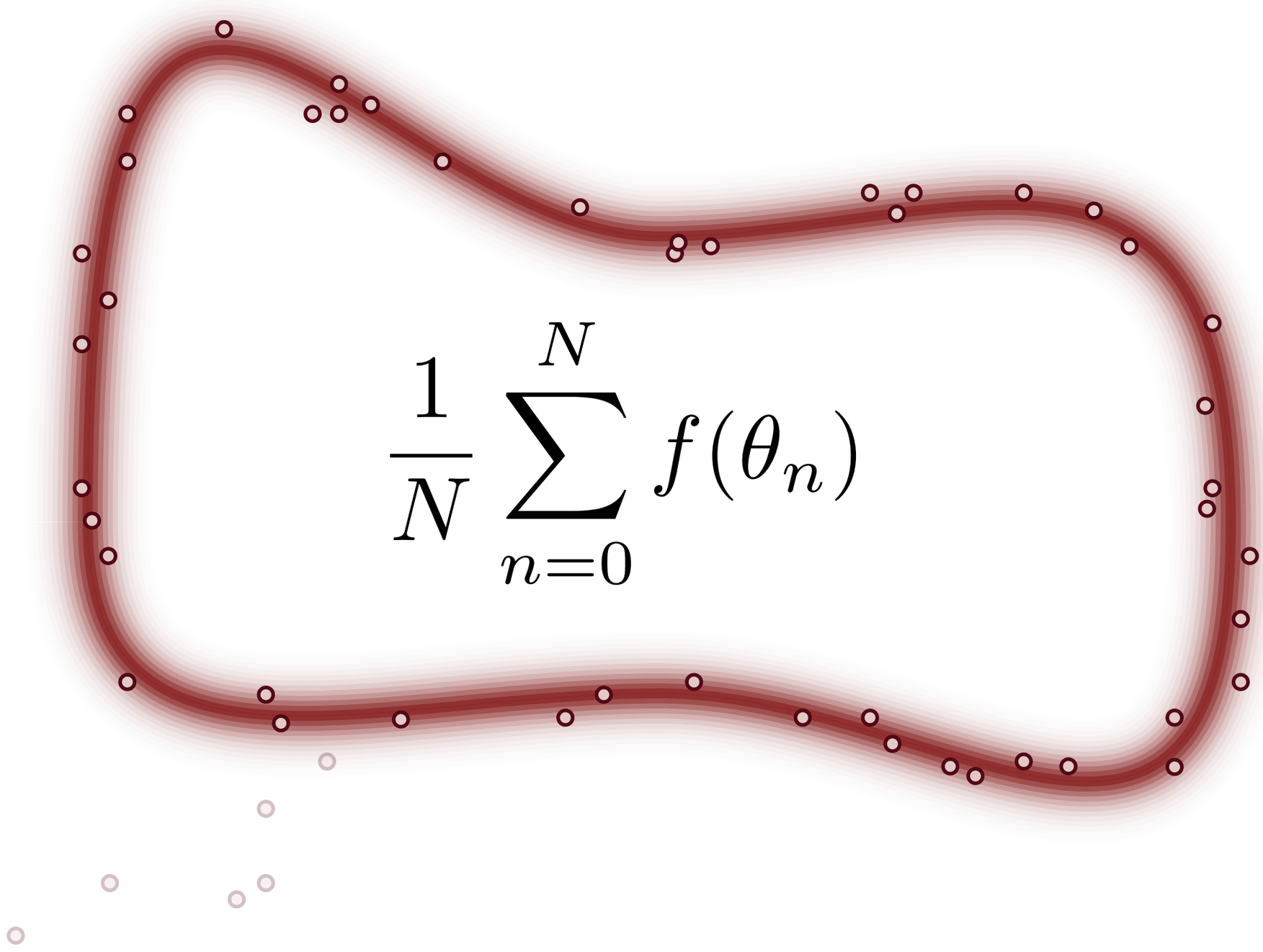
Once it has found the typical set the Markov chain then explores it, or *mixes*.



If run long enough the Markov chain defines consistent *Markov Chain Monte Carlo estimators*.



If run long enough the Markov chain defines consistent *Markov Chain Monte Carlo estimators.*

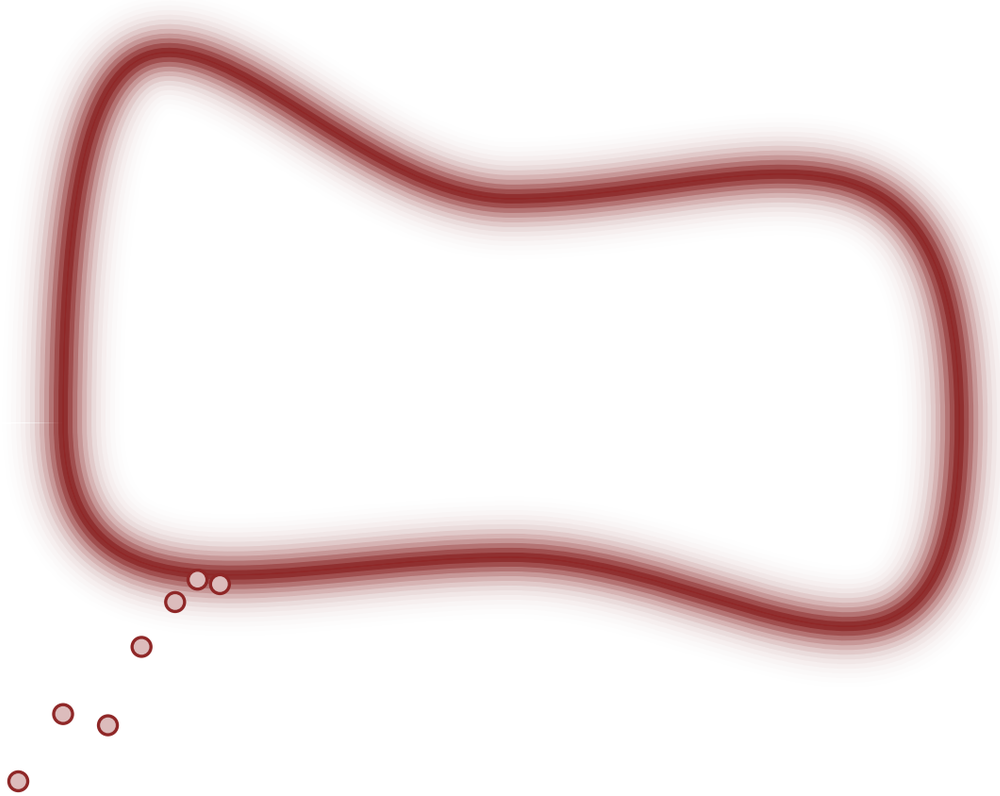


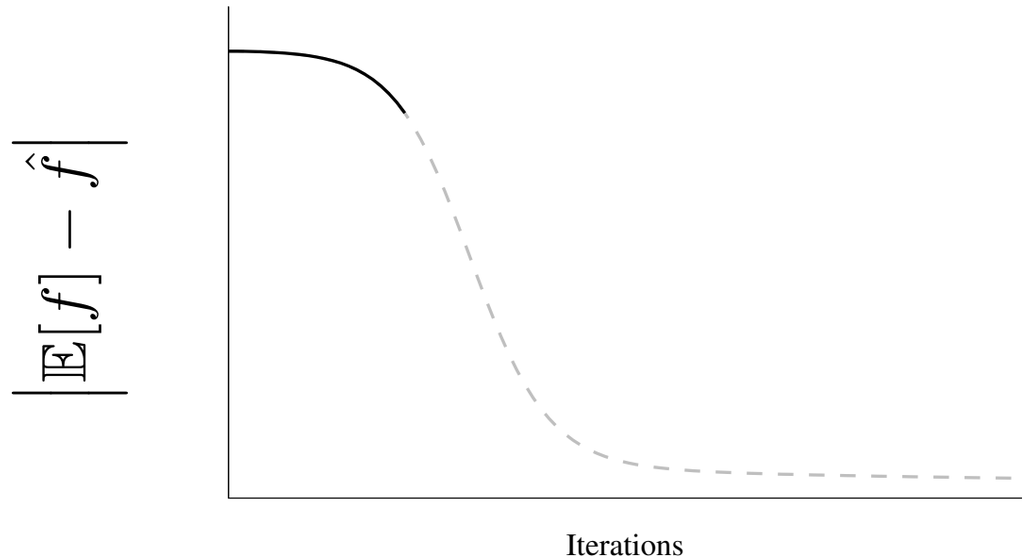
$$\frac{1}{N}\sum_{n=0}^{N} f(\theta_n)$$

If run long enough the Markov chain defines consistent *Markov Chain Monte Carlo estimators.*

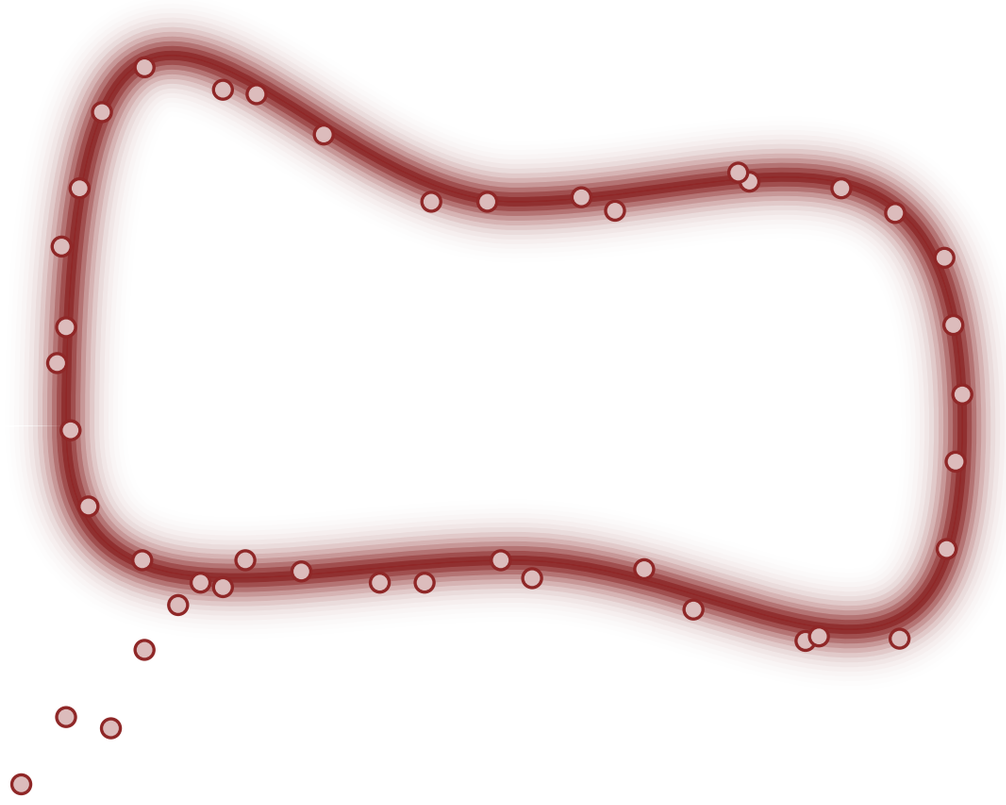
$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N} f(\theta_n) \to \mathbb{E}[f]$$

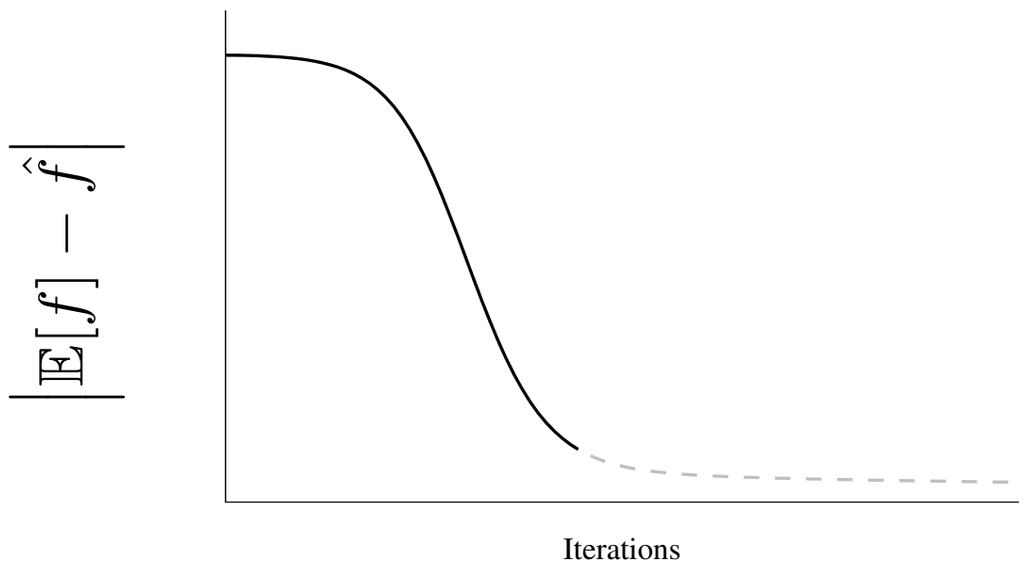
As the Markov chain mixes, MCMC estimators *usually* converge rapidly to the true expectations.



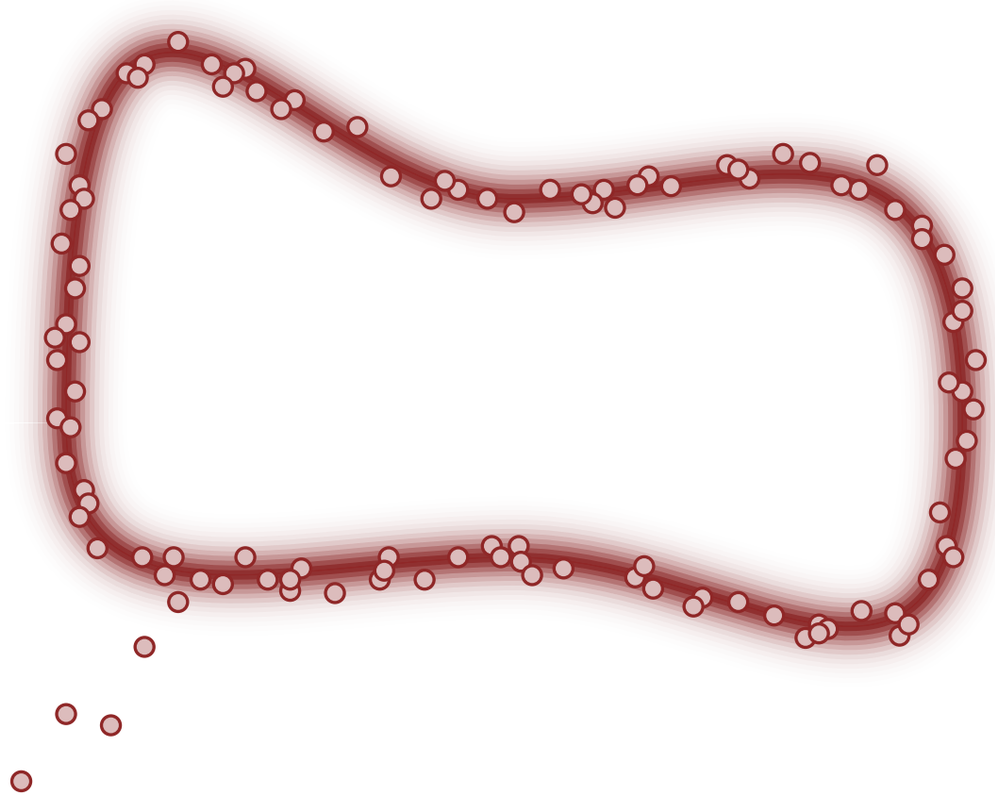


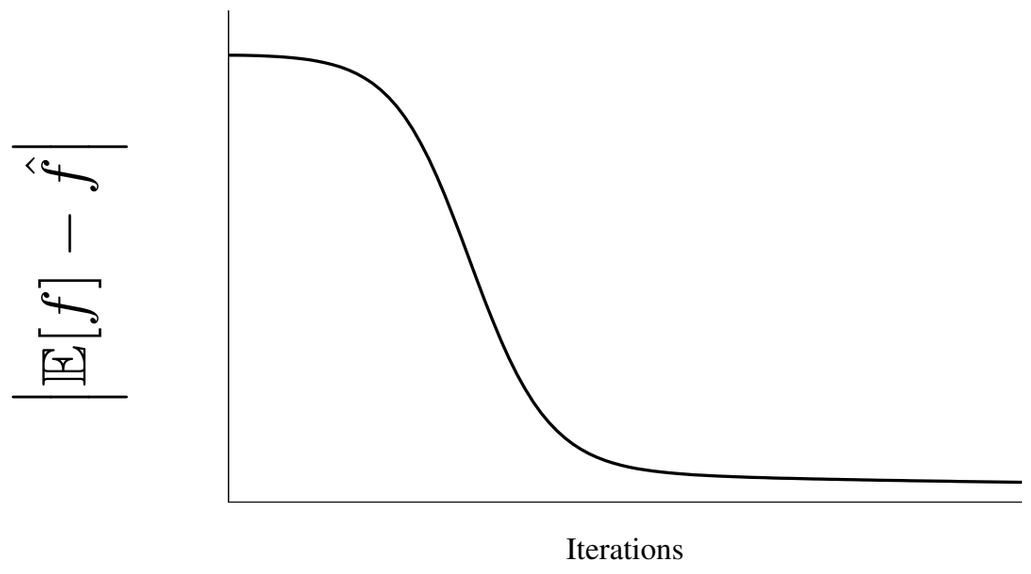
As the Markov chain mixes, MCMC estimators *usually* converge rapidly to the true expectations.



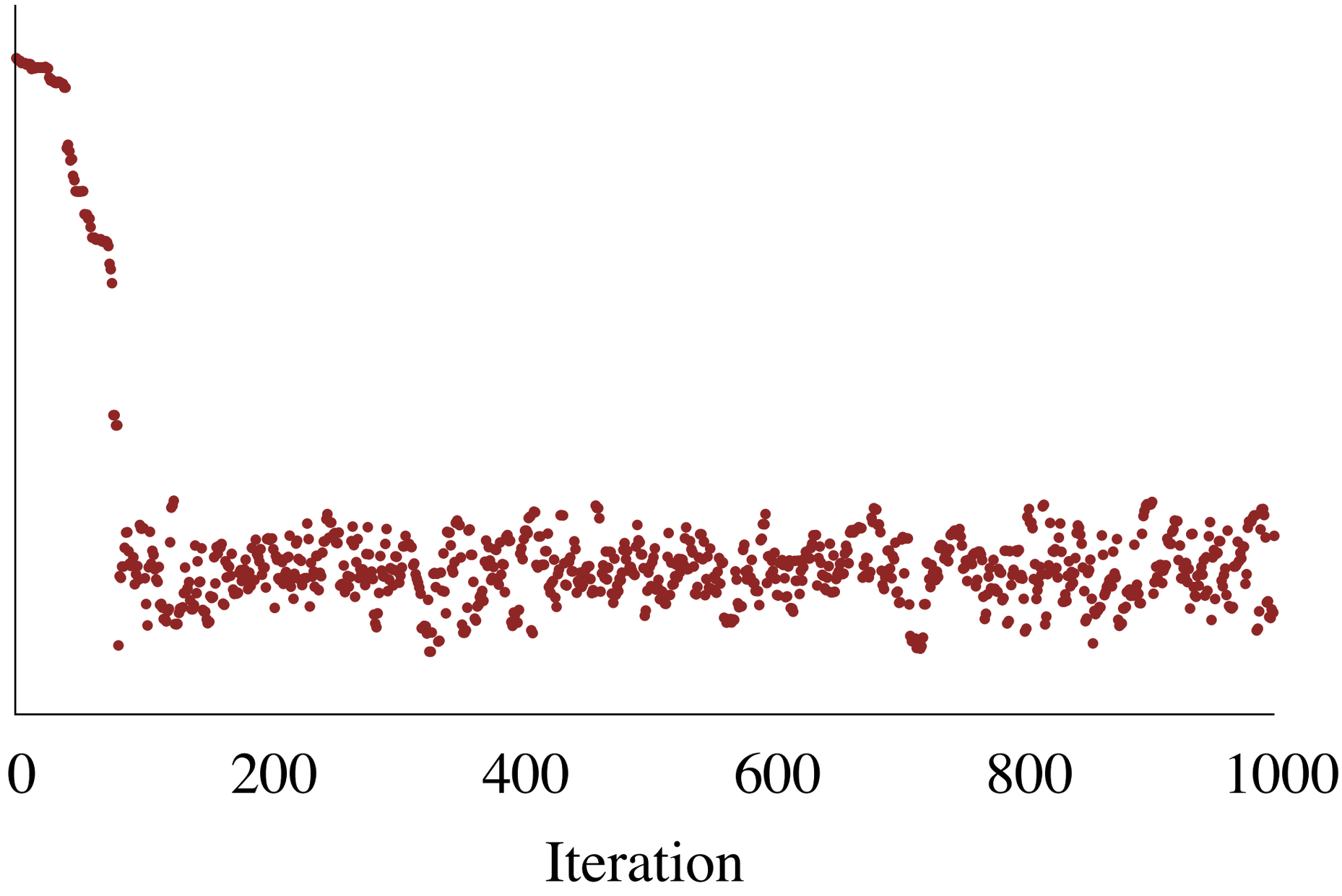


As the Markov chain mixes, MCMC estimators *usually* converge rapidly to the true expectations.

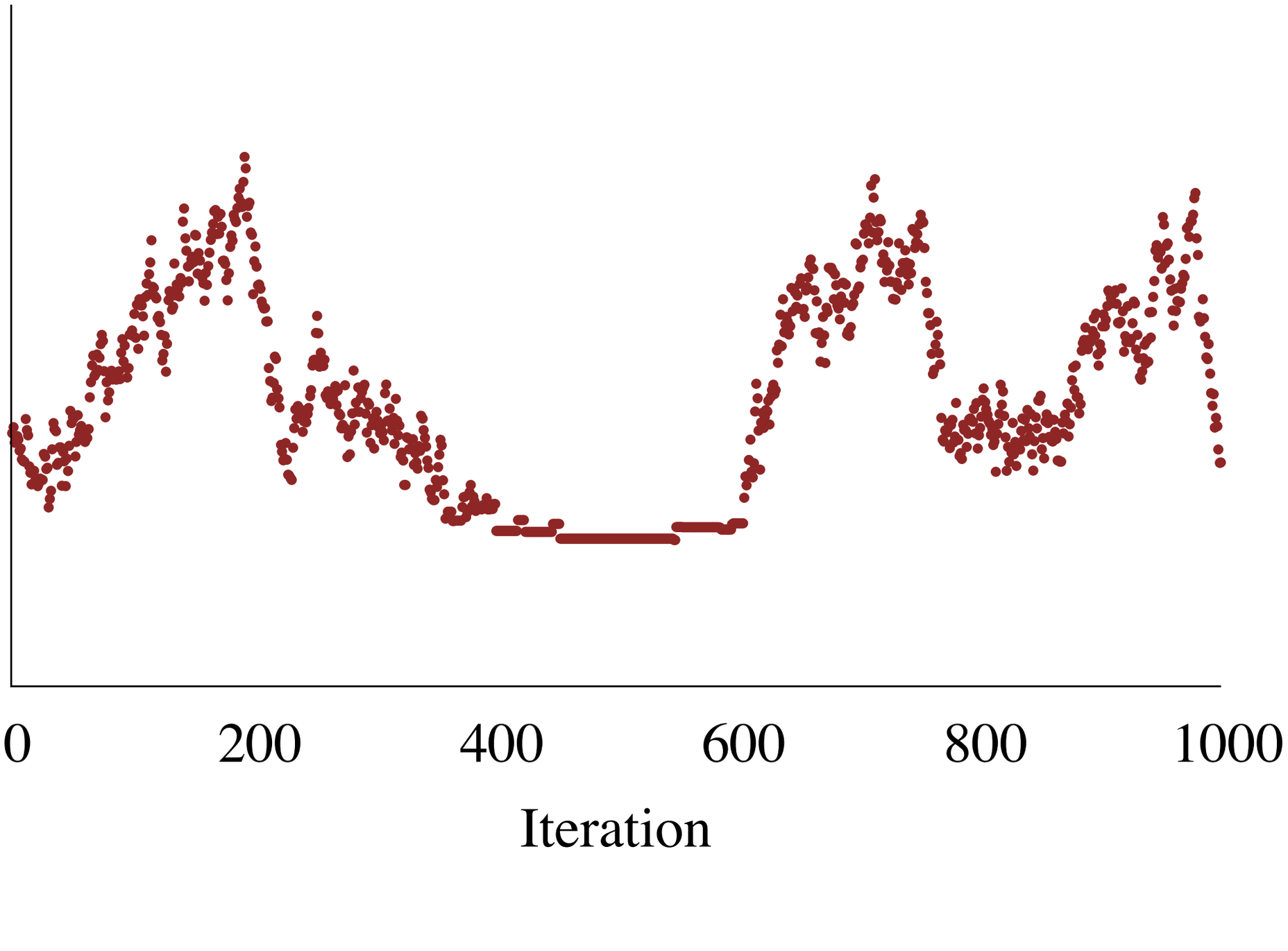




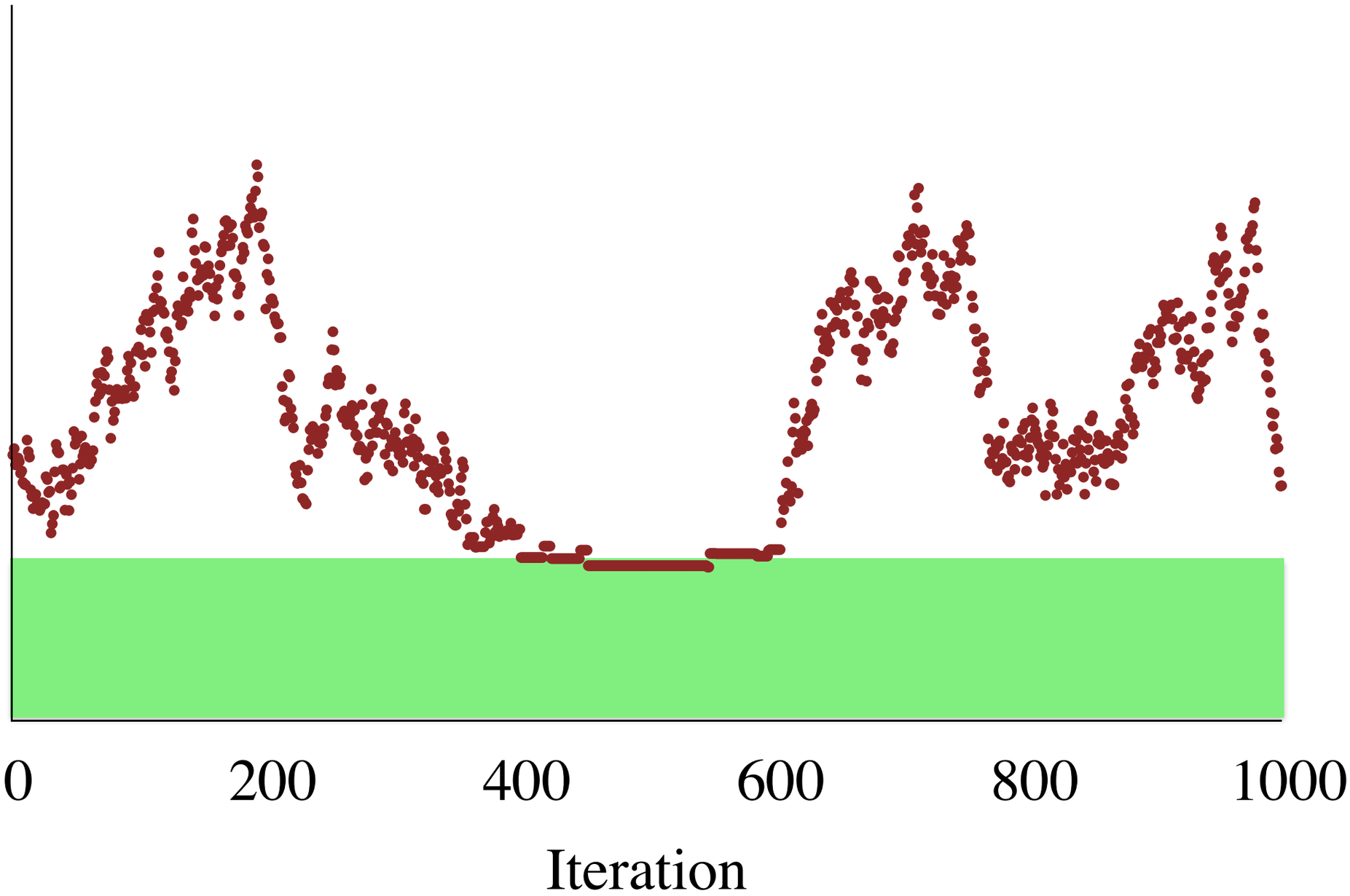
But convergence of MCMC estimators need not always be so well-behaved!



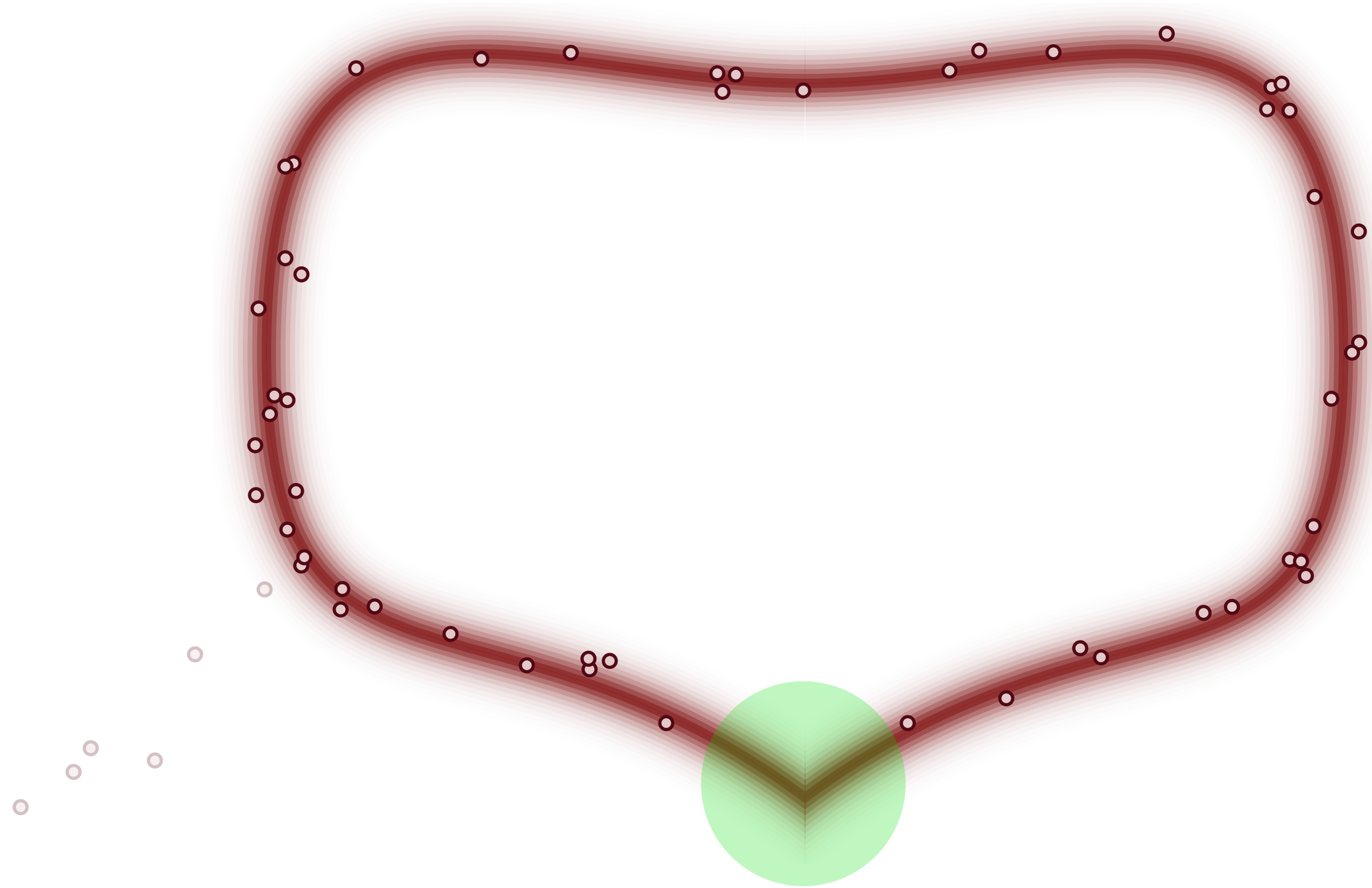
But convergence of MCMC estimators need not always be so well-behaved!



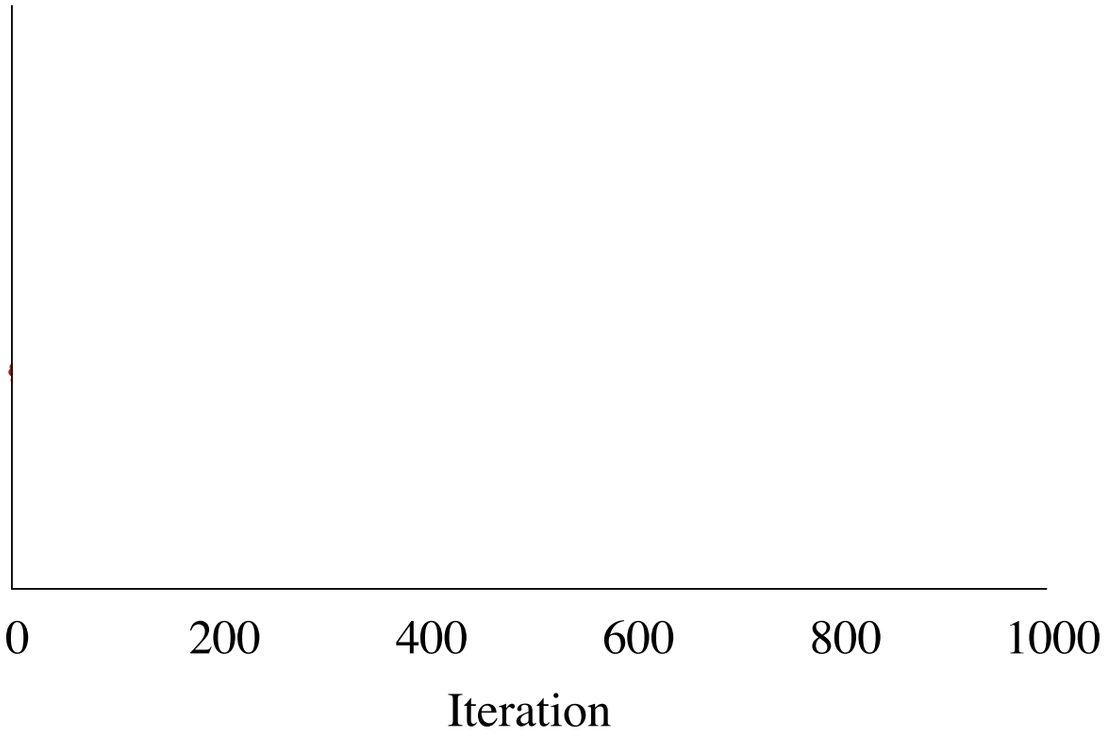
But convergence of MCMC estimators need not always be so well-behaved!

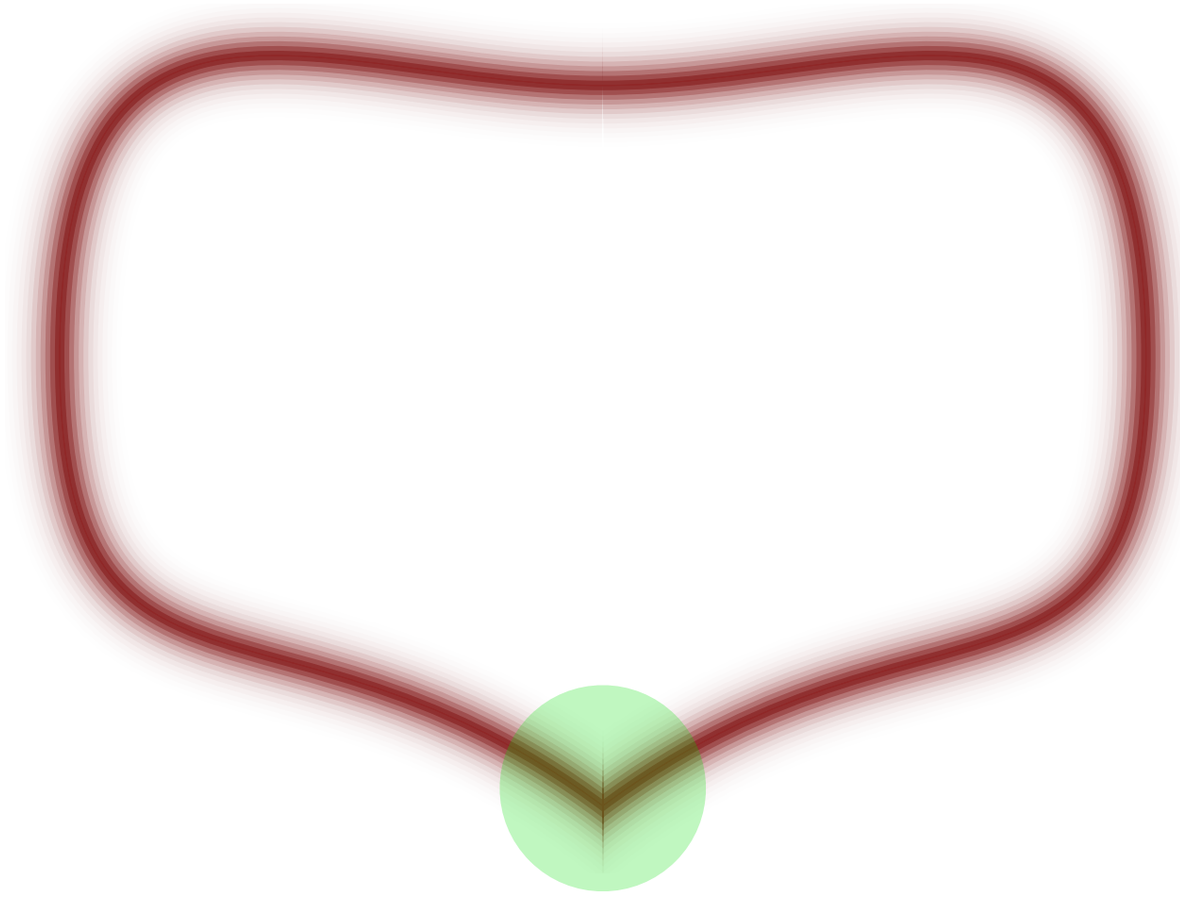


Markov chains are not always able
to explore the entire typical set!

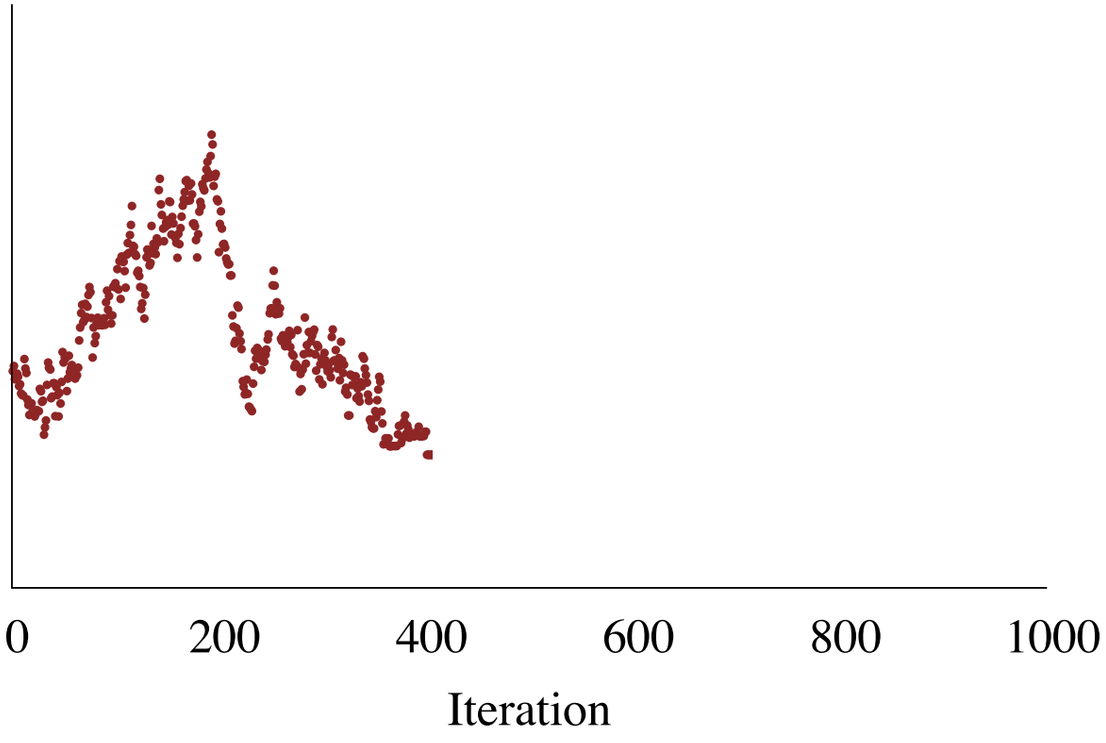


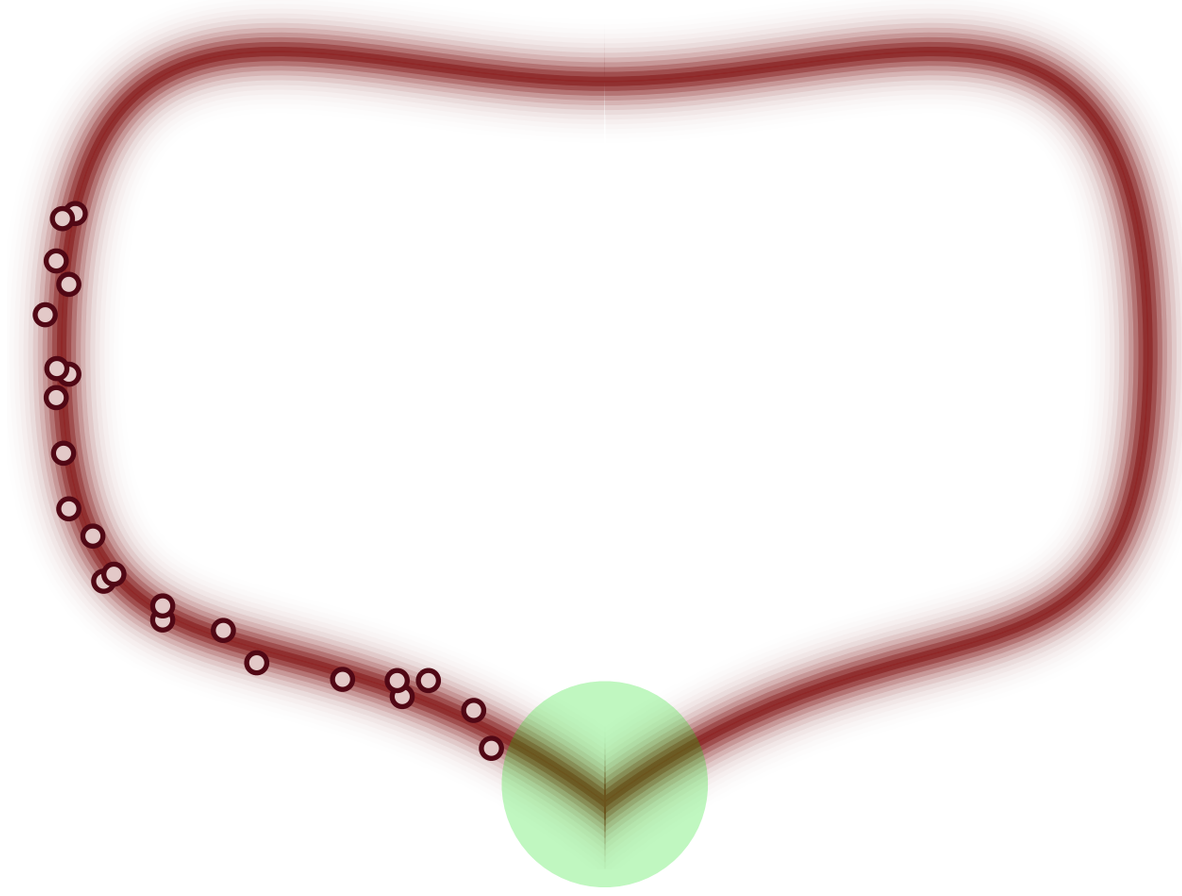
Markov chains are not always able
to explore the entire typical set!



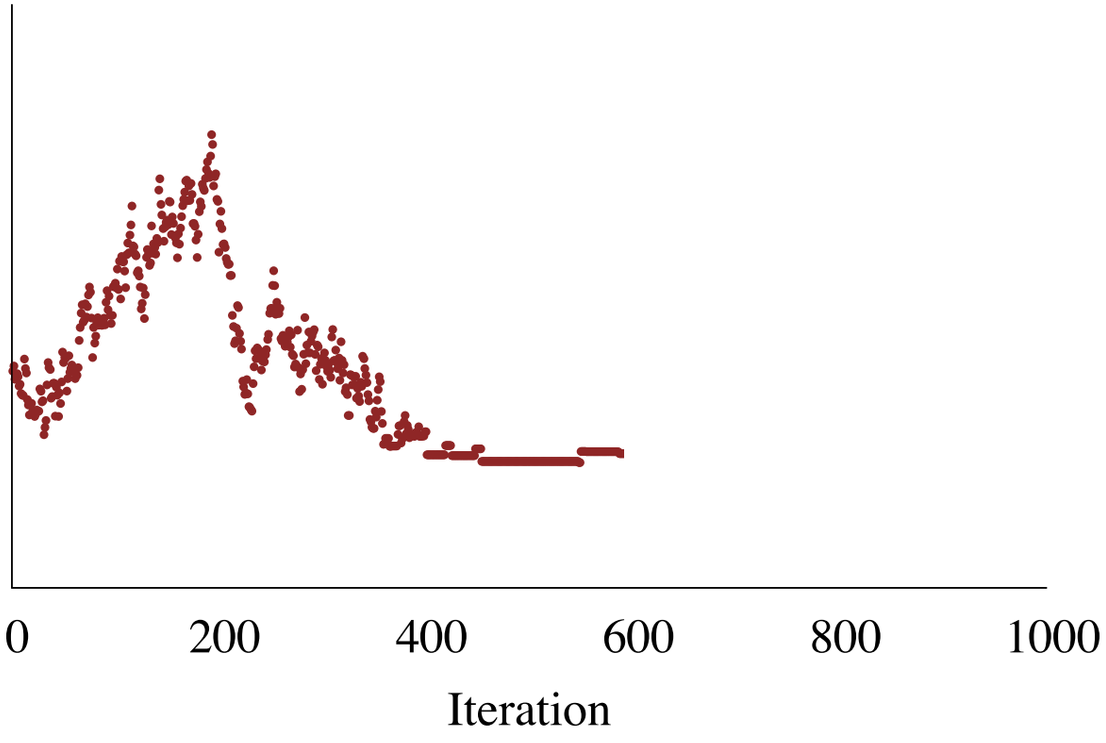


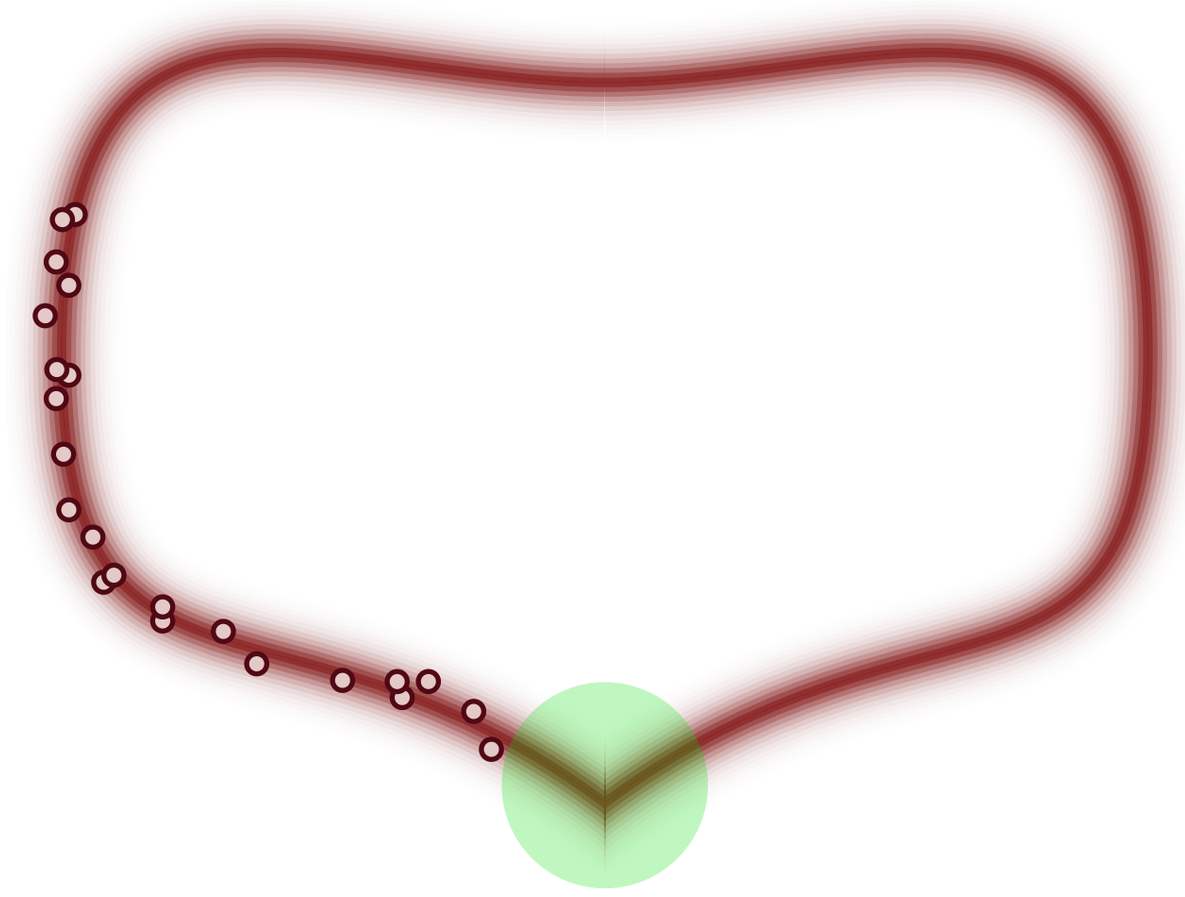
Markov chains are not always able
to explore the entire typical set!



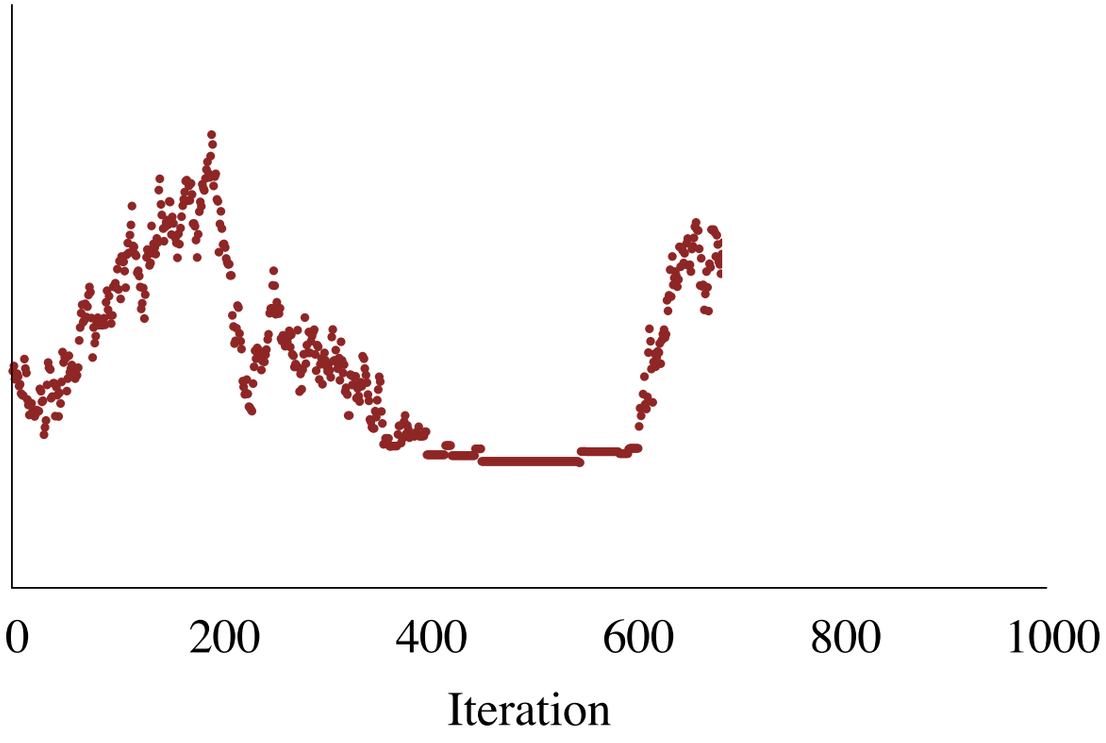


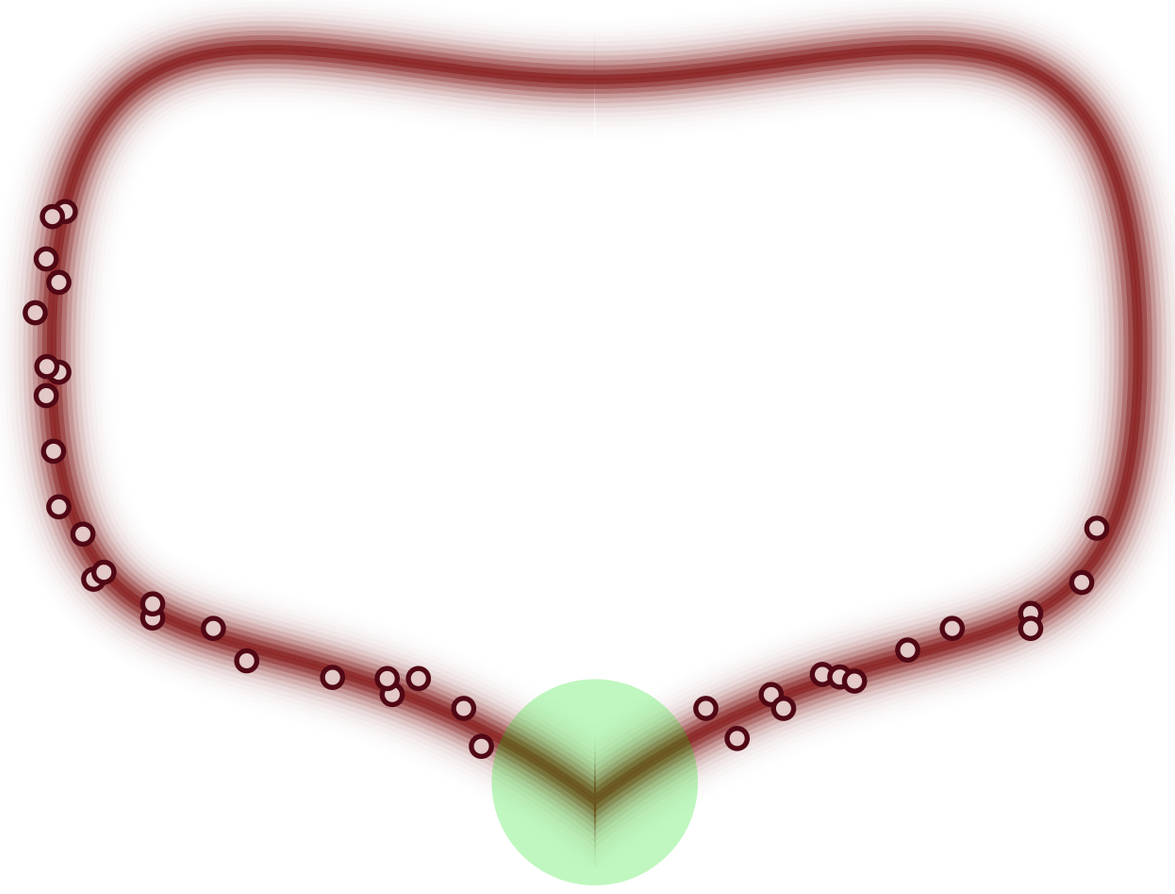
Markov chains are not always able
to explore the entire typical set!





Markov chains are not always able
to explore the entire typical set!





The problem is that Markov Chain Monte Carlo estimators are only *asymptotically* unbiased.

$$\lim_{N \to \infty} \frac{1}{N} \sum_{n=0}^{N} f(\theta_n) \to \mathbb{E}[f]$$

Accurate estimation with only a finite number of transitions requires strong *ergodicity* conditions.

$$\hat{f} = \frac{1}{N} \sum_{n=1}^{N} f(\theta_i)$$

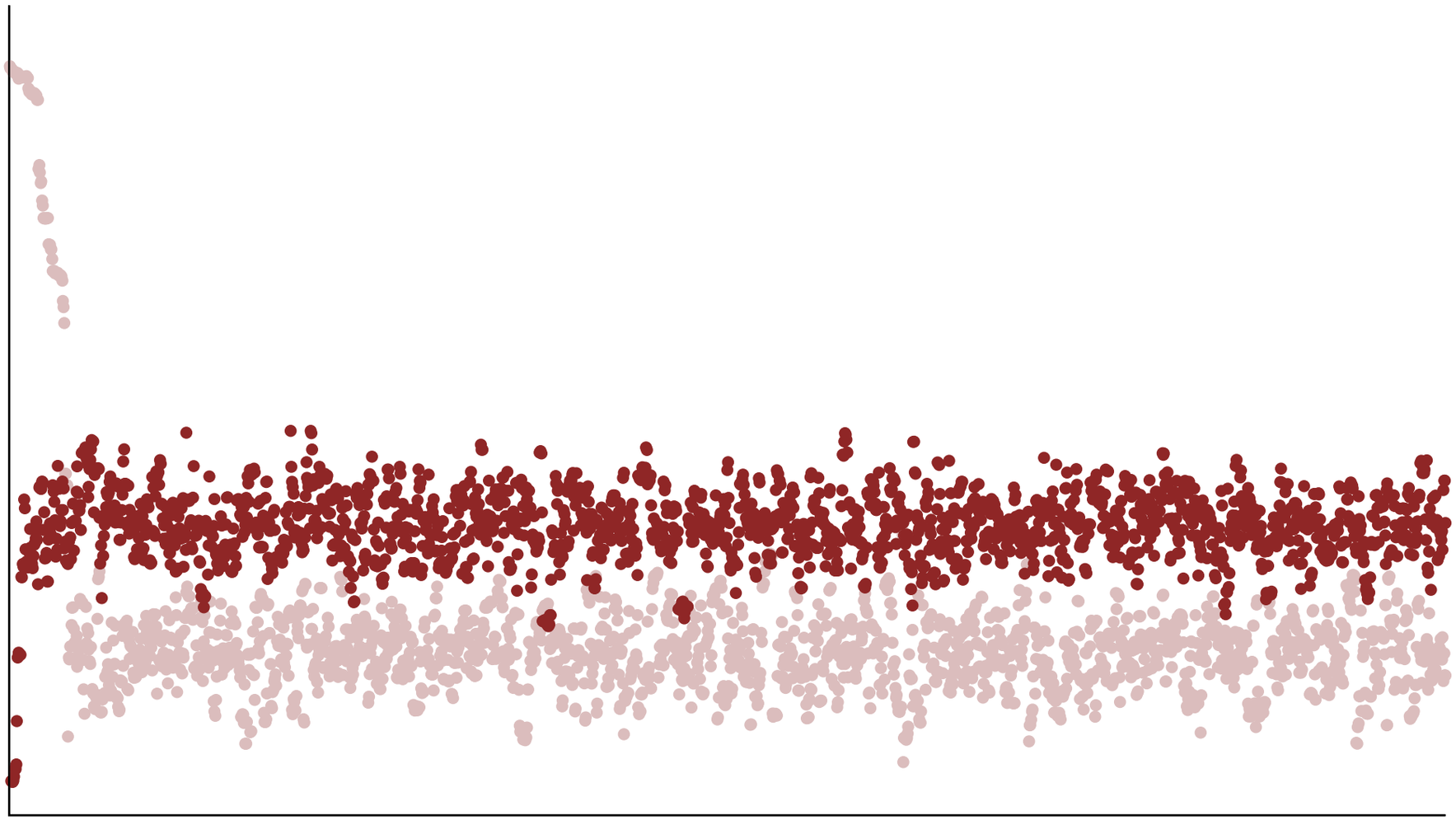
Accurate estimation with only a finite number of transitions requires strong *ergodicity* conditions.

$$\hat{f} = \frac{1}{N} \sum_{n=1}^{N} f(\theta_i)$$

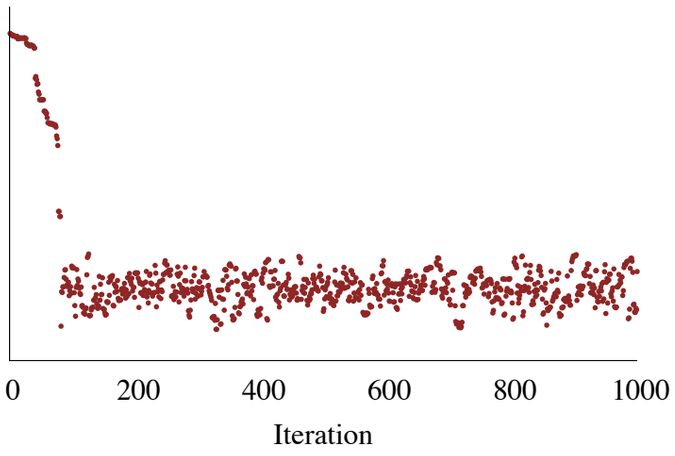
$$\hat{f} \sim \mathcal{N}\left(\mathbb{E}[f], \mathrm{MCSE}^2\right)$$

$$\mathrm{MCSE}^2 = \frac{\mathrm{Var}(f)}{\mathrm{ESS}}$$

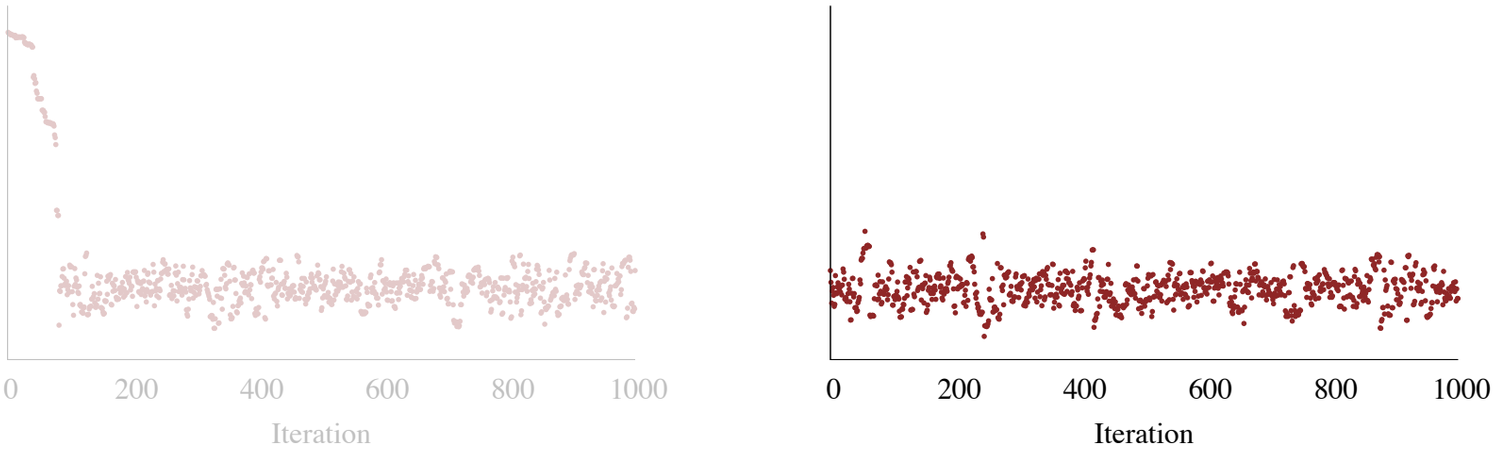
Markov Chain Monte Carlo Best Practices



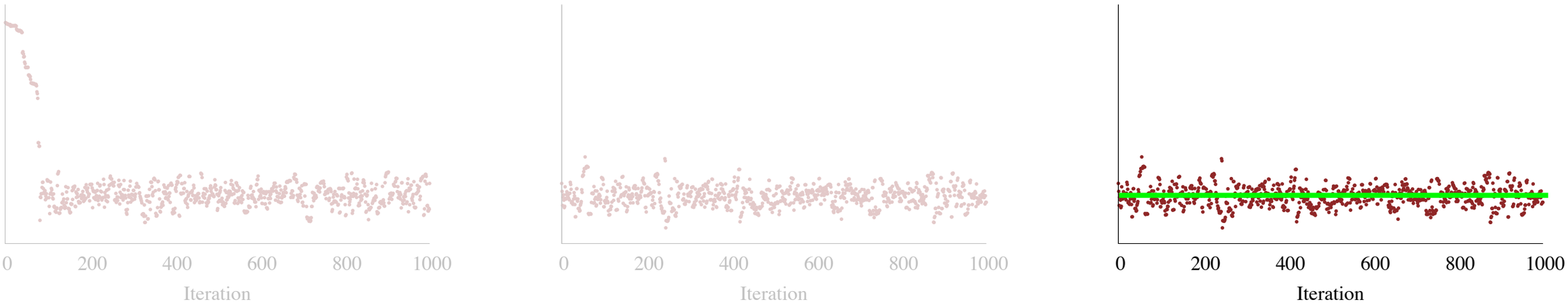
Markov chain Monte Carlo is best executed in three stages: warmup, sampling, and analysis.



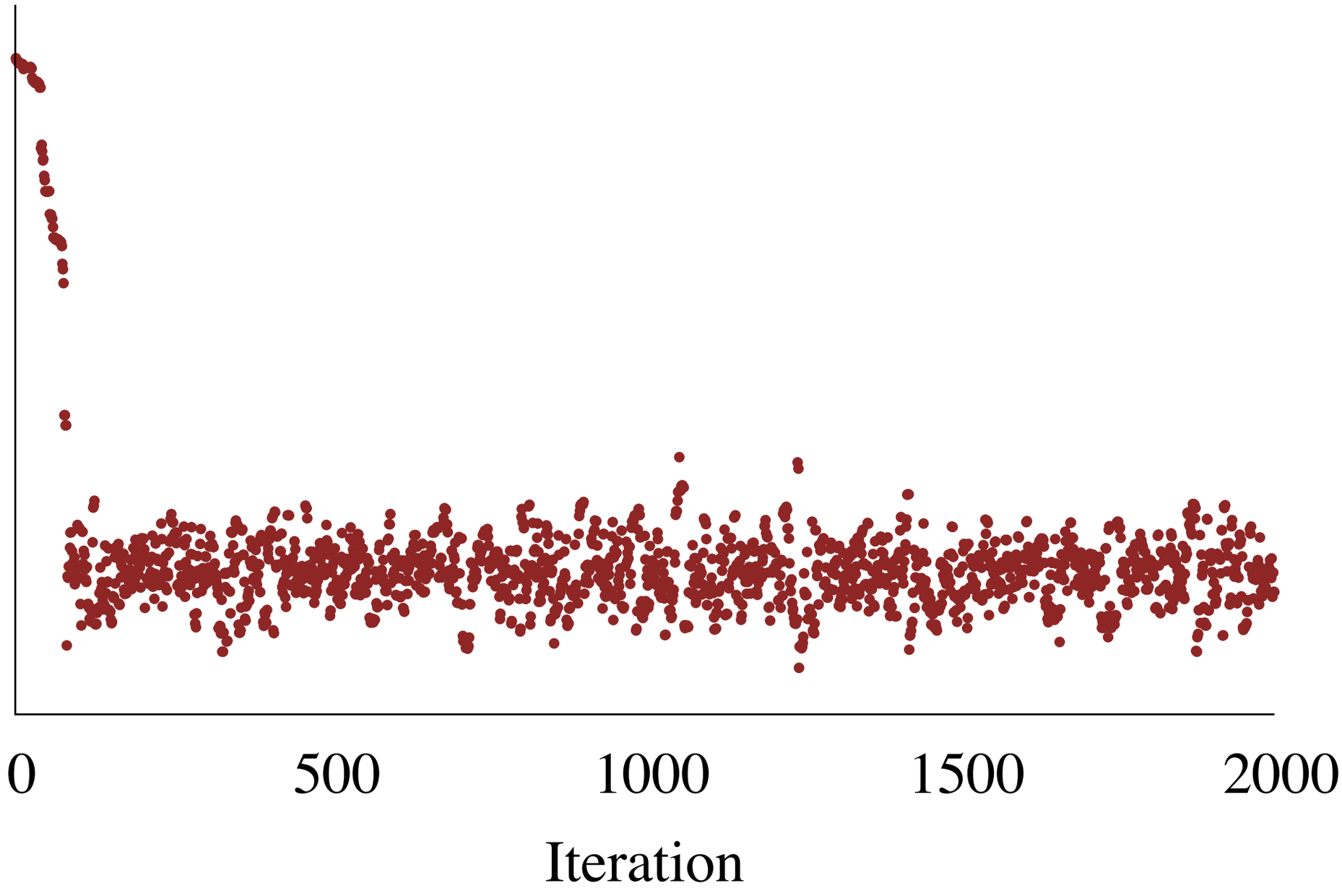
Markov chain Monte Carlo is best executed in three stages: warmup, sampling, and analysis.



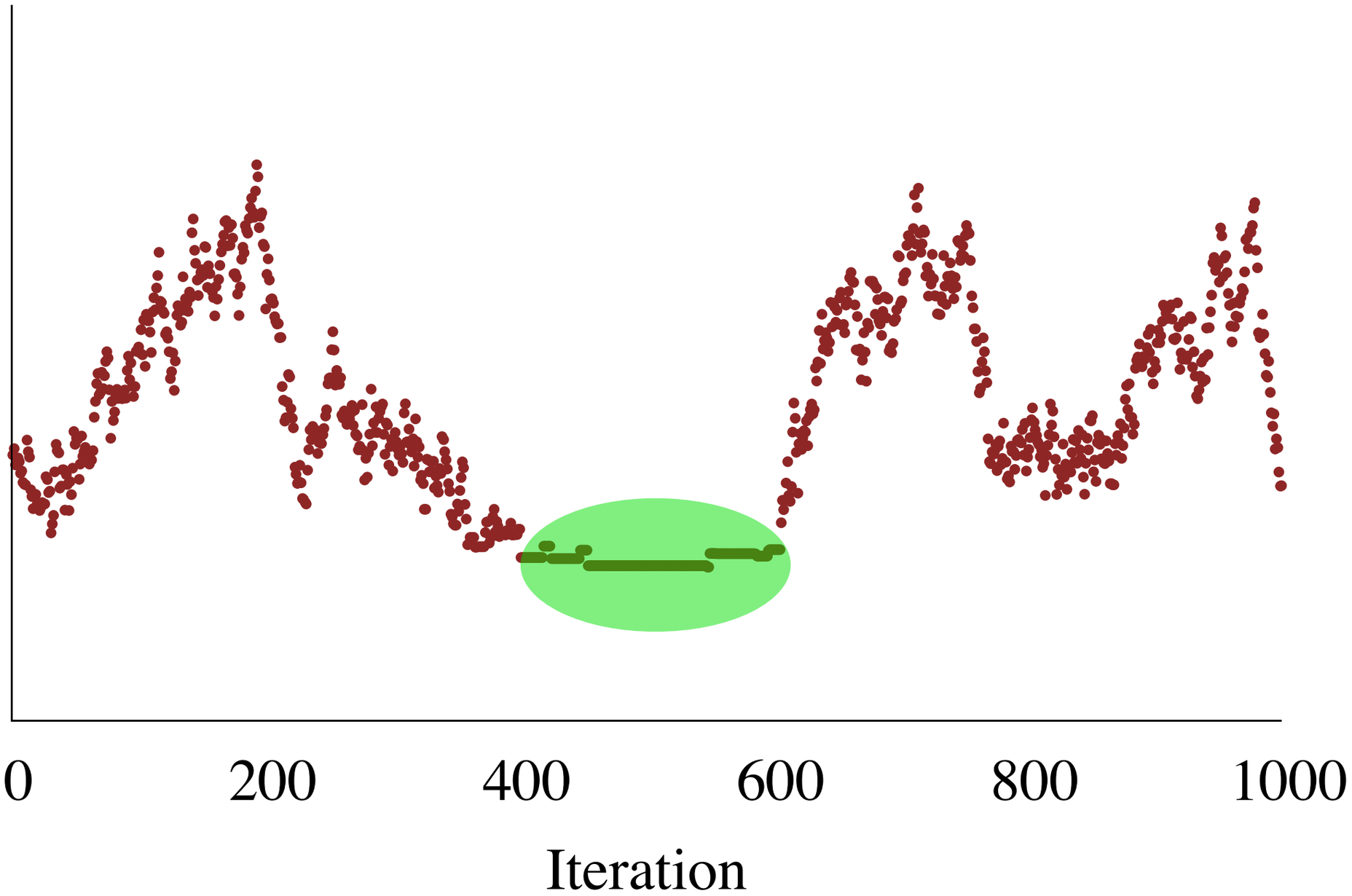
Markov chain Monte Carlo is best executed in three stages: warmup, sampling, and analysis.



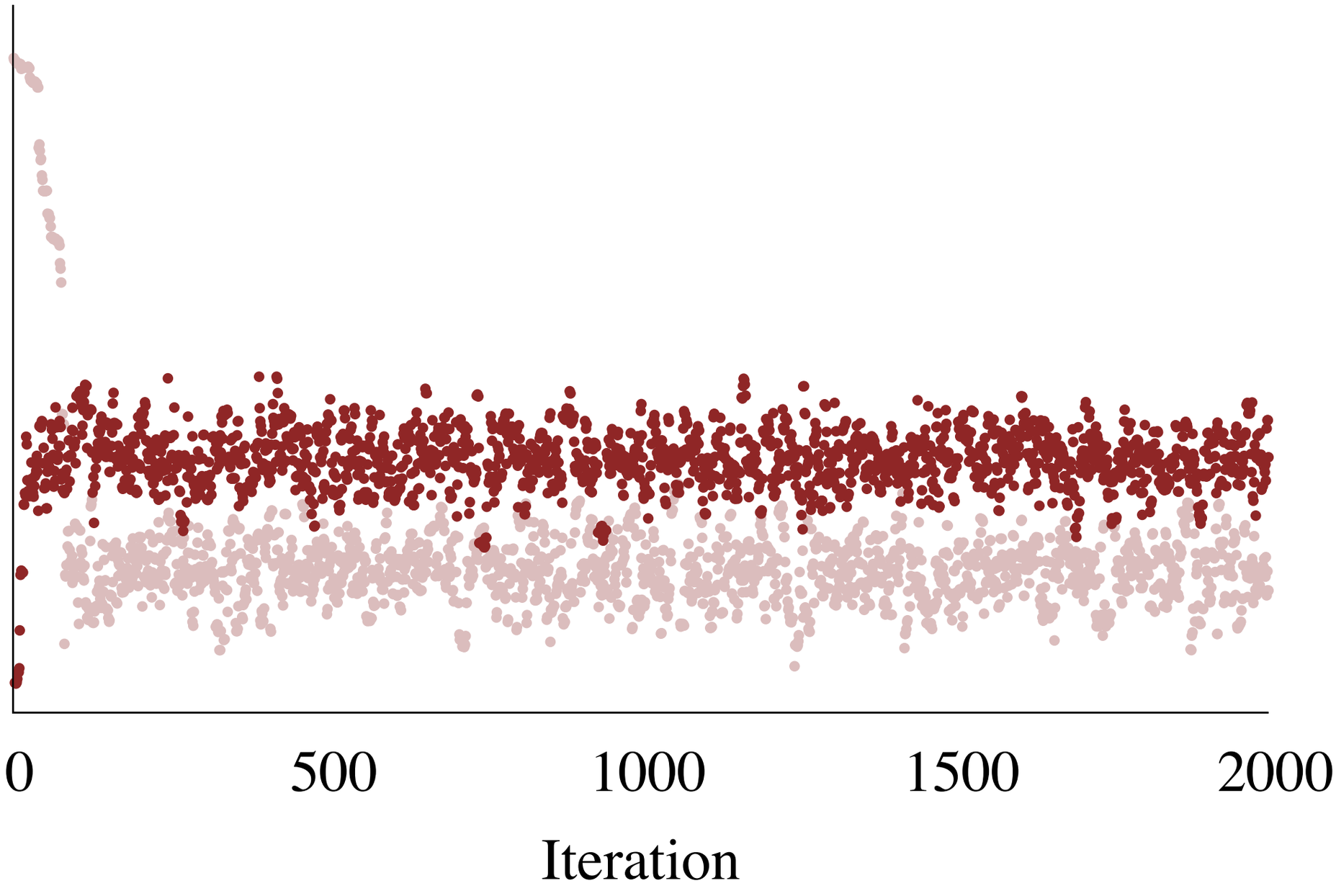
The first step in analyzing the chains is to verify that they found and then completely explored the typical set.



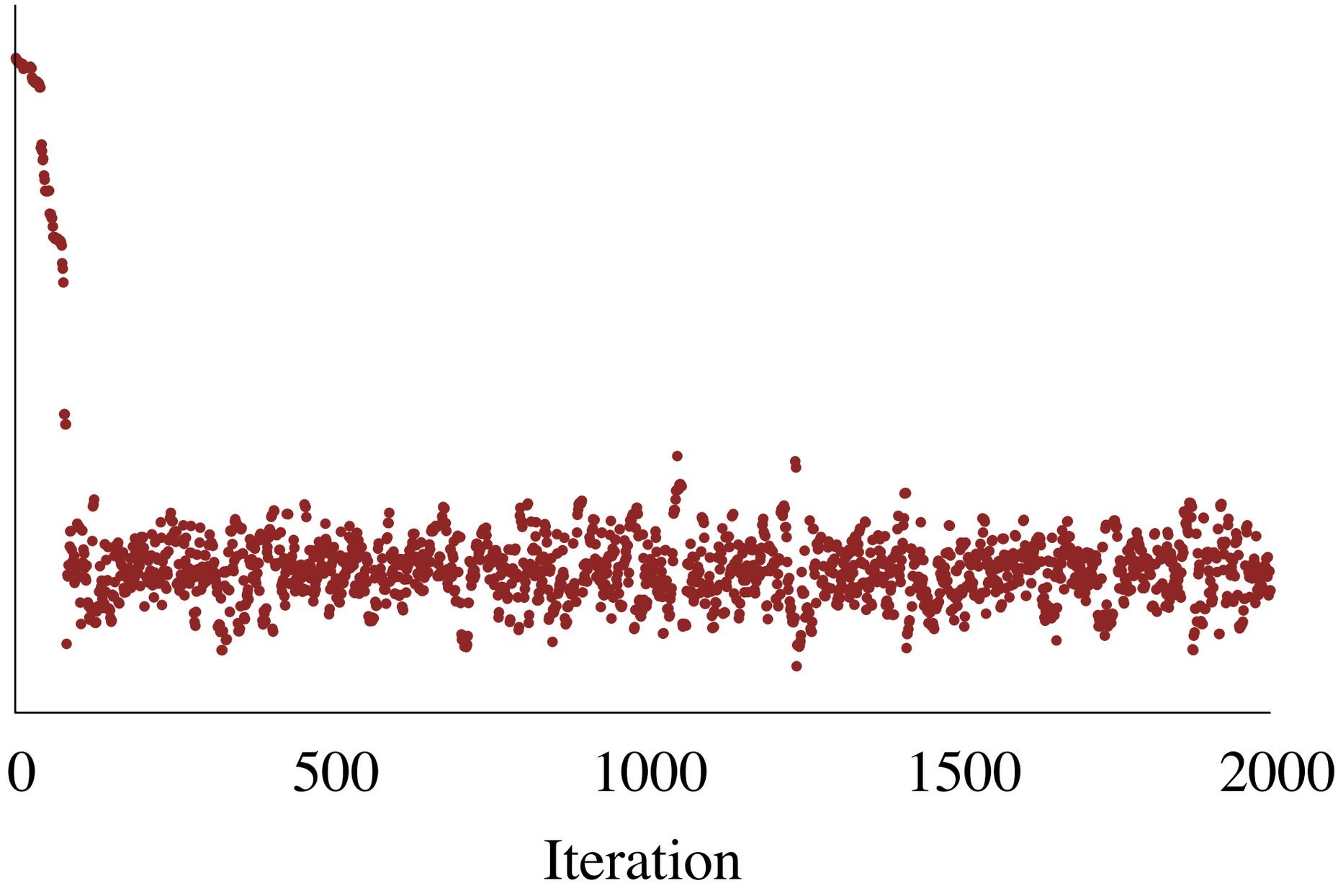
For example, exploration of the typical set can be obstructed by regions of high curvature.



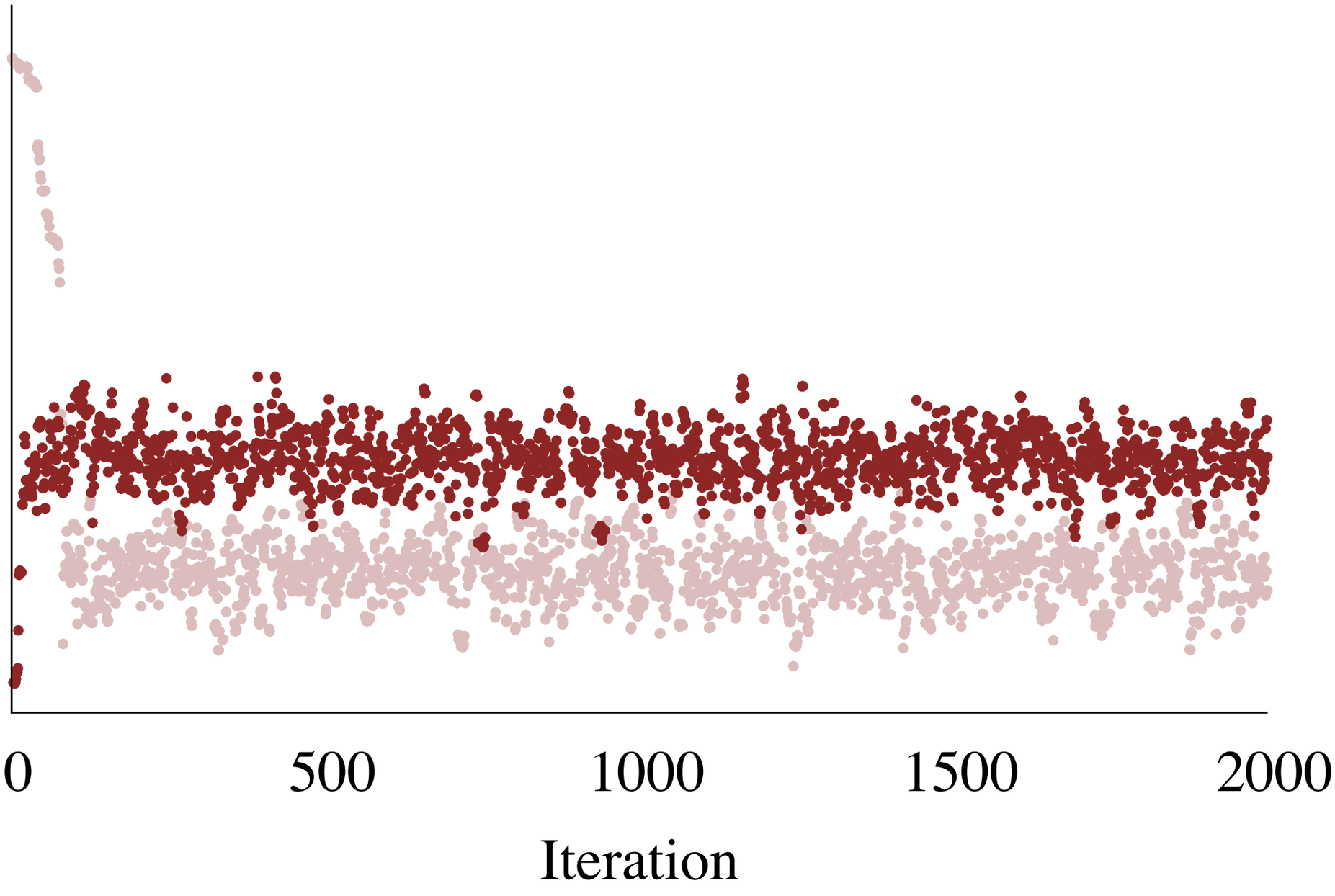
Disconnected typical sets, which occur in multimodal distributions, are especially problematic.



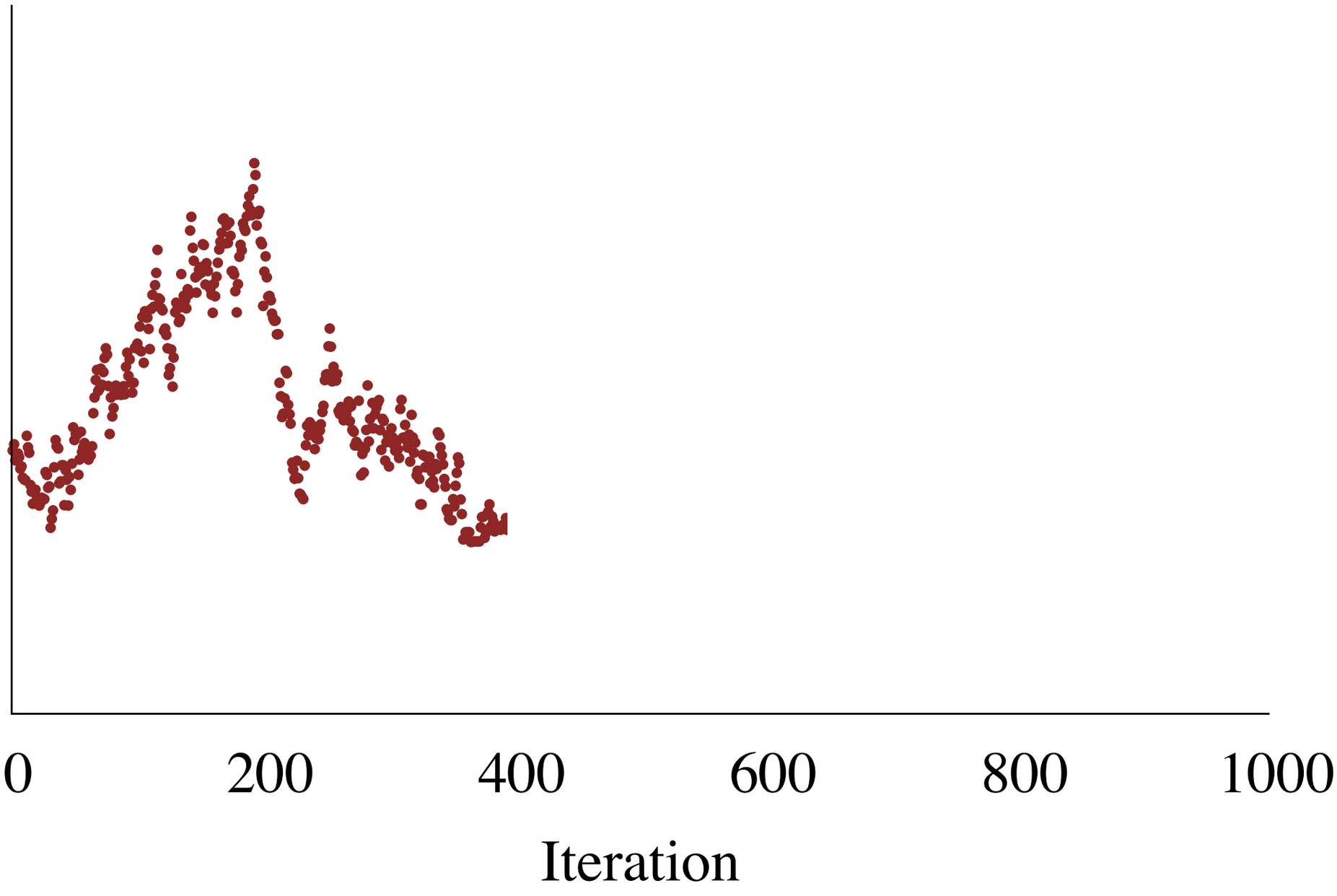
How can we verify that there were no obstructions?
Visual diagnostics are certainly appealing...



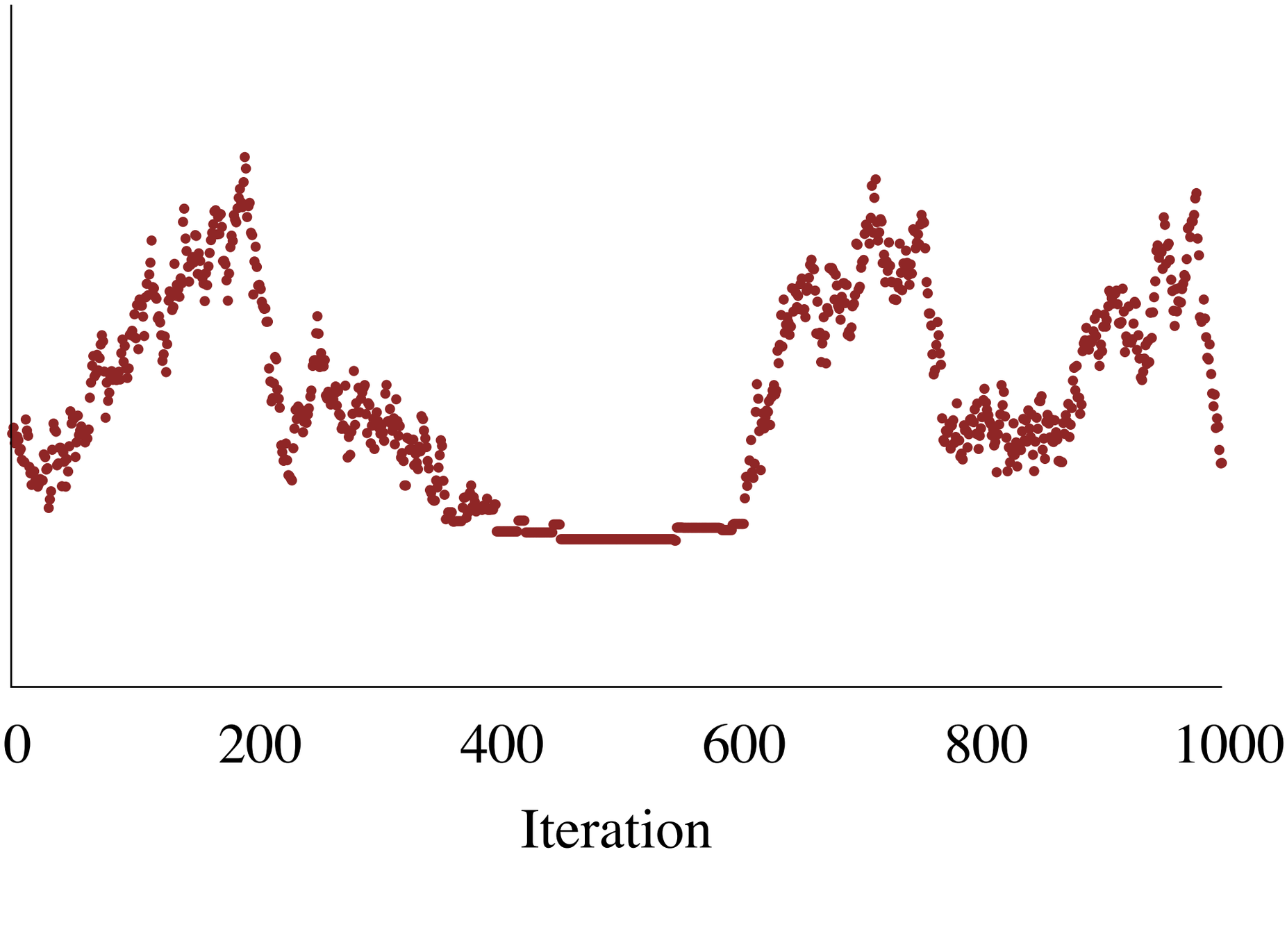
Unfortunately visual diagnostics can be misleading.



Unfortunately visual diagnostics can be misleading.

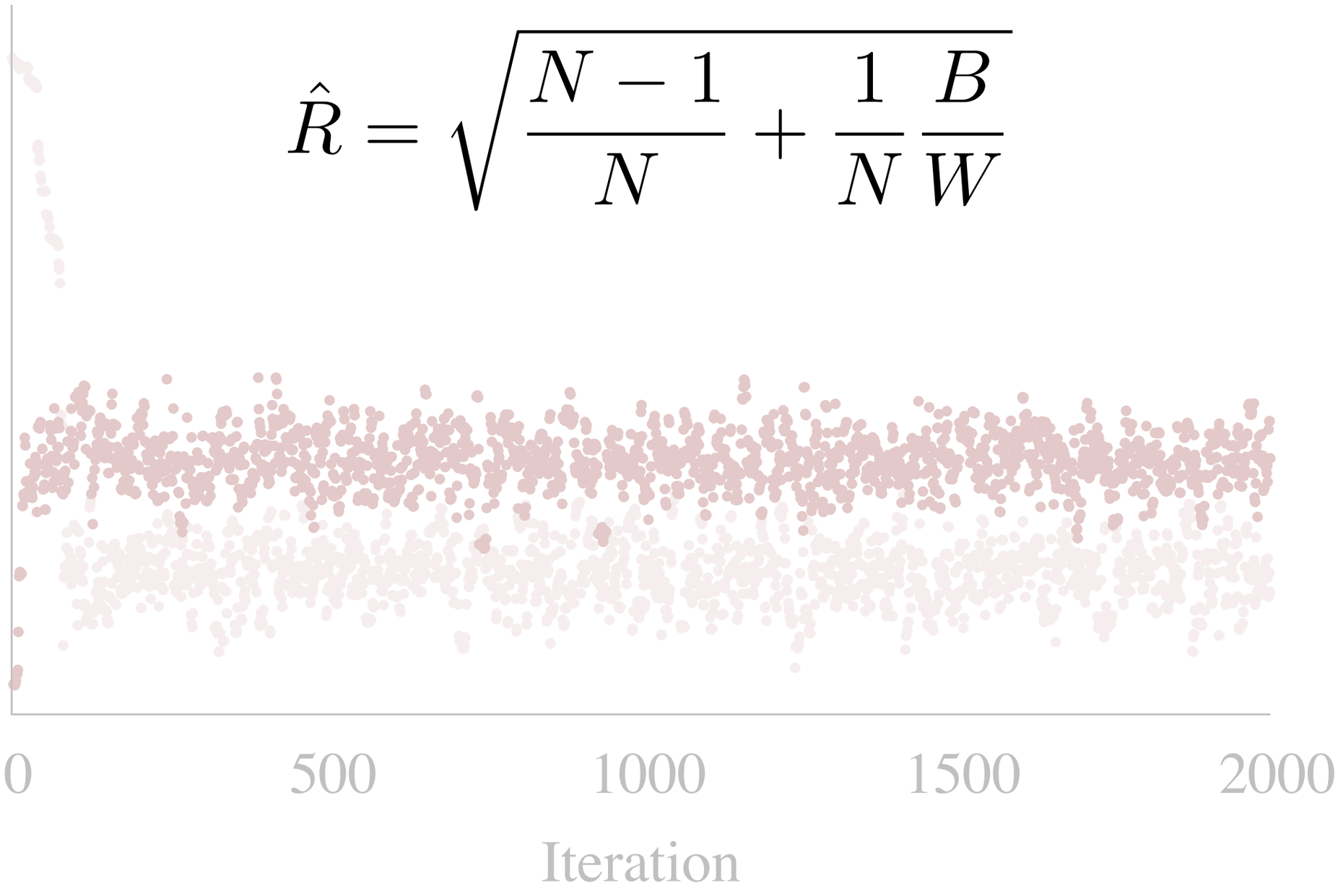


Unfortunately visual diagnostics can be misleading.



The best strategy is to run multiple chains from diffuse initializations and compare then using the *Rhat* statistic.

$$\hat{R} = \sqrt{\frac{N-1}{N} + \frac{1}{N}\frac{B}{W}}$$



The best strategy is to run multiple chains from diffuse initializations and compare then using the *Rhat* statistic.

```
Inference for Stan model: example_model
1 chains: each with iter=(1000); warmup=(0); thin=(1); 1000 iterations saved.

Warmup took (0.034) seconds, 0.034 seconds total
Sampling took (0.039) seconds, 0.039 seconds total
```

	Mean	MCSE	StdDev	5%	50%	95%	N_Eff	N_Eff/s	R_hat
lp__	5.0	5.7e-02	9.6e-01	3.1	5.2	5.9	287	7431	1.0
accept_stat__	0.93	2.7e-03	8.6e-02	0.76	0.96	1.0	1000	25869	1.0
stepsize__	0.78	3.1e-15	2.2e-15	0.78	0.78	0.78	0.50	13	1.0
treedepth__	2.0	2.0e-02	5.7e-01	1.0	2.0	3.0	778	20124	1.0
n_leapfrog__	3.4	5.8e-02	1.8e+00	1.0	3.0	7.0	950	24588	1.0
n_divergent__	0.00	0.0e+00	0.0e+00	0.00	0.00	0.00	1000	25869	1.0
x	20	4.3e-02	1.0e+00	18	20	22	568	14697	1.0

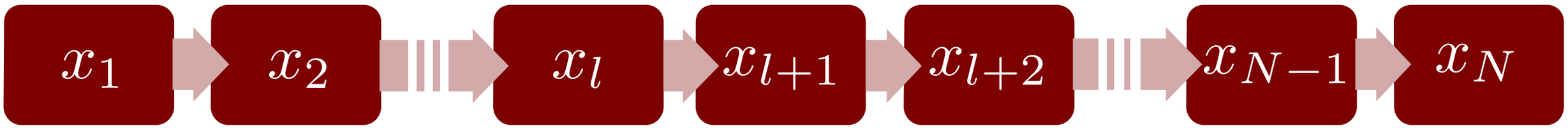
If there are no indications of pathologies then we can move on to quantifying the accuracy of our estimates.

$$\hat{f} = \frac{1}{N} \sum_{n=1}^{N} f(\theta_i)$$

$$\hat{f} \sim \mathcal{N}\left(\mathbb{E}[f], \mathrm{MCSE}^2\right)$$

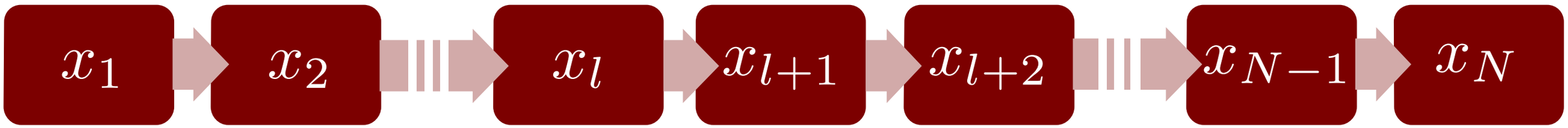
$$\mathrm{MCSE}^2 = \frac{\mathrm{Var}(f)}{\mathrm{ESS}}$$

We first quantify the correlations in the Markov chains by estimating their *autocorrelations*.



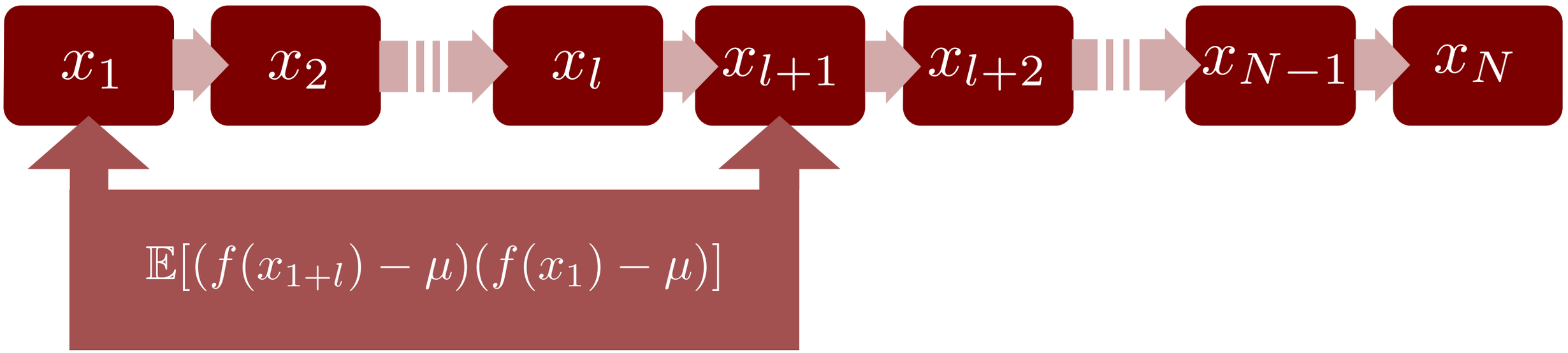
We first quantify the correlations in the Markov chains by estimating their *autocorrelations*.

$$\mu = \mathbb{E}[f(x)]$$



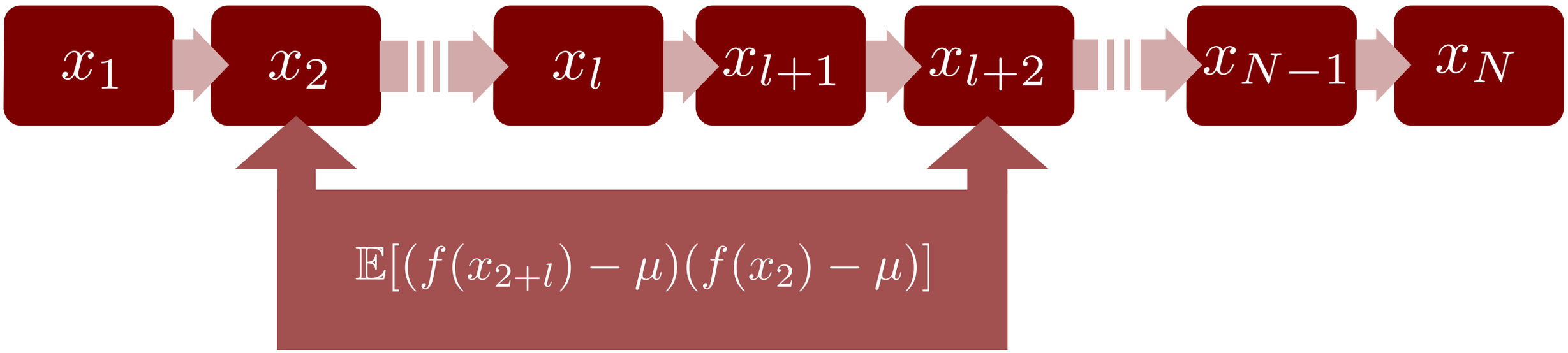
We first quantify the correlations in the Markov chains by estimating their *autocorrelations*.

$$\mu = \mathbb{E}[f(x)]$$



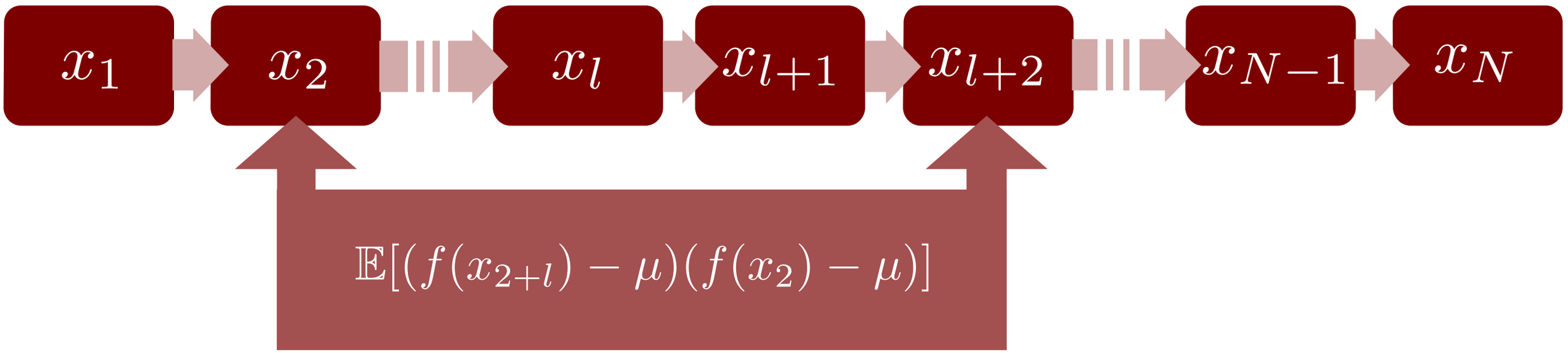
We first quantify the correlations in the Markov chains by estimating their *autocorrelations*.

$$\mu = \mathbb{E}[f(x)]$$



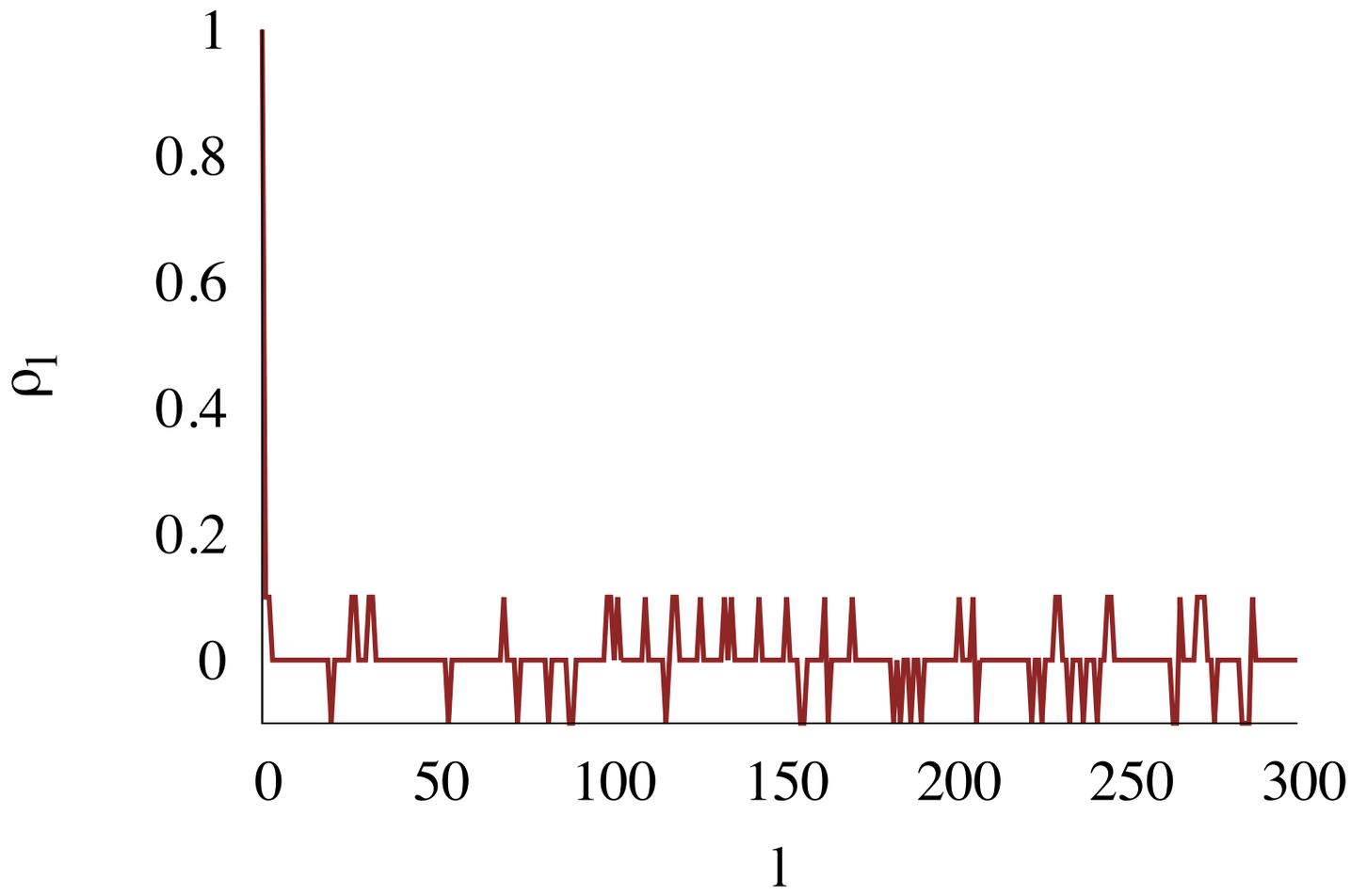
We first quantify the correlations in the Markov chains by estimating their *autocorrelations*.

$$\mu = \mathbb{E}[f(x)]$$

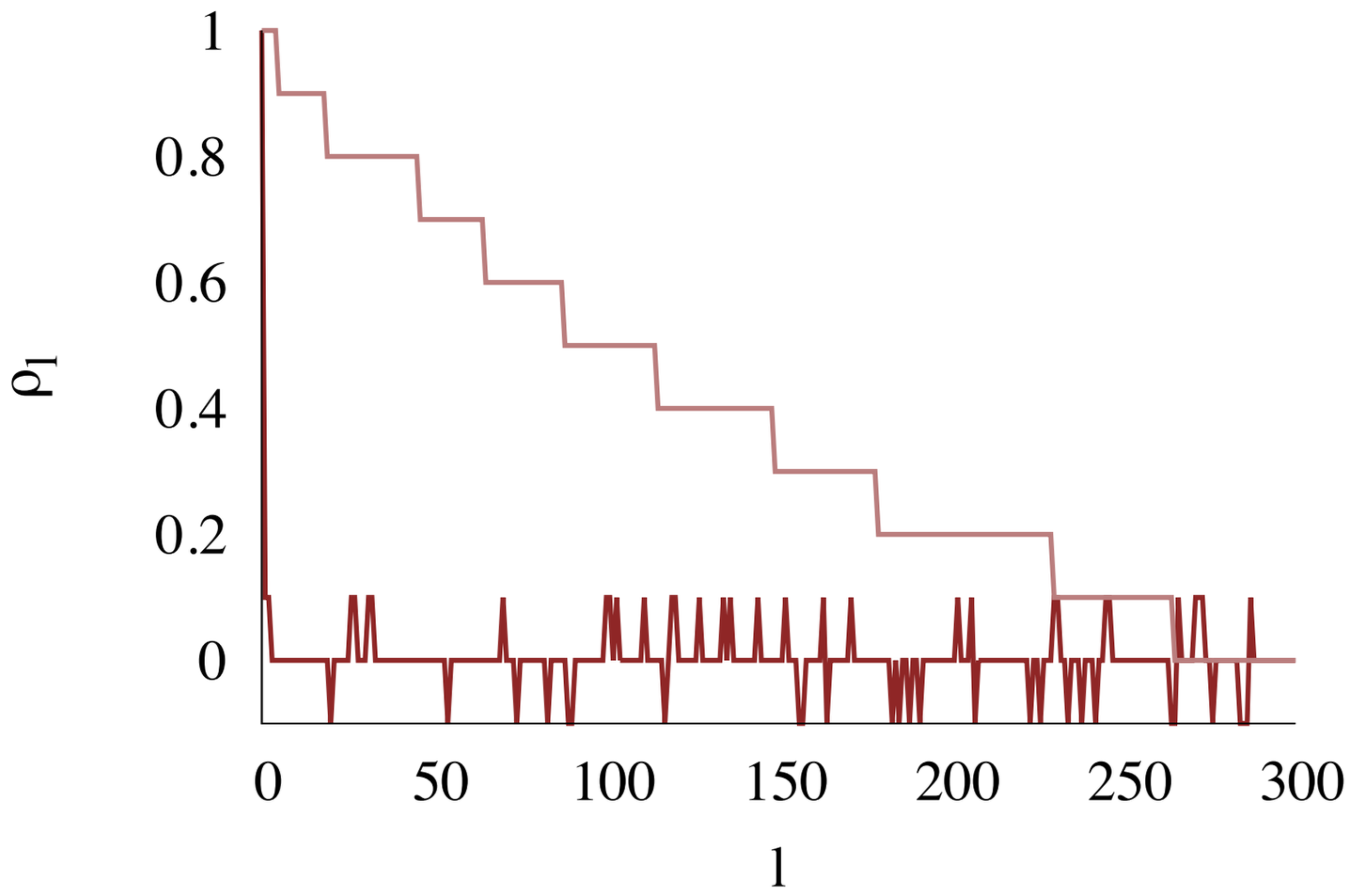


$$\rho_l = \frac{\sum_{m=1}^{N} \mathbb{E}[(f(x_{m+l}) - \mu)(f(x_m) - \mu)]}{\mathrm{Var}(f)}$$

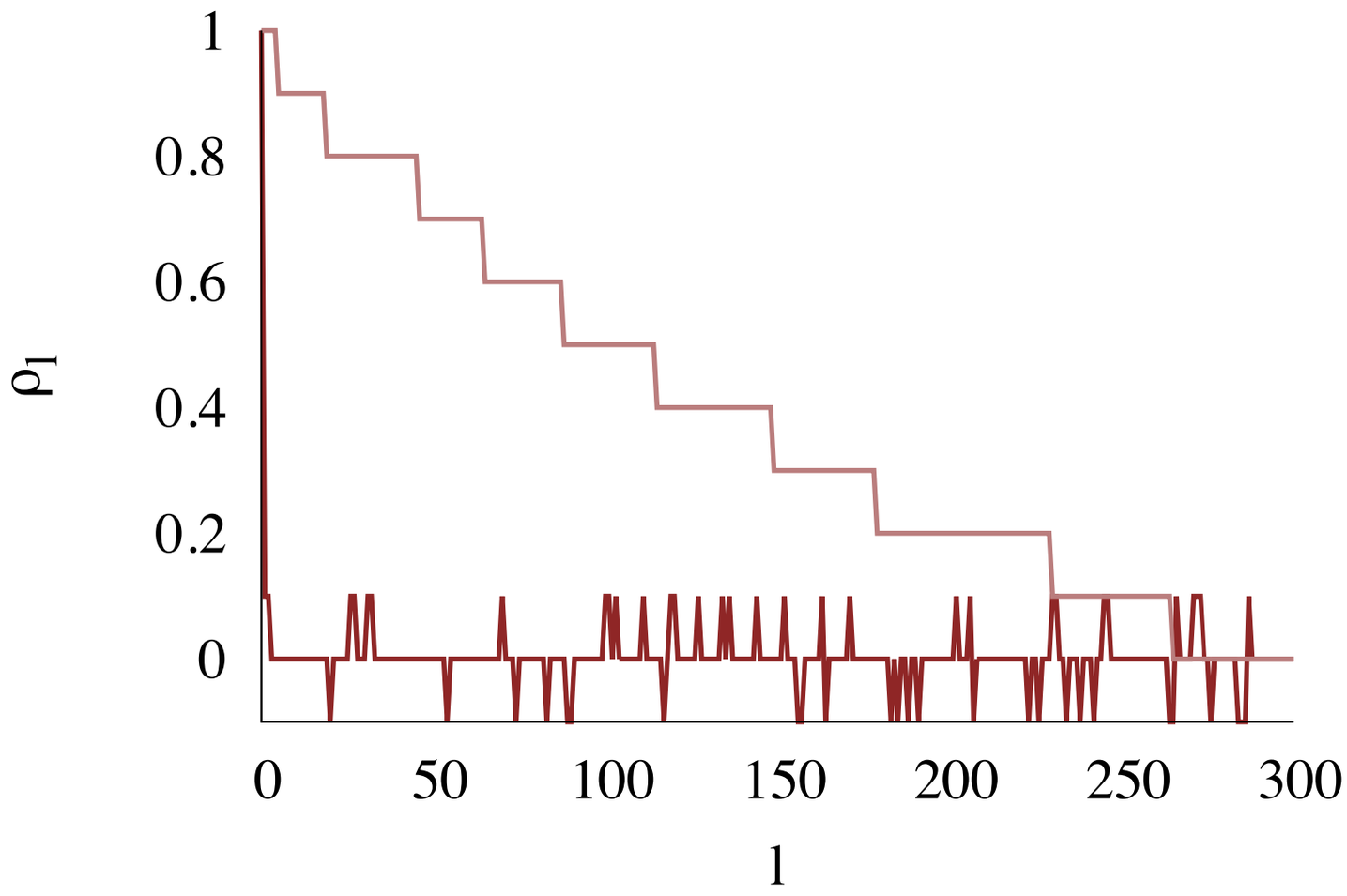
We first quantify the correlations in the Markov chains by estimating their *autocorrelations*.



We first quantify the correlations in the Markov chains by estimating their *autocorrelations*.



The sum of the autocorrelations, known as the *Effective Sample Size*, then quantifies the overall power of the chain.



$$\text{ESS} = \frac{N}{1 + 2\sum_{l=1}^{N} \rho_l}$$

Finally, we can construct an MCMC estimator of any pertinent function as well as an estimate of its error.

$$\hat{f} = \frac{1}{N} \sum_{n=1}^{N} f(\theta_i)$$

$$\hat{f} \sim \mathcal{N}\left(\mathbb{E}[f], \mathrm{MCSE}^2\right)$$

$$\mathrm{MCSE}^2 = \frac{\mathrm{Var}(f)}{\mathrm{ESS}}$$

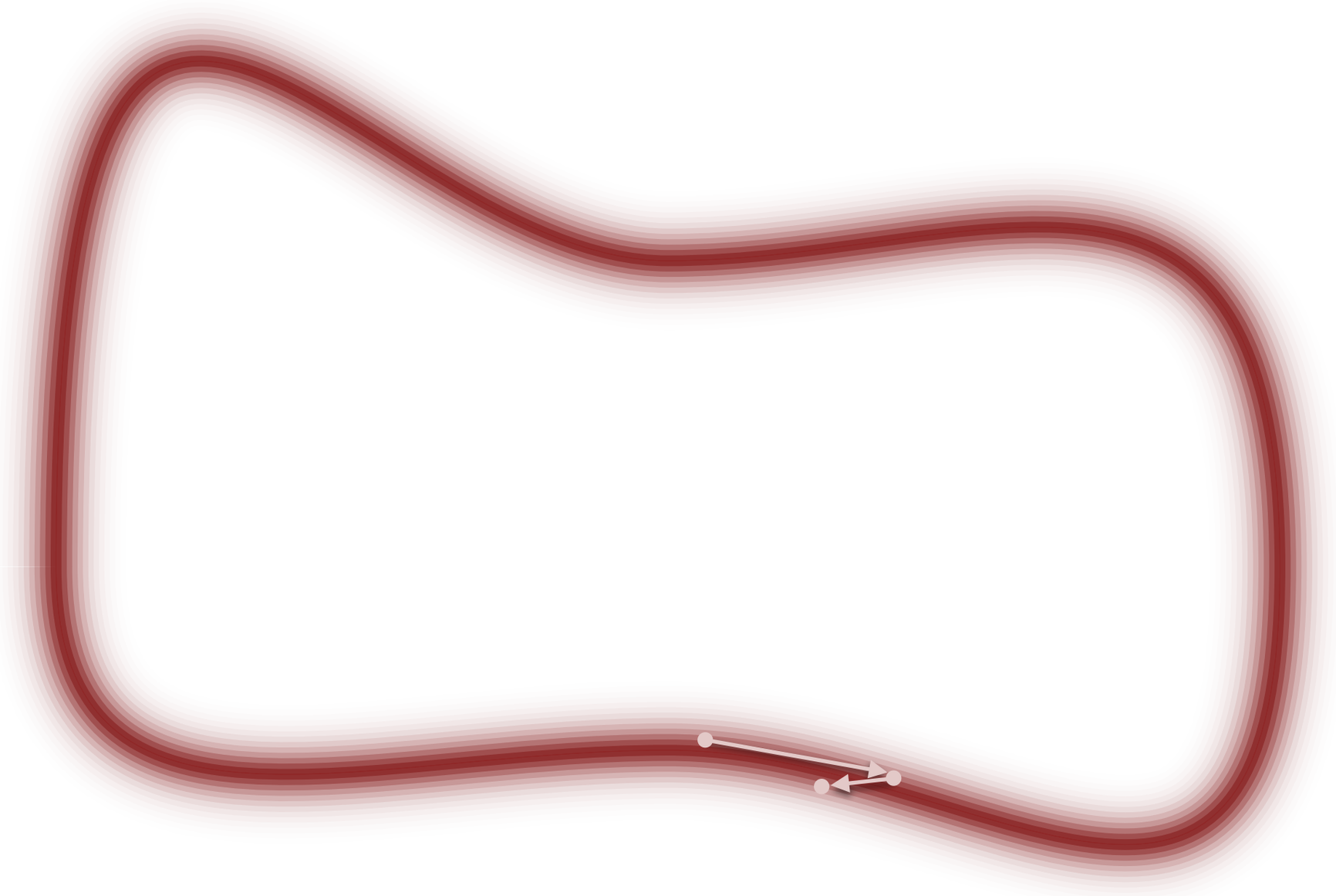
Finally, we can construct an MCMC estimator of any pertinent function as well as an estimate of its error.

```
Inference for Stan model: example_model
1 chains: each with iter=(1000); warmup=(0); thin=(1); 1000 iterations saved.

Warmup took (0.034) seconds, 0.034 seconds total
Sampling took (0.039) seconds, 0.039 seconds total
```

	Mean	MCSE	StdDev	5%	50%	95%	N_Eff	N_Eff/s	R_hat
lp__	5.0	5.7e-02	9.6e-01	3.1	5.2	5.9	287	7431	1.0
accept_stat__	0.93	2.7e-03	8.6e-02	0.76	0.96	1.0	1000	25869	1.0
stepsize__	0.78	3.1e-15	2.2e-15	0.78	0.78	0.78	0.50	13	1.0
treedepth__	2.0	2.0e-02	5.7e-01	1.0	2.0	3.0	778	20124	1.0
n_leapfrog__	3.4	5.8e-02	1.8e+00	1.0	3.0	7.0	950	24588	1.0
n_divergent__	0.00	0.0e+00	0.0e+00	0.00	0.00	0.00	1000	25869	1.0
x	20	4.3e-02	1.0e+00	18	20	22	568	14697	1.0

Explicit Implementations of Markov Chain Monte Carlo



The previous discussion assumes that we can construct a Markov chain that targets our specific posterior.

$$\pi_S(\theta|\widetilde{\mathcal{D}}) = \int \mathrm{d}\theta' \, T(\theta, \theta') \pi_S(\theta'|\widetilde{\mathcal{D}})$$

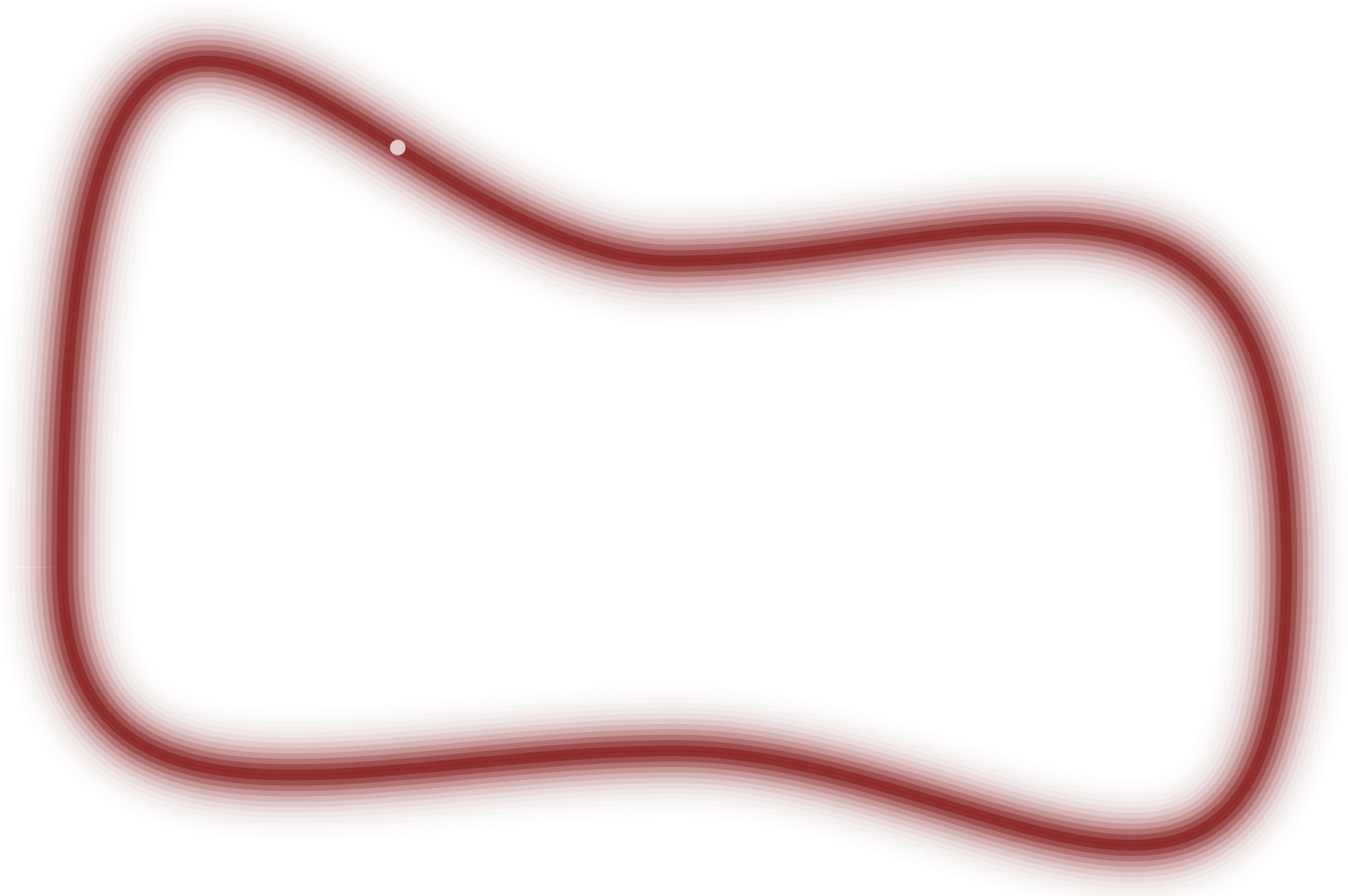
To simplify notation let's drop the explicit dependence on the data for the remainder of this lecture.

$$\pi(\theta) = \int \mathrm{d}\theta' \, T(\theta, \theta') \, \pi(\theta')$$

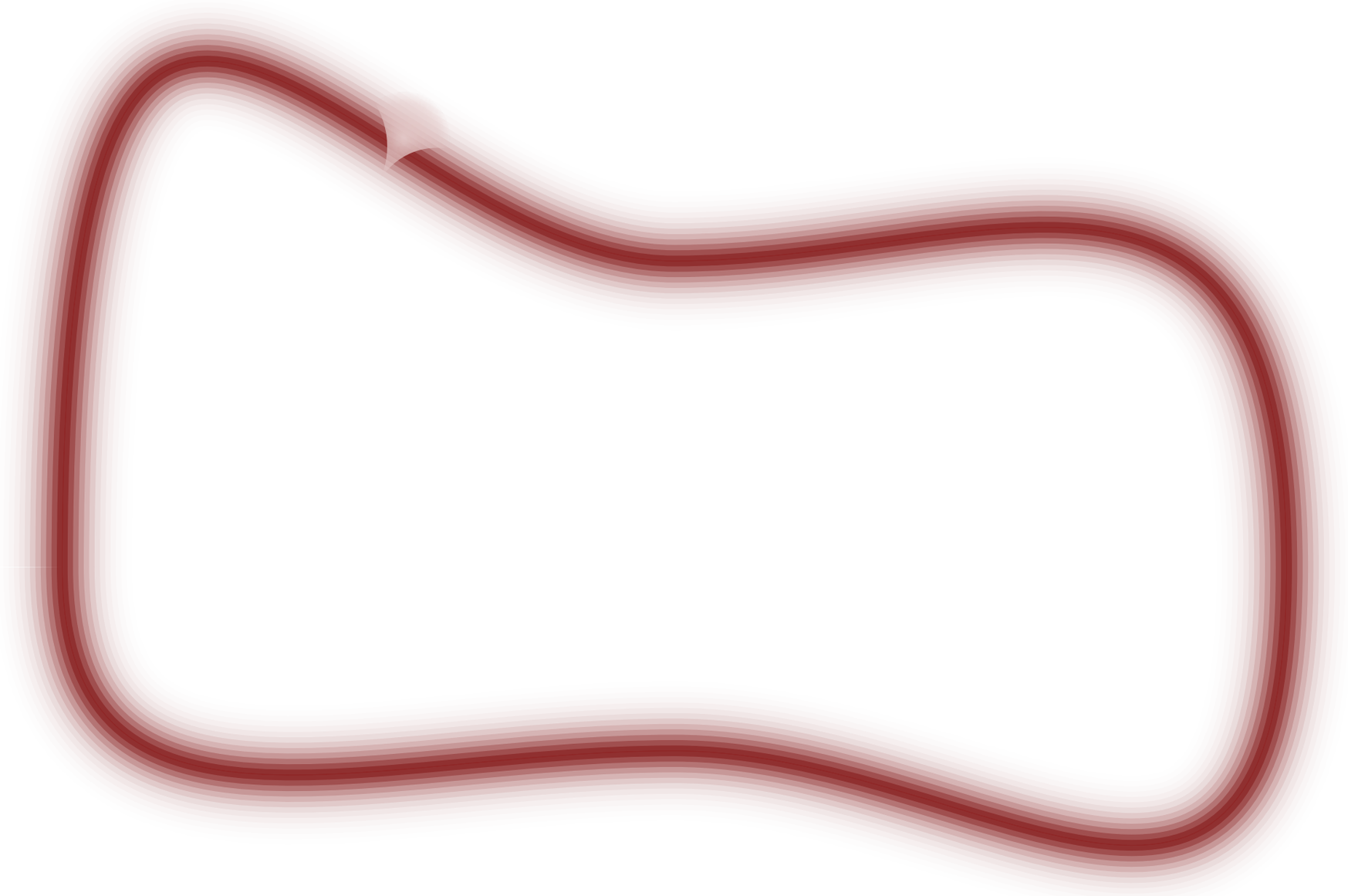
One way to construct a chain is Random Walk Metropolis which explores the posterior with a "guided" diffusion.

$$T(\theta, \theta') = \mathcal{N}\left(\theta' | \theta, \sigma^2\right) \min\left(1, \frac{\pi(\theta')}{\pi(\theta)}\right)$$

Unfortunately this naive diffusion explores high-dimensional target distributions inefficiently.

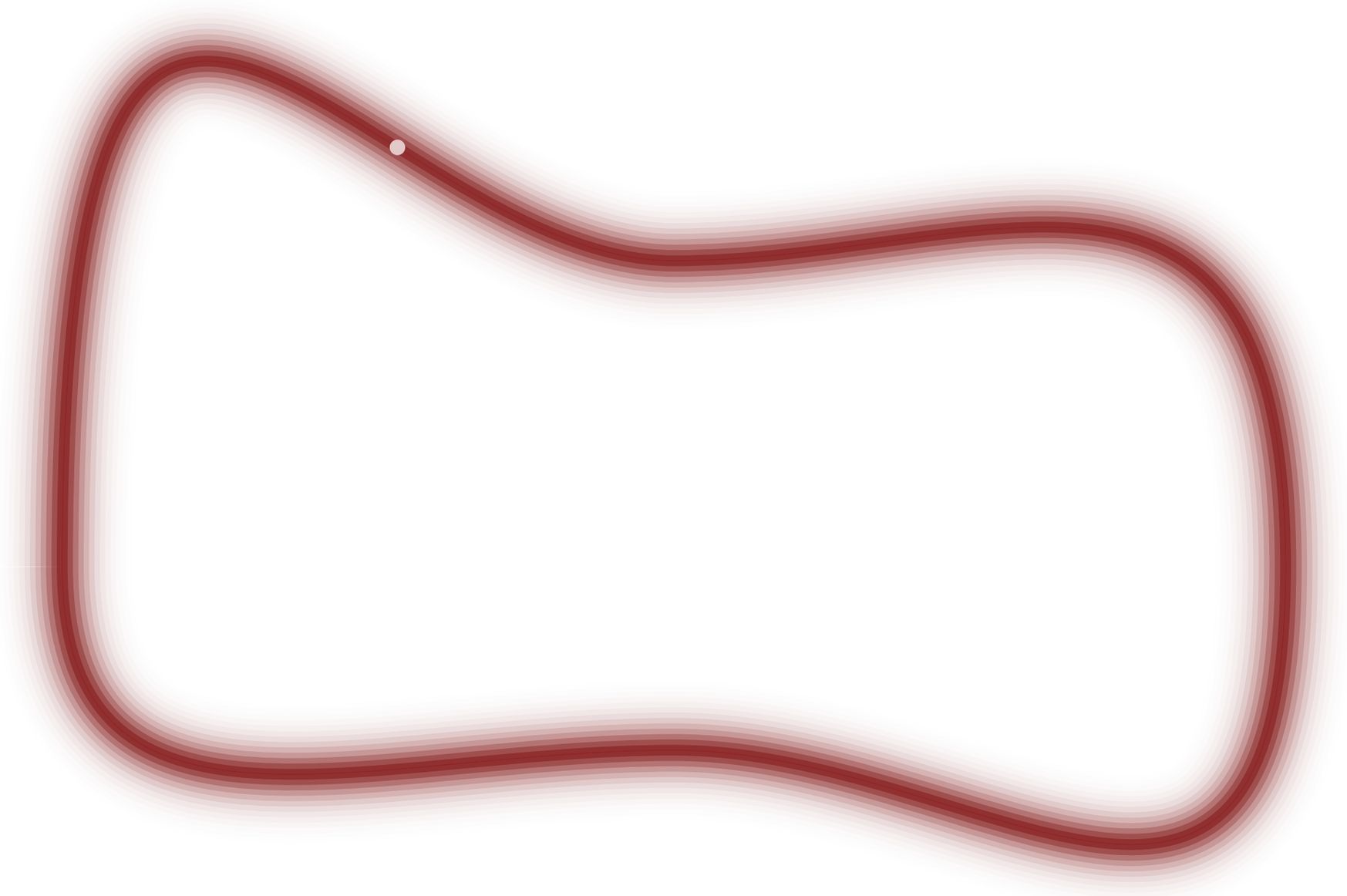


Unfortunately this naive diffusion explores high-dimensional target distributions inefficiently.



Unfortunately this naive diffusion explores high-dimensional target distributions inefficiently.

In order to efficiently scale to complex posteriors, MCMC needs *coherent* exploration of the typical set.

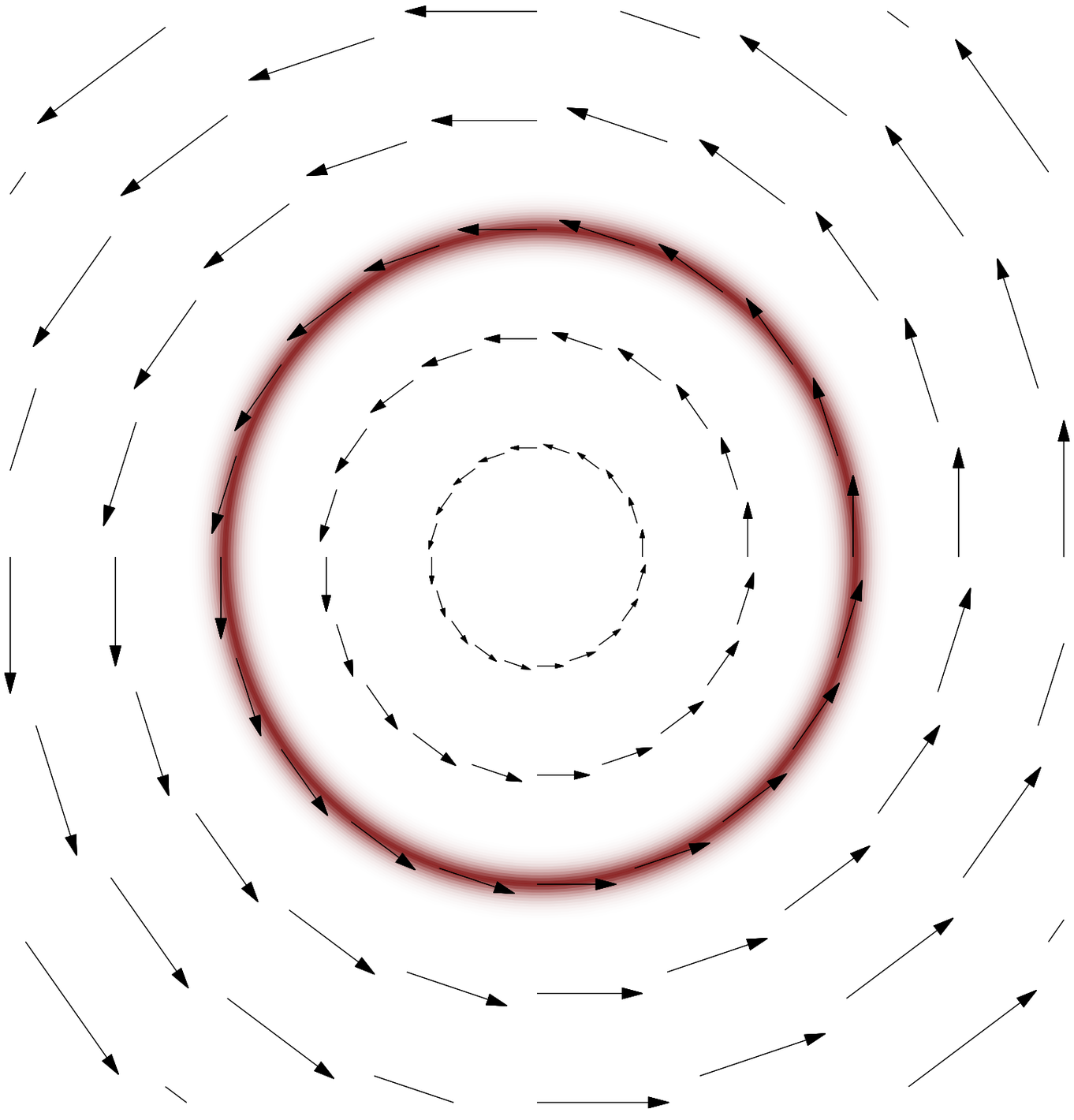


In order to efficiently scale to complex posteriors, MCMC needs *coherent* exploration of the typical set.

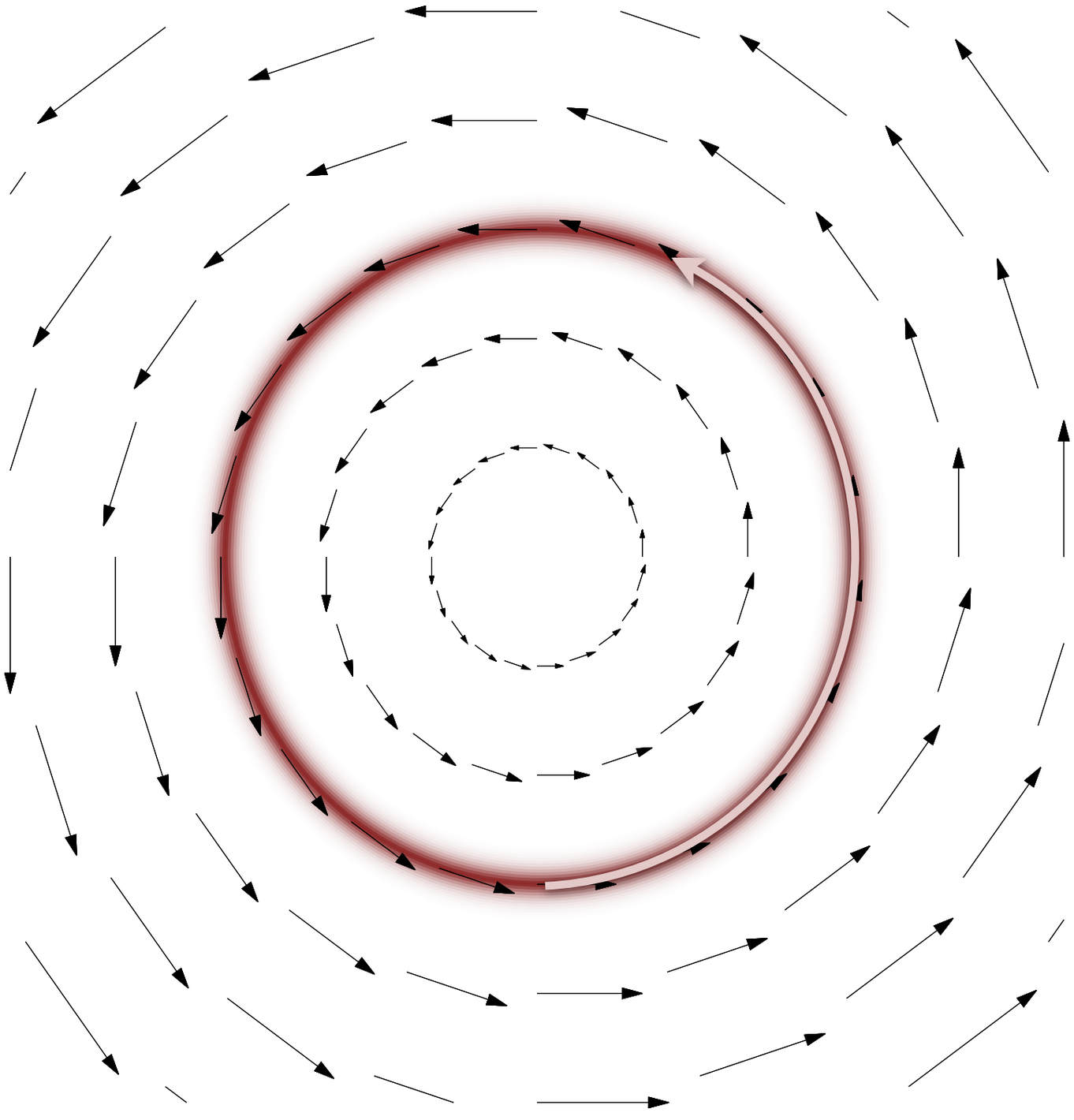
Hamiltonian Monte Carlo

One way to construct the desired exploration is to integrate along a *vector field* aligned with the typical set.

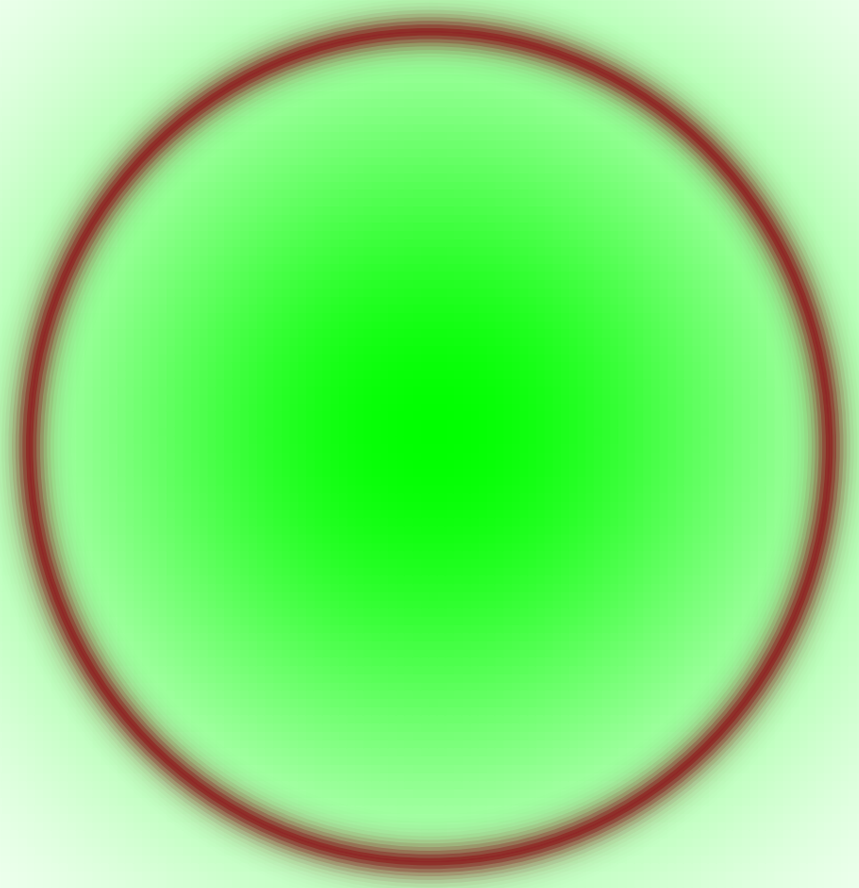
One way to construct the desired exploration is to integrate along a *vector field* aligned with the typical set.



One way to construct the desired exploration is to integrate along a *vector field* aligned with the typical set.

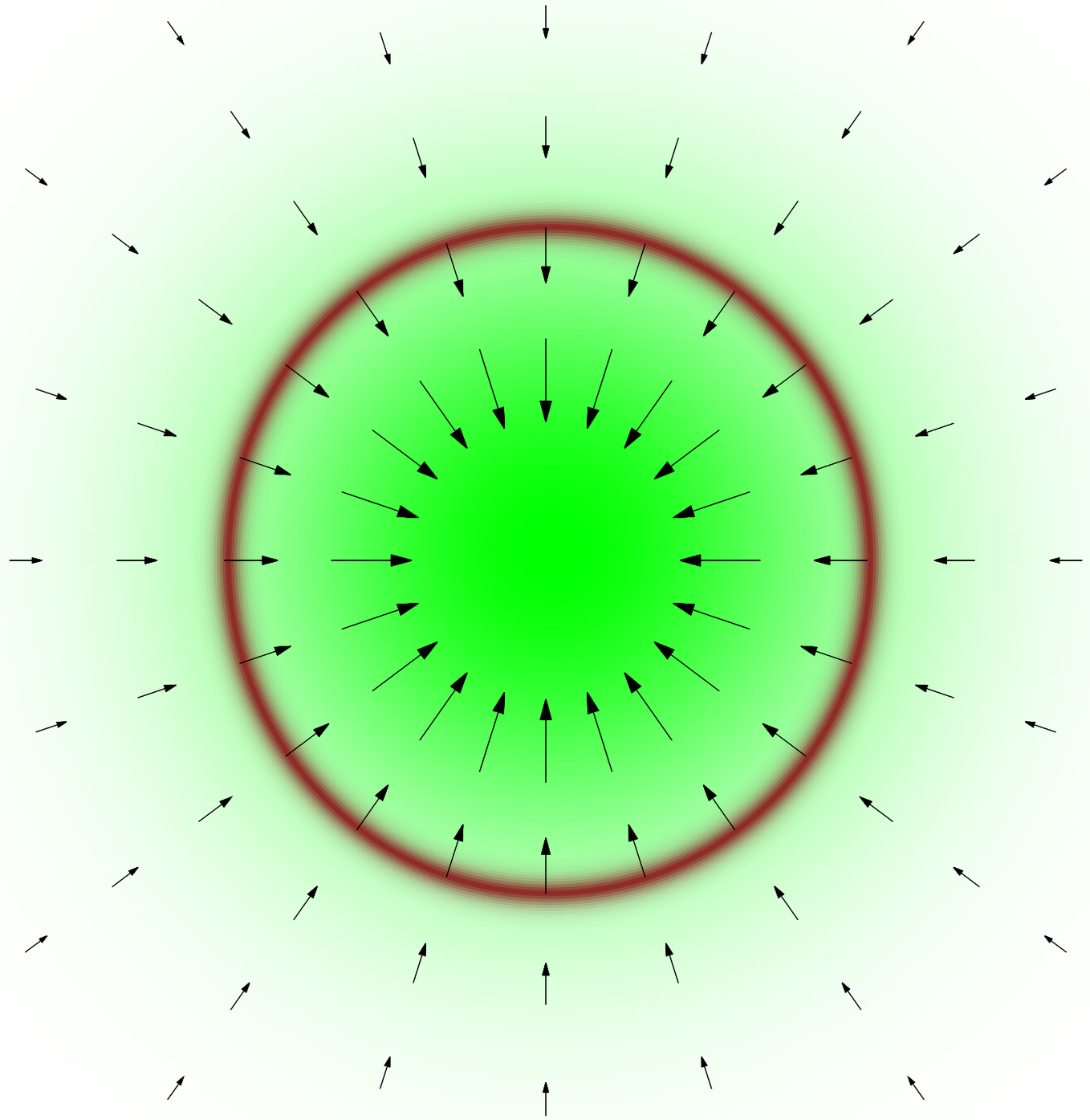


Creating the desired vector field requires transforming available vector fields, such as the gradient.



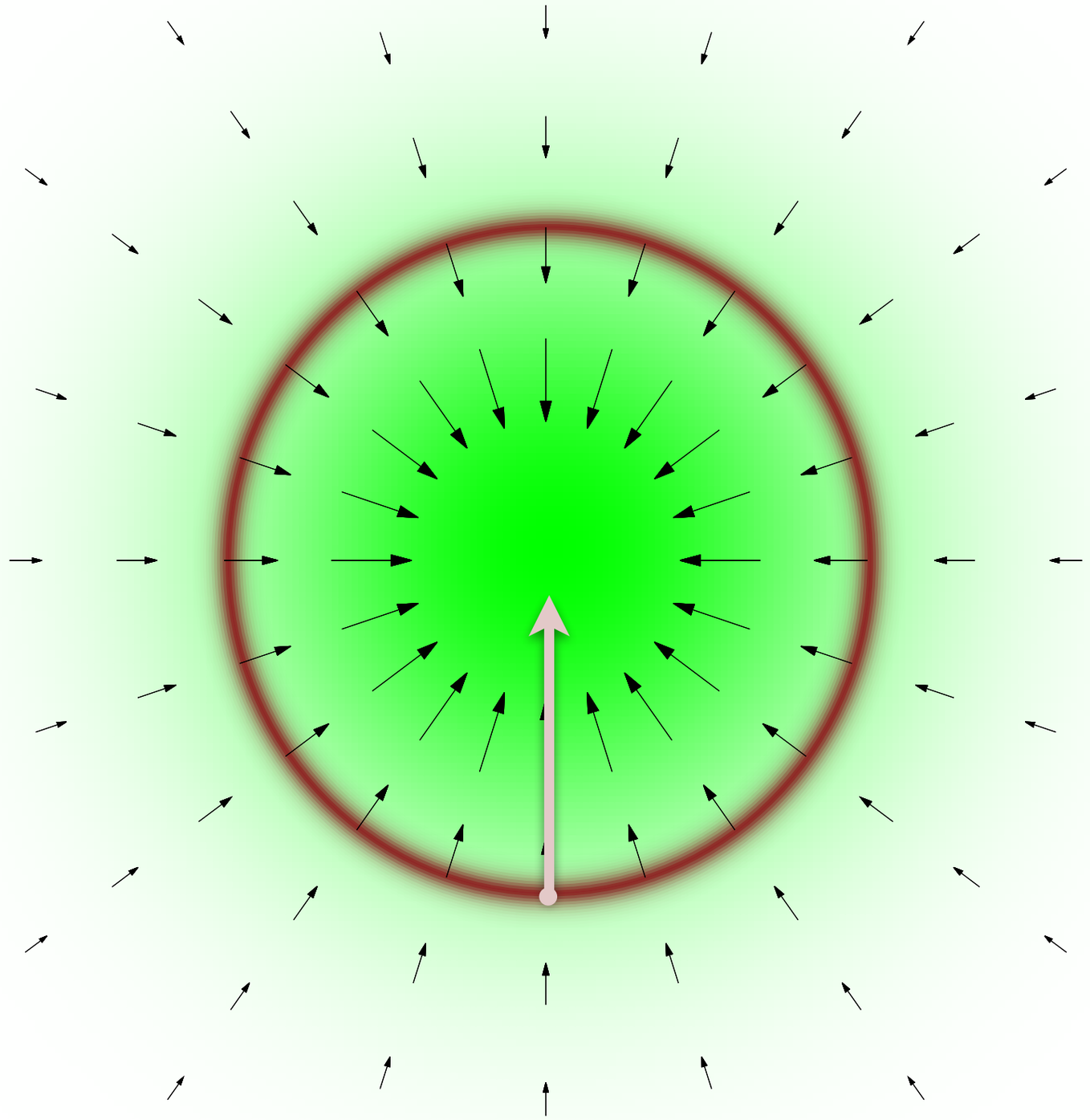
$\pi(\theta)$

Creating the desired vector field requires transforming available vector fields, such as the gradient.



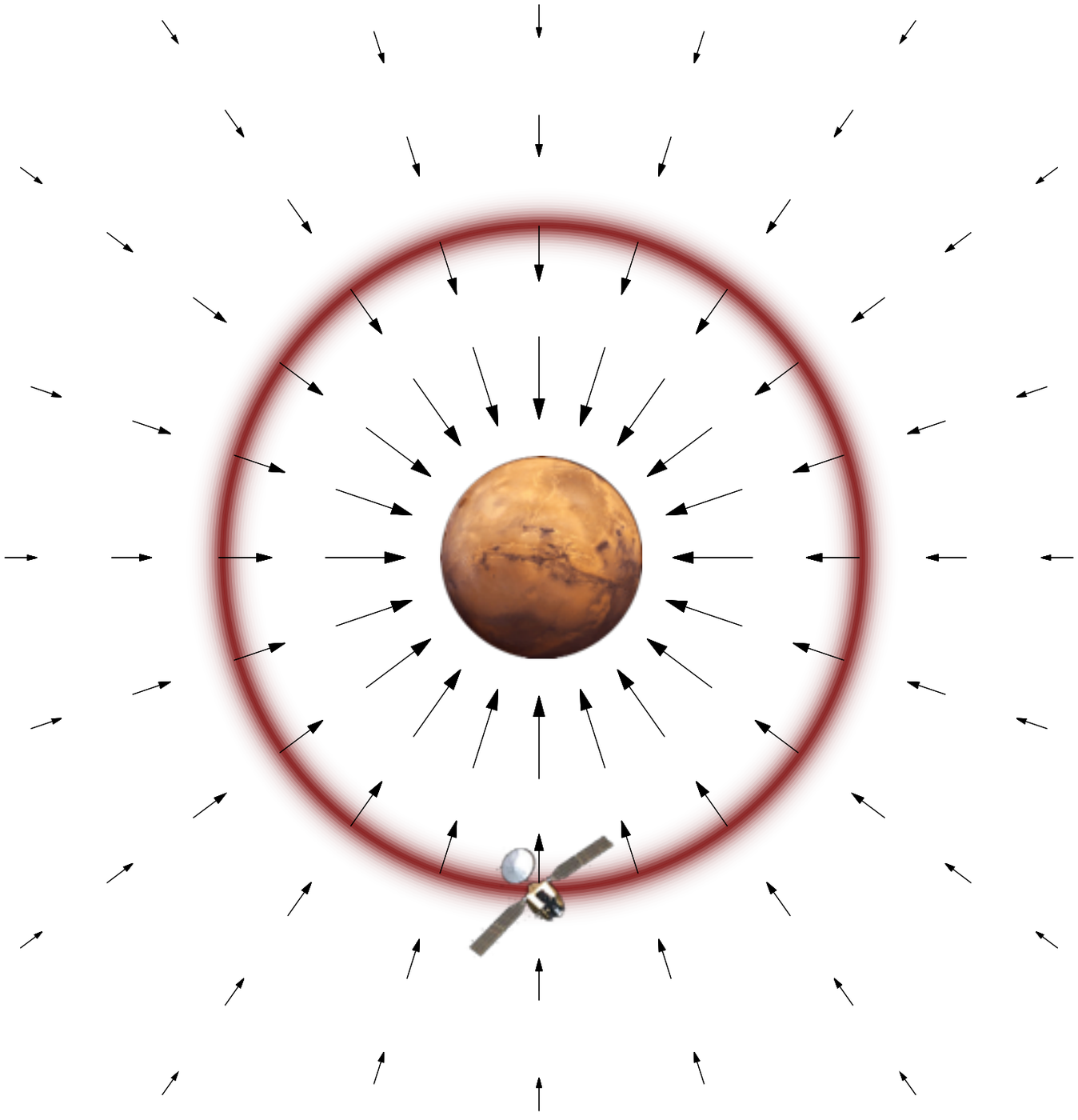
$$\frac{\partial \pi(\theta)}{\partial \theta}$$

Creating the desired vector field requires transforming available vector fields, such as the gradient.

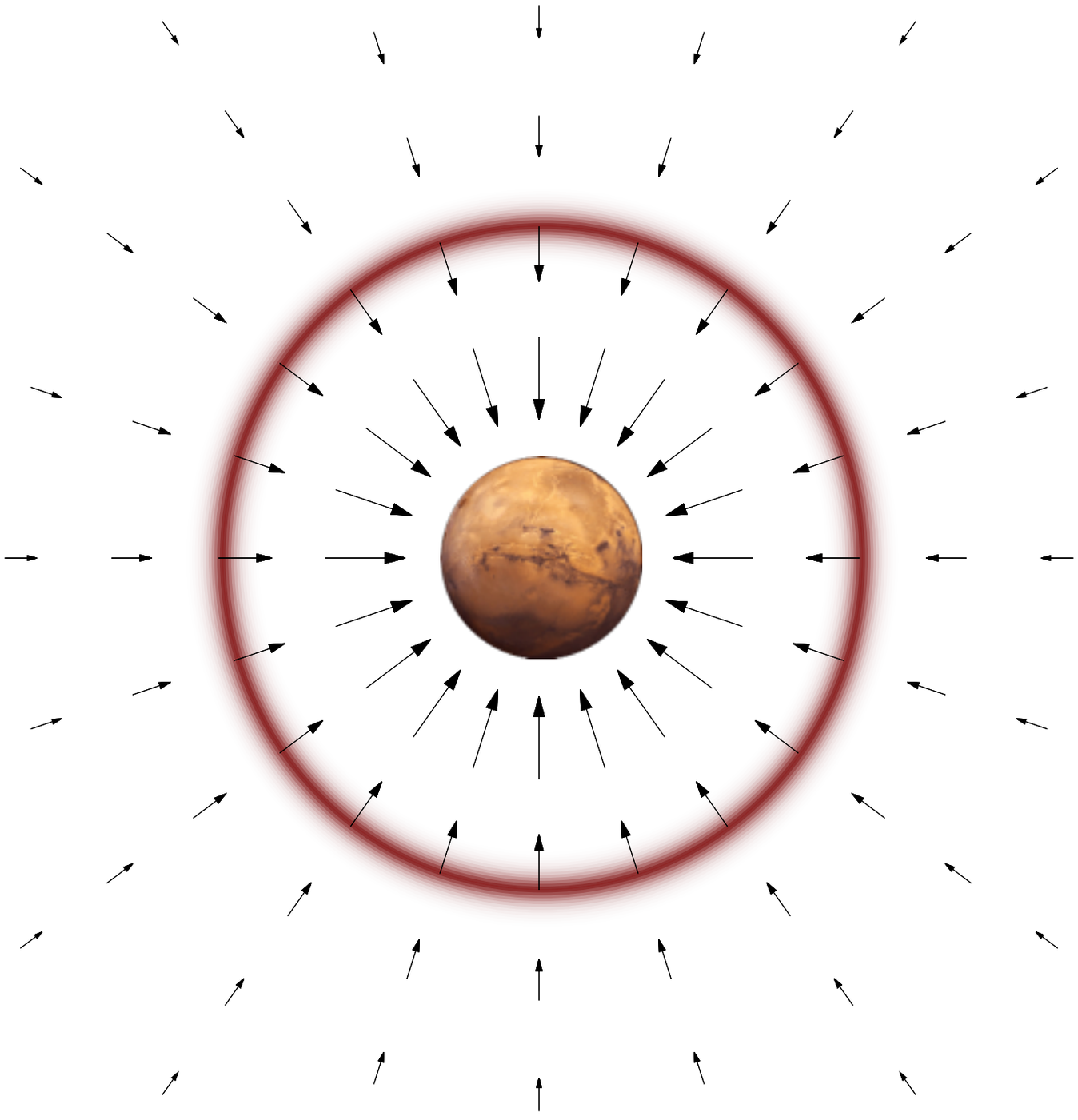


$$\frac{\partial \pi(\theta)}{\partial \theta}$$

Differential geometry informs this transformation, although a physical analogy can be more intuitive.



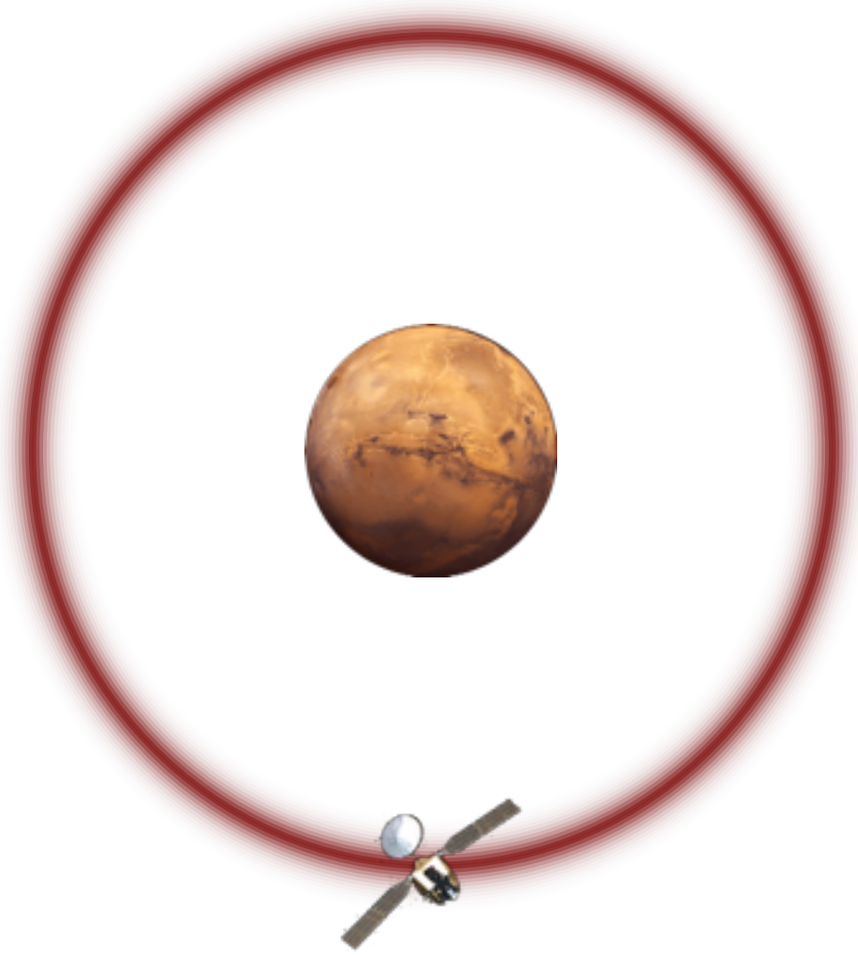
Differential geometry informs this transformation, although a physical analogy can be more intuitive.



We need to add *momentum* in just the right way.
Too little and we still crash into the planet.

We need to add *momentum* in just the right way.
Too little and we still crash into the planet.

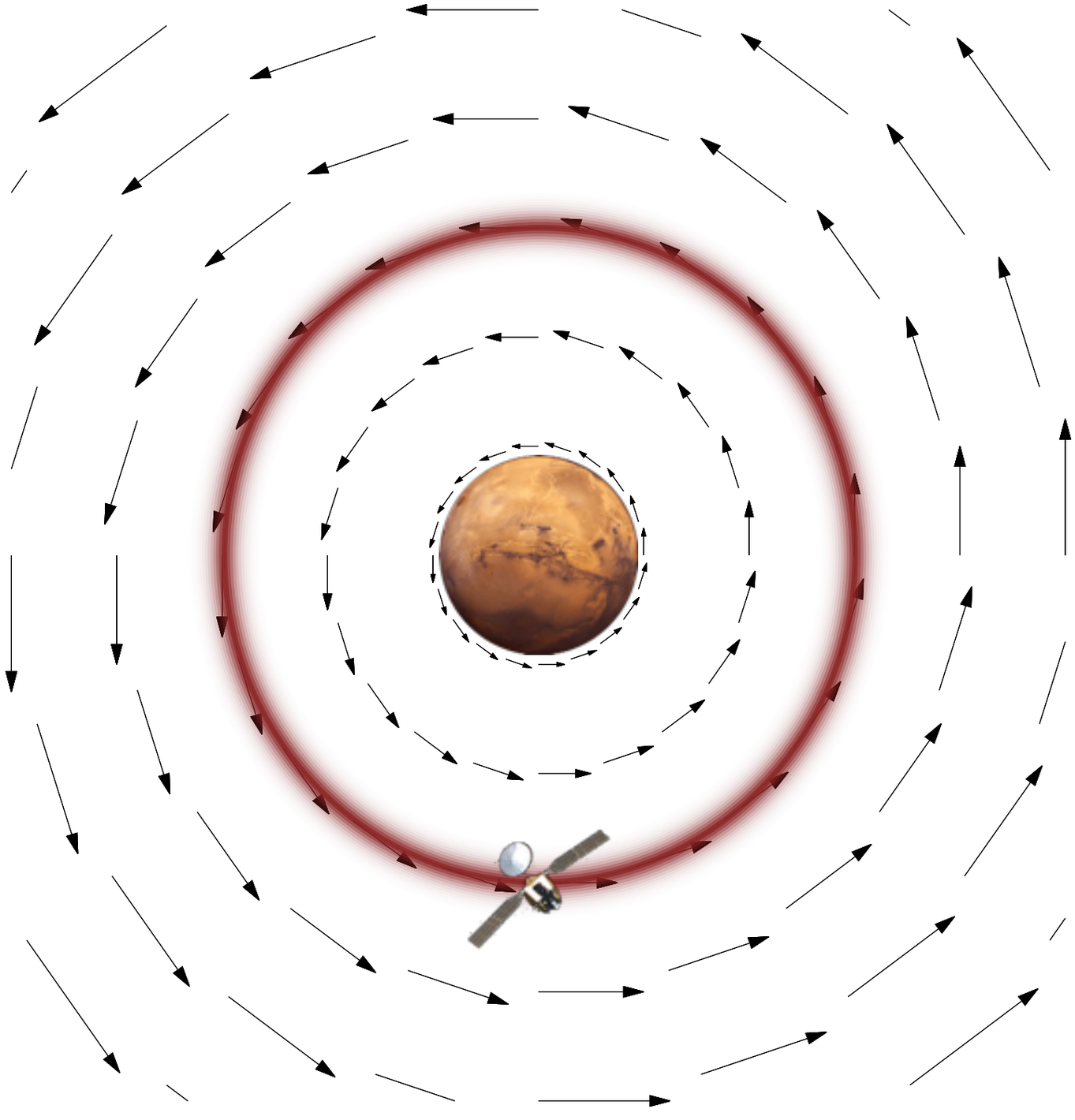
Too much and we fly off to infinity.



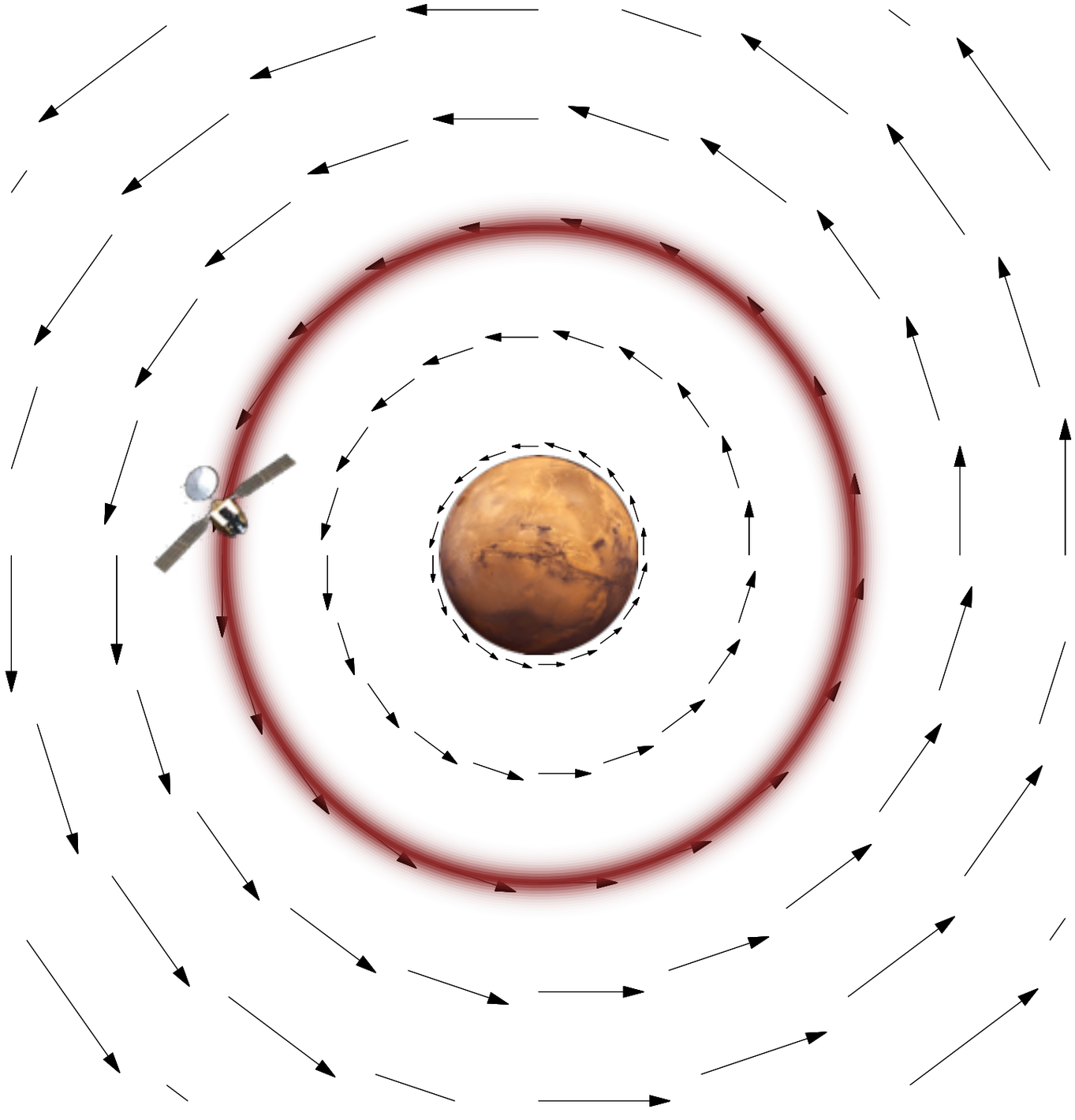
Too much and we fly off to infinity.

Just enough, however, aligns the gradients with the typical set and yields the desired orbital trajectory.

Just enough, however, aligns the gradients with the typical set and yields the desired orbital trajectory.



Just enough, however, aligns the gradients with the typical set and yields the desired orbital trajectory.



Hamiltonian Monte Carlo is the unique procedure for adding momenta to yield coherent exploration.

$$\theta \rightarrow (\theta, p)$$

Hamiltonian Monte Carlo is the unique procedure for adding momenta to yield coherent exploration.

$$\theta \rightarrow (\theta, p)$$

$$\pi(\theta) \rightarrow \pi(\theta, p)$$

Hamiltonian Monte Carlo is the unique procedure for adding momenta to yield coherent exploration.

$$\theta \to (\theta, p)$$

$$\pi(\theta) \to \pi(\theta, p) = \pi(p|\theta)\,\pi(\theta)$$

The expanded target distribution defines a *Hamiltonian* that decomposes into a *potential energy* and *kinetic energy*.

$$\pi(p|\theta)\,\pi(\theta)$$

The expanded target distribution defines a *Hamiltonian* that decomposes into a *potential energy* and *kinetic energy*.

$$H(\theta, p) = -\log \pi(p|\theta)\, \pi(\theta)$$

The expanded target distribution defines a *Hamiltonian* that decomposes into a *potential energy* and *kinetic energy*.

$$H(\theta, p) = -\log \pi(p|\theta)\, \pi(\theta)$$

$$= -\log \pi(p|\theta) - \log \pi(\theta)$$

The expanded target distribution defines a *Hamiltonian* that decomposes into a *potential energy* and *kinetic energy*.

$$H(\theta, p) = -\log \pi(p|\theta)\, \pi(\theta)$$

$$= -\log \pi(p|\theta) - \log \pi(\theta)$$

$$T$$

The expanded target distribution defines a *Hamiltonian* that decomposes into a *potential energy* and *kinetic energy*.

$$H(\theta, p) = -\log \pi(p|\theta)\, \pi(\theta)$$

$$= \underbrace{-\log \pi(p|\theta)}_{T} \underbrace{- \log \pi(\theta)}_{V}$$

Any choice of kinetic energy generates coherent exploration through the expanded system.

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = \frac{\partial T}{\partial p}$$

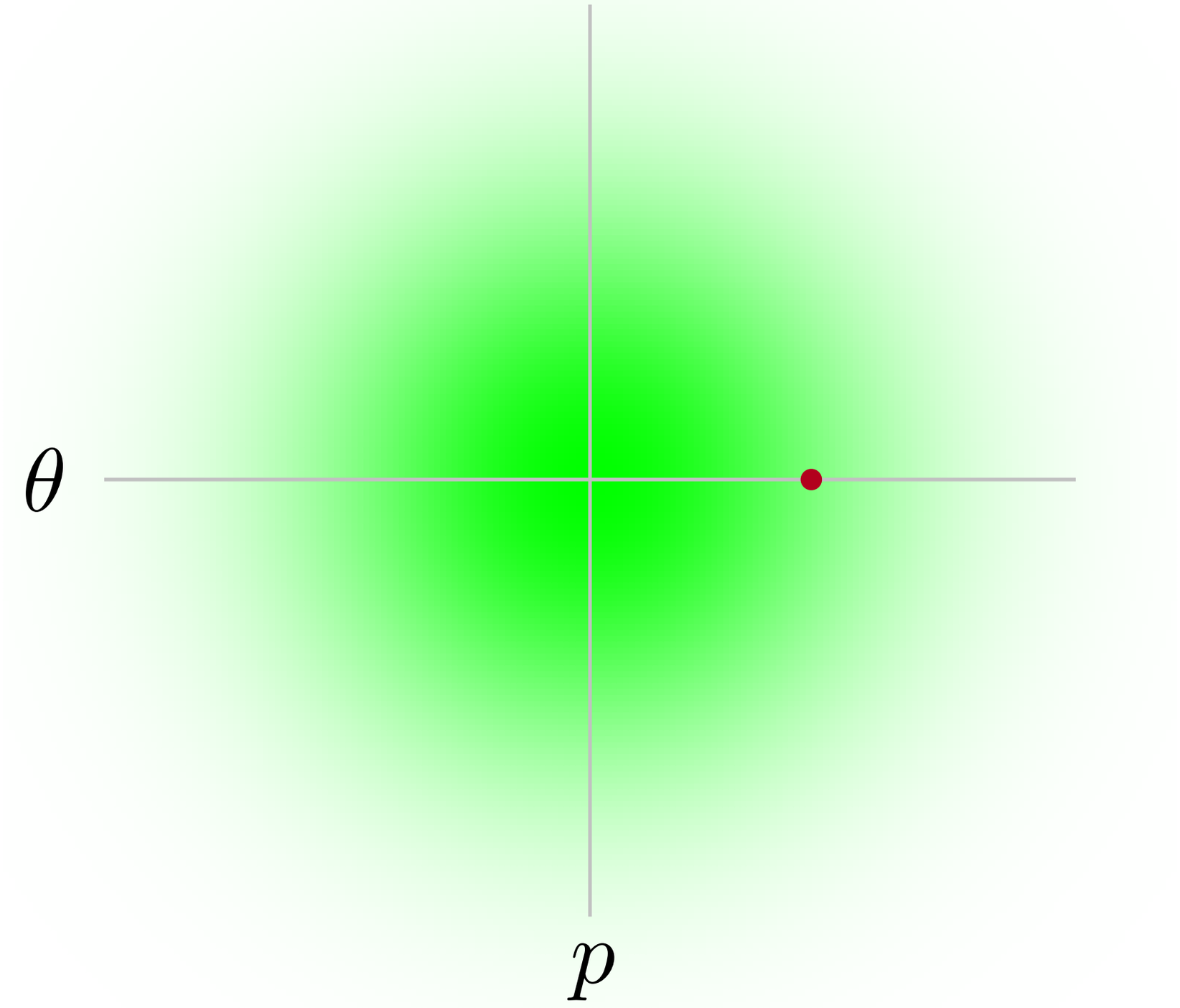
$$\frac{\mathrm{d}p}{\mathrm{d}t} = -\frac{\partial T}{\partial \theta} - \frac{\partial V}{\partial \theta}$$

Any choice of kinetic energy generates coherent exploration through the expanded system.

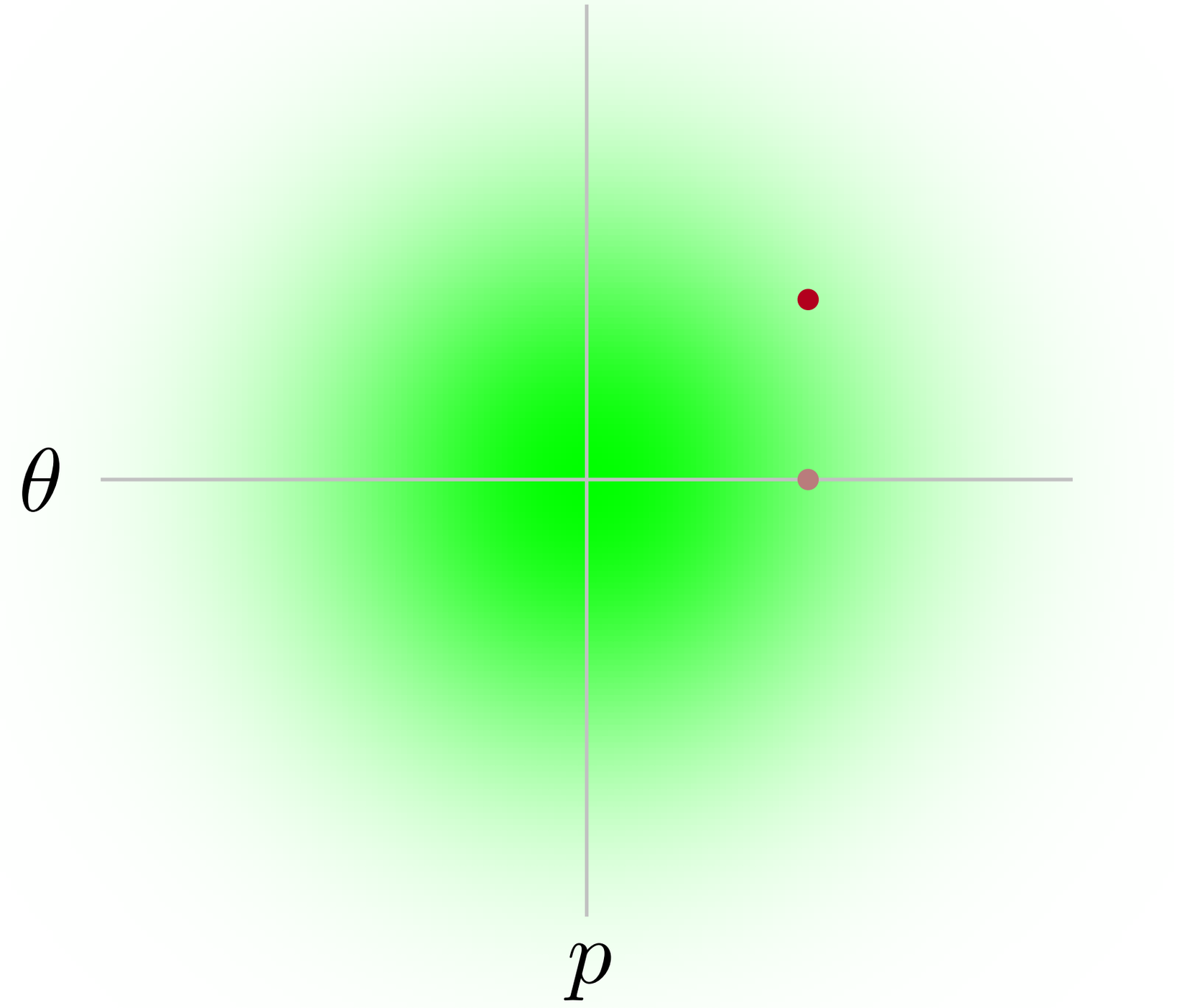
$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = \frac{\partial T}{\partial p}$$

$$\frac{\mathrm{d}p}{\mathrm{d}t} = -\frac{\partial T}{\partial \theta} - \frac{\partial V}{\partial \theta}$$

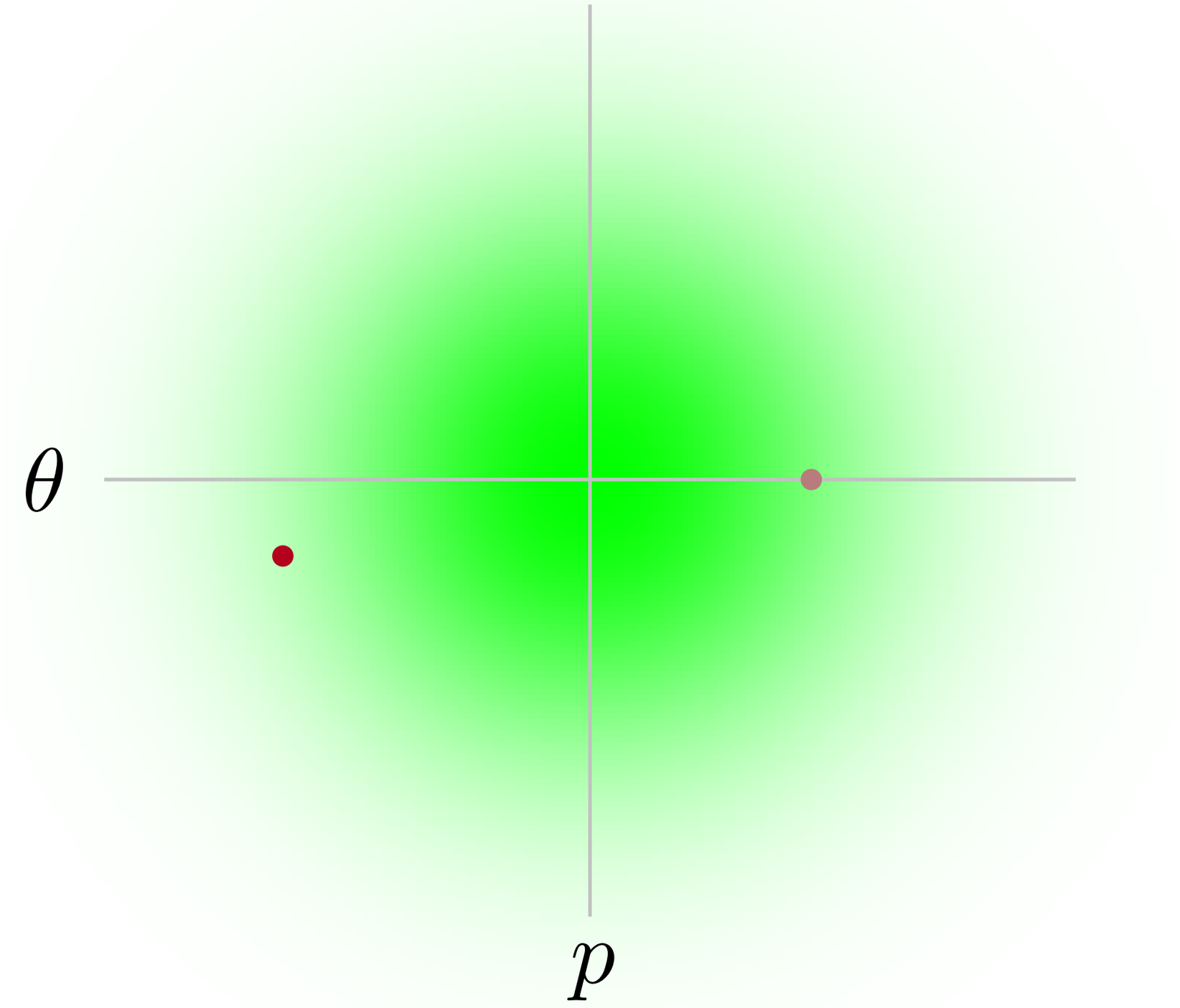
These trajectories then allow us to define a Markov transition on our original parameter space.



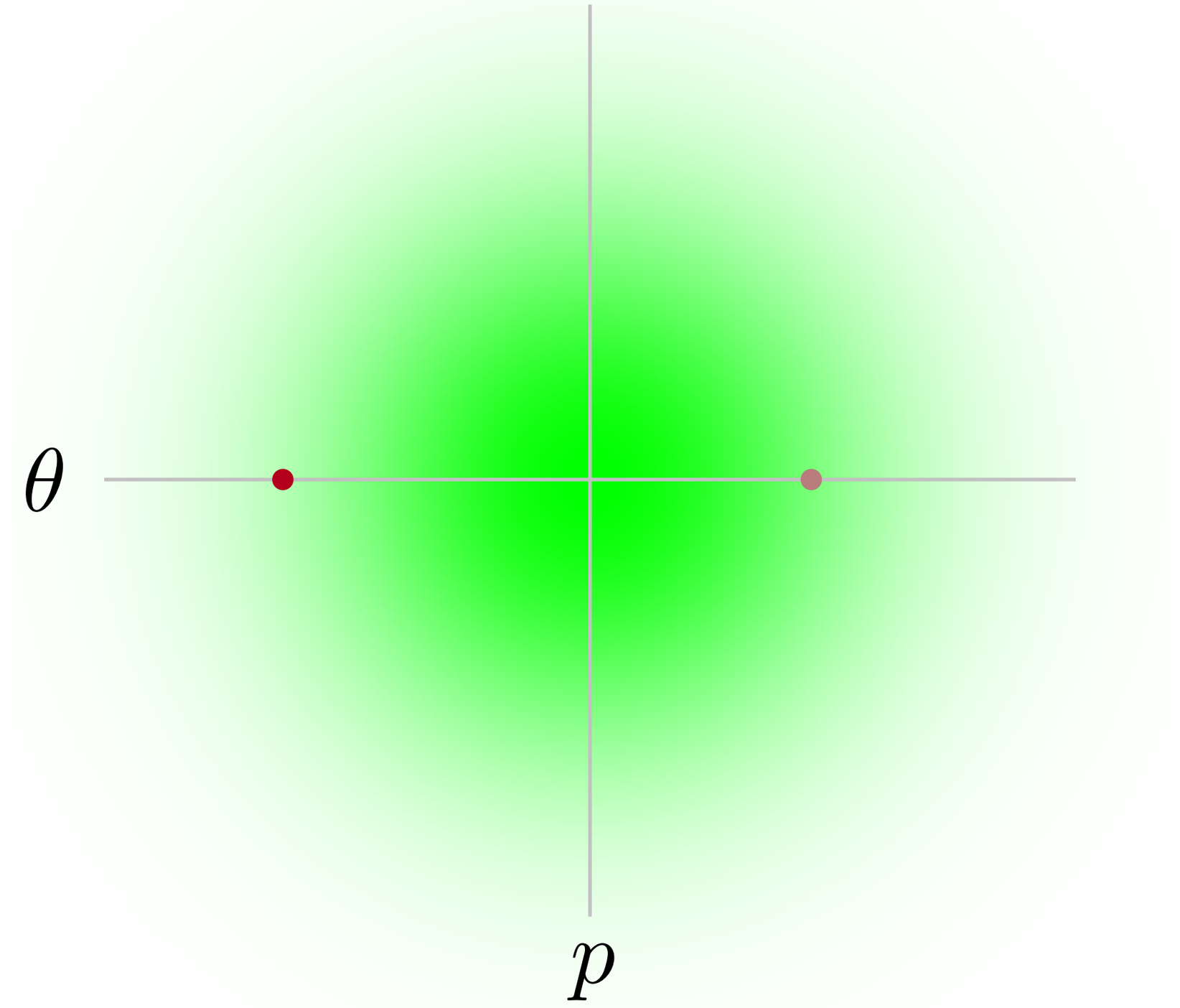
These trajectories then allow us to define a Markov transition on our original parameter space.



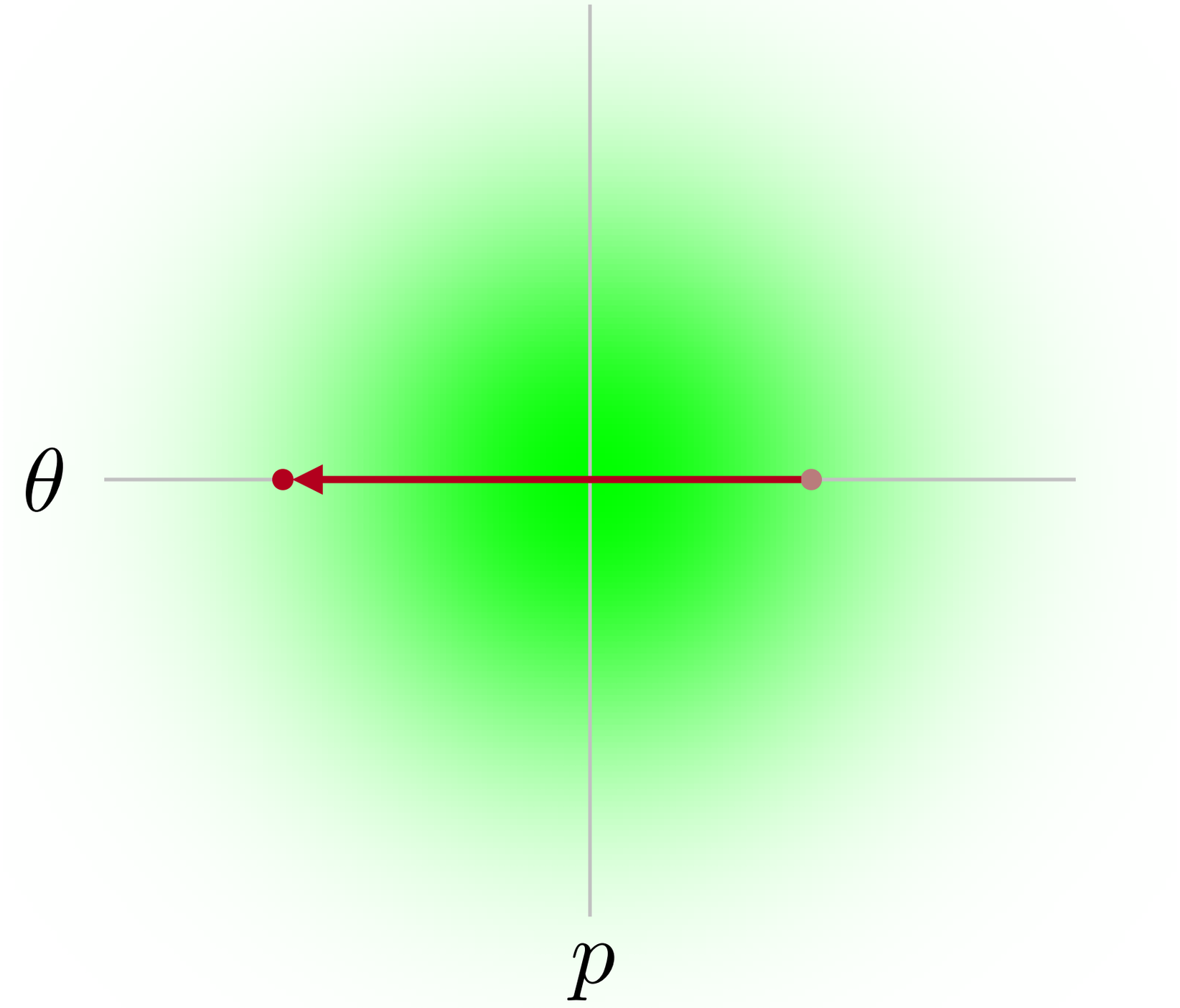
These trajectories then allow us to define a Markov transition on our original parameter space.



These trajectories then allow us to define a Markov transition on our original parameter space.



These trajectories then allow us to define a Markov transition on our original parameter space.



In practice we have to use numerical integrators to approximate the true Hamiltonian trajectories.

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = \frac{\partial T}{\partial p}$$

$$\frac{\mathrm{d}p}{\mathrm{d}t} = -\frac{\partial T}{\partial \theta} - \frac{\partial V}{\partial \theta}$$

In practice we have to use numerical integrators to approximate the true Hamiltonian trajectories.

$$\theta \rightarrow \theta + \epsilon \frac{\partial T}{\partial p}$$

$$p \rightarrow p - \epsilon \left(\frac{\partial T}{\partial \theta} + \frac{\partial V}{\partial \theta} \right)$$

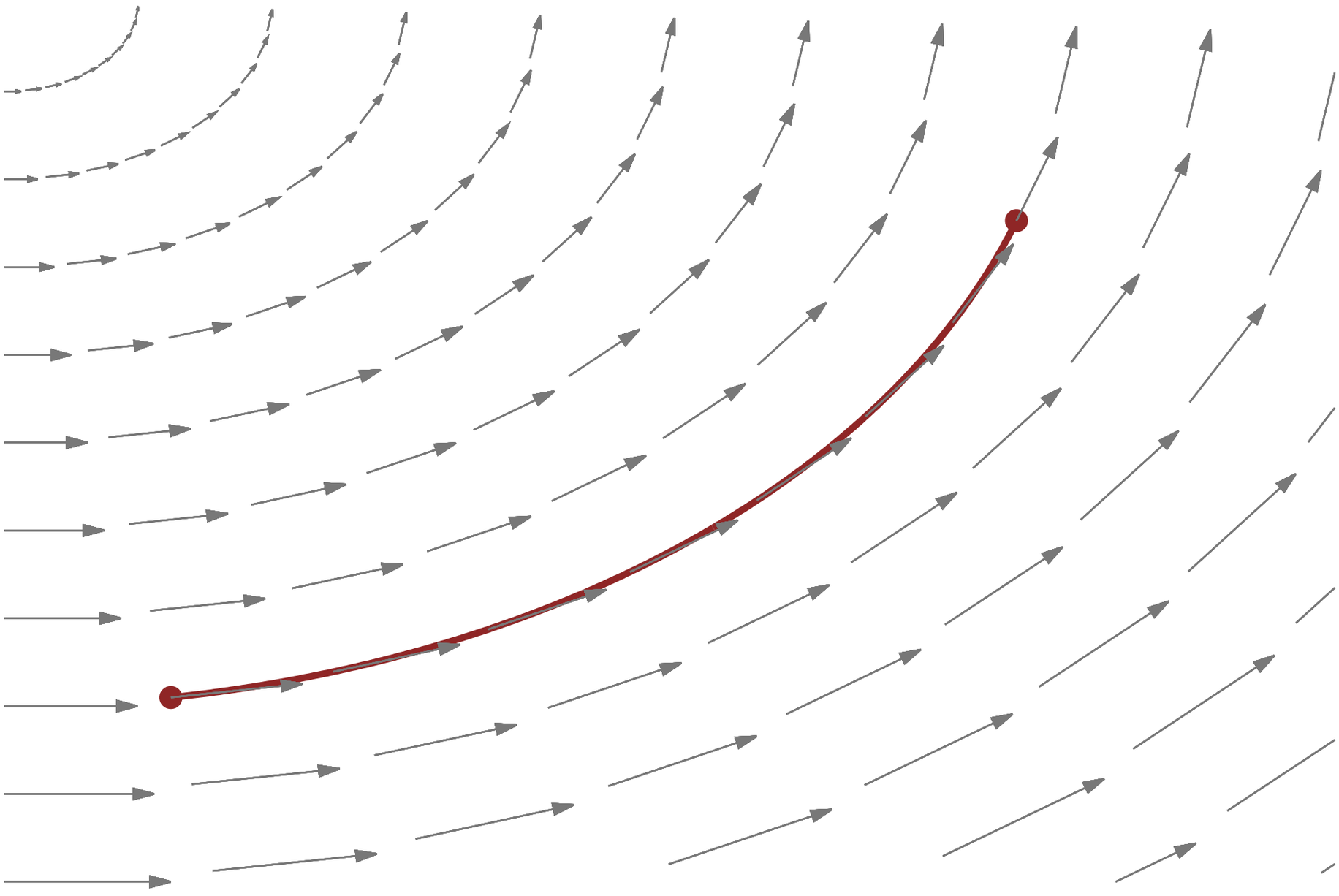
In practice we have to use numerical integrators to approximate the true Hamiltonian trajectories.

$$\theta \to \theta + \epsilon \frac{\partial T}{\partial p}$$

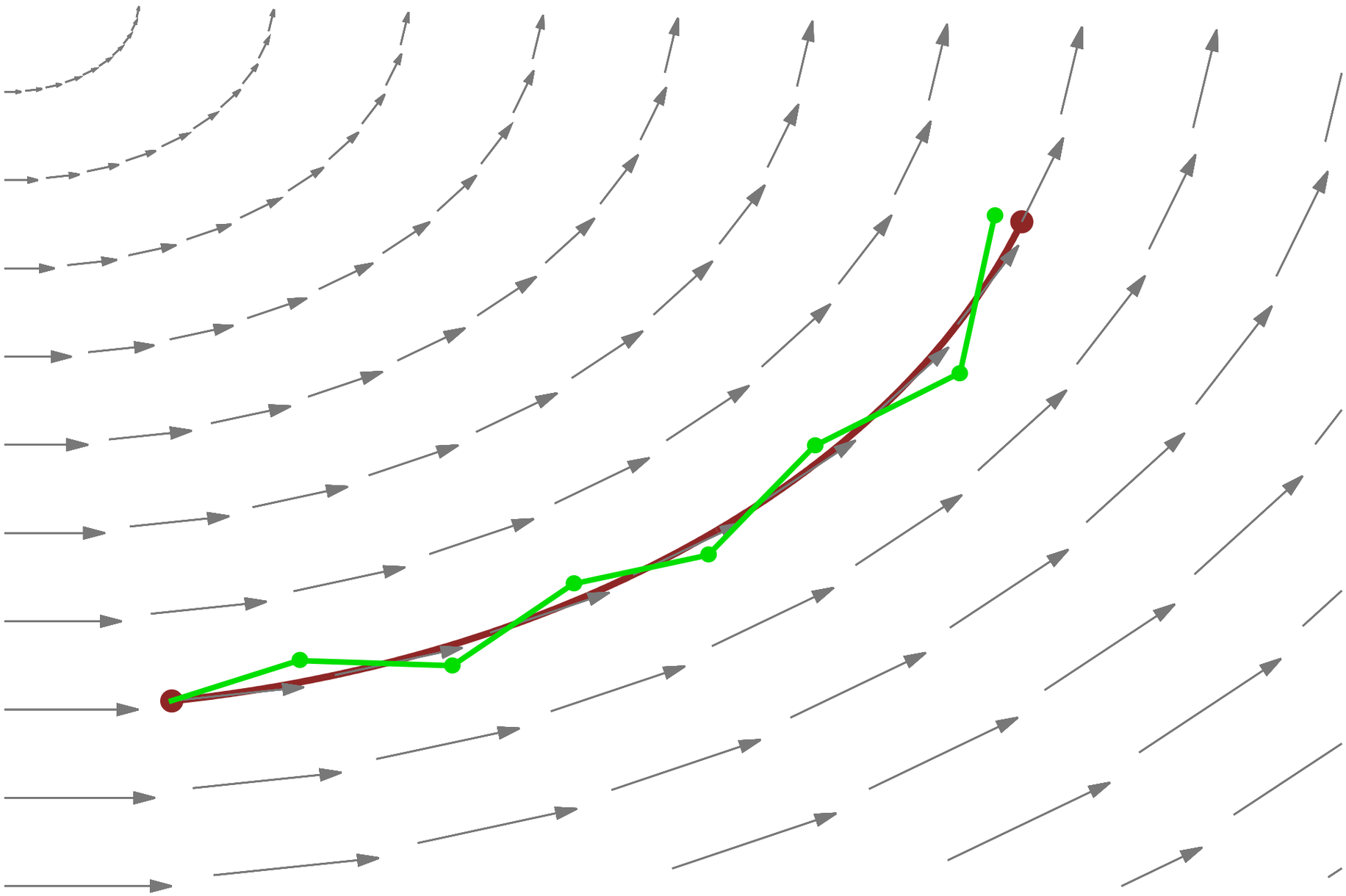
$$p \to p - \epsilon \left(\frac{\partial T}{\partial \theta} + \frac{\partial V}{\partial \theta} \right)$$

$$\pi(\text{accept}) = \min\left(1, \frac{\pi(\theta', p')}{\pi(\theta, p)}\right)$$

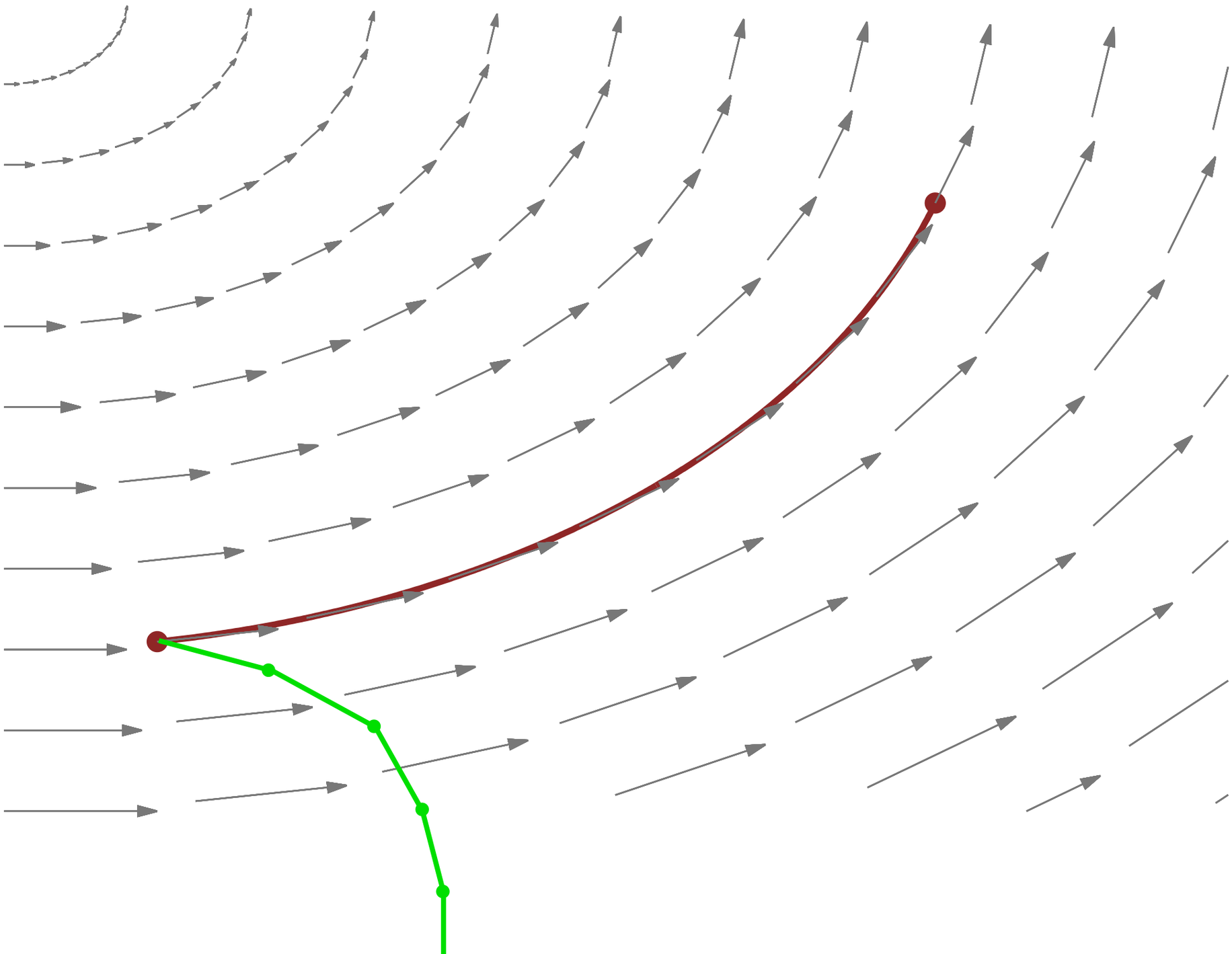
Symplectic integrators very accurately simulate these Hamiltonian systems, even in high dimensions.



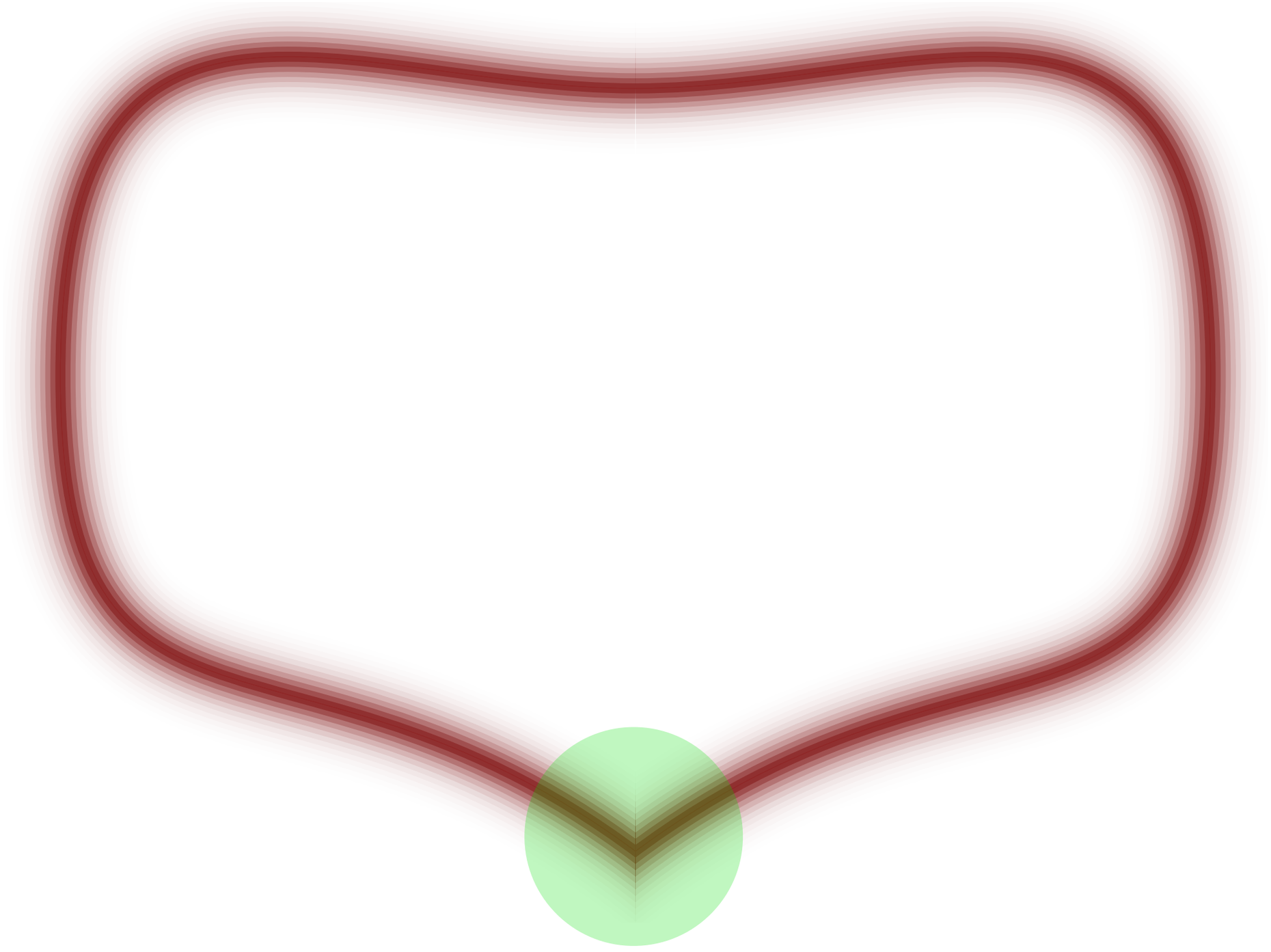
Symplectic integrators very accurately simulate these Hamiltonian systems, even in high dimensions.



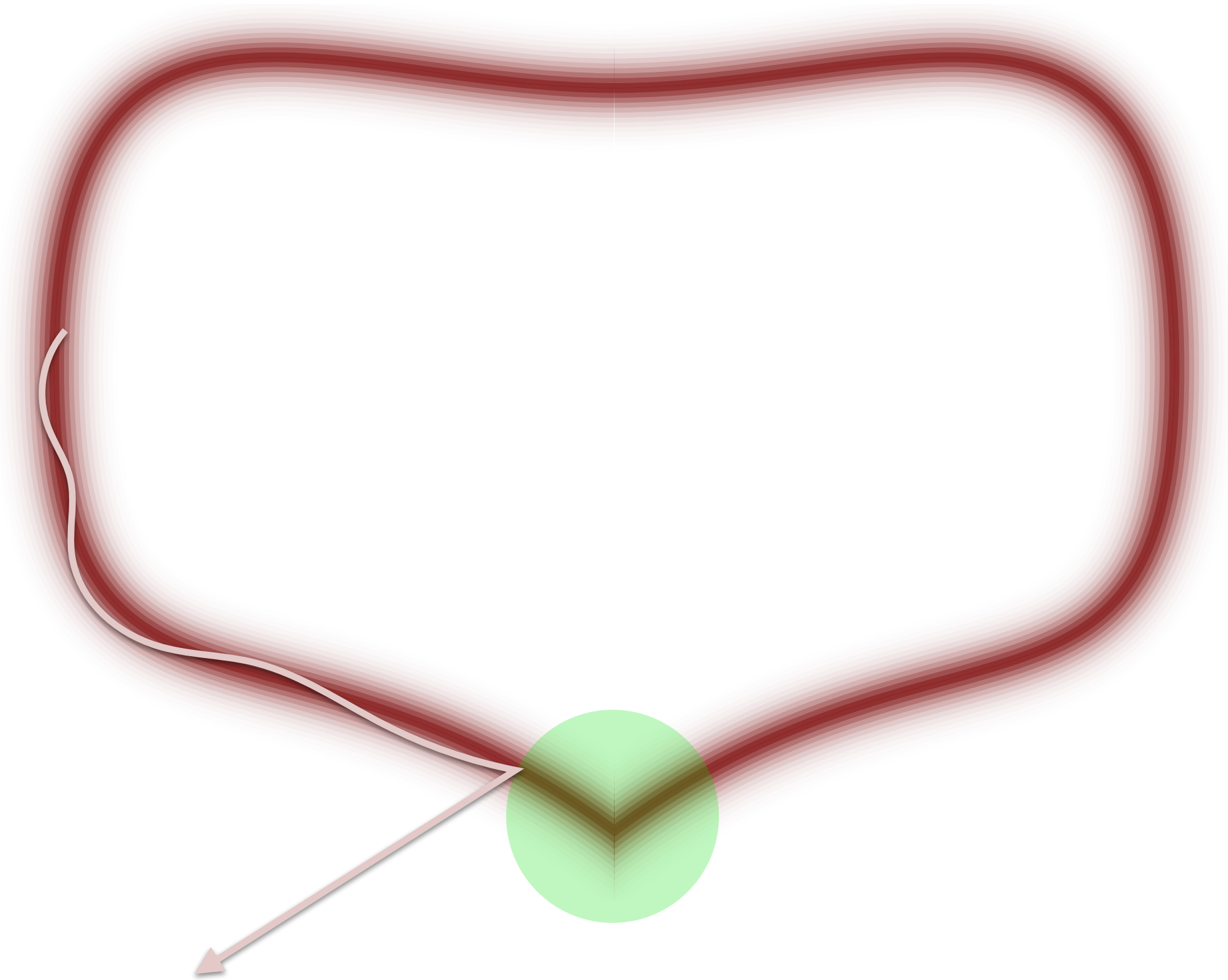
Although symplectic integrators are mostly well-behaved, sometimes they can *diverge*.



Fruitfully for us, these divergences tend to be caused by the same pathologies that can bias our MCMC estimates.



Fruitfully for us, these divergences tend to be caused by the same pathologies that can bias our MCMC estimates.



Software tools like Stan implement MCMC, allowing users to focus on building models and analyzing chains.