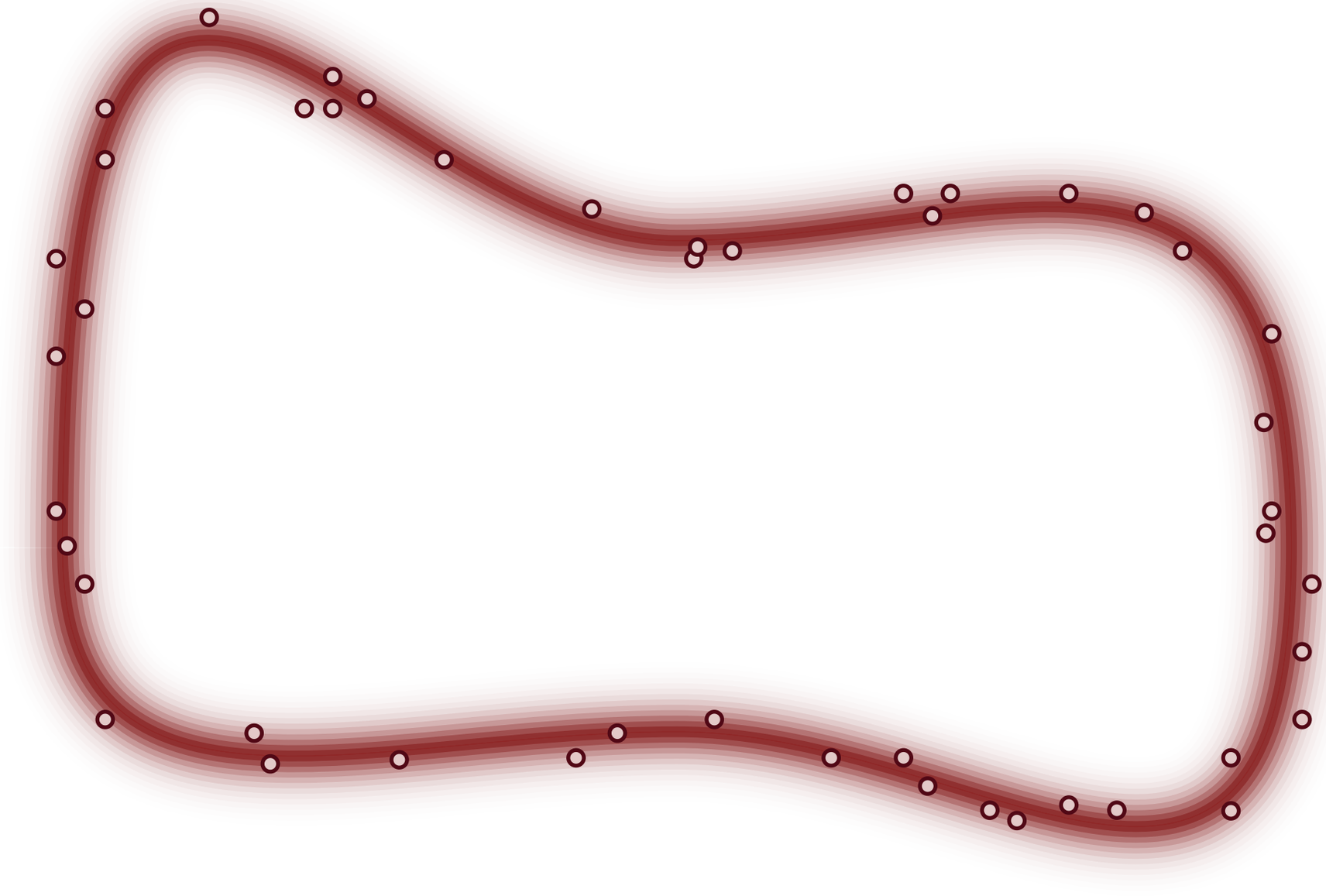
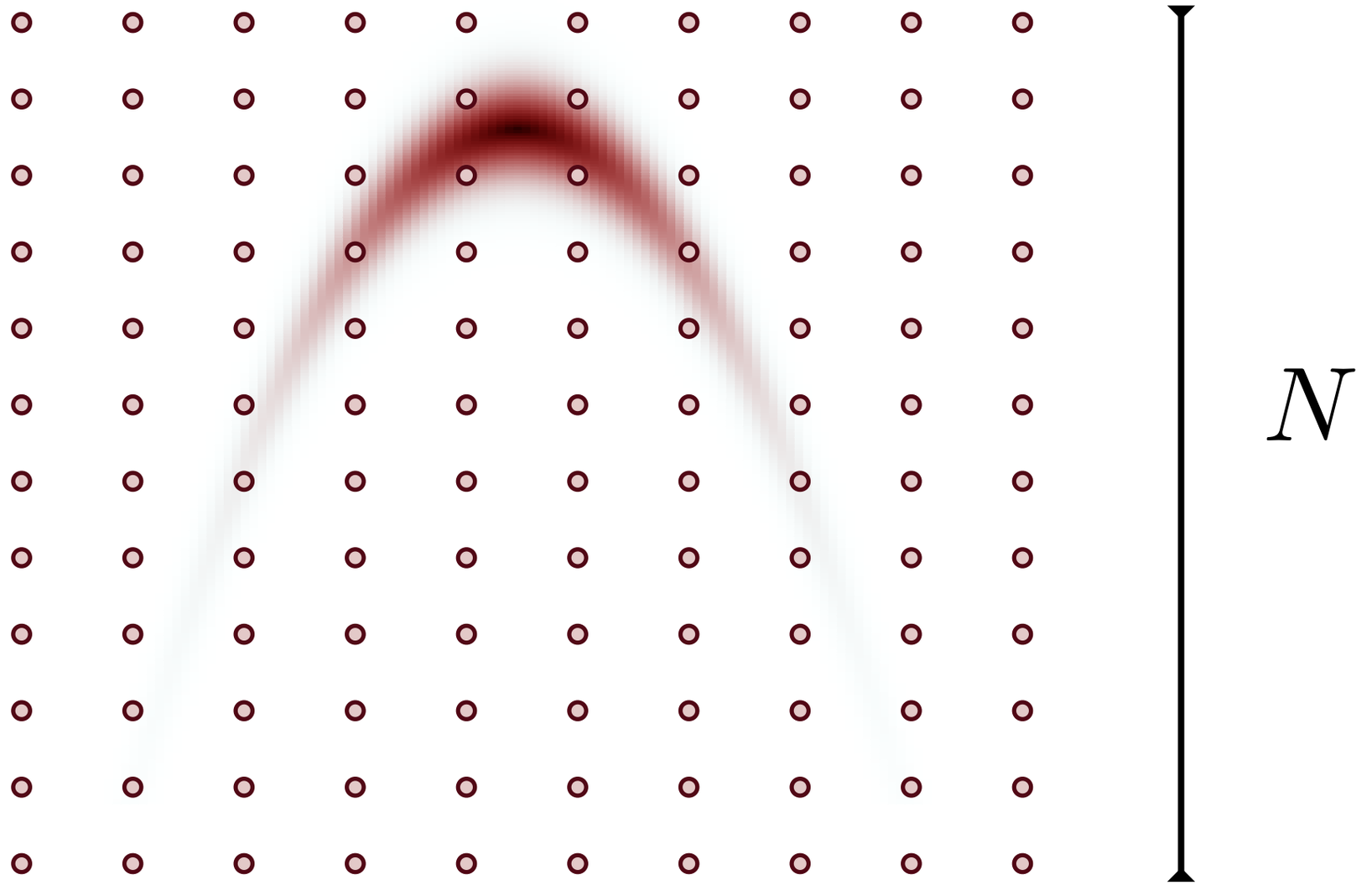
Bayesian Computation and Markov chain Monte Carlo



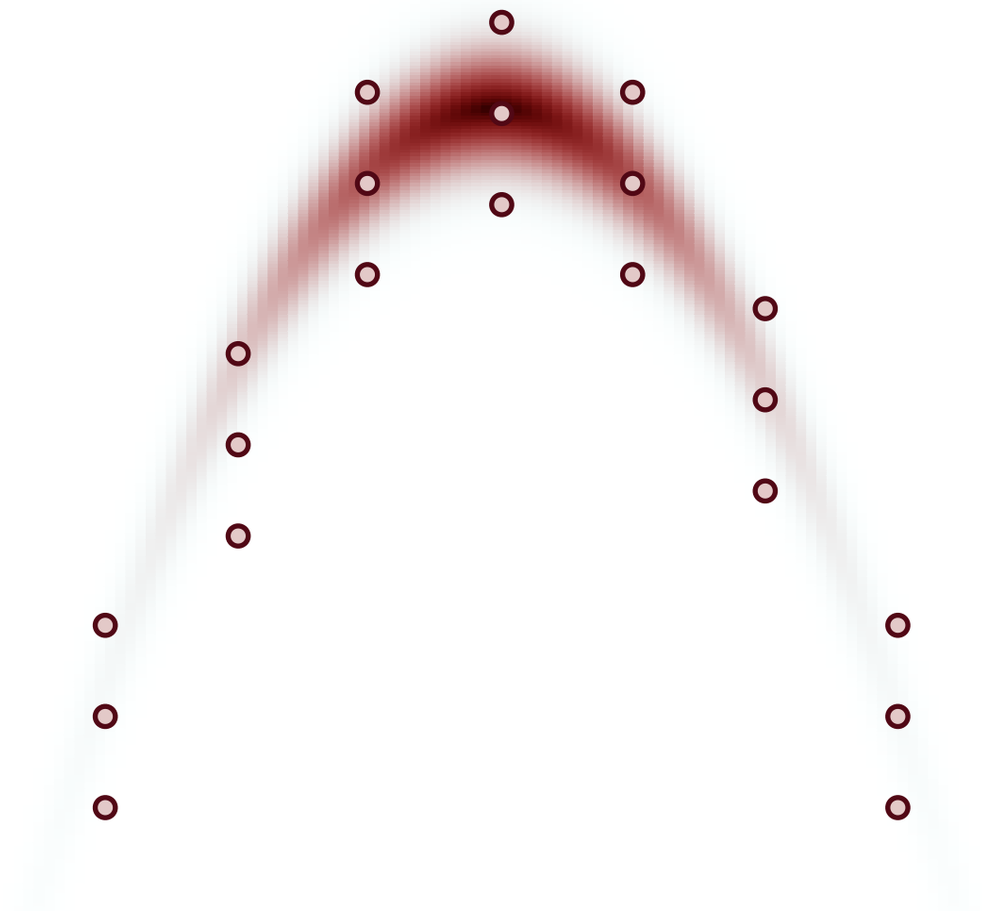
Once we've built a model, Bayesian computation reduces to evaluating expectations, or integrals.

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

Unfortunately, the cost of naive numerical integration scales exponentially with the dimension of the posterior.



To be efficient we need to focus computation on the relevant neighborhoods of parameter space.



But exactly which neighborhoods end up contributing most to arbitrary expectations?

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

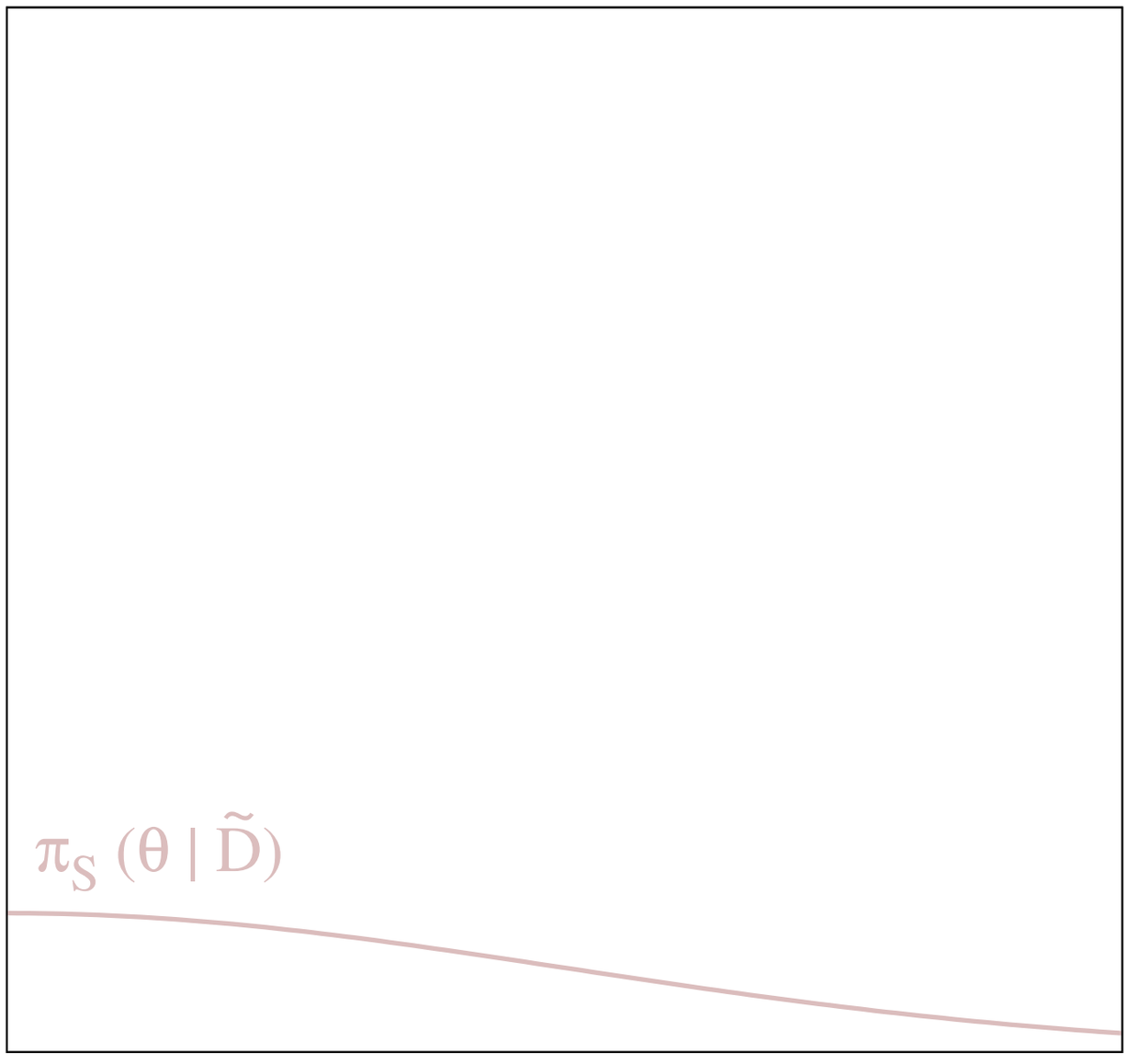
But exactly which neighborhoods end up contributing most to arbitrary expectations?

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

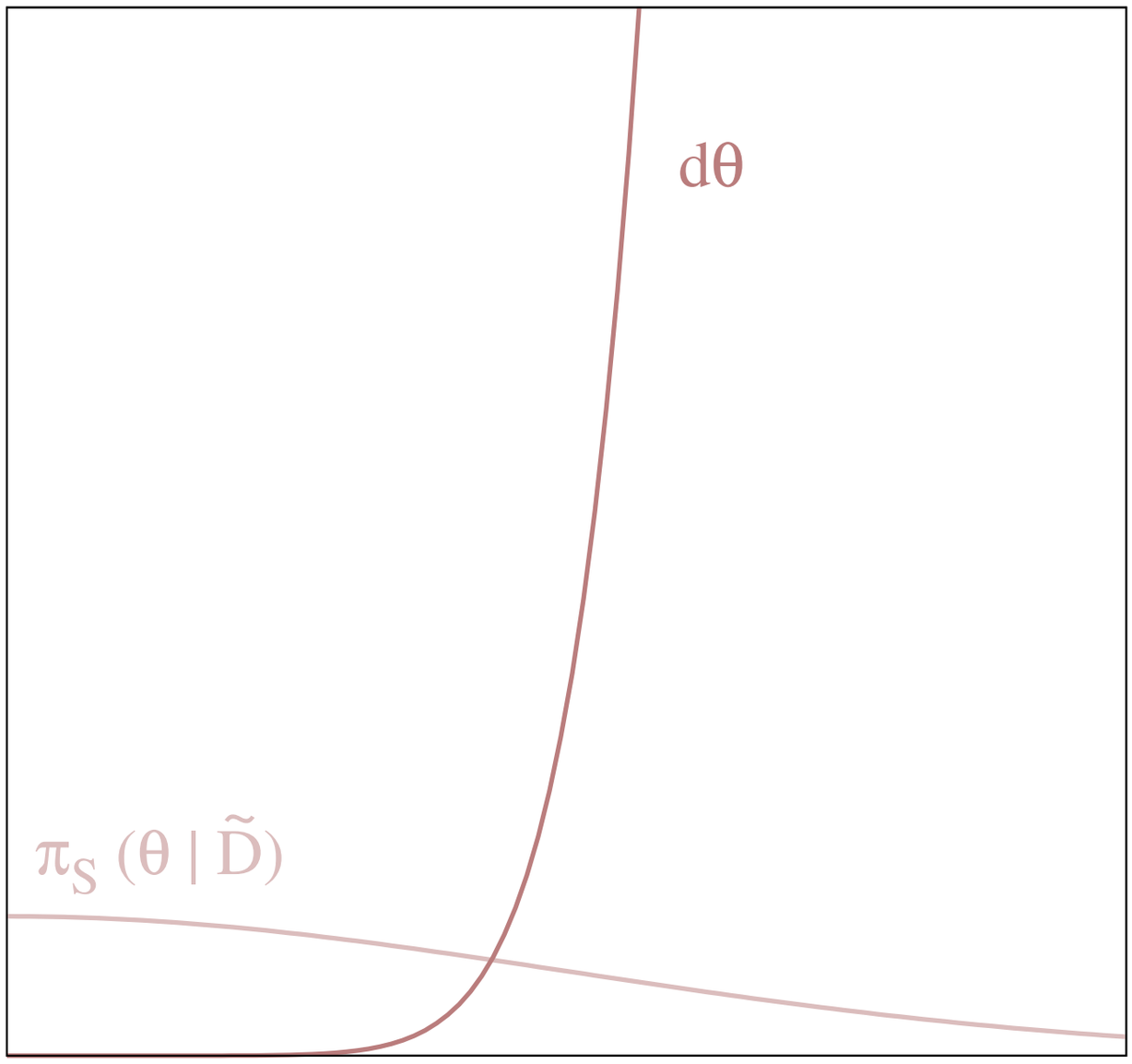
But exactly which neighborhoods end up contributing most to arbitrary expectations?

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

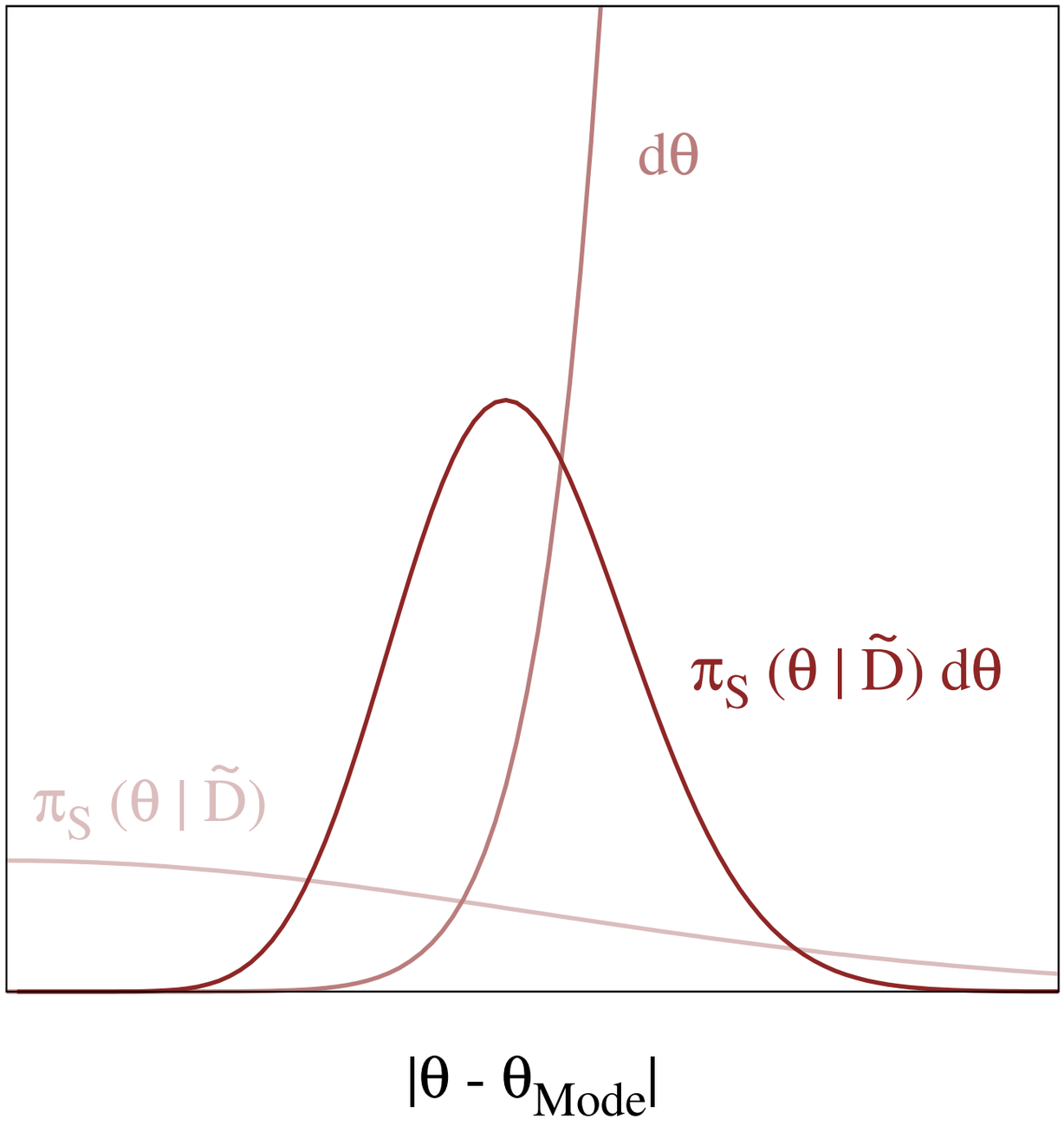
Relevant neighborhoods, however, are defined not by probability density but rather by probability mass.



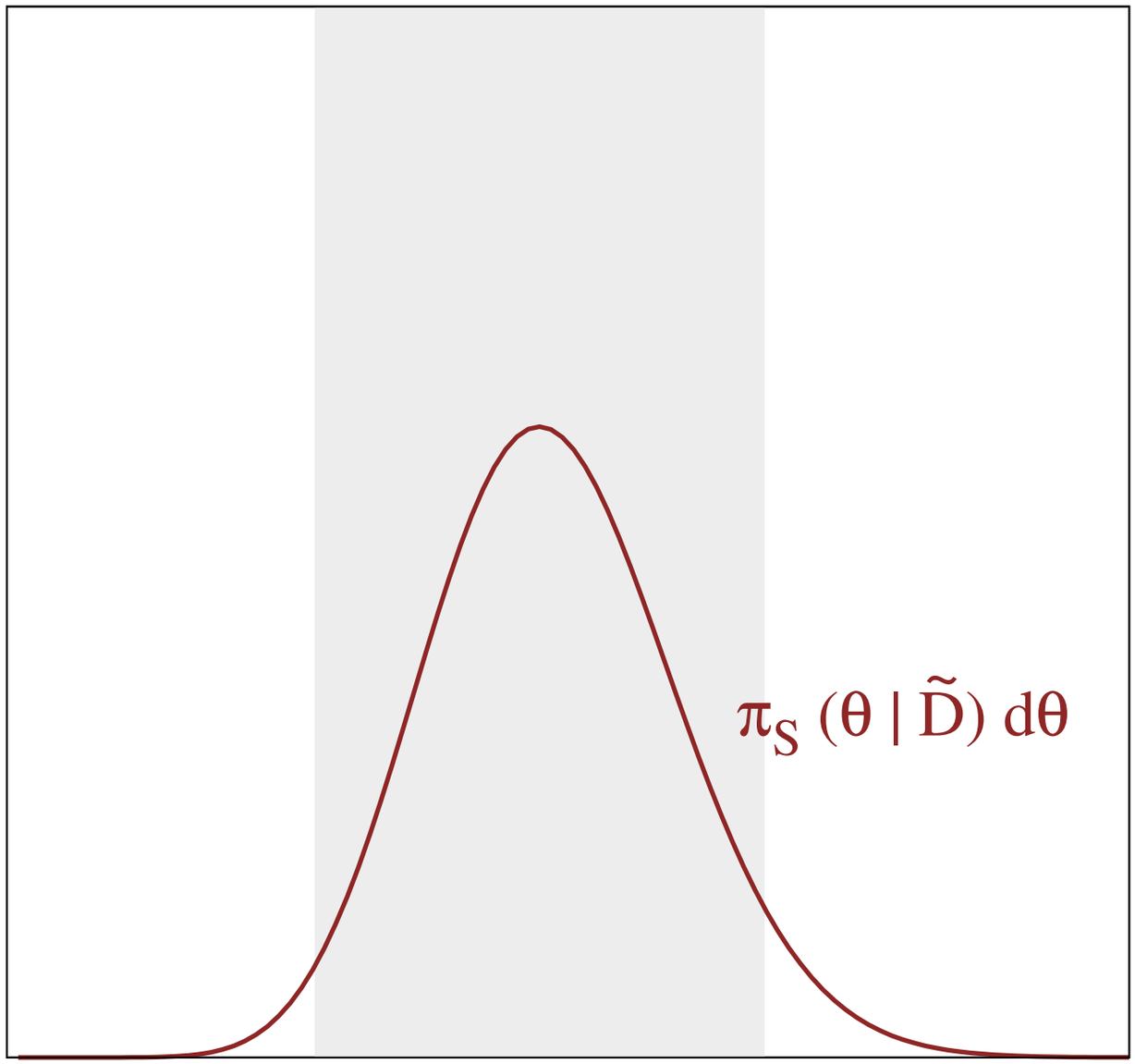
Relevant neighborhoods, however, are defined not by probability density but rather by probability mass.



Relevant neighborhoods, however, are defined not by probability density but rather by probability mass.



Probability mass concentrates on a hypersurface called the *typical set* that surrounds the mode.



This *concentration of measure* into a narrow typical set frustrates the accurate estimation of integrals.

To accurately estimate expectations we need a method for numerically finding and then exploring the typical set.

Deterministic

Modal Estimators

Laplace Estimators

Variational Estimators

...

Stochastic

Rejection Sampling

Importance Sampling

Markov Chain Monte Carlo

...

To accurately estimate expectations we need a method for numerically finding and then exploring the typical set.

Deterministic

Modal Estimators

Laplace Estimators

Variational Estimators

...

Stochastic

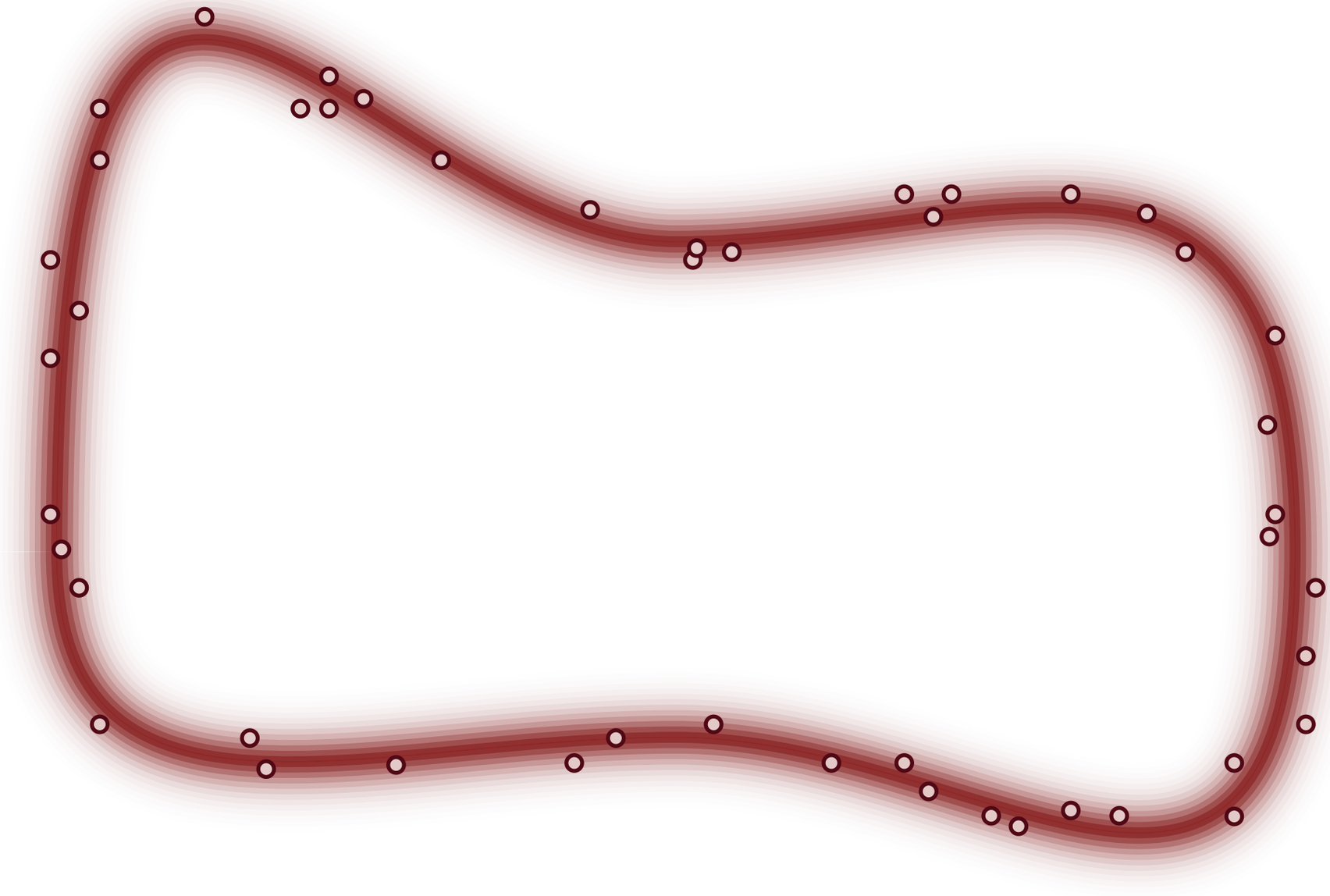
Rejection Sampling

Importance Sampling

Markov Chain Monte Carlo

...

Markov chain Monte Carlo



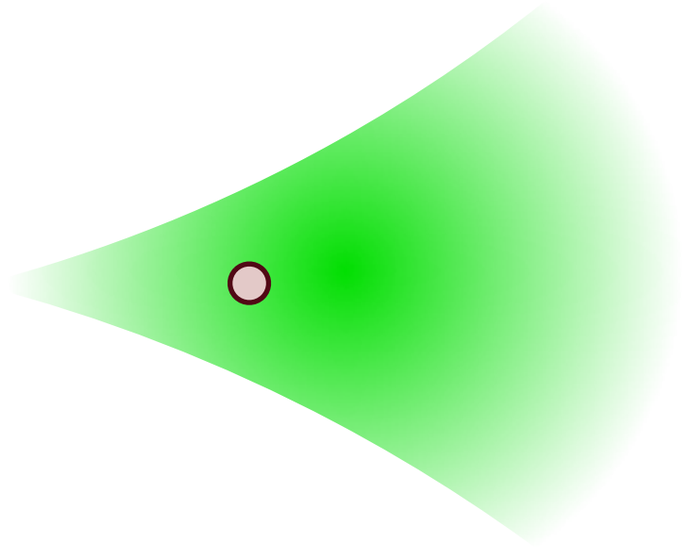
A Markov transition that targets our desired distribution naturally concentrates towards probability mass.

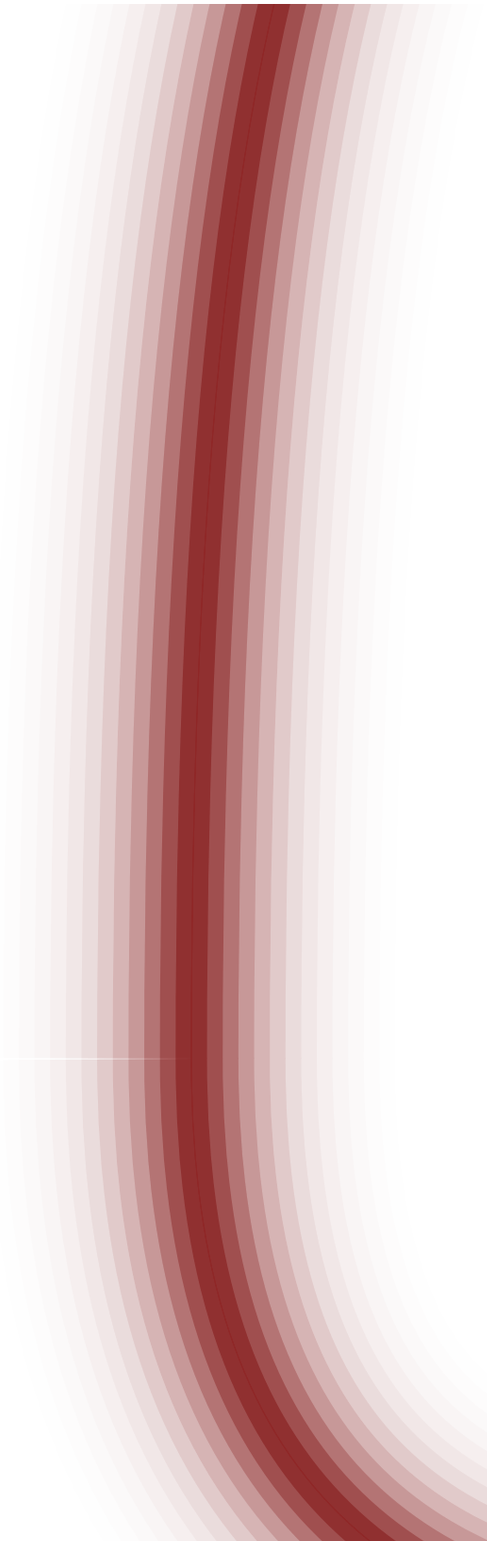
$$T(\theta \mid \theta')$$

A Markov transition that targets our desired distribution naturally concentrates towards probability mass.

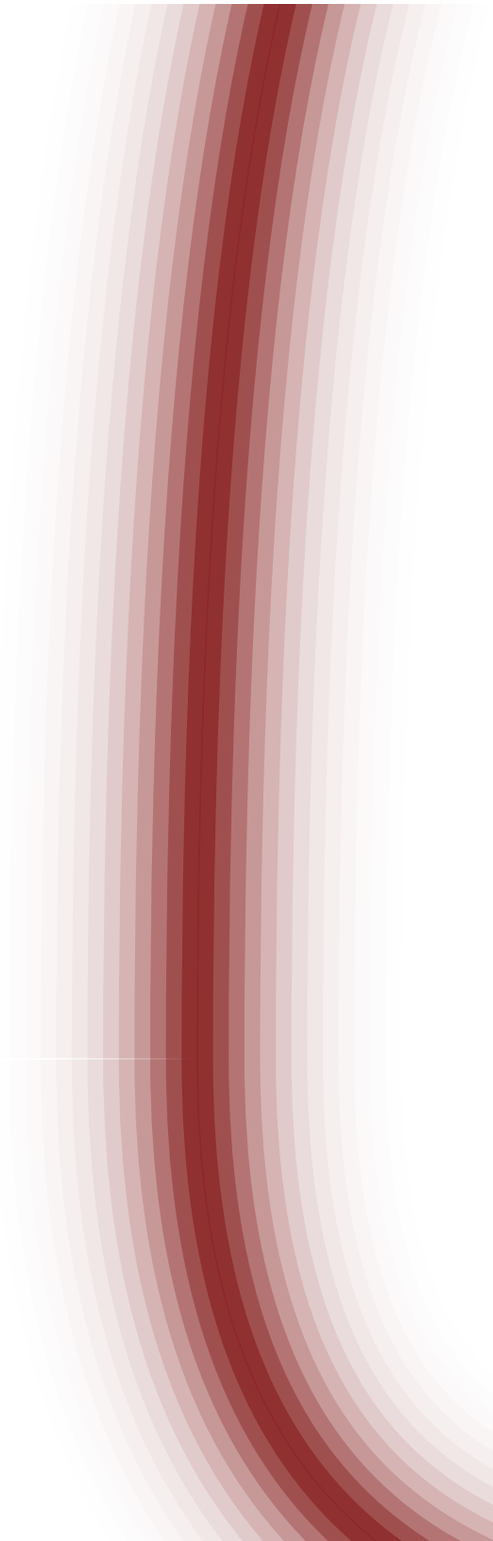
$$\pi_S(\theta \mid \widetilde{\mathcal{D}}) = \int \mathrm{d}\theta' \, T(\theta \mid \theta') \pi_S(\theta' \mid \widetilde{\mathcal{D}})$$

A Markov transition that targets our desired distribution naturally concentrates towards probability mass.

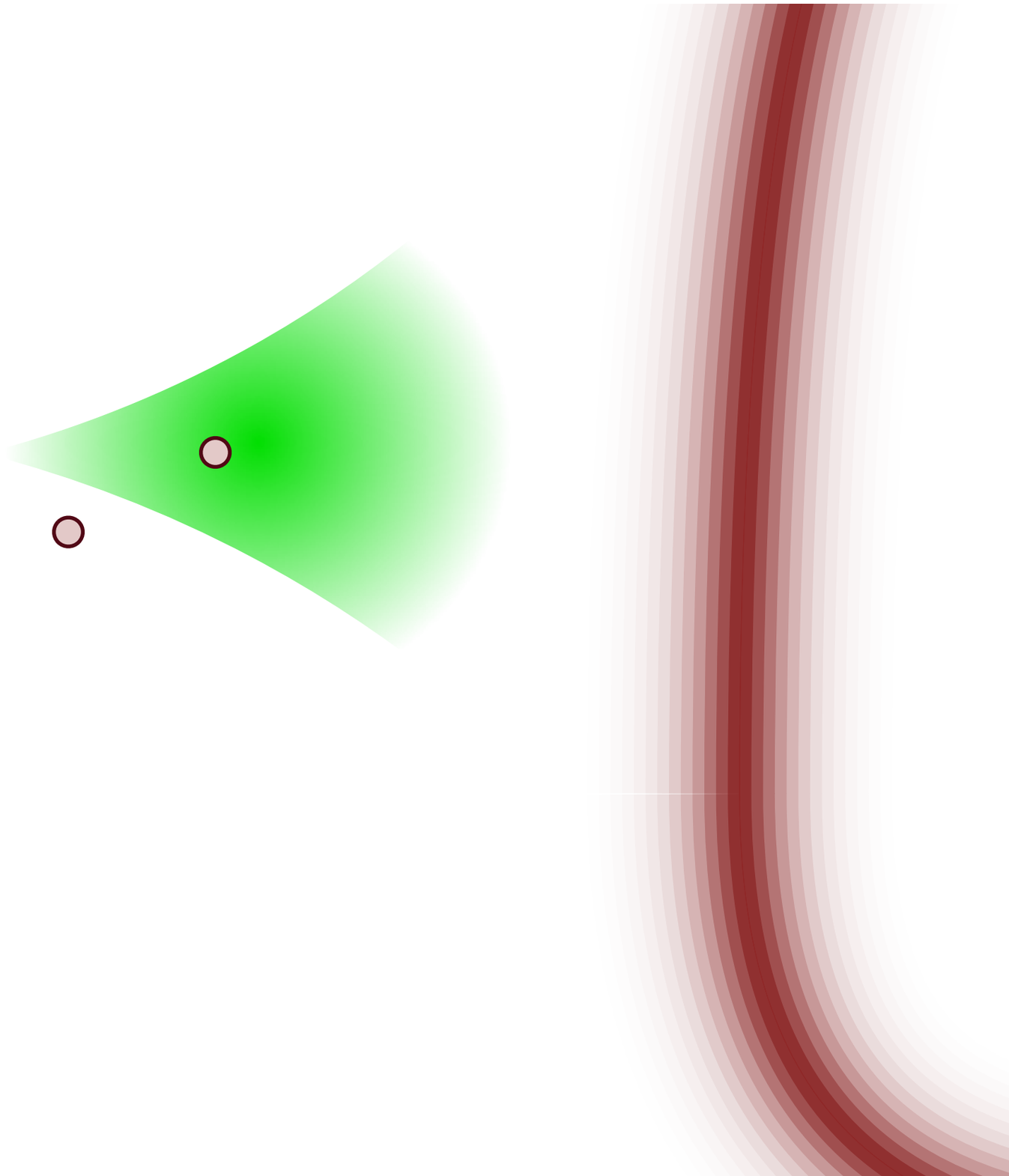




A Markov transition that targets our desired distribution naturally concentrates towards probability mass.

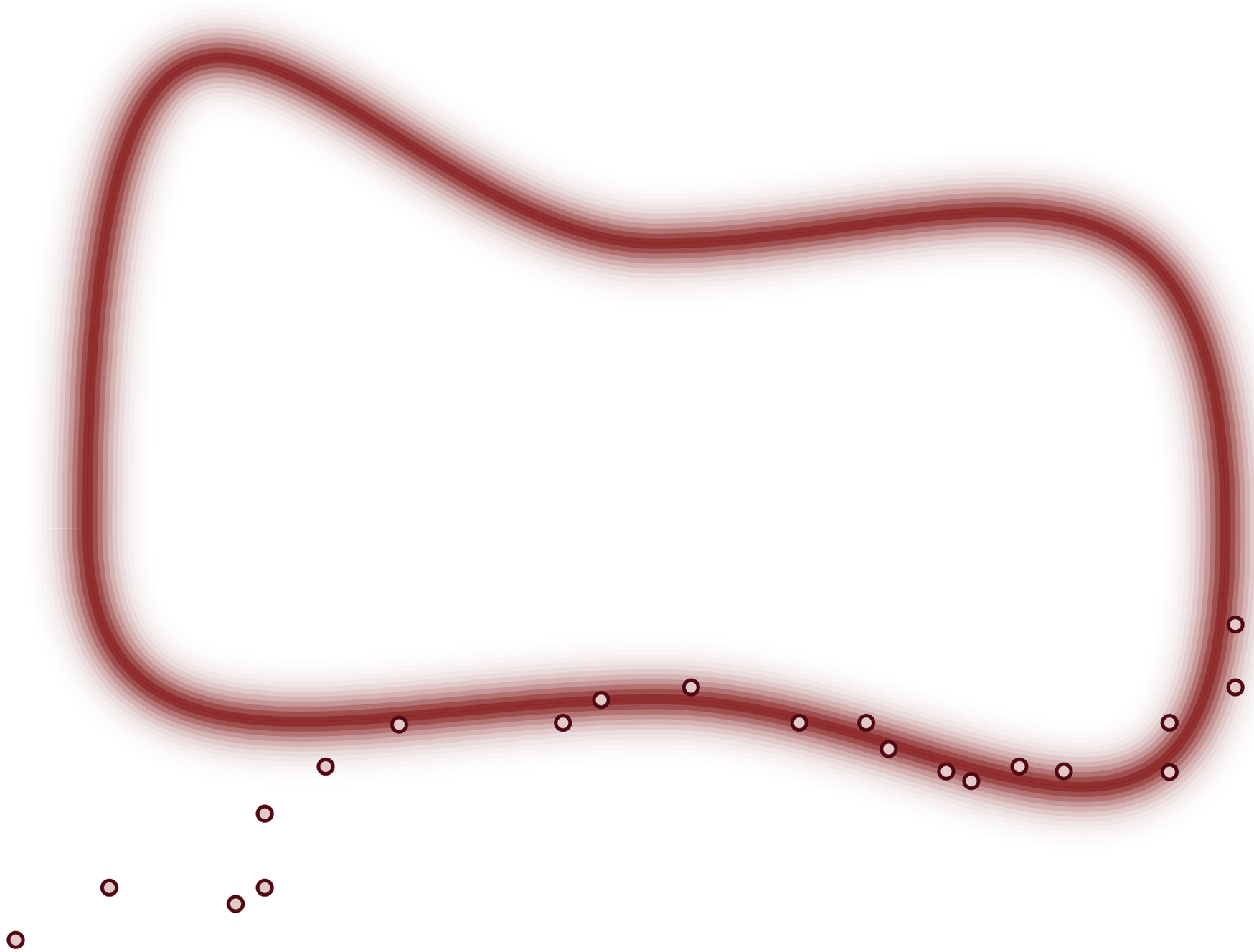


A Markov transition that targets our desired distribution naturally concentrates towards probability mass.

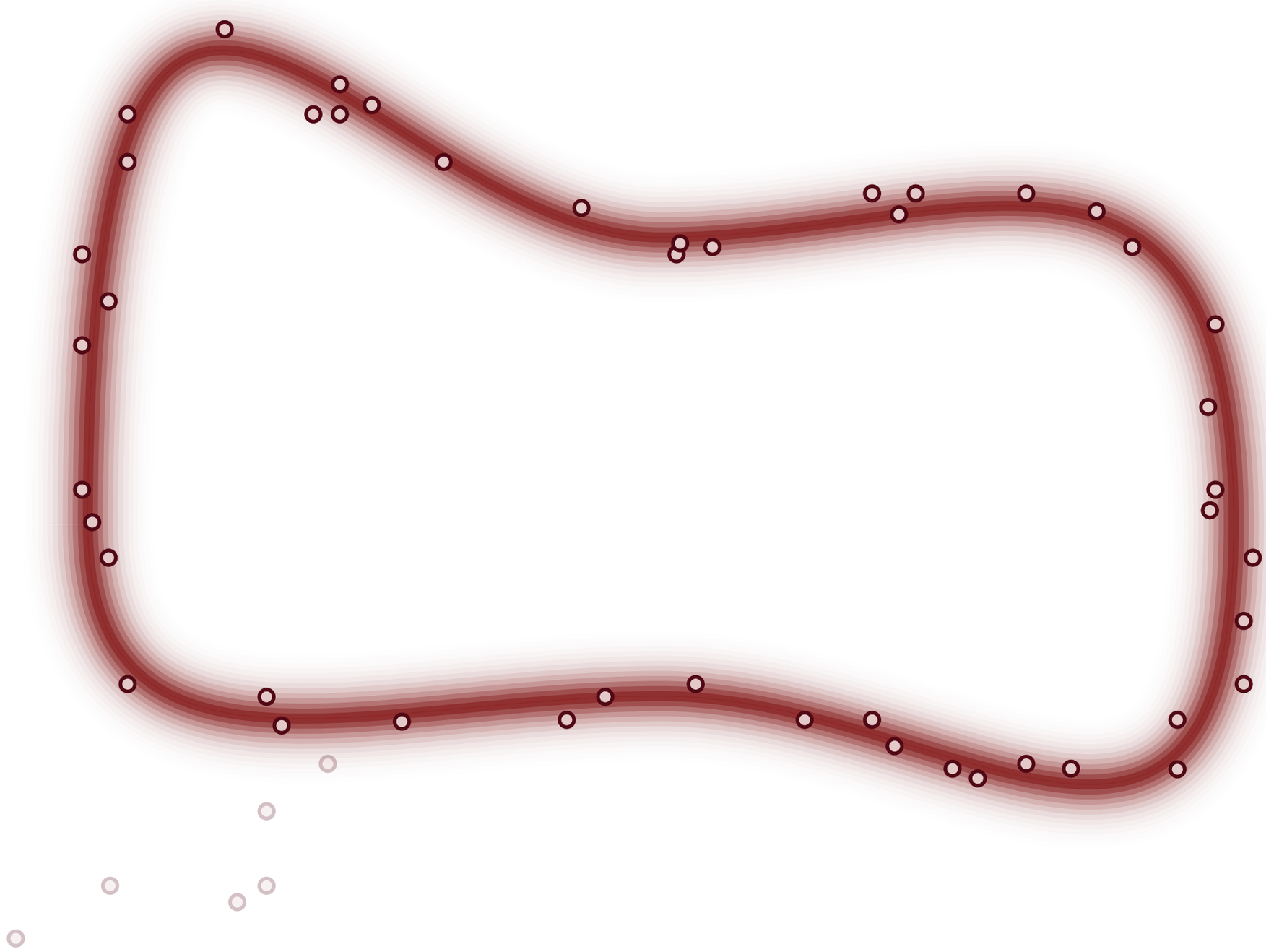


One approach is to use Markov chains as a generic scheme for finding and then exploring typical sets.

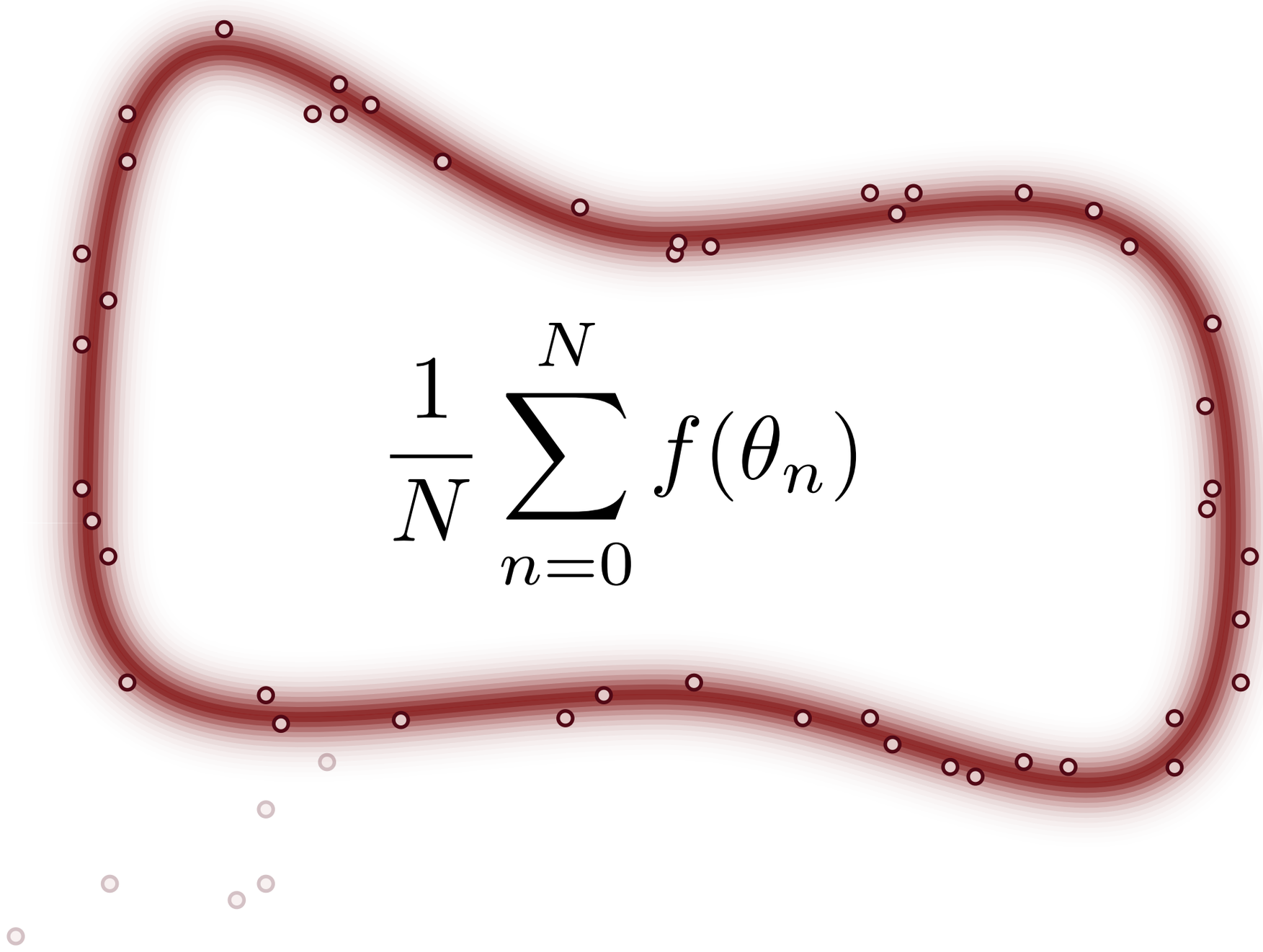
One approach is to use Markov chains as a generic scheme for finding and then exploring typical sets.



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*.

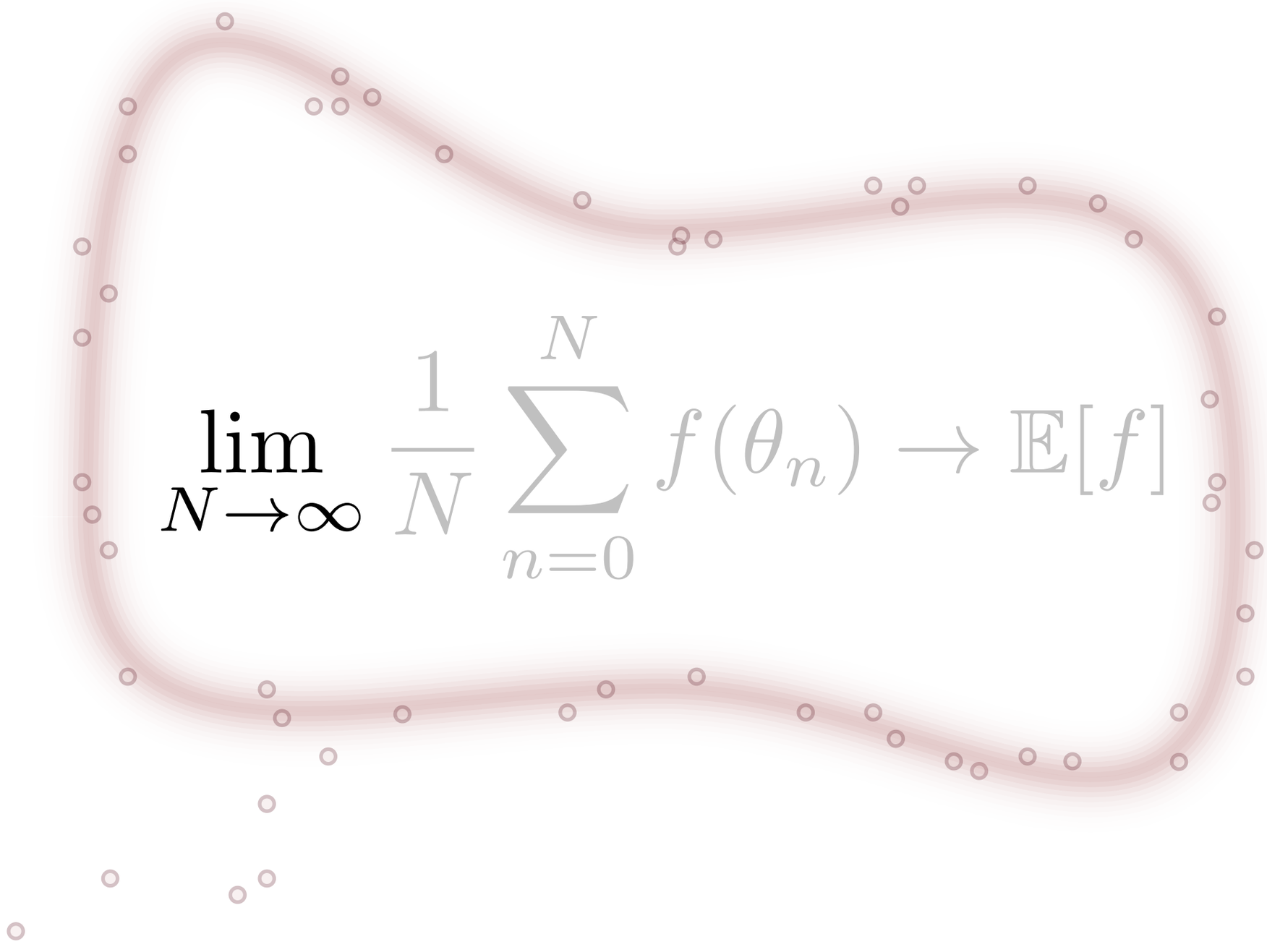


$$\frac{1}{N}\sum_{n=0}^{N} f(\theta_n)$$

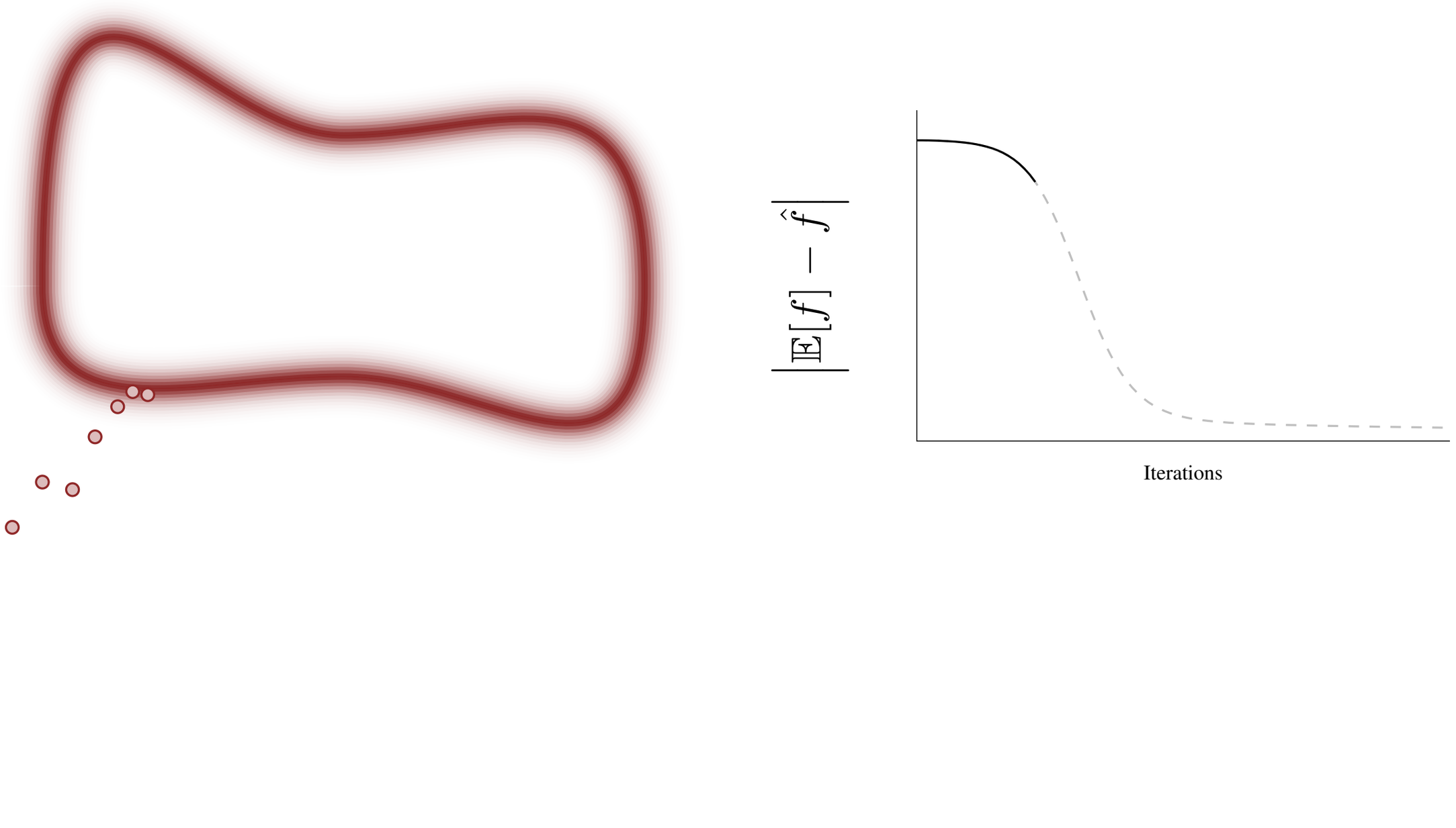
If run long enough, the Markov chain defines consistent *Markov Chain Monte Carlo estimators*.

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

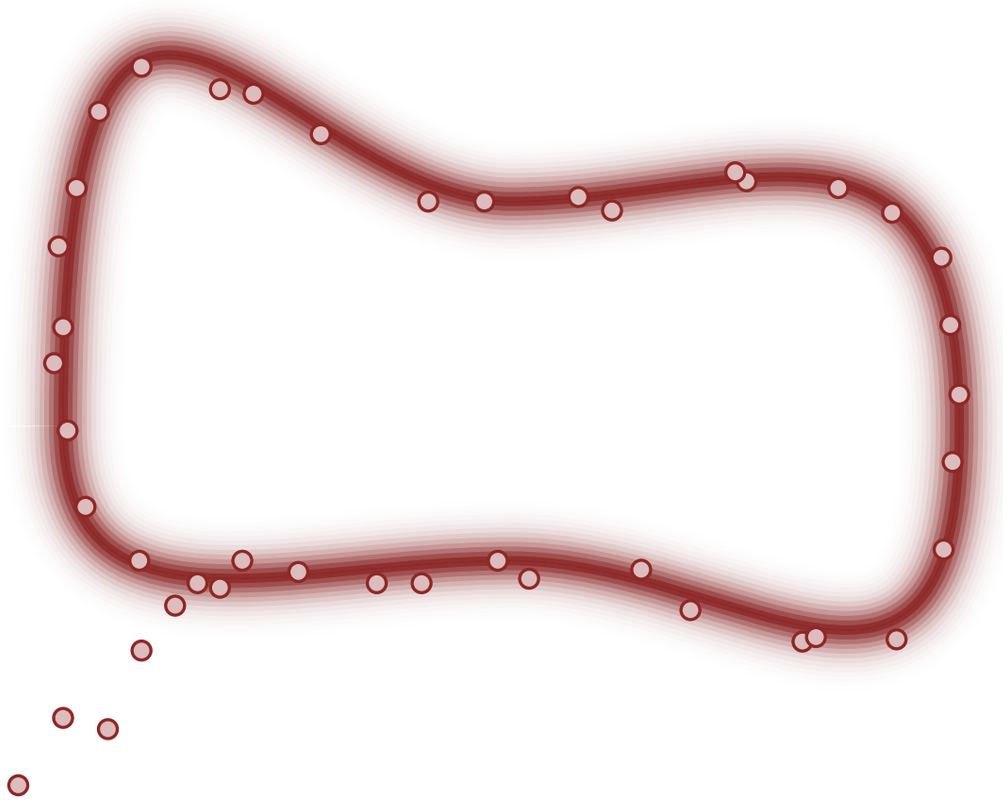
But just how long is long enough to achieve sufficiently accurate estimates in a given application?

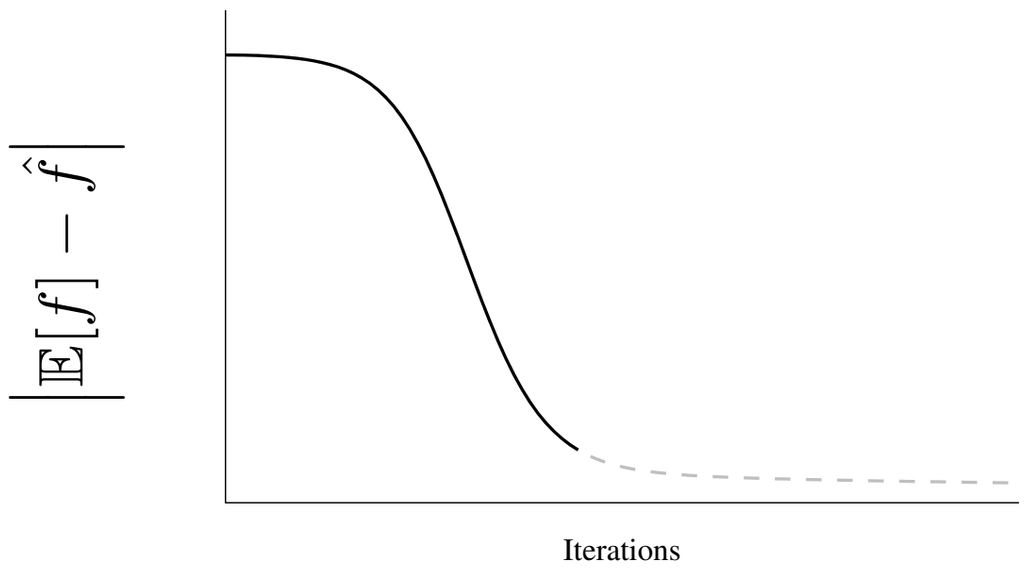


Under ideal conditions, MCMC estimators converge to the true expectations in a very practical progression.

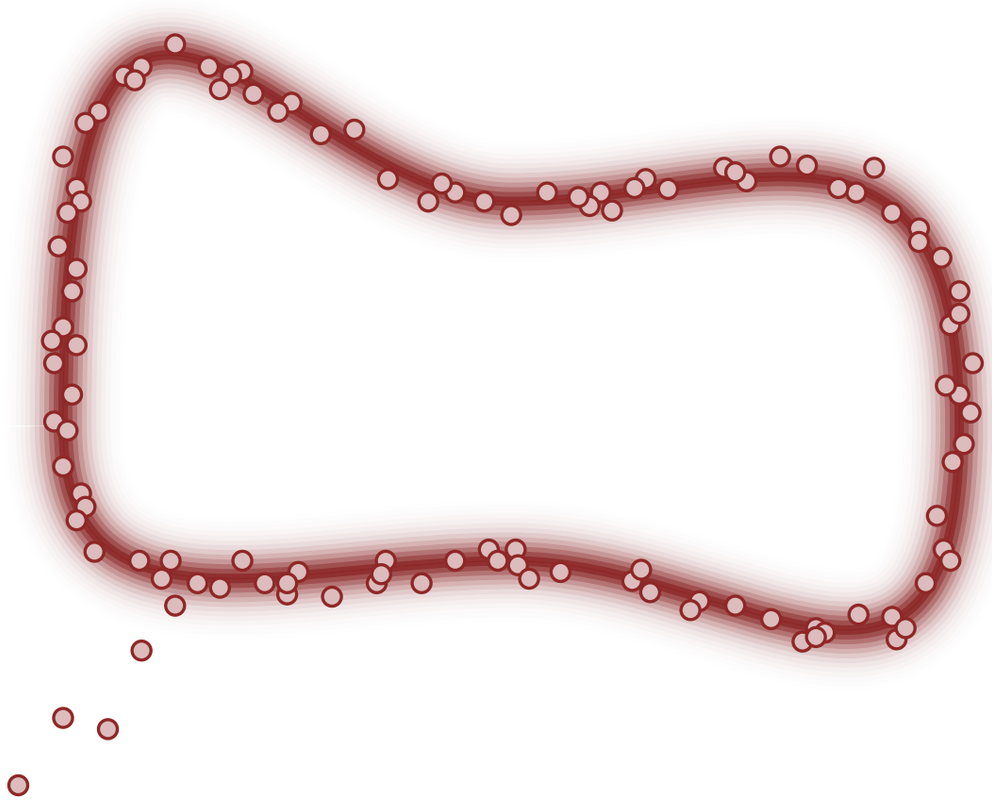


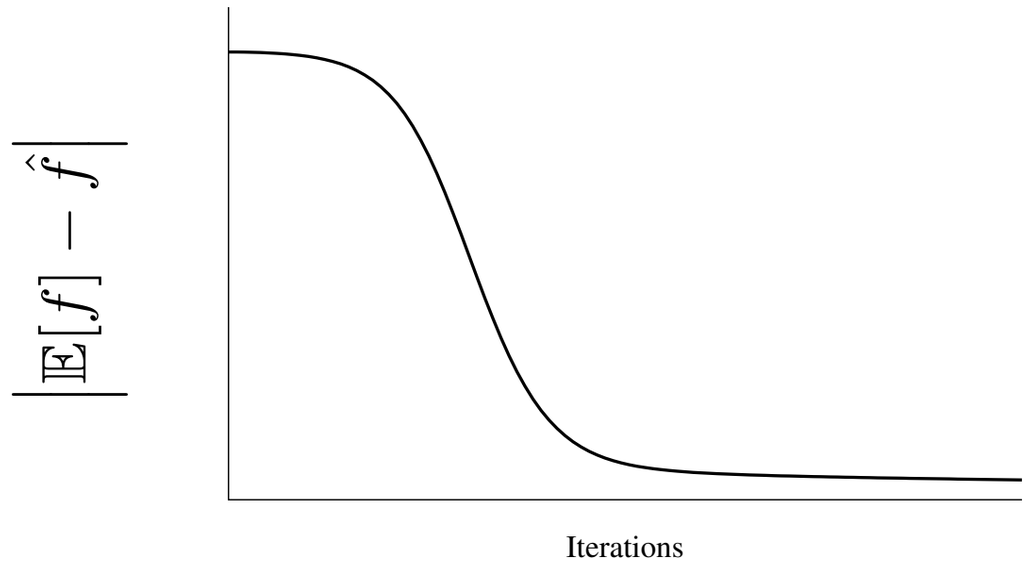
Under ideal conditions, MCMC estimators converge to the true expectations in a very practical progression.



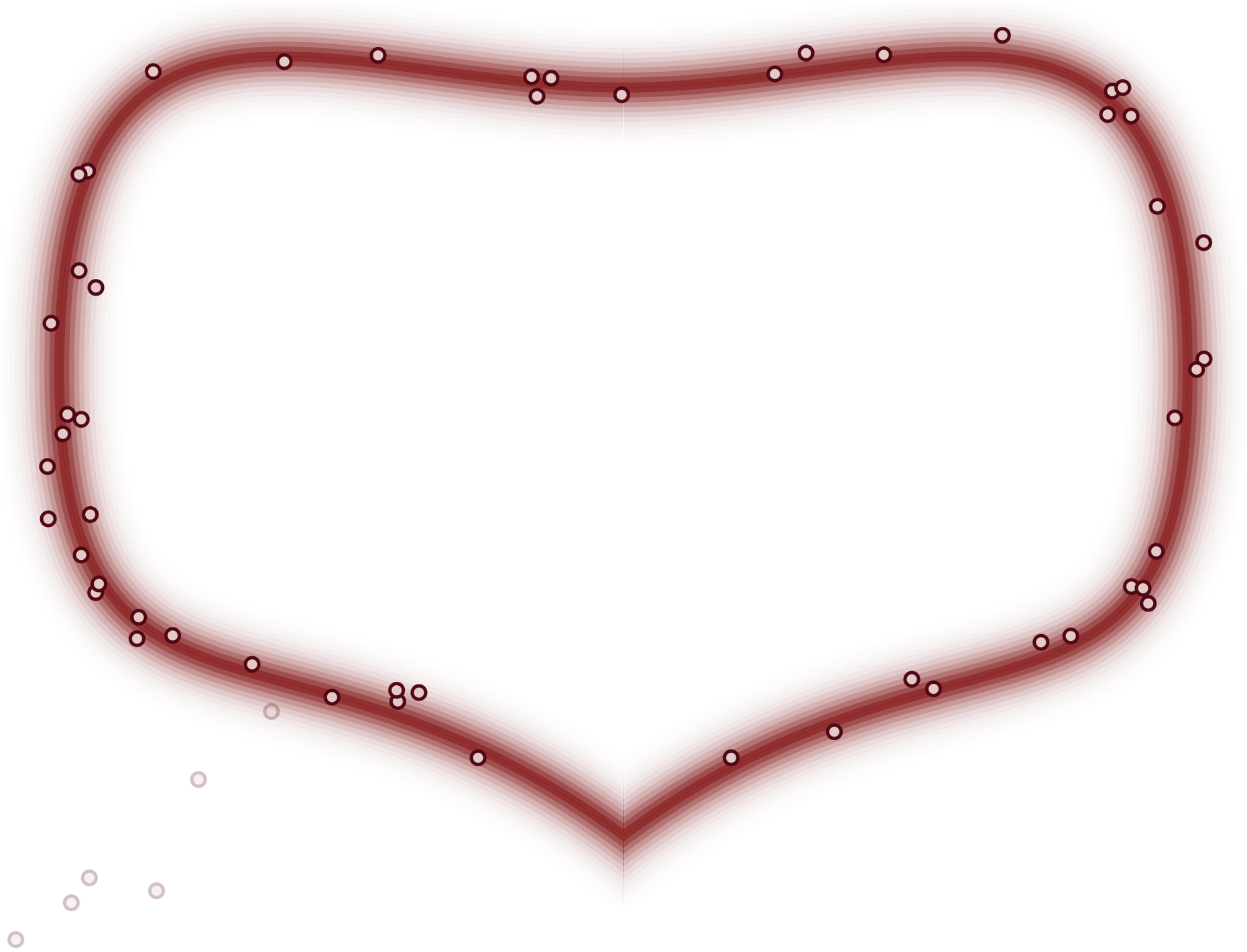


Under ideal conditions, MCMC estimators converge to the true expectations in a very practical progression.

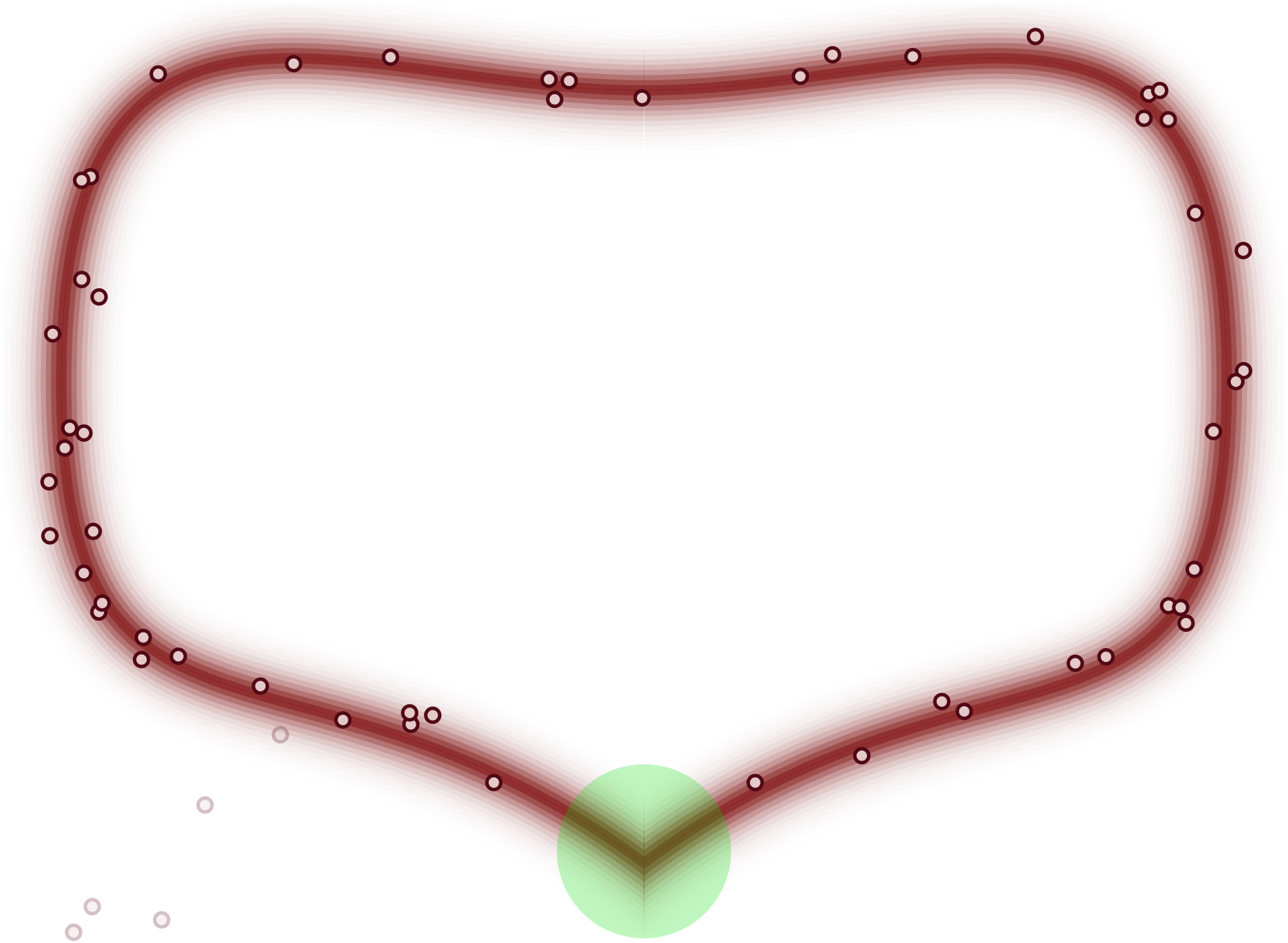




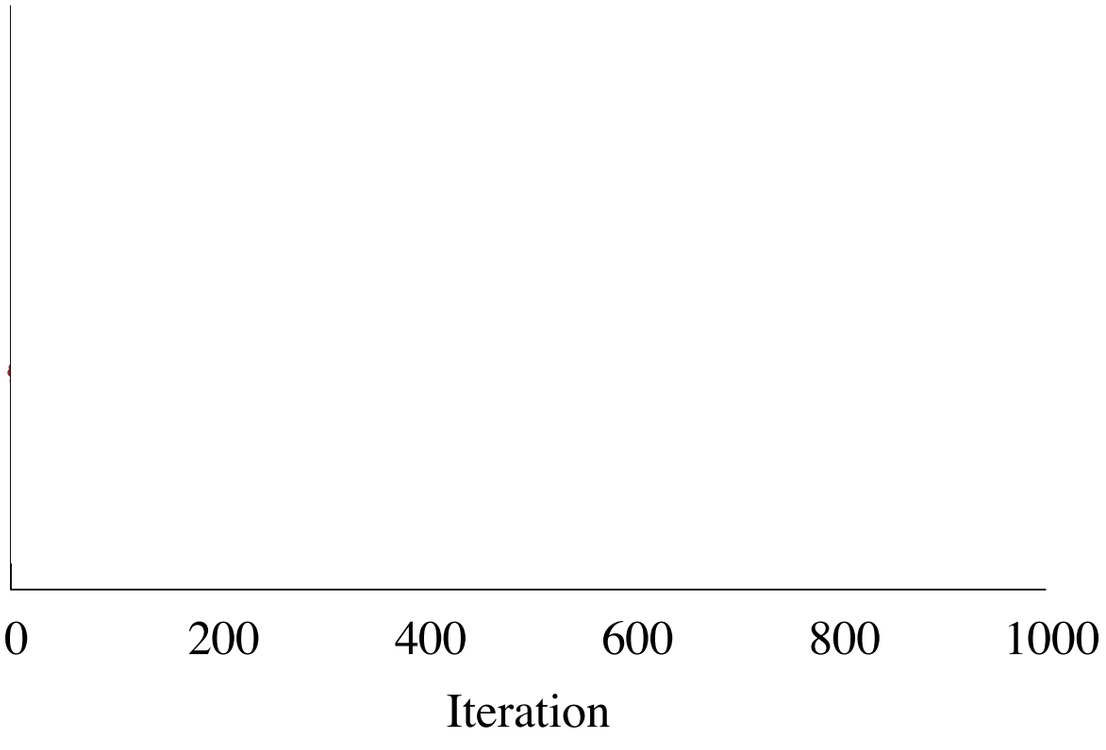
There are many pathological posterior geometries,
however, that spoil these ideal conditions.

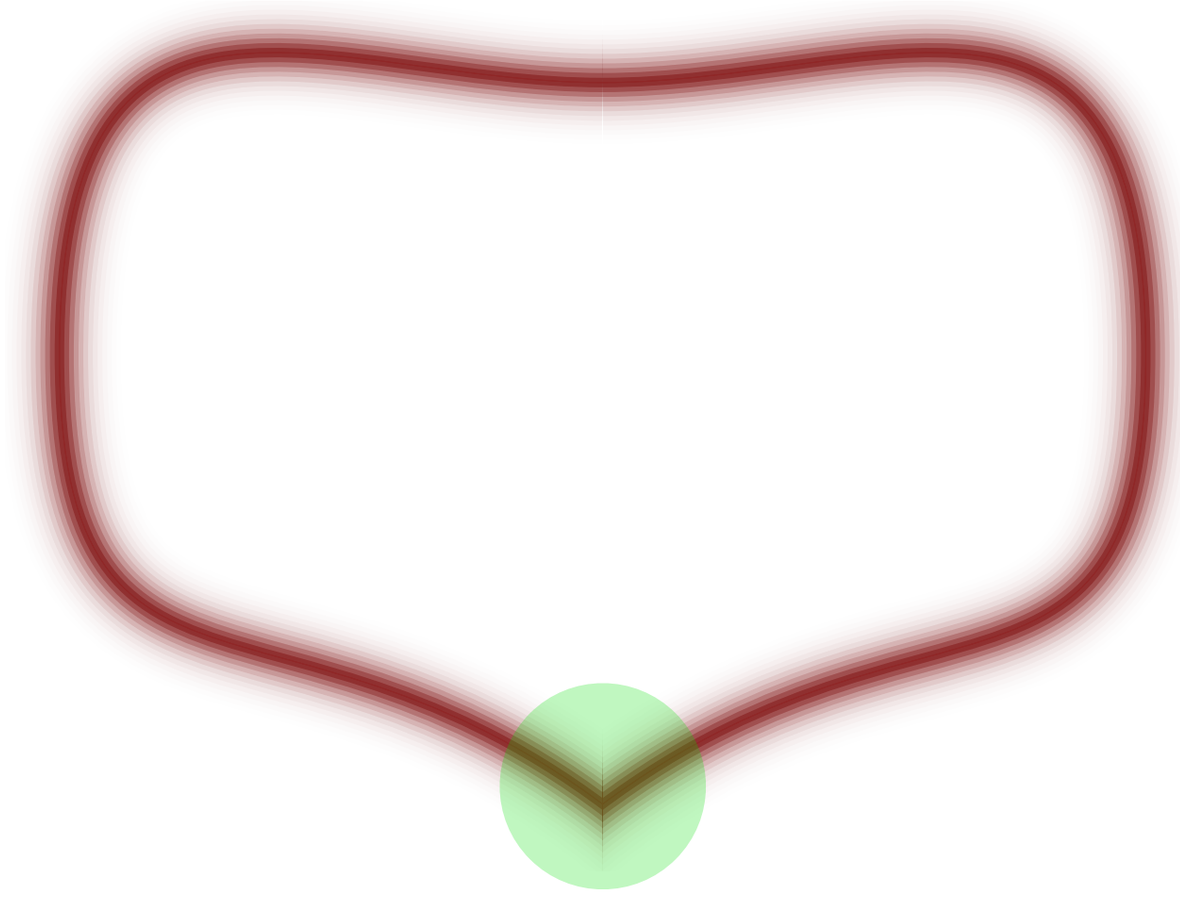


There are many pathological posterior geometries, however, that spoil these ideal conditions.

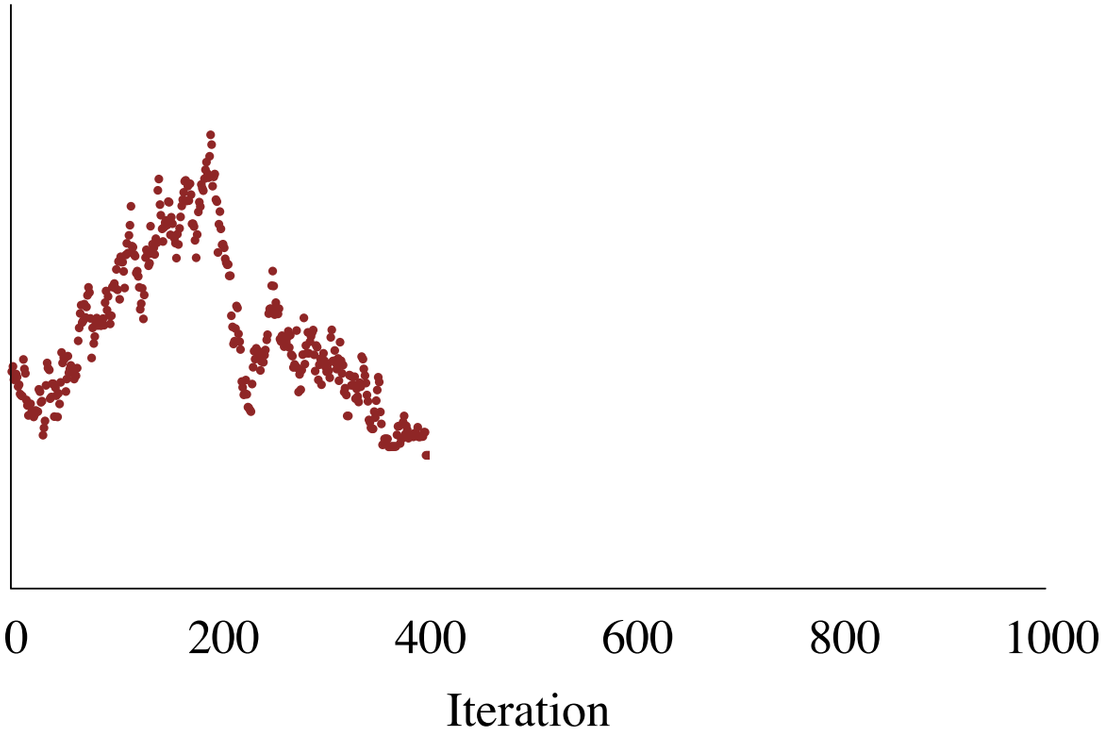


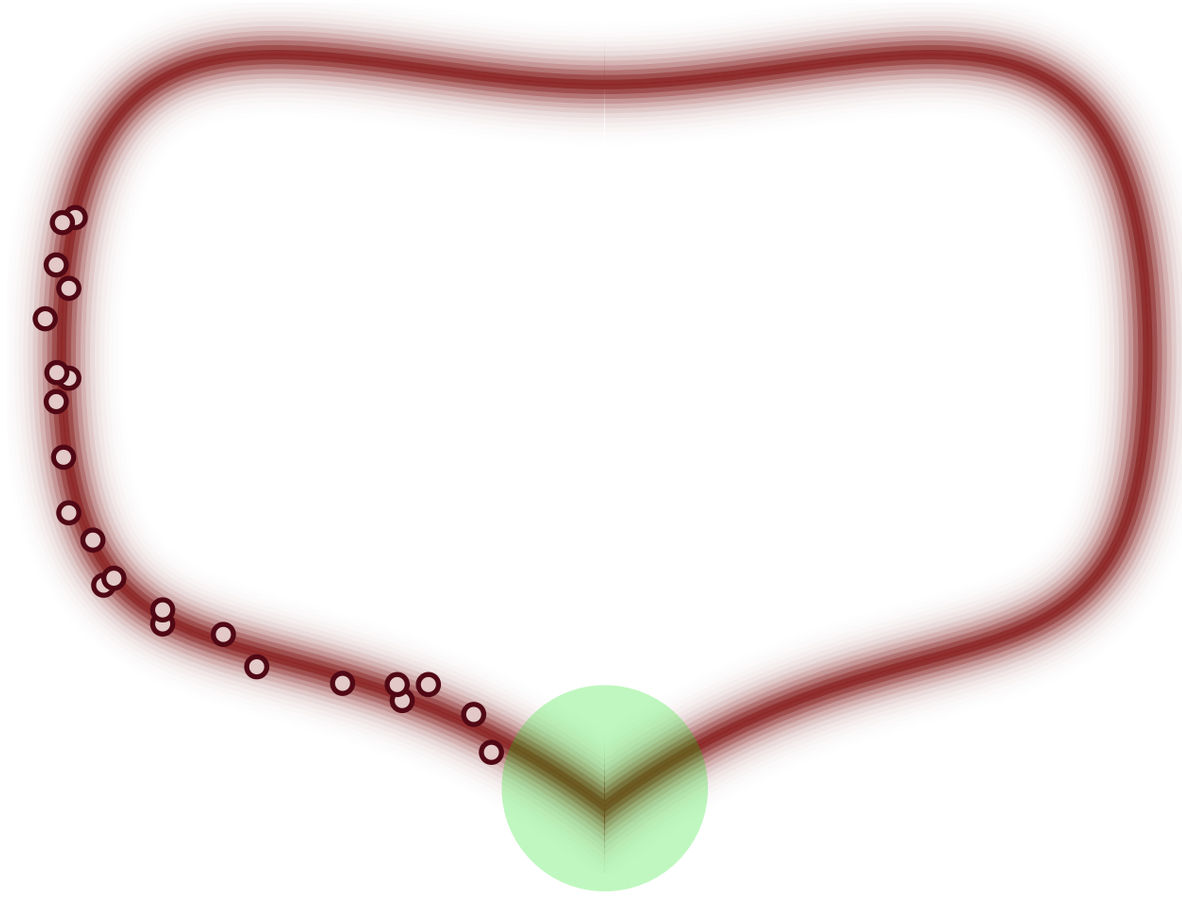
There are many pathological posterior geometries,
however, that spoil these ideal conditions.



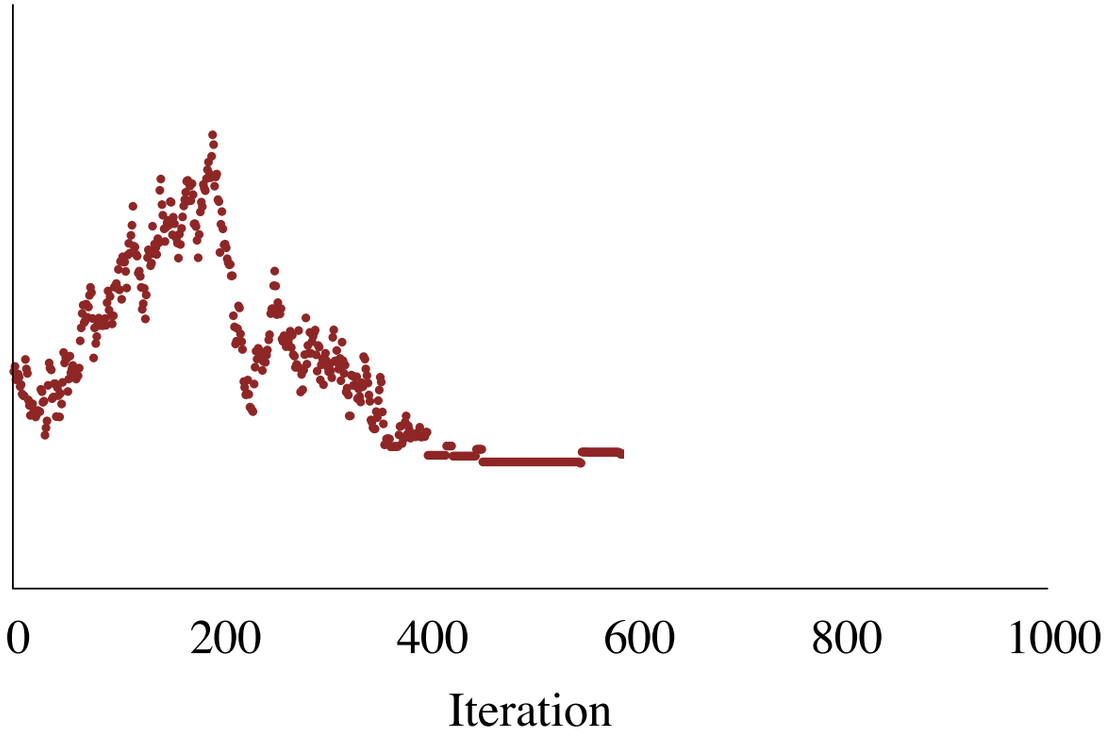


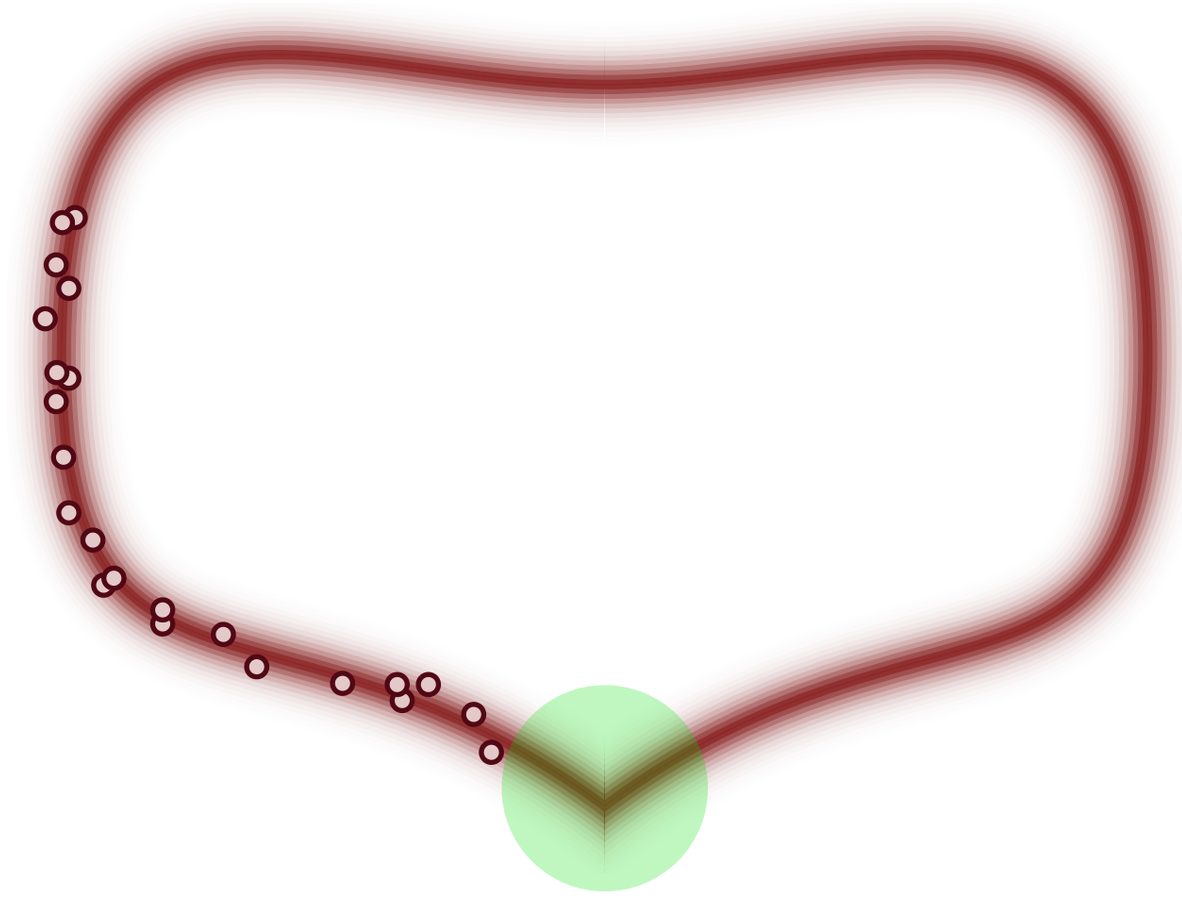
There are many pathological posterior geometries,
however, that spoil these ideal conditions.



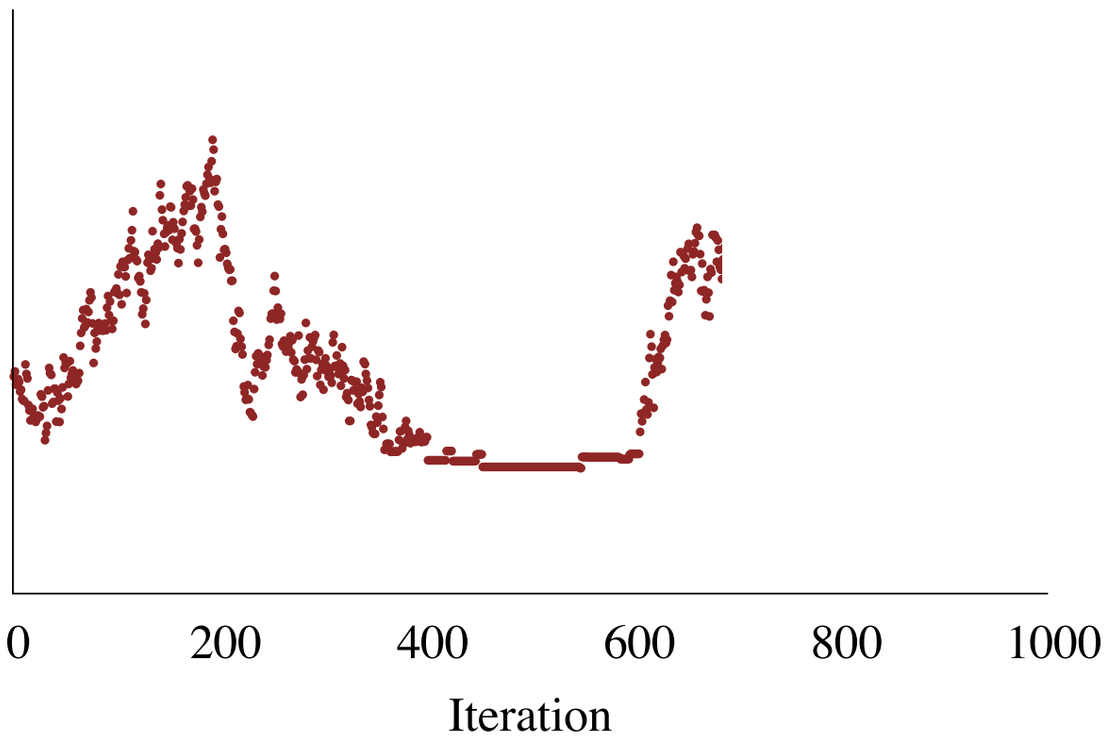


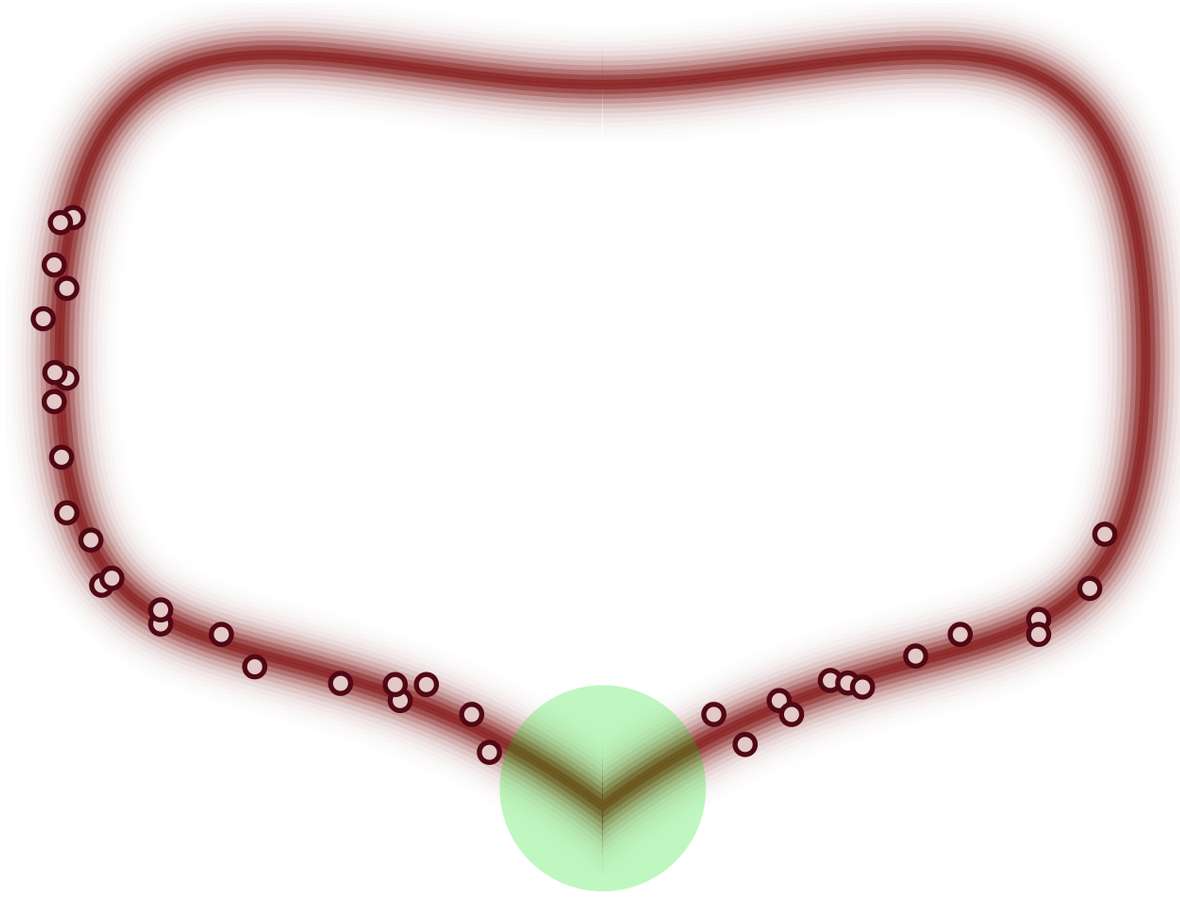
There are many pathological posterior geometries,
however, that spoil these ideal conditions.





There are many pathological posterior geometries,
however, that spoil these ideal conditions.





Geometric ergodicity ensures that there are no posterior pathologies obstructing accurate MCMC estimation.

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

Geometric ergodicity ensures that there are no posterior pathologies obstructing accurate MCMC estimation.

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

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

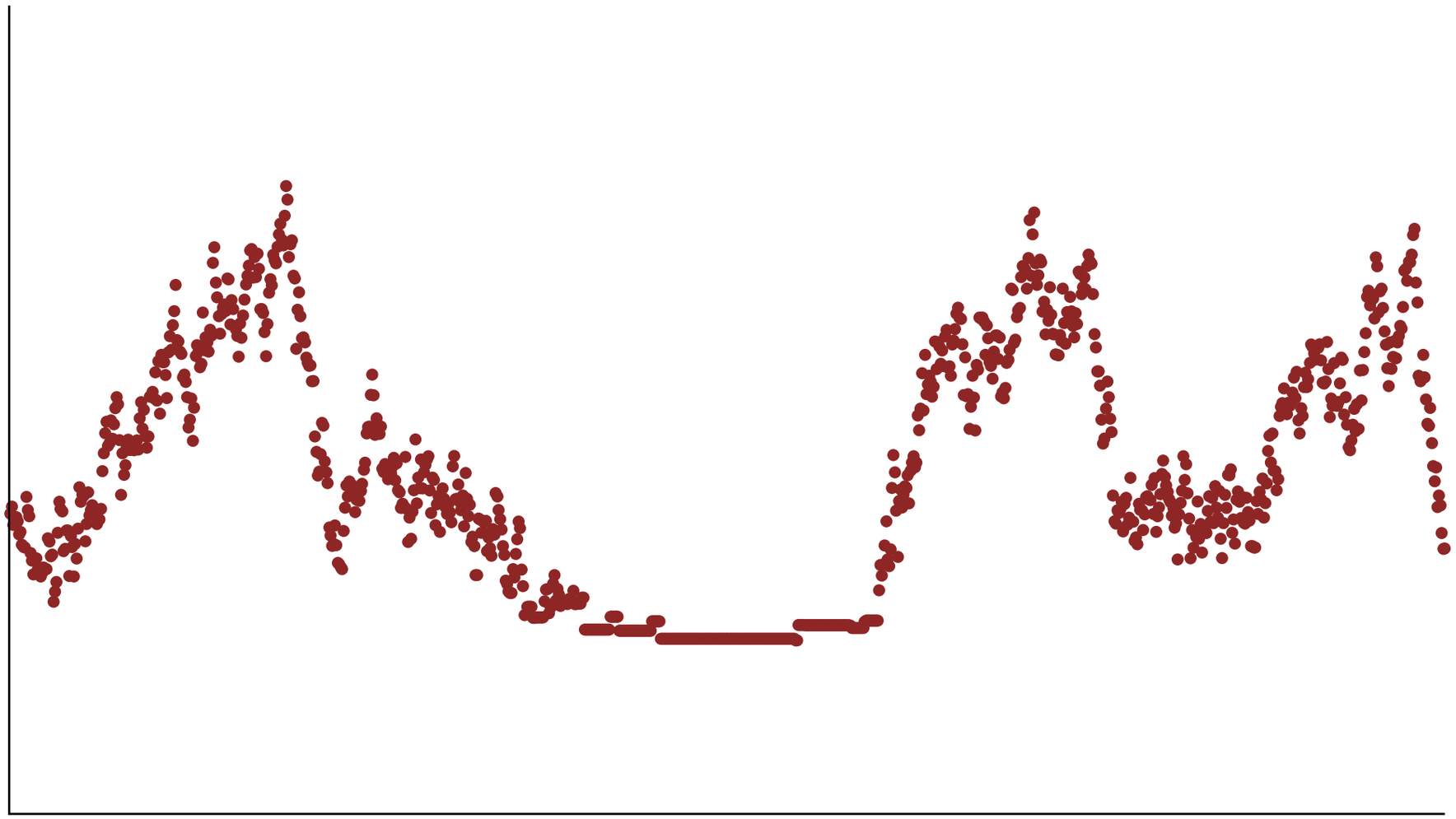
Geometric ergodicity ensures that there are no posterior pathologies obstructing accurate MCMC estimation.

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

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

$$\mathrm{MCSE}^2 = \frac{\mathrm{Var}[f]}{\mathrm{ESS}[f]}$$

Diagnosing Inadequate Convergence



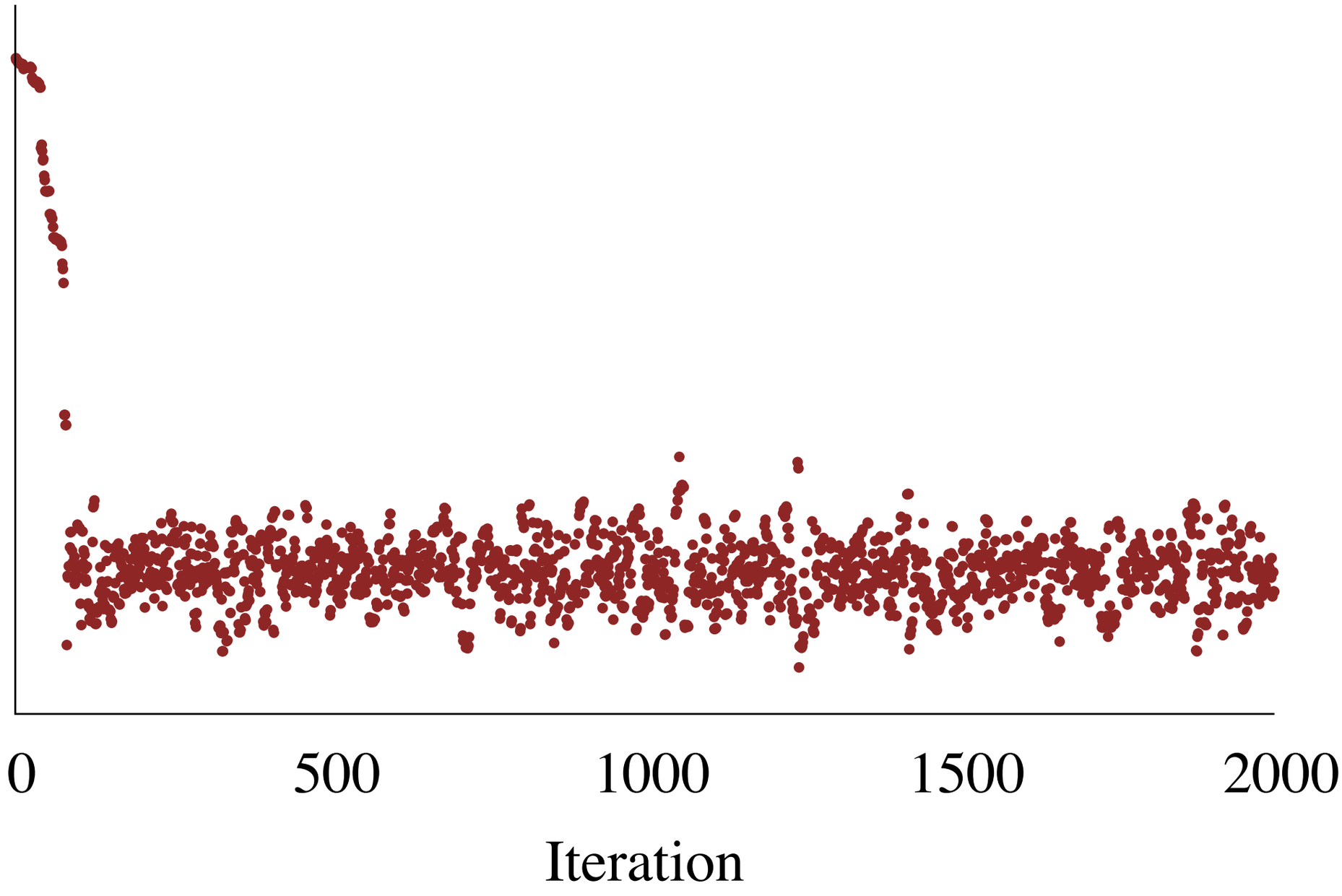
How do we verify that not only geometric ergodicity holds but also our Markov chains have converged?

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

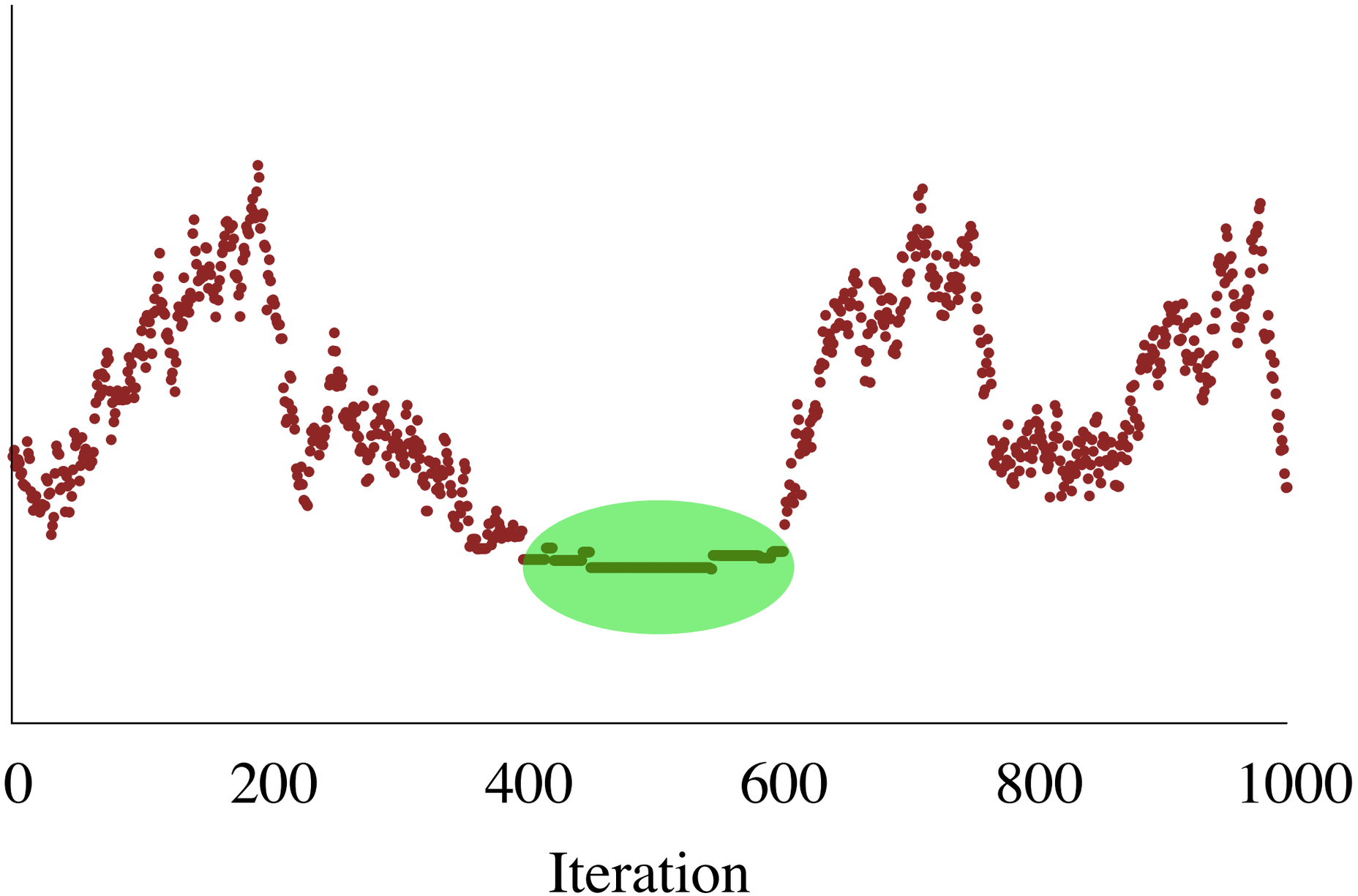
$$\hat{f} \sim \mathcal{N}\left(\mathbb{E}[f], \mathrm{MCSE}^2\right)$$

$$\mathrm{MCSE}^2 = \frac{\mathrm{Var}[f]}{\mathrm{ESS}[f]}$$

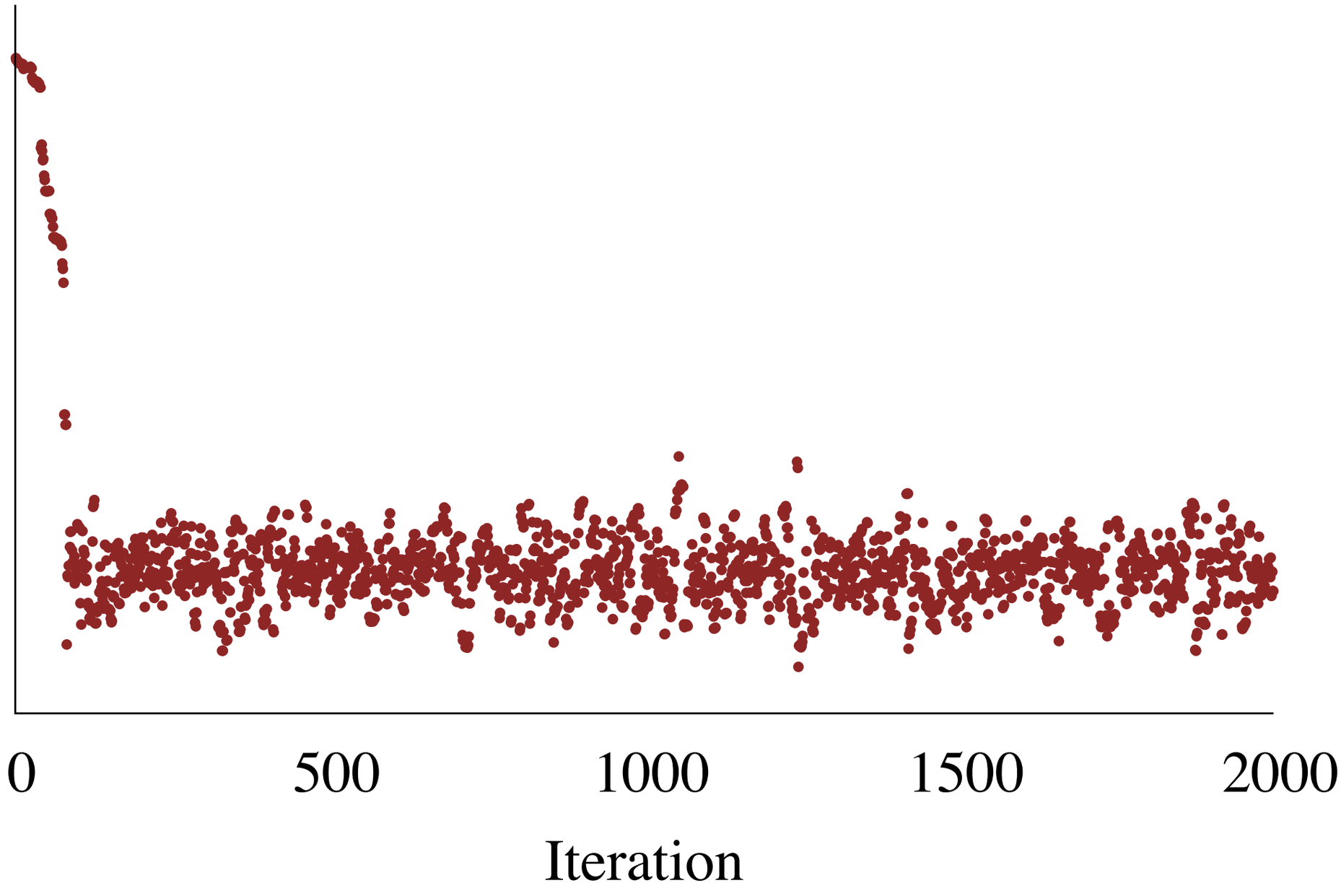
Visual diagnostics of *trace plots* is one particularly immediate option.



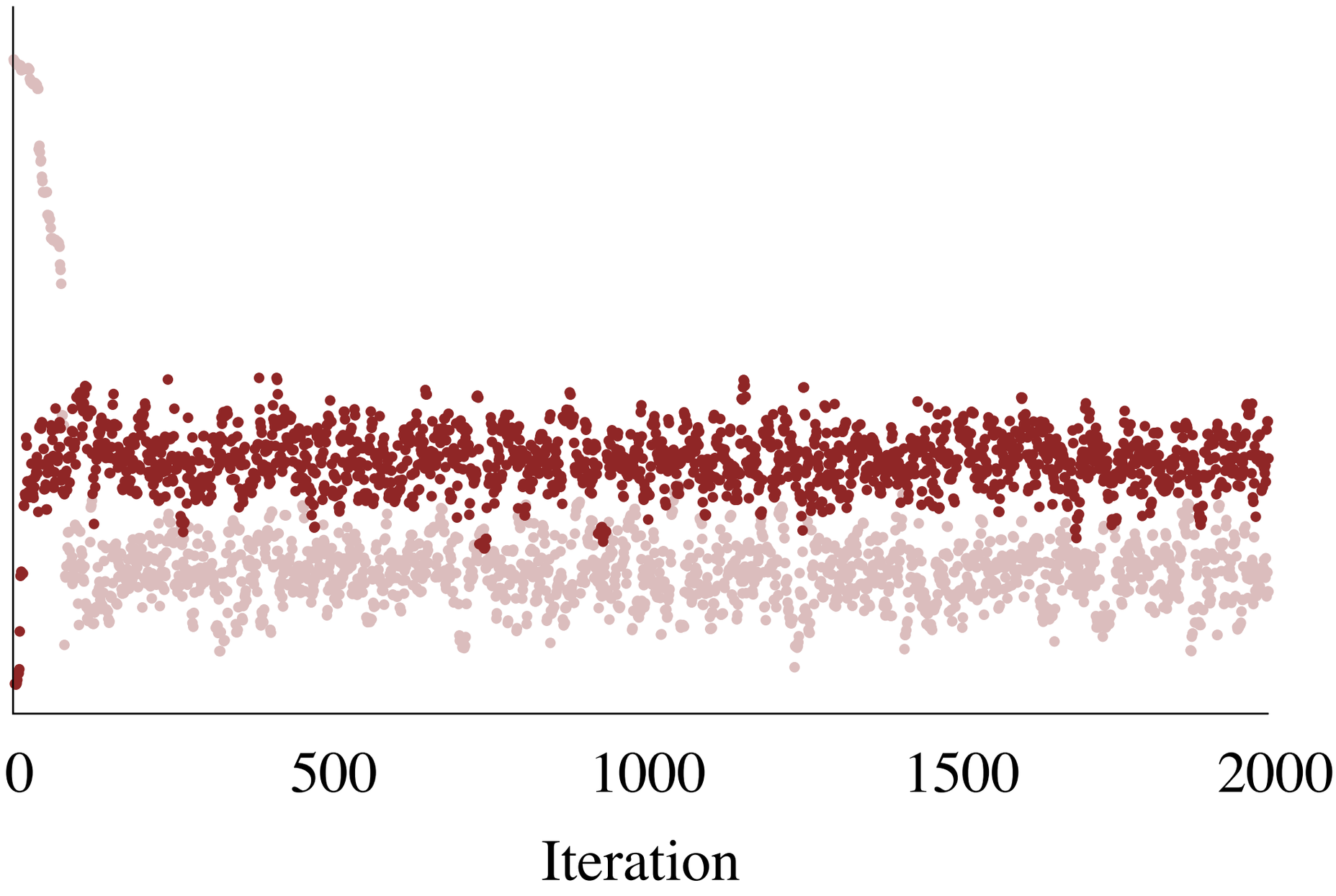
For example, we might identify regions of high curvature where the Markov chains stick.



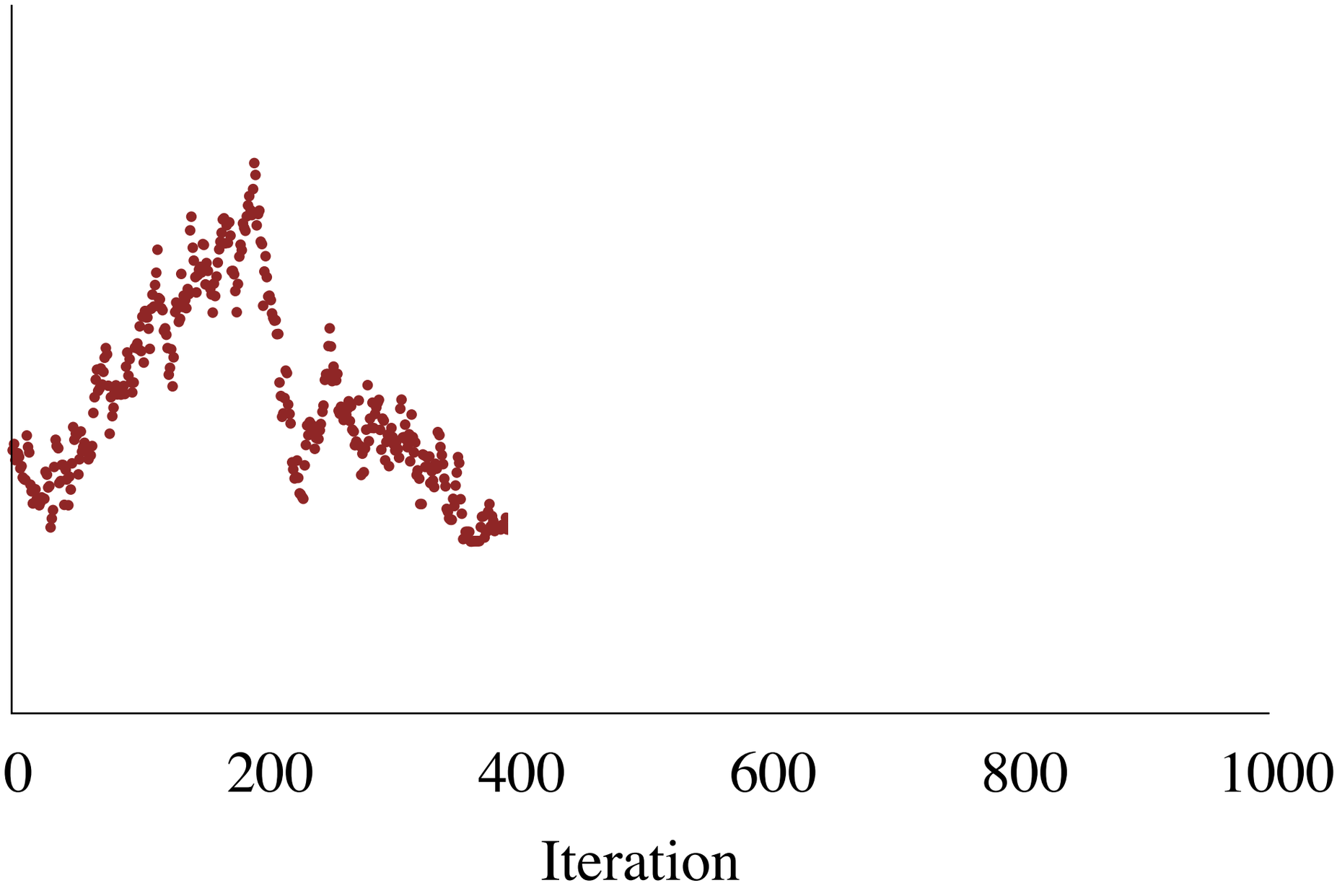
Unfortunately visual diagnostics can be misleading.



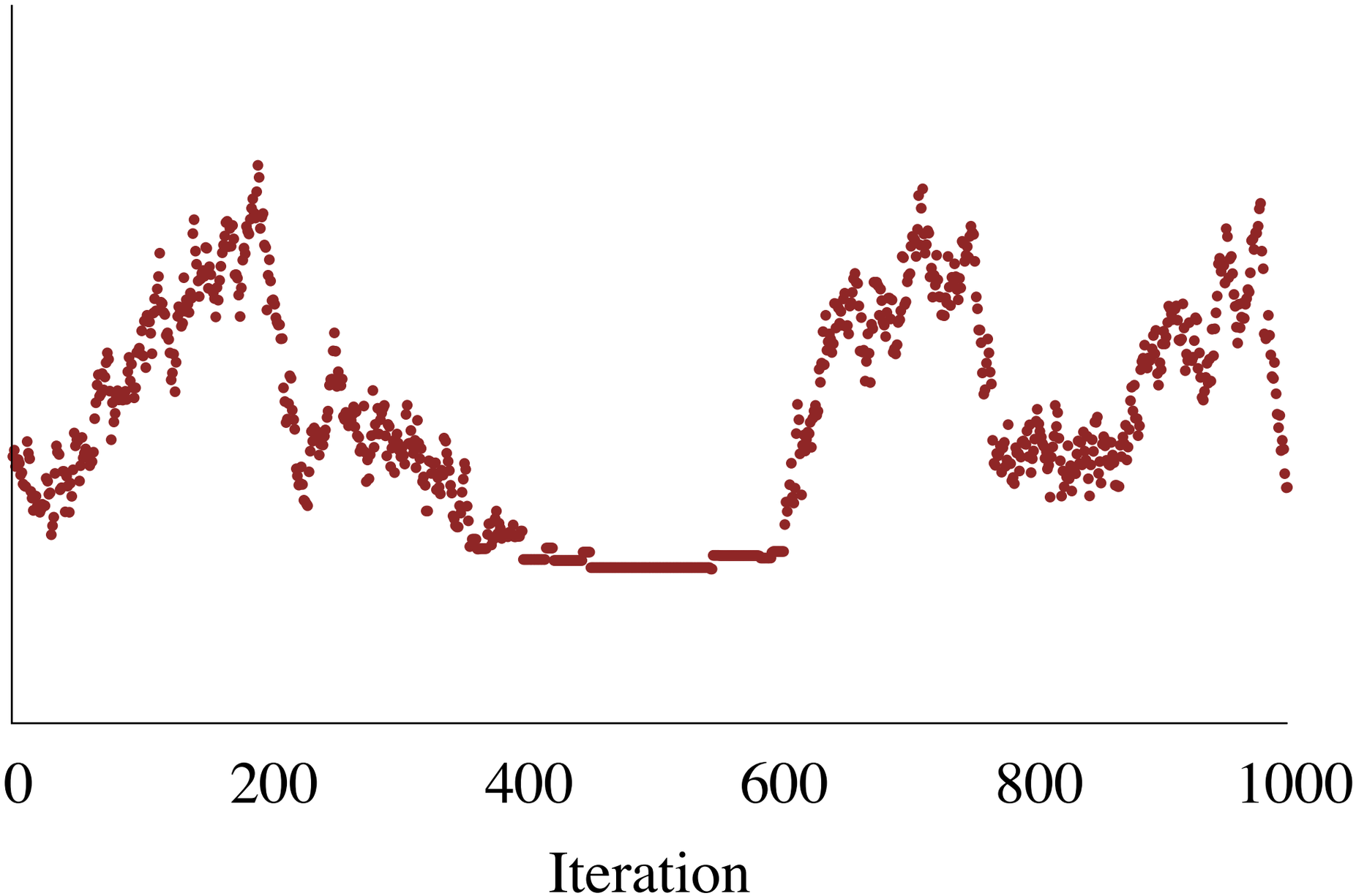
Unfortunately visual diagnostics can be misleading.



Unfortunately visual diagnostics can be misleading.

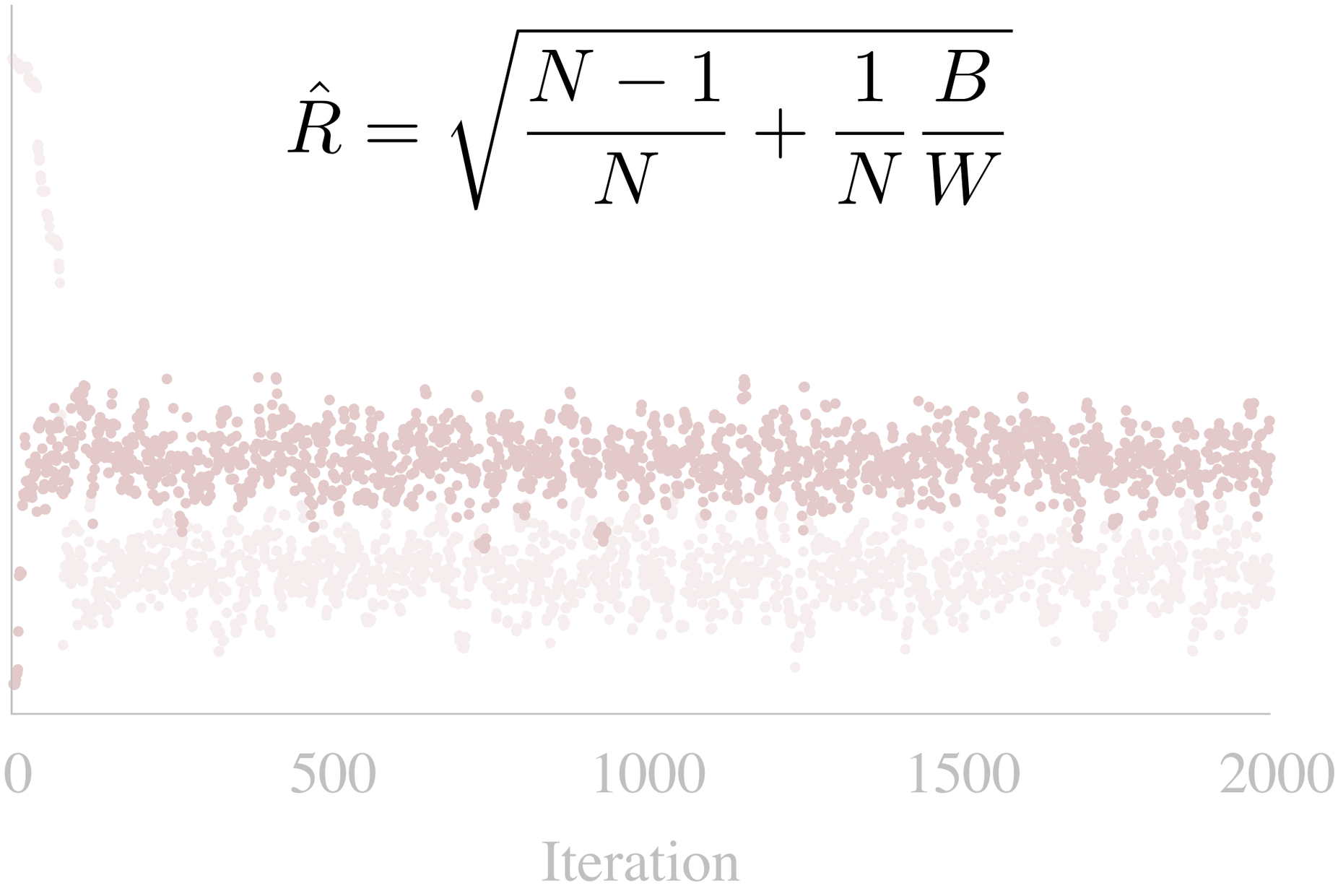


Unfortunately visual diagnostics can be misleading.



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 generic strategy is to run *multiple* chains from diffuse initializations and compare them using split Rhat.

```
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
divergent__	0.00	0.0e+00	0.0e+00	0.00	0.00	0.00	1000	25869	1.0
theta	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.

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

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

$$\mathrm{MCSE}^2 = \frac{\mathrm{Var}[f]}{\mathrm{ESS}[f]}$$

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
theta	20	4.3e-02	1.0e+00	18	20	22	568	14697	1.0

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
theta	20	4.3e-02	1.0e+00	18	20	22	568	14697	1.0

Hamiltonian Monte Carlo

The previous discussion presumed the existence of a Markov chain that targets our specific posterior.

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

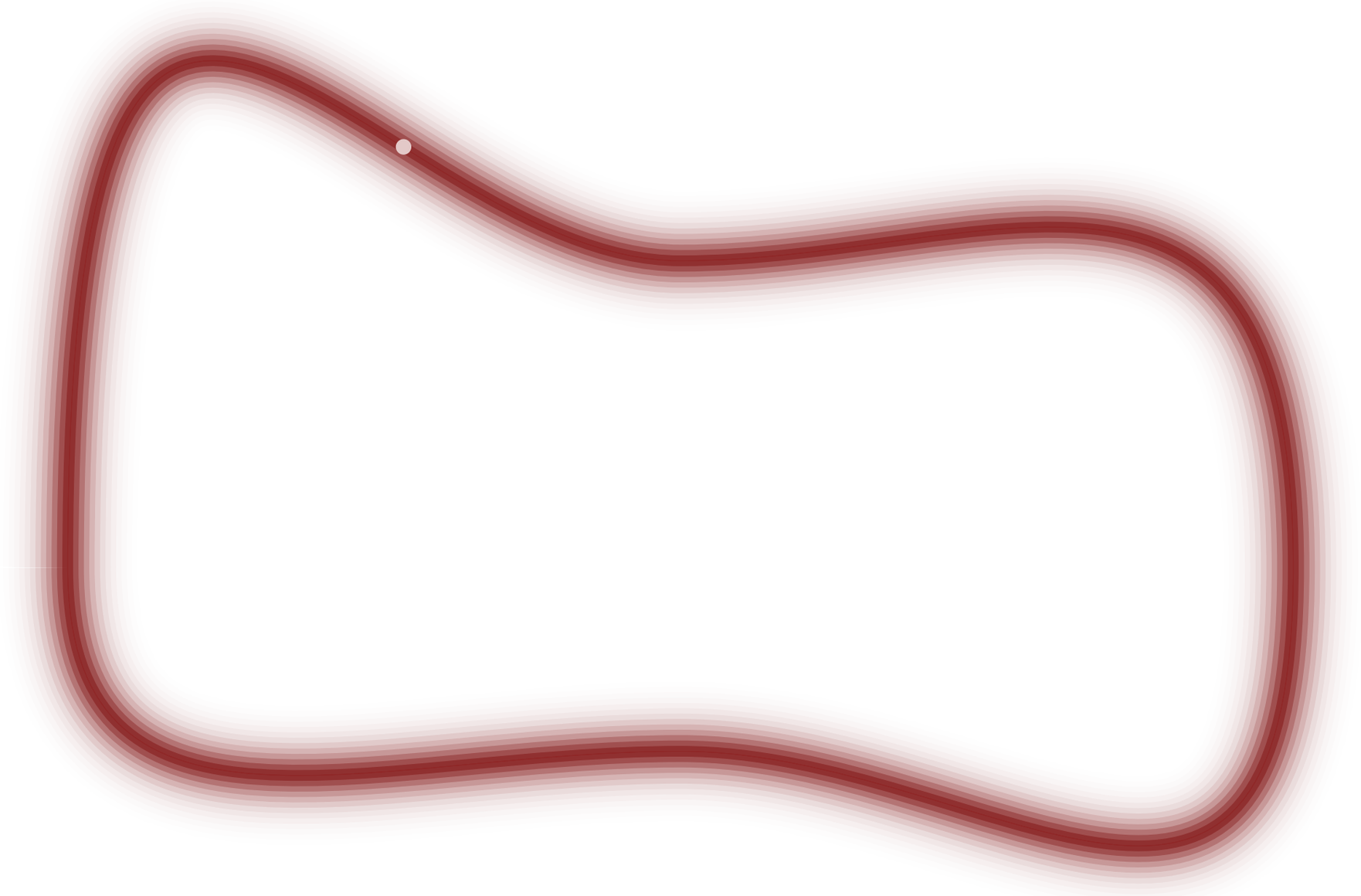
To simplify notation a bit let's ignore the explicit dependence on the data and the small world.

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

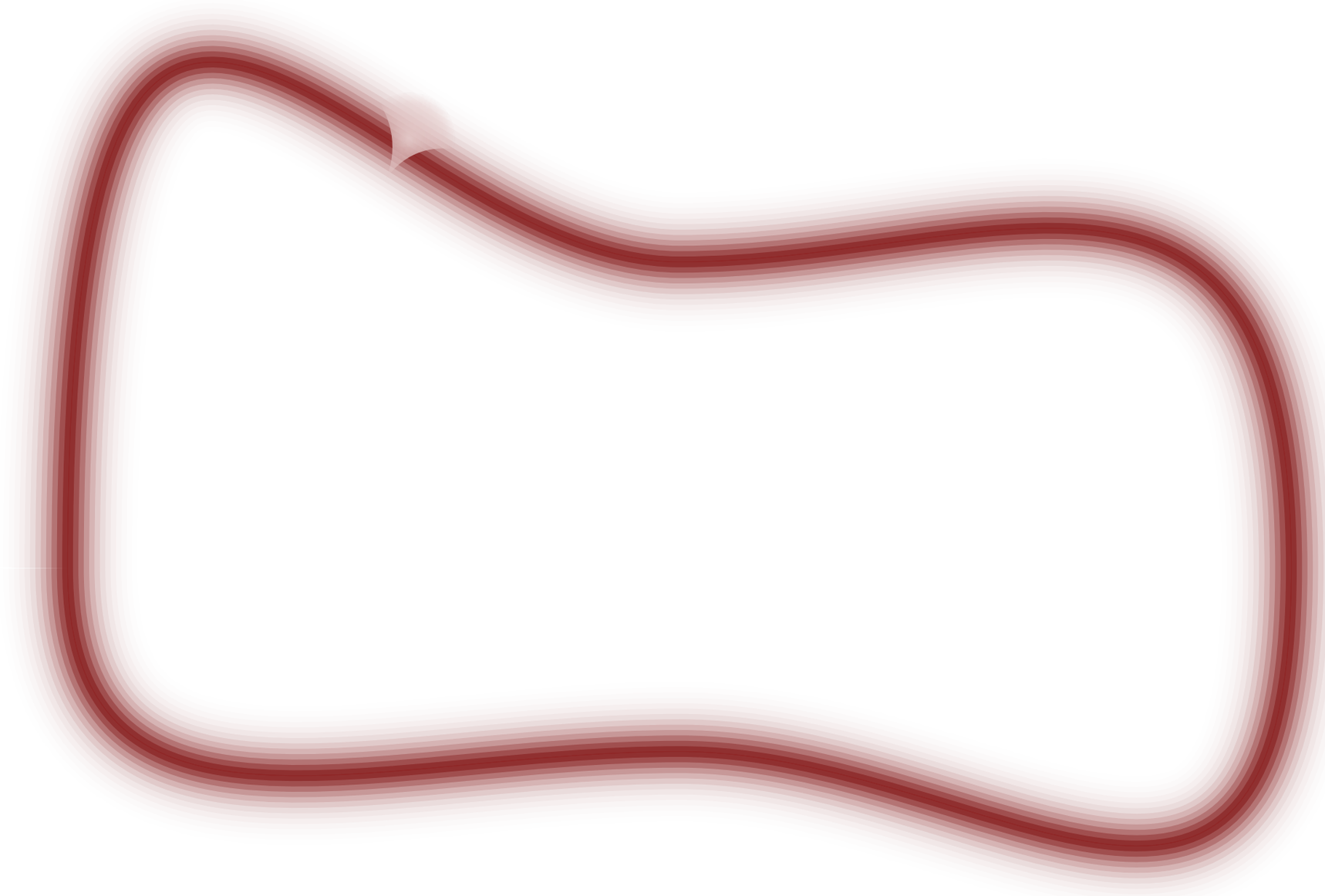
One way to construct a chain is Random Walk Metropolis which explores the posterior with a "guided" diffusion.

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

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

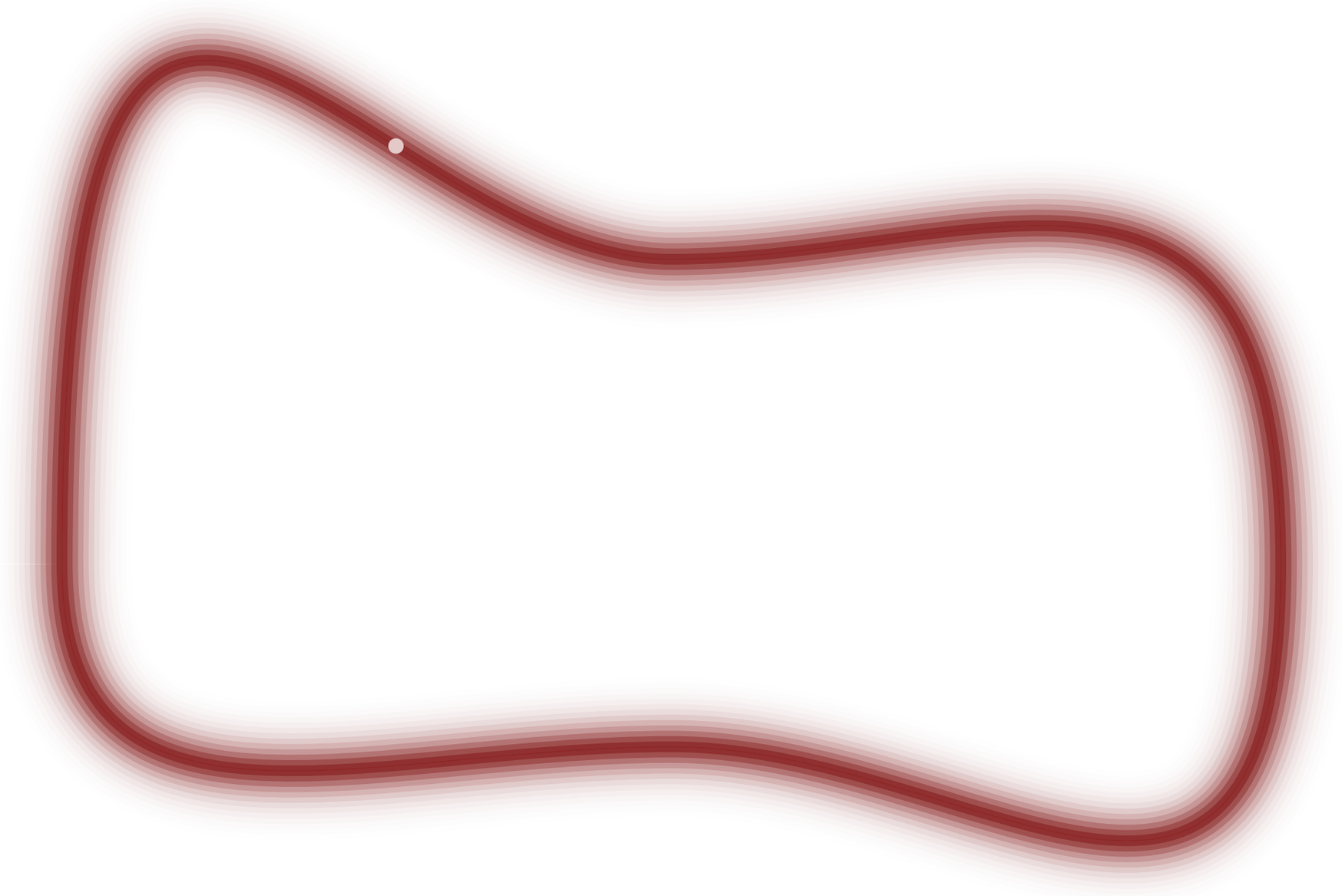


Unfortunately this naive diffusion explores high-dimensional target distributions inefficiently.



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

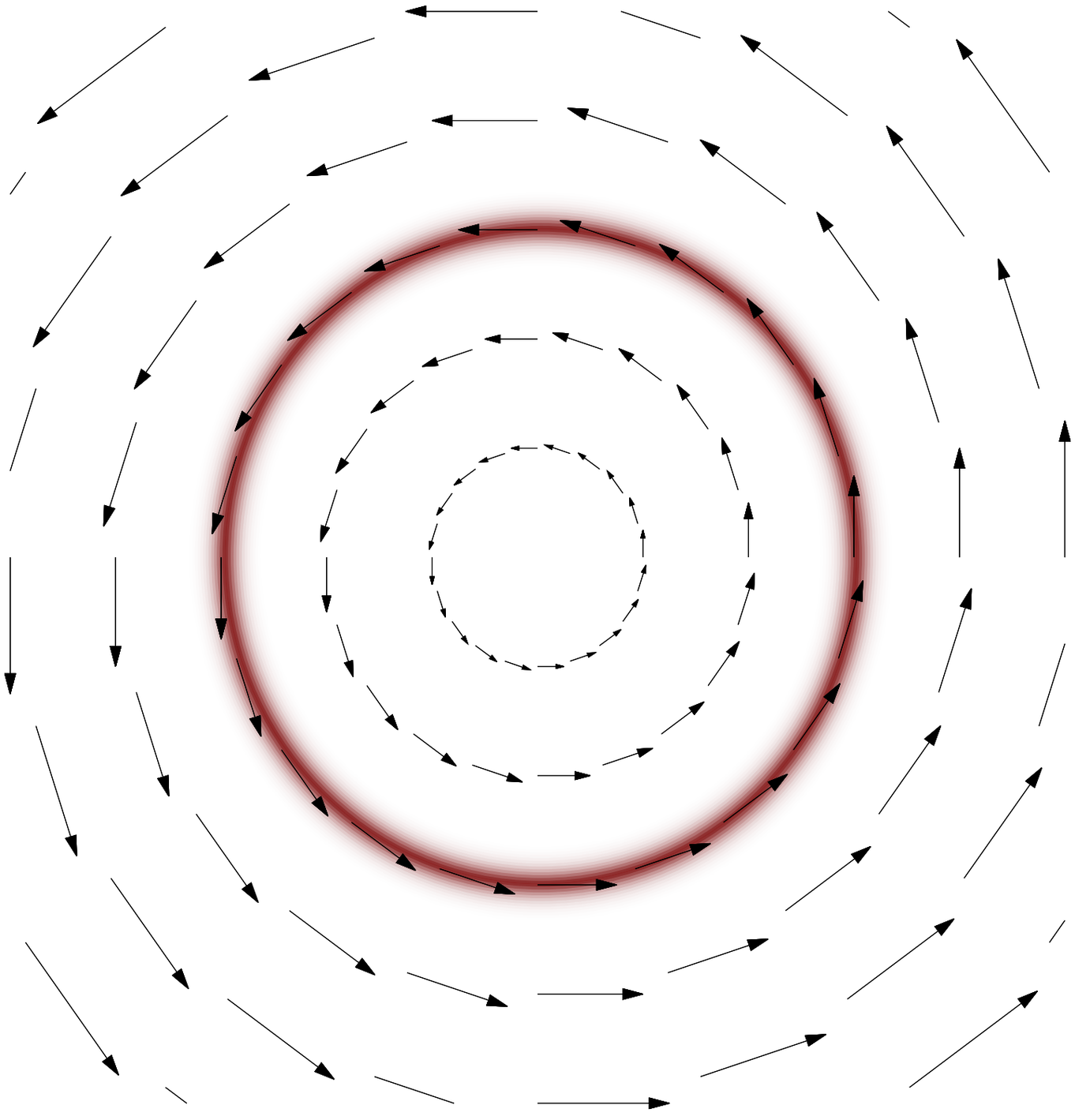
In order to efficiently scale to complex posteriors, MCMC needs *coherent* exploration of the typical set.



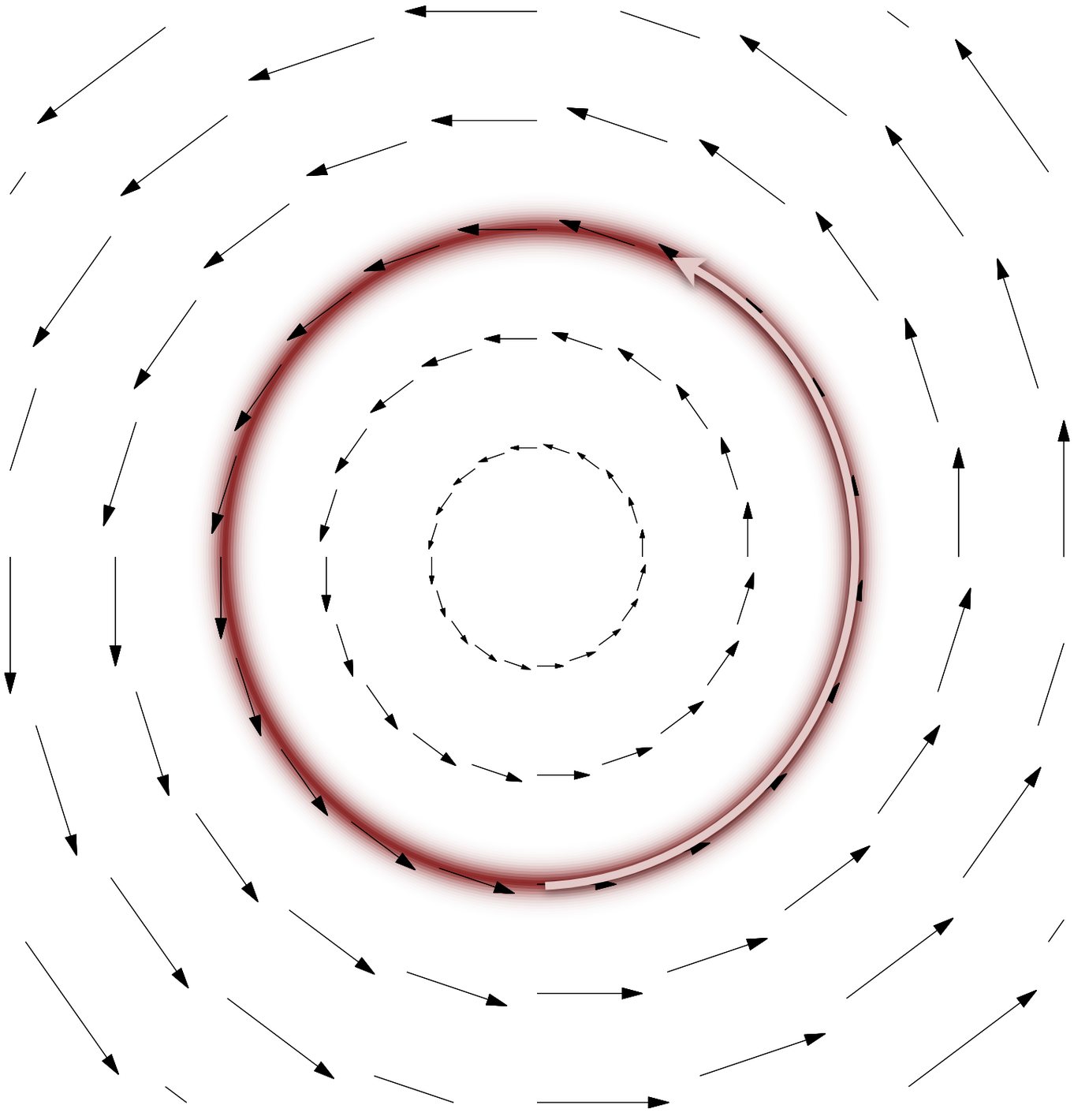
In order to efficiently scale to complex posteriors, MCMC needs *coherent* exploration of the typical set.

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

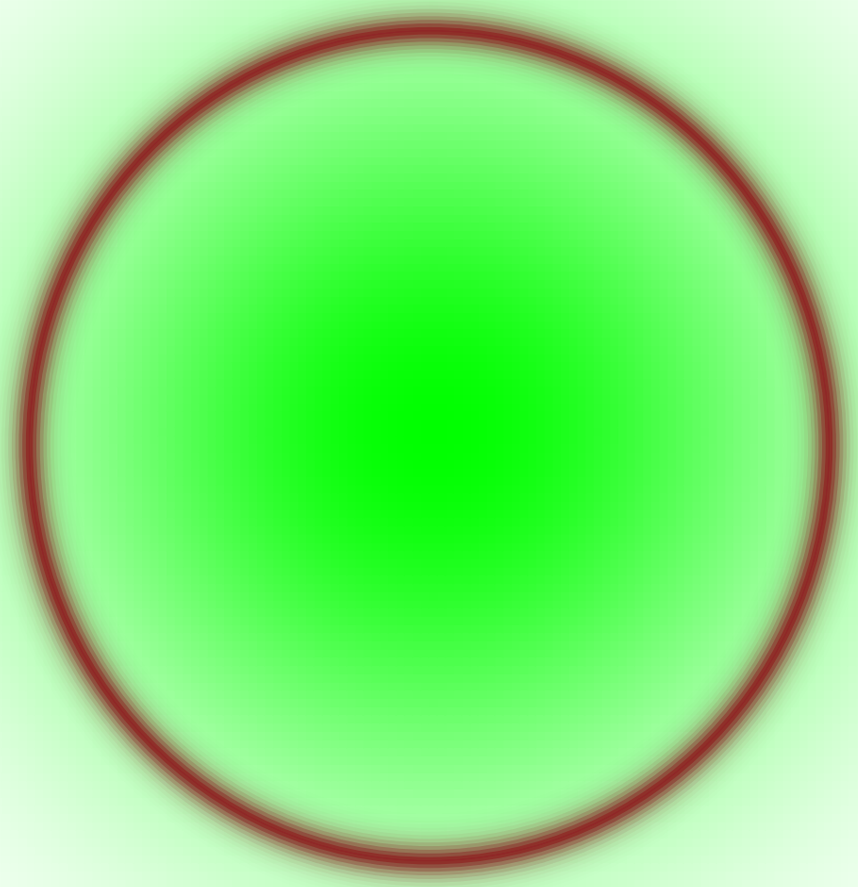
One way to construct the desired exploration is to integrate along a *vector field* aligned with the typical set.



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

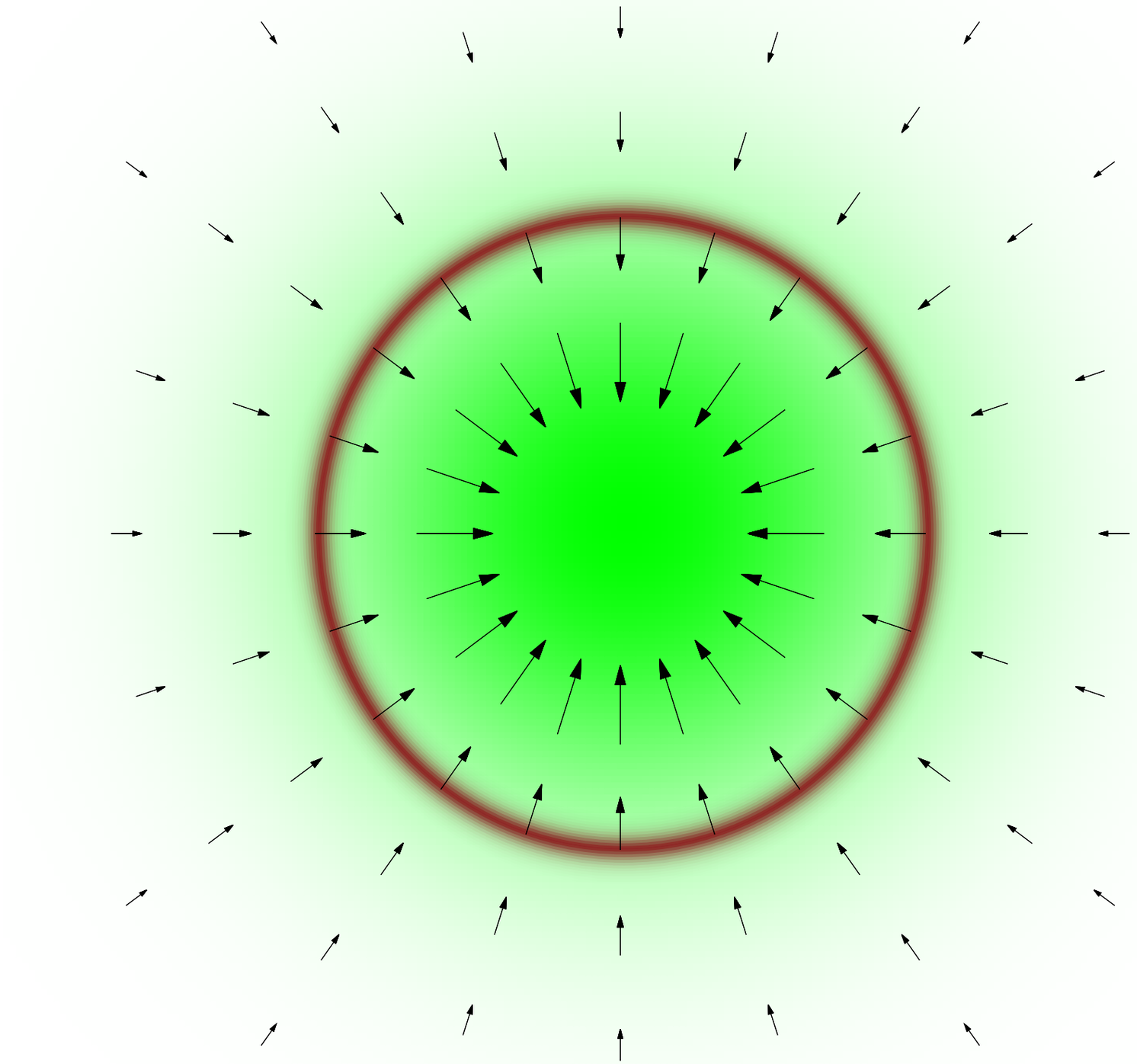


Creating the desired vector field requires transforming available vector fields, such as the gradient.



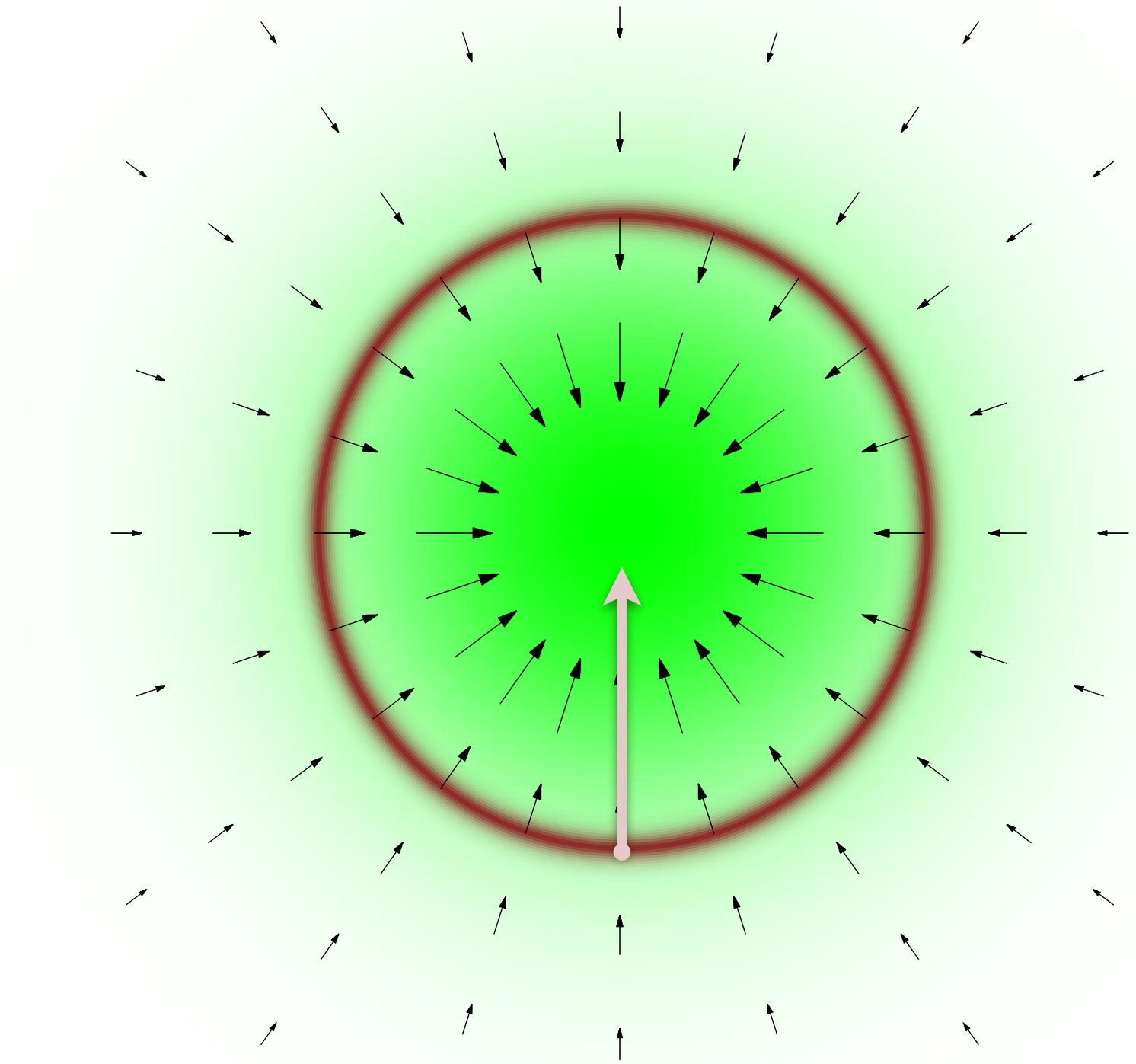
$\pi(\theta)$

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



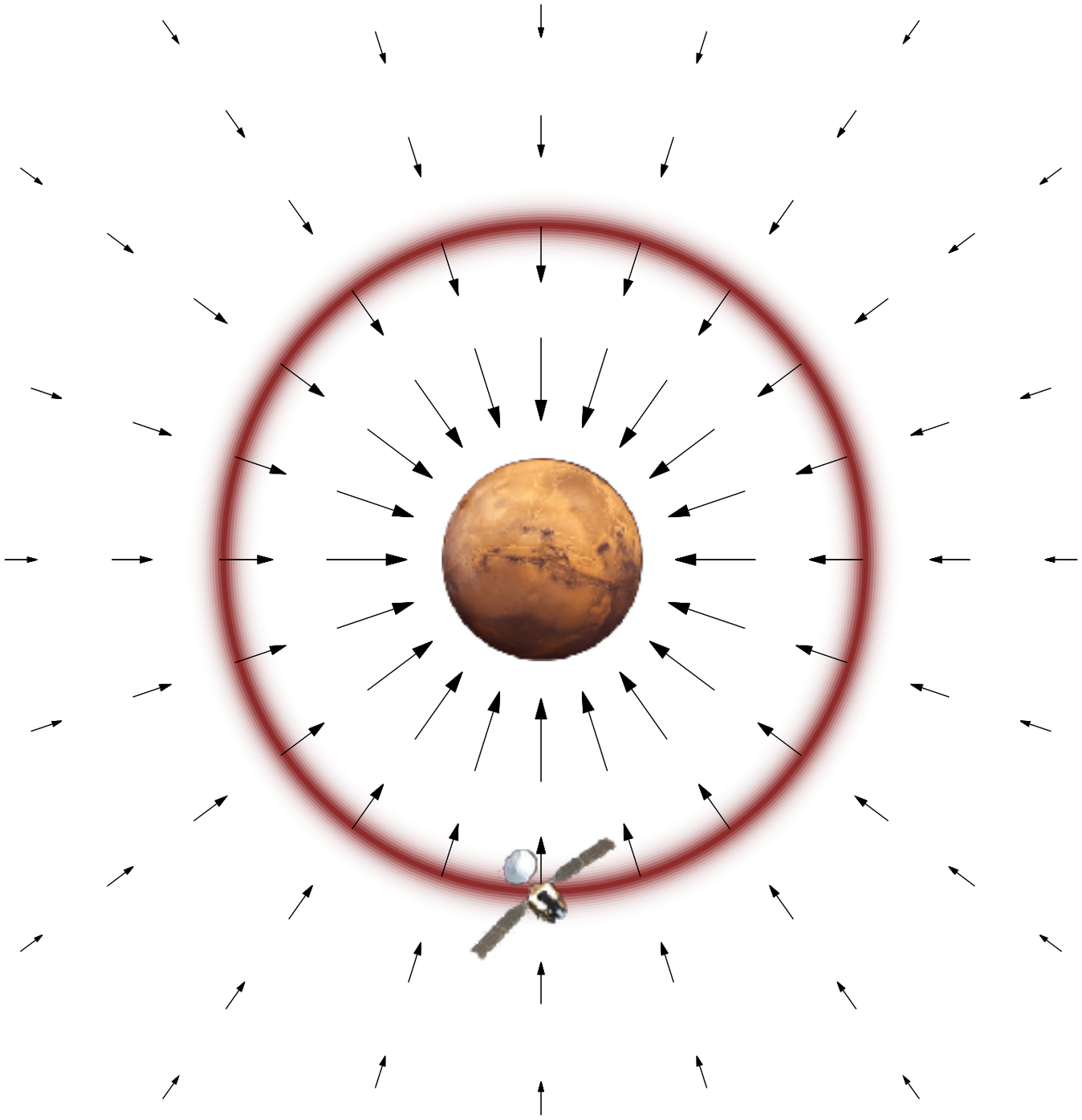
$$\frac{\partial \pi(\theta)}{\partial \theta}$$

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

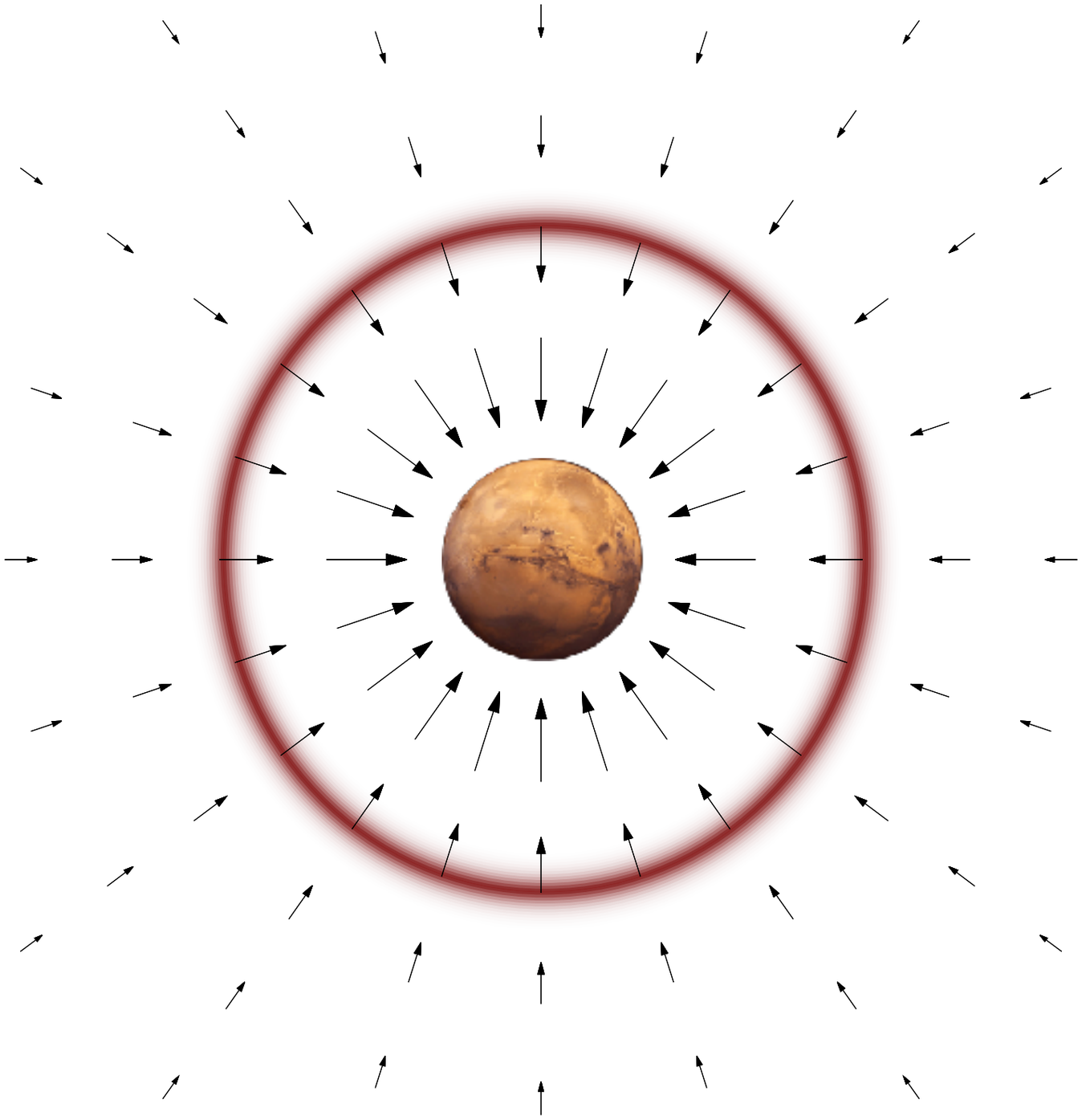


$$\frac{\partial \pi(\theta)}{\partial \theta}$$

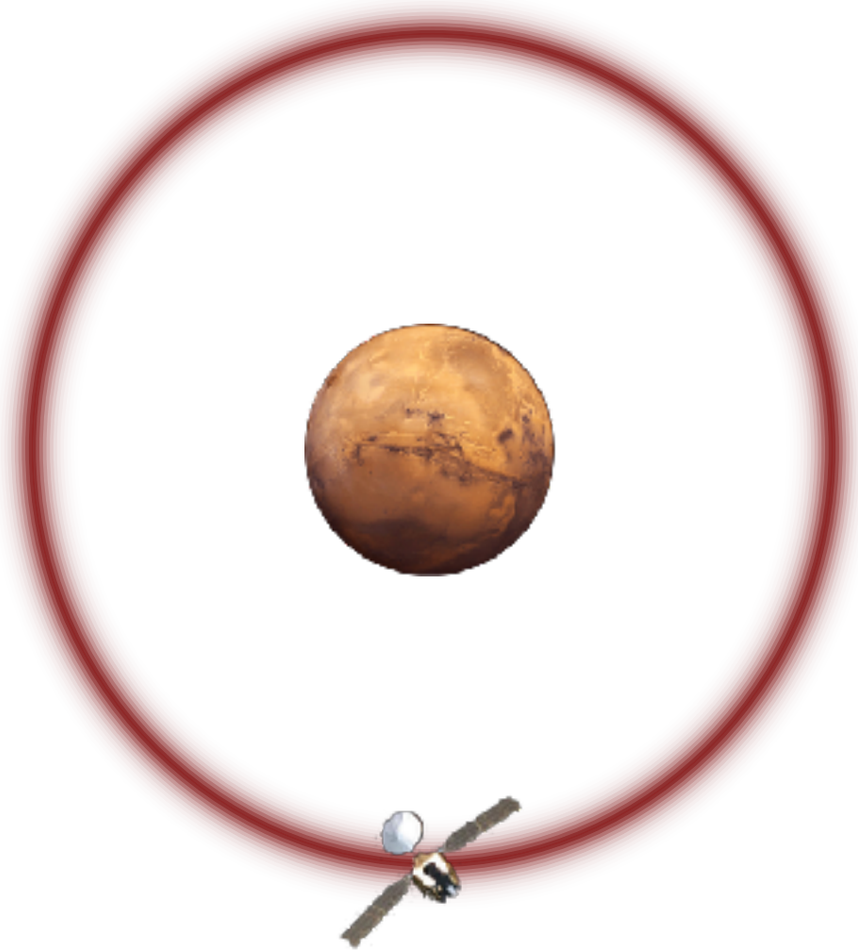
Differential geometry informs this transformation, although a physical analogy can be more intuitive.



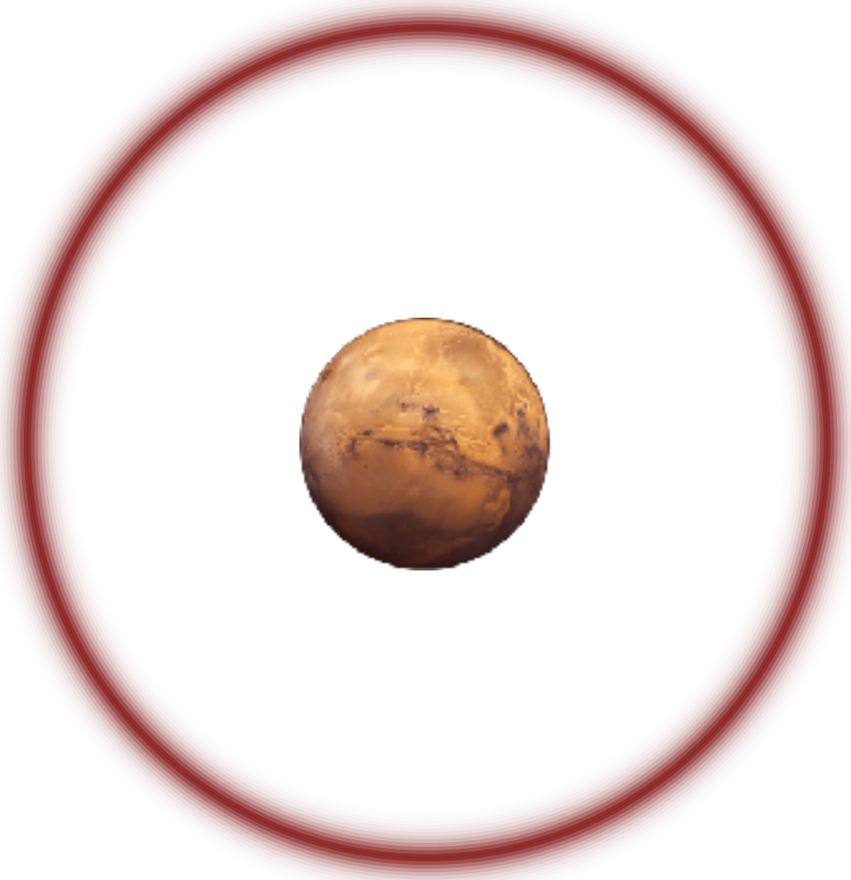
Differential geometry informs this transformation, although a physical analogy can be more intuitive.



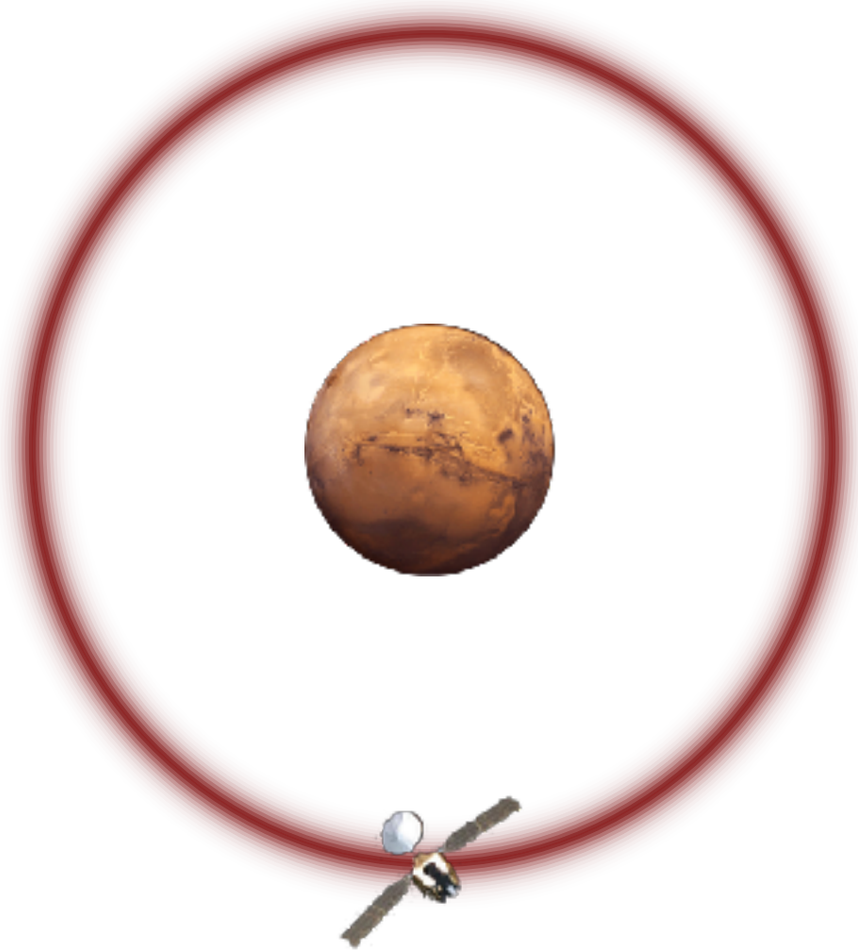
We need to add *momentum* in just the right way.
Too little and we still crash into the planet.



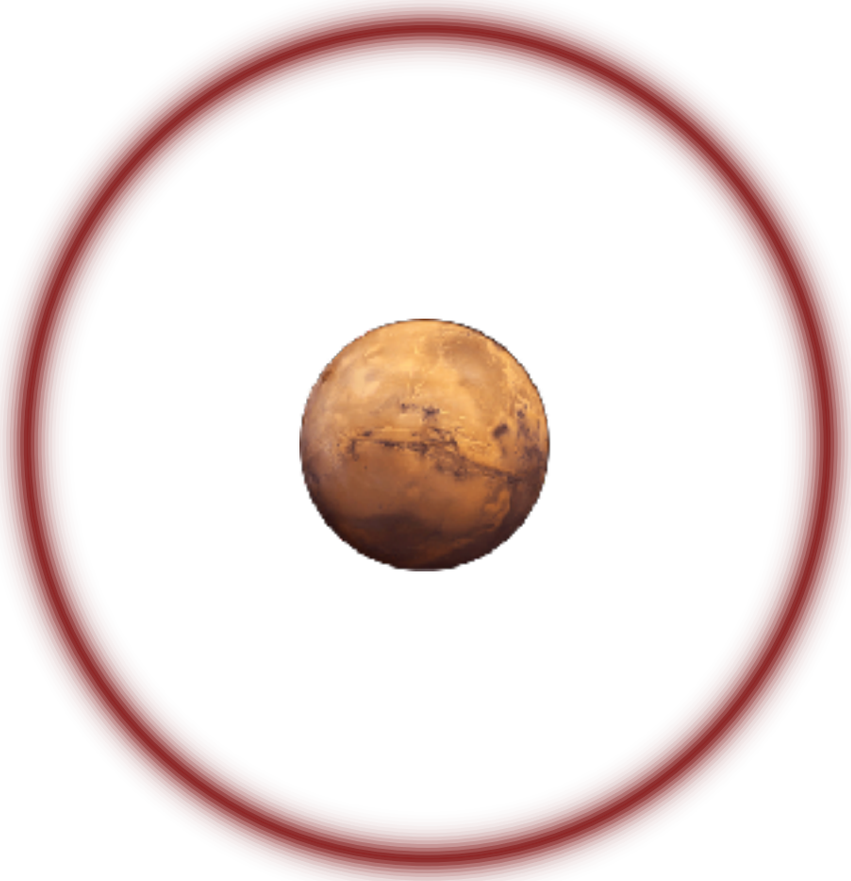
We need to add *momentum* in just the right way.
Too little and we still crash into the planet.



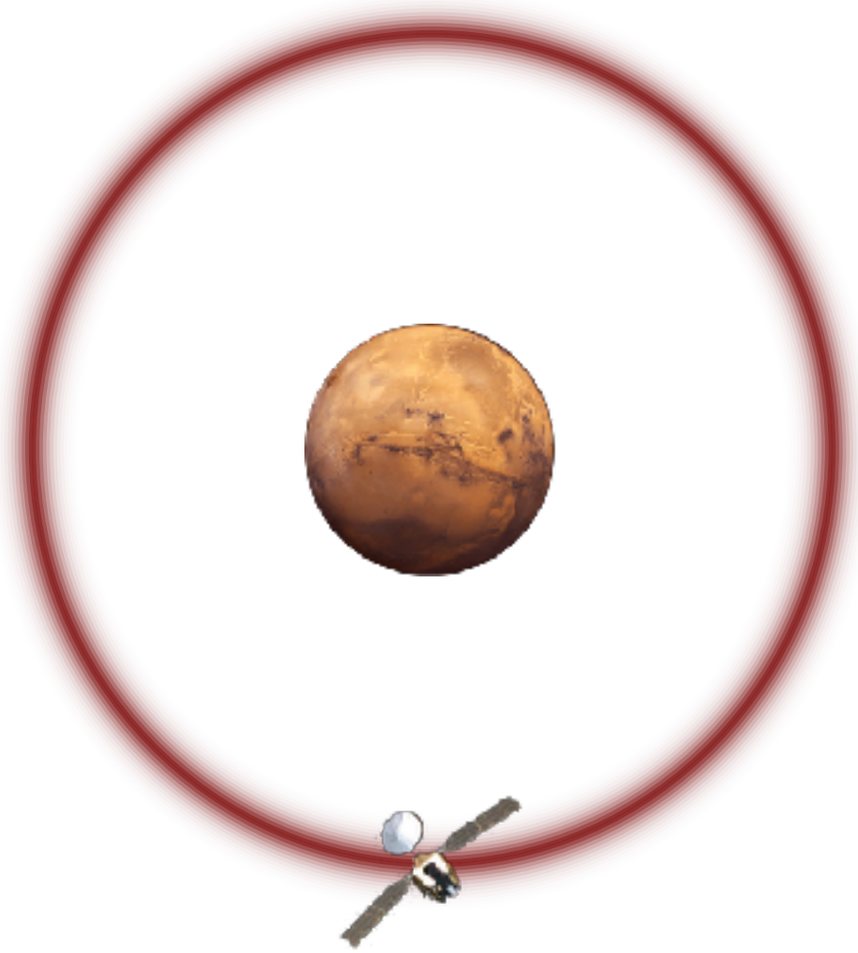
Too much and we fly off to infinity.



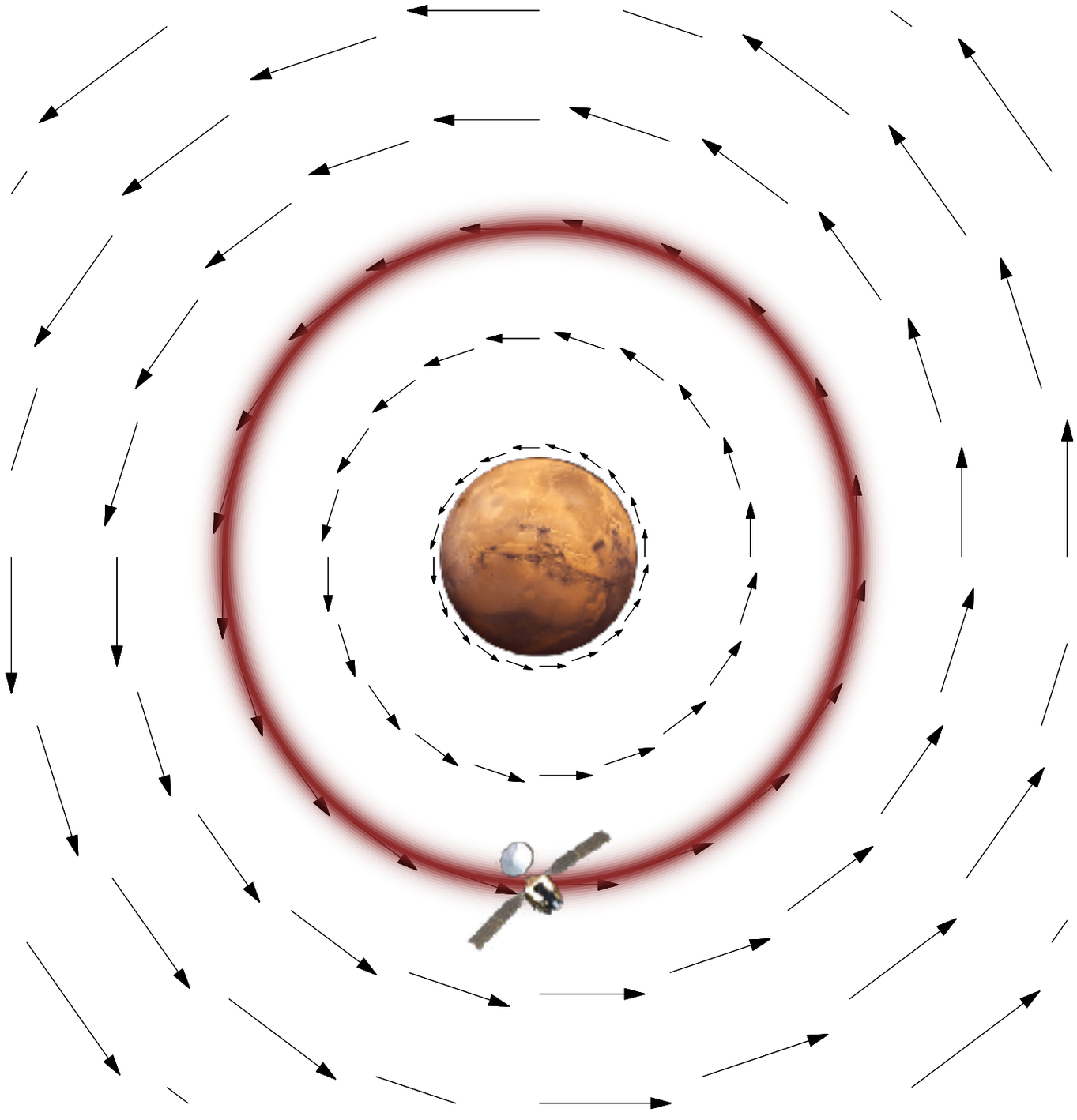
Too much and we fly off to infinity.



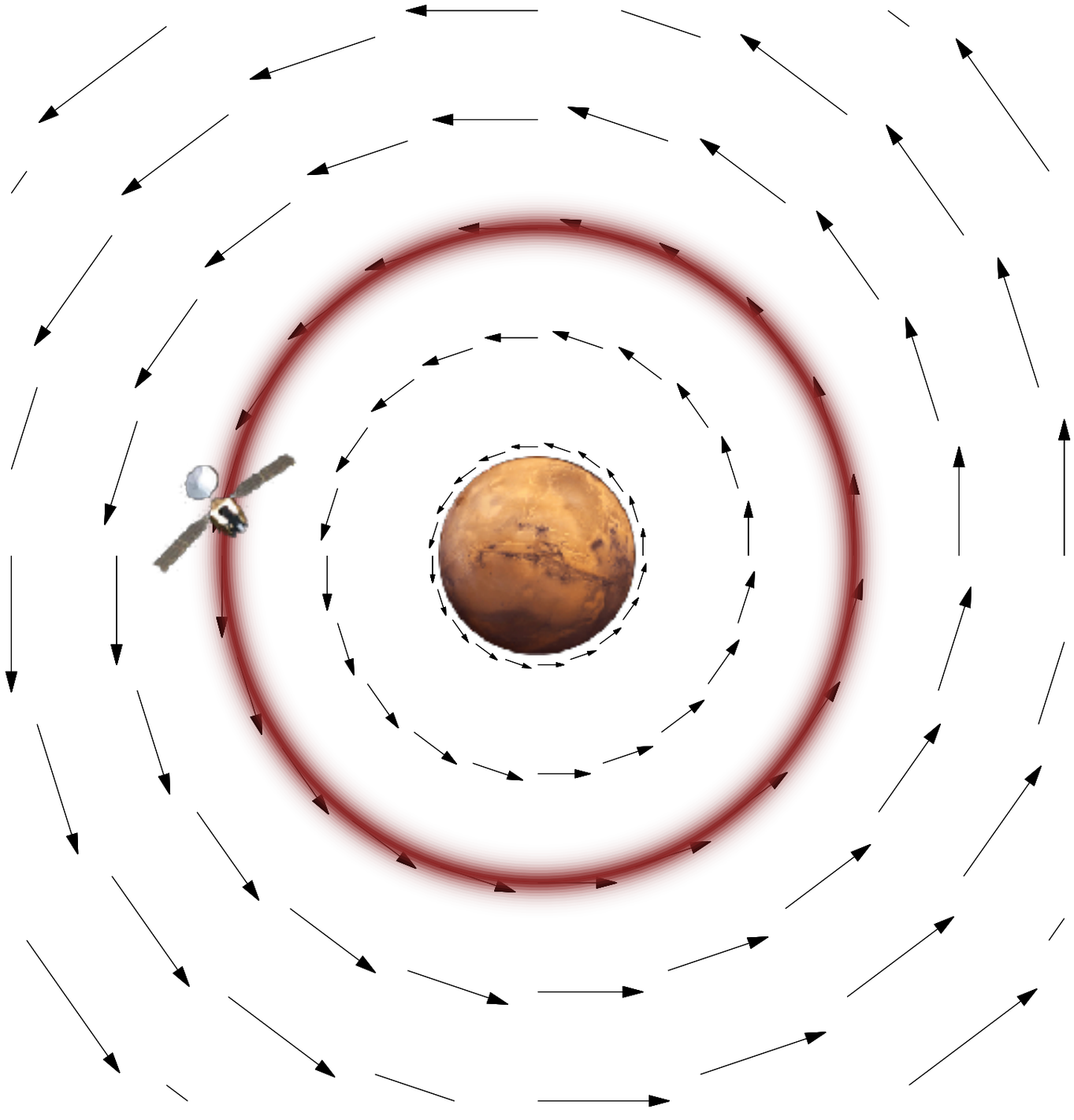
Just enough, however, aligns the gradients with the typical set and yields the desired orbital trajectory.



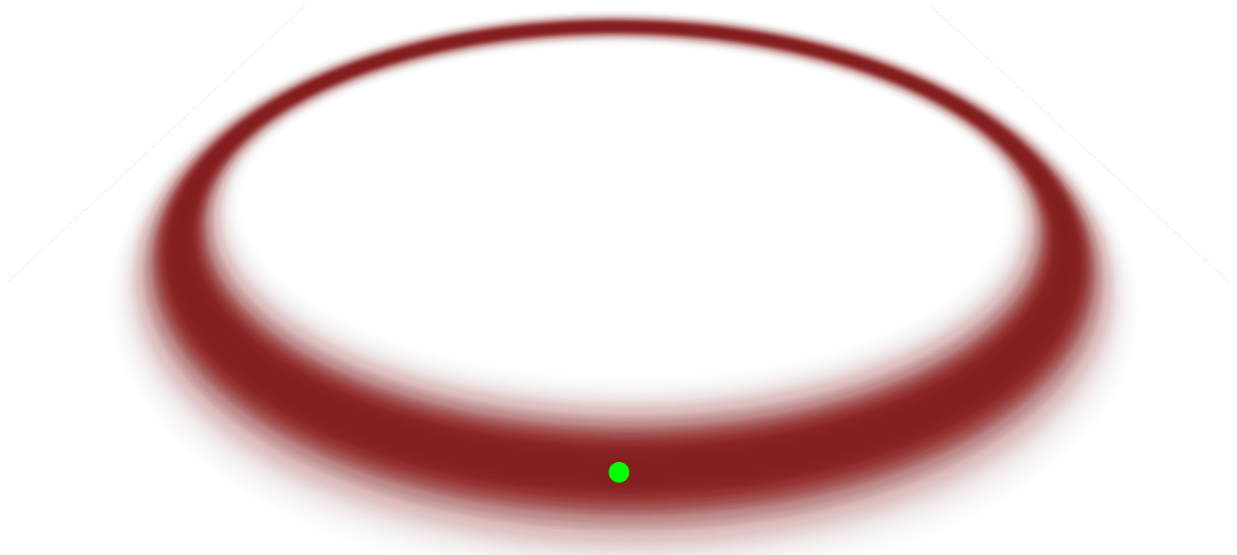
Just enough, however, aligns the gradients with the typical set and yields the desired orbital trajectory.



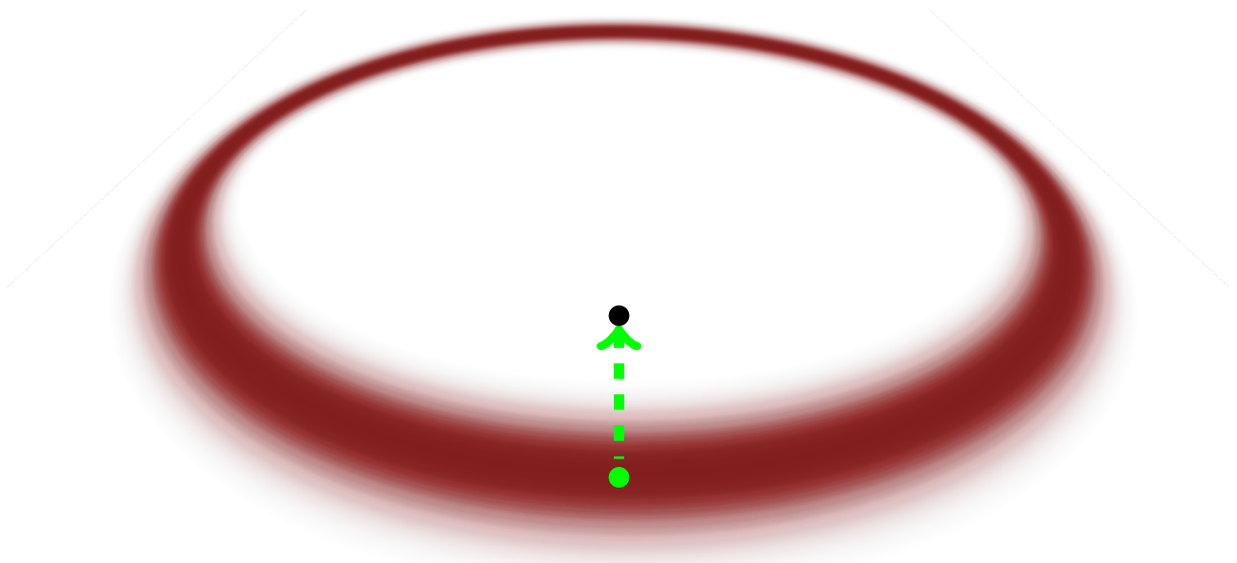
Just enough, however, aligns the gradients with the typical set and yields the desired orbital trajectory.



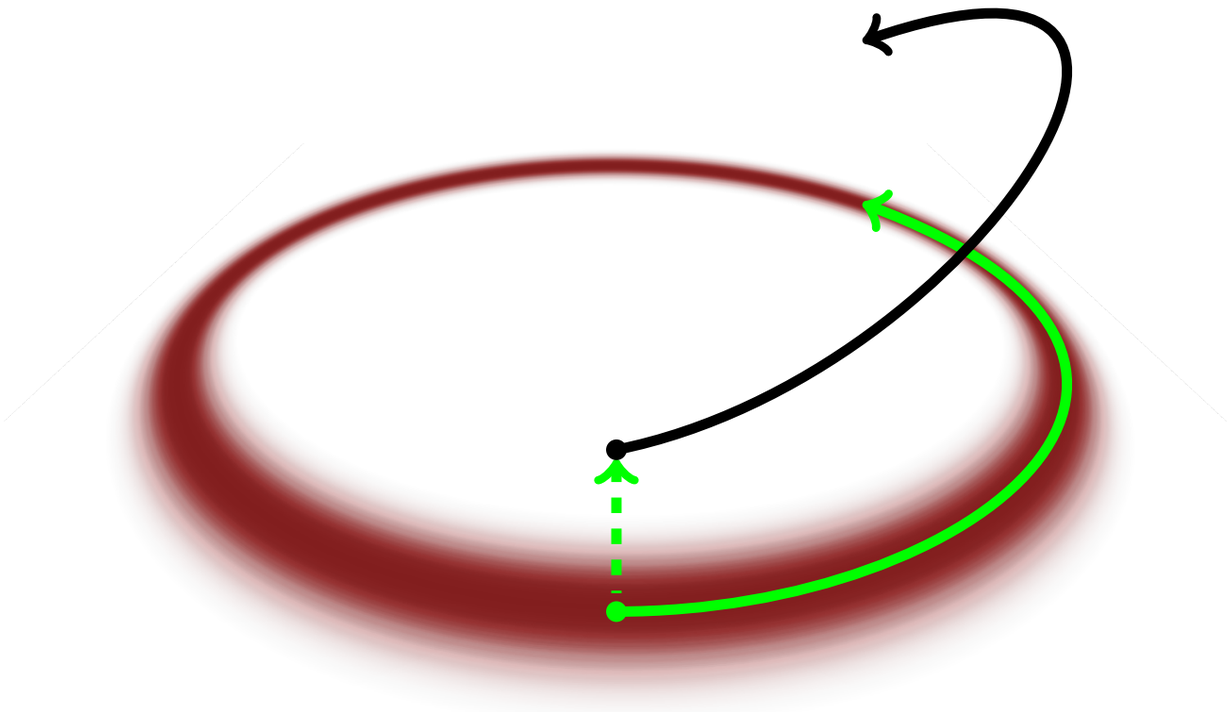
Hamiltonian Monte Carlo yields fast, and *robust,* exploration of the distributions common in practice.



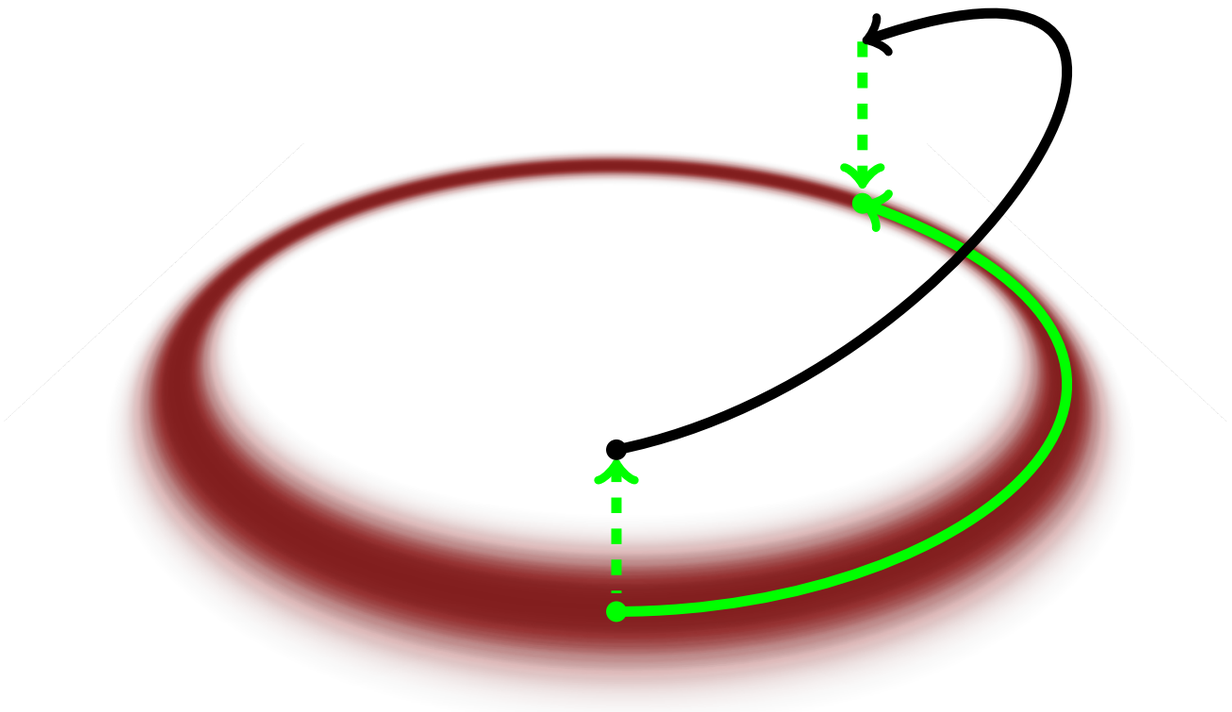
Hamiltonian Monte Carlo yields fast, and *robust,* exploration of the distributions common in practice.



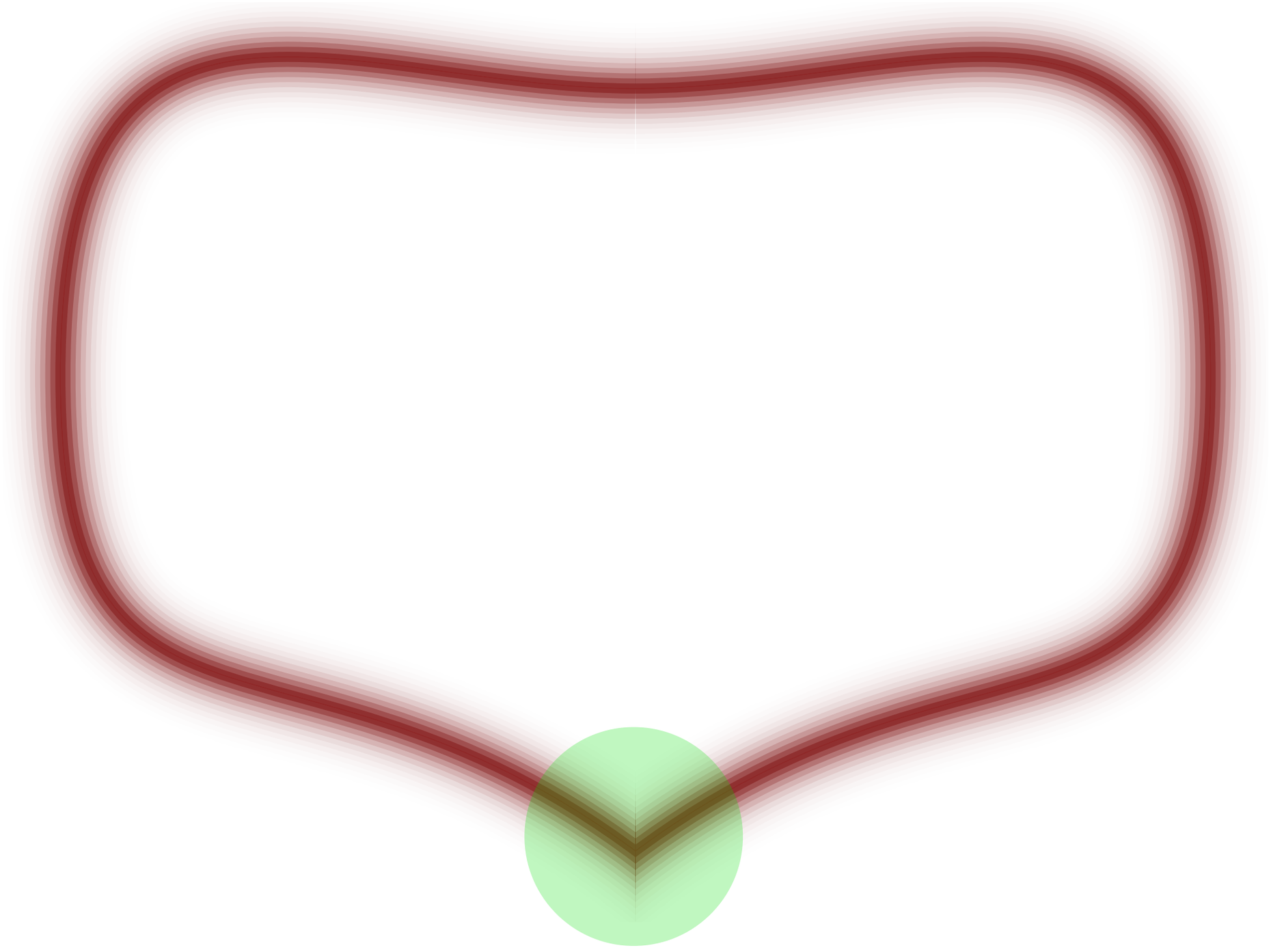
Hamiltonian Monte Carlo yields fast, and *robust,* exploration of the distributions common in practice.



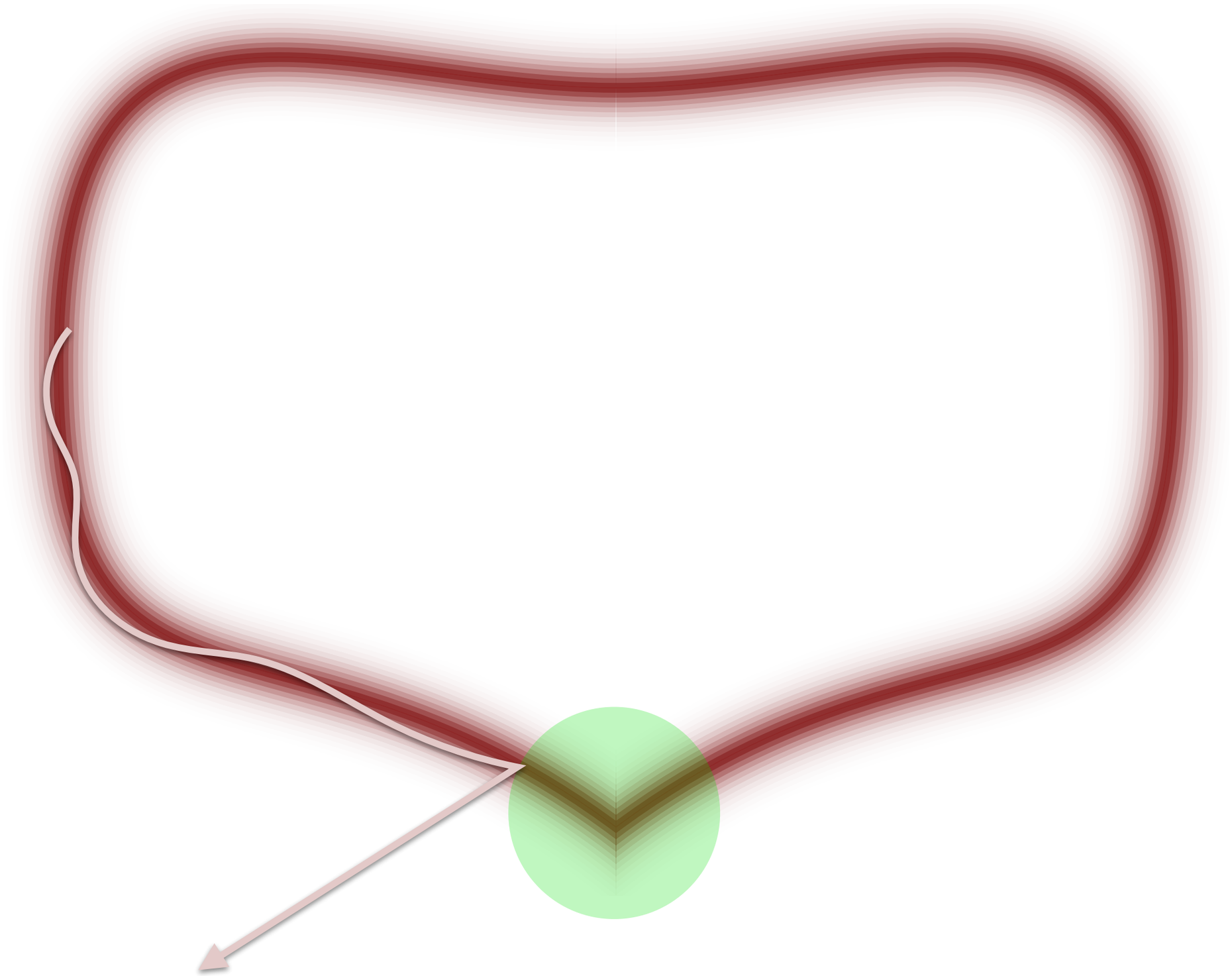
Hamiltonian Monte Carlo yields fast, and *robust,* exploration of the distributions common in practice.



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.



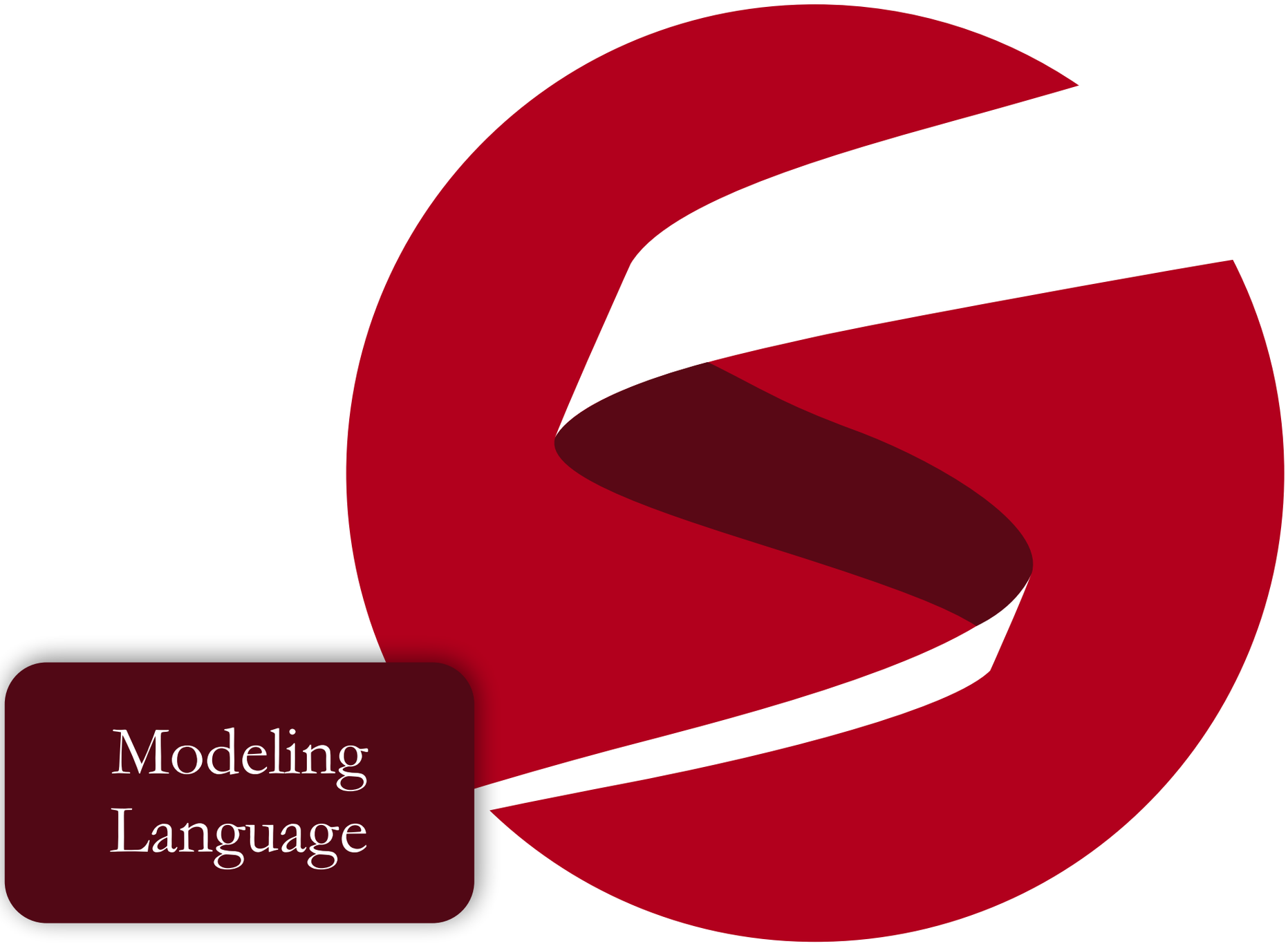
These distinctive behaviors can then be used to construct novel diagnostics of geometric ergodicity.

```
Warning messages:
1: There were 2 chains where the estimated Bayesian
Fraction of Missing Information was low. See
http://mc-stan.org/misc/warnings.html#bfmi-low
```

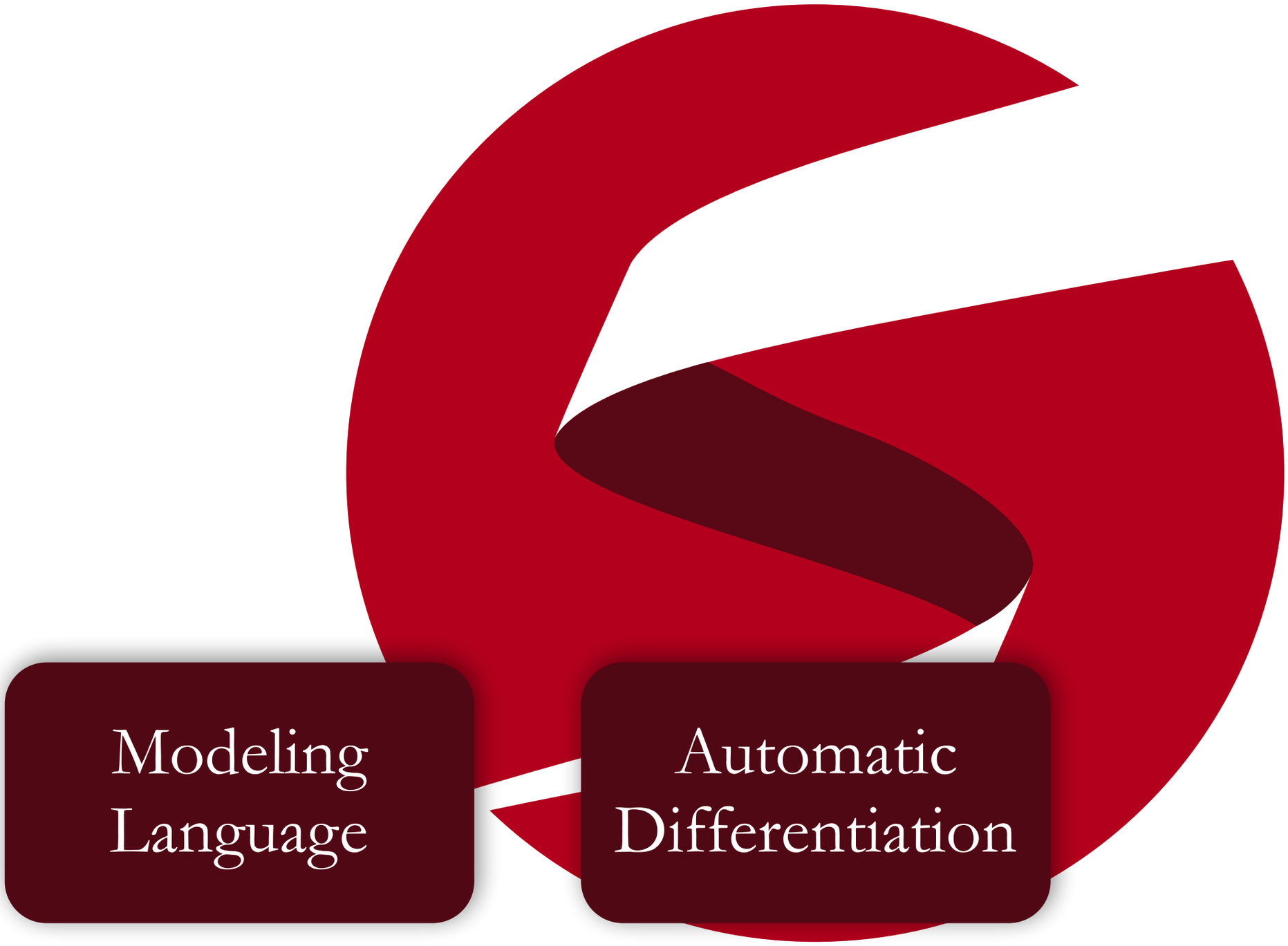
```
Warning messages:
1: There were 2 divergent transitions after warmup.
Increasing adapt_delta above 0.8 may help. See
http://mc-stan.org/misc/warnings.html#divergent-
transitions-after-warmup
```

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

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



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



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