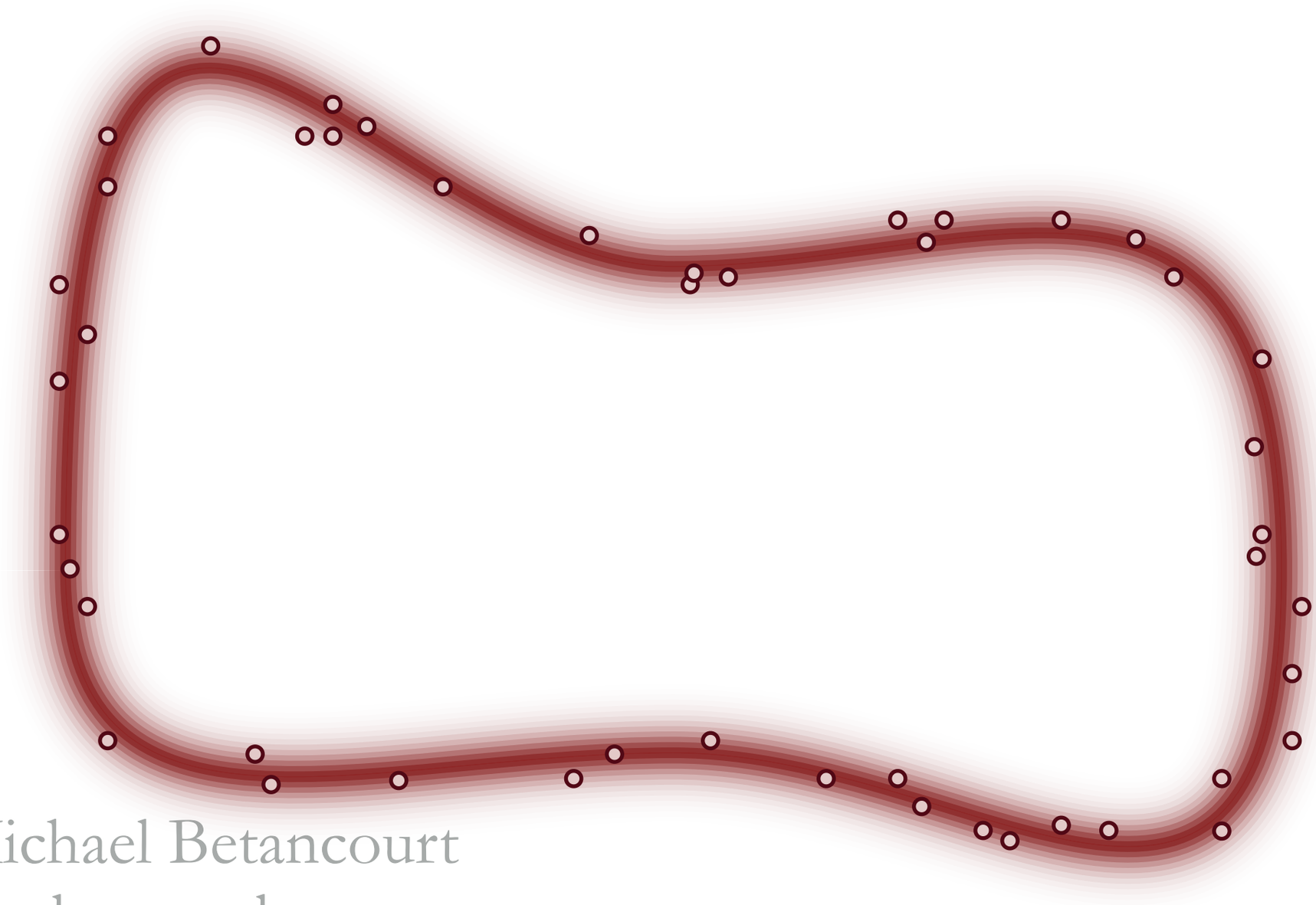
Markov chain Monte Carlo



Michael Betancourt @betanalpha
Symplectomorphic, LLC

Machine Learning Summer School
London, United Kingdom
July 23, 2019

We can also use it to analyze particular algorithms and build intuition about their robustness, or lack thereof.

Deterministic

Modal Estimators

Laplace Estimators

Variational Estimators

...

Stochastic

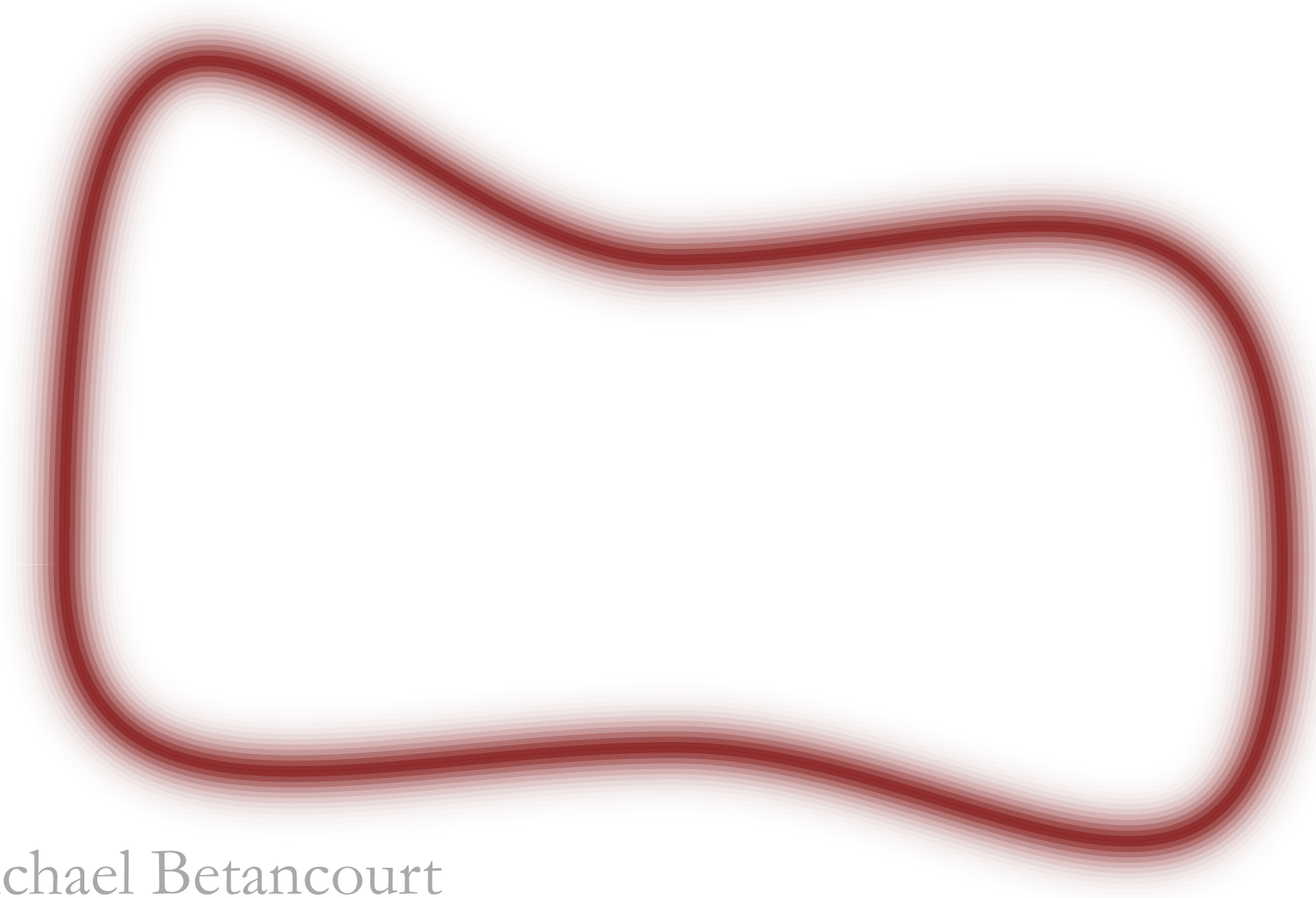
Importance Sampling

Monte Carlo

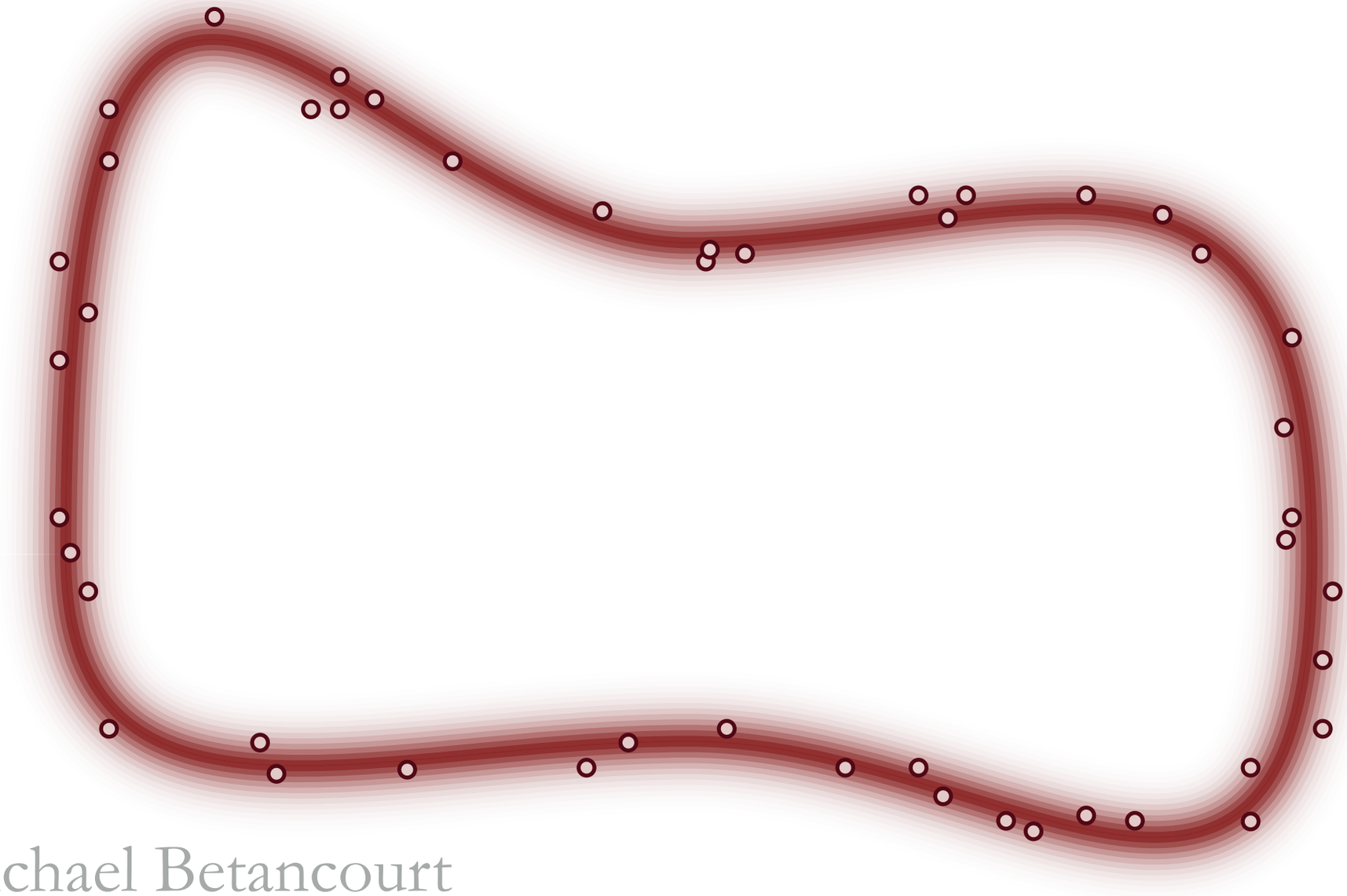
Markov Chain Monte Carlo

...

Monte Carlo methods quantify the typical set using *exact* samples drawn from the target distribution.



Monte Carlo methods quantify the typical set using *exact* samples drawn from the target distribution.



Monte Carlo estimators average a given function over these samples to approximate the expectation value.

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

Subject to mild conditions on the true expectation values these estimators enjoy a *central limit theorem*.

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

$$\hat{f} \sim \mathcal{N}(\mathbb{E}[f], \text{MC-SE}[f])$$

Subject to mild conditions on the true expectation values these estimators enjoy a *central limit theorem*.

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

$$\hat{f} \sim \mathcal{N}(\mathbb{E}[f], \text{MC-SE}[f])$$

$$\text{MC-SE}[f] = \sqrt{\frac{\text{Var}[f]}{N}}$$

Subject to mild conditions on the true expectation values these estimators enjoy a *central limit theorem*.

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

$$\hat{f} \sim \mathcal{N}(\mathbb{E}[f], \text{MC-SE}[f])$$

$$\text{MC-SE}[f] = \sqrt{\frac{\text{Var}[f]}{N}}$$

While exact samples are typically impractical to generate we can extend this approach to *Markov chain Monte Carlo*.

Deterministic
Modal Estimators
Laplace Estimators
Variational Estimators
...

Stochastic
Importance Sampling
Monte Carlo
Markov Chain Monte Carlo
...

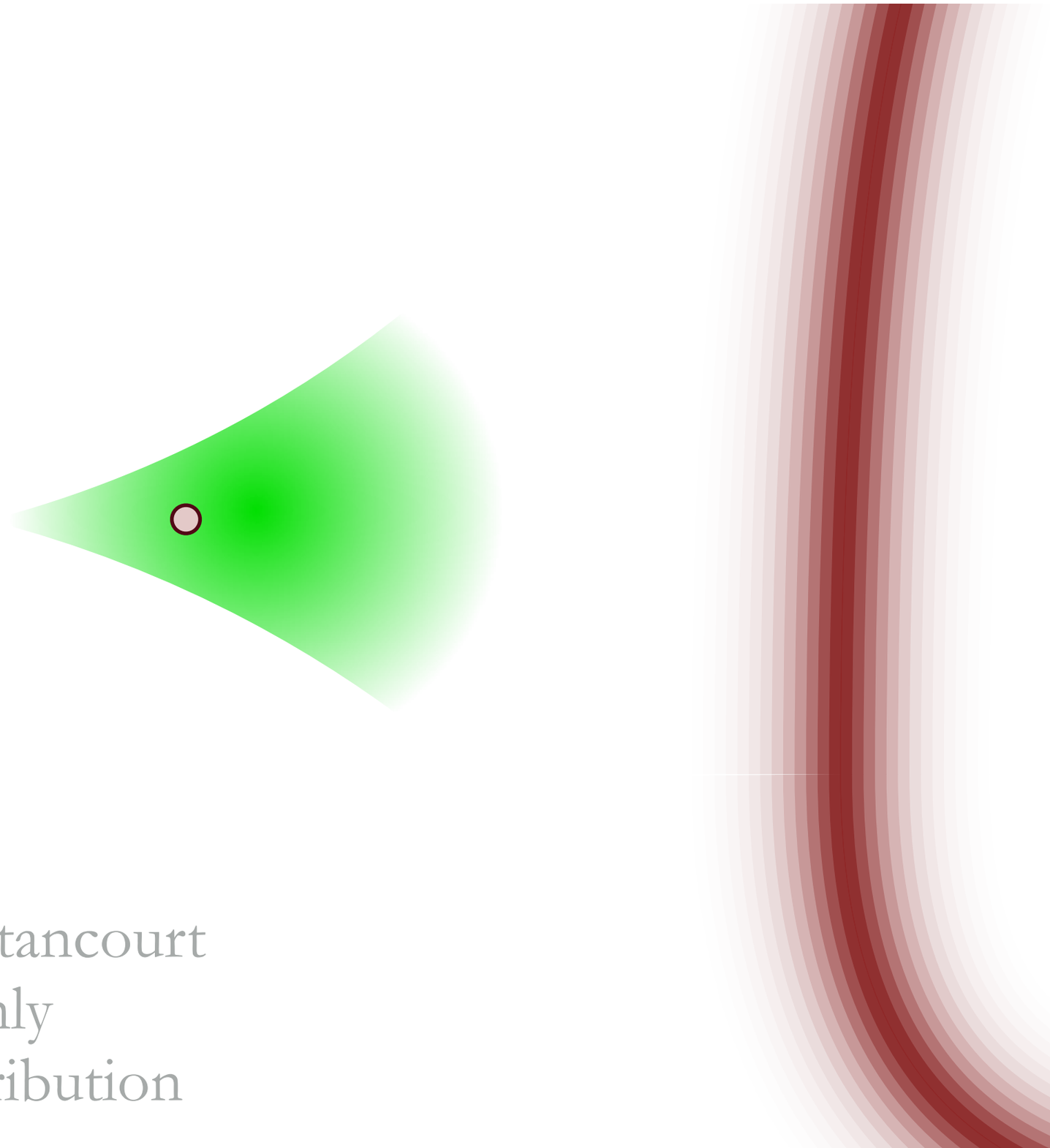
A Markov transition that *targets* a particular distribution naturally concentrates towards its probability mass.

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

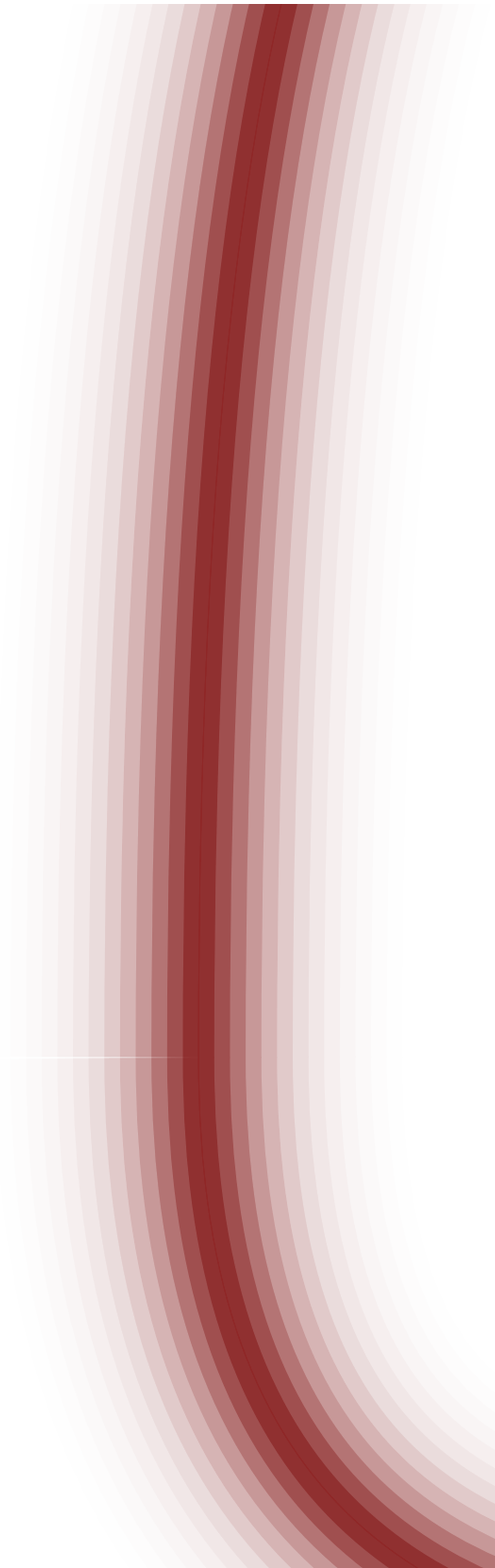
A Markov transition that *targets* a particular distribution naturally concentrates towards its probability mass.

$$\pi_S(\theta \mid \tilde{y}) = \int \mathrm{d}\theta' \, \pi_S(\theta' \mid \tilde{y}) \, T(\theta \mid \theta')$$

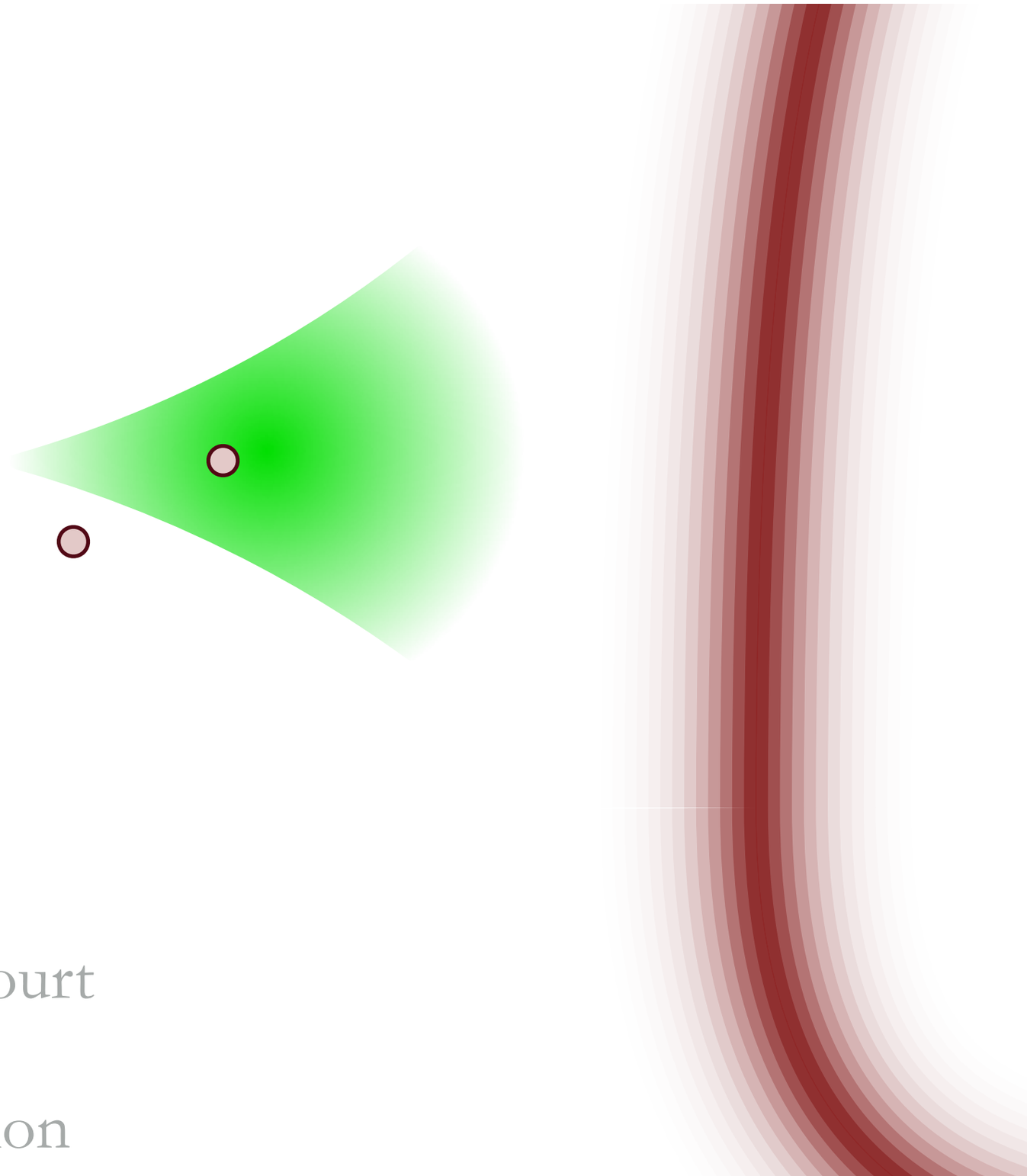
A Markov transition that *targets* a particular distribution naturally concentrates towards its probability mass.



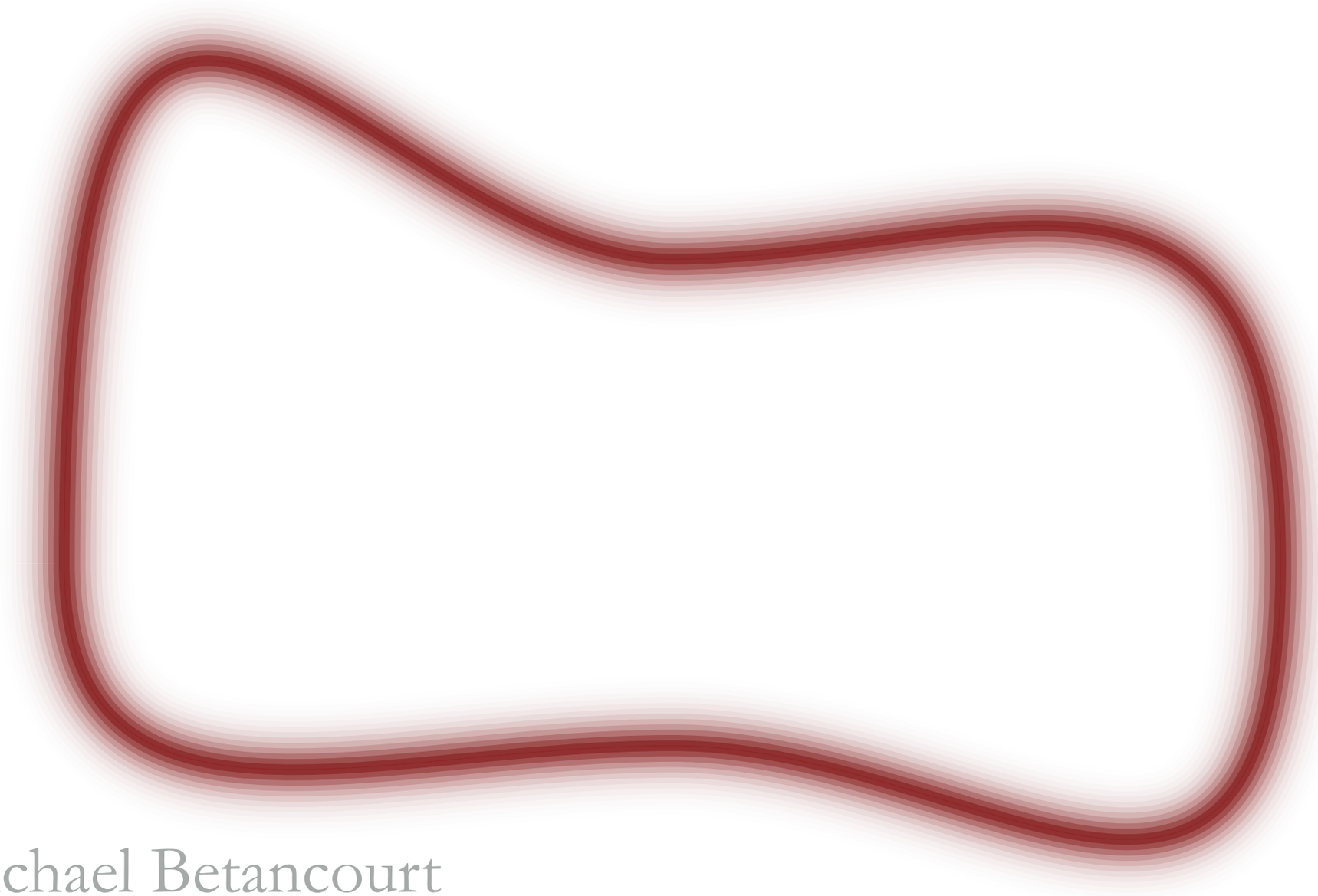
A Markov transition that *targets* a particular distribution naturally concentrates towards its probability mass.



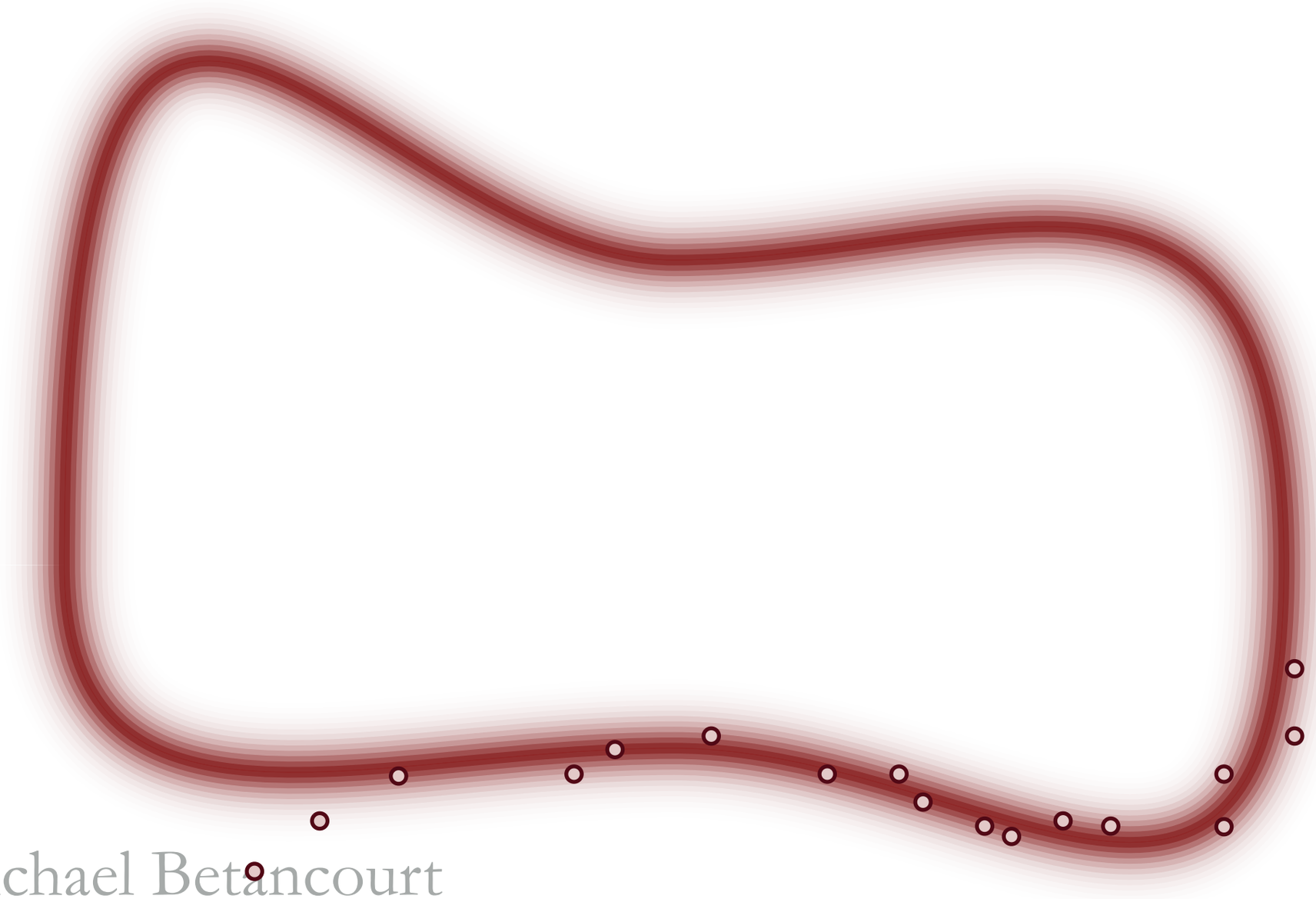
A Markov transition that *targets* a particular distribution naturally concentrates towards its probability mass.



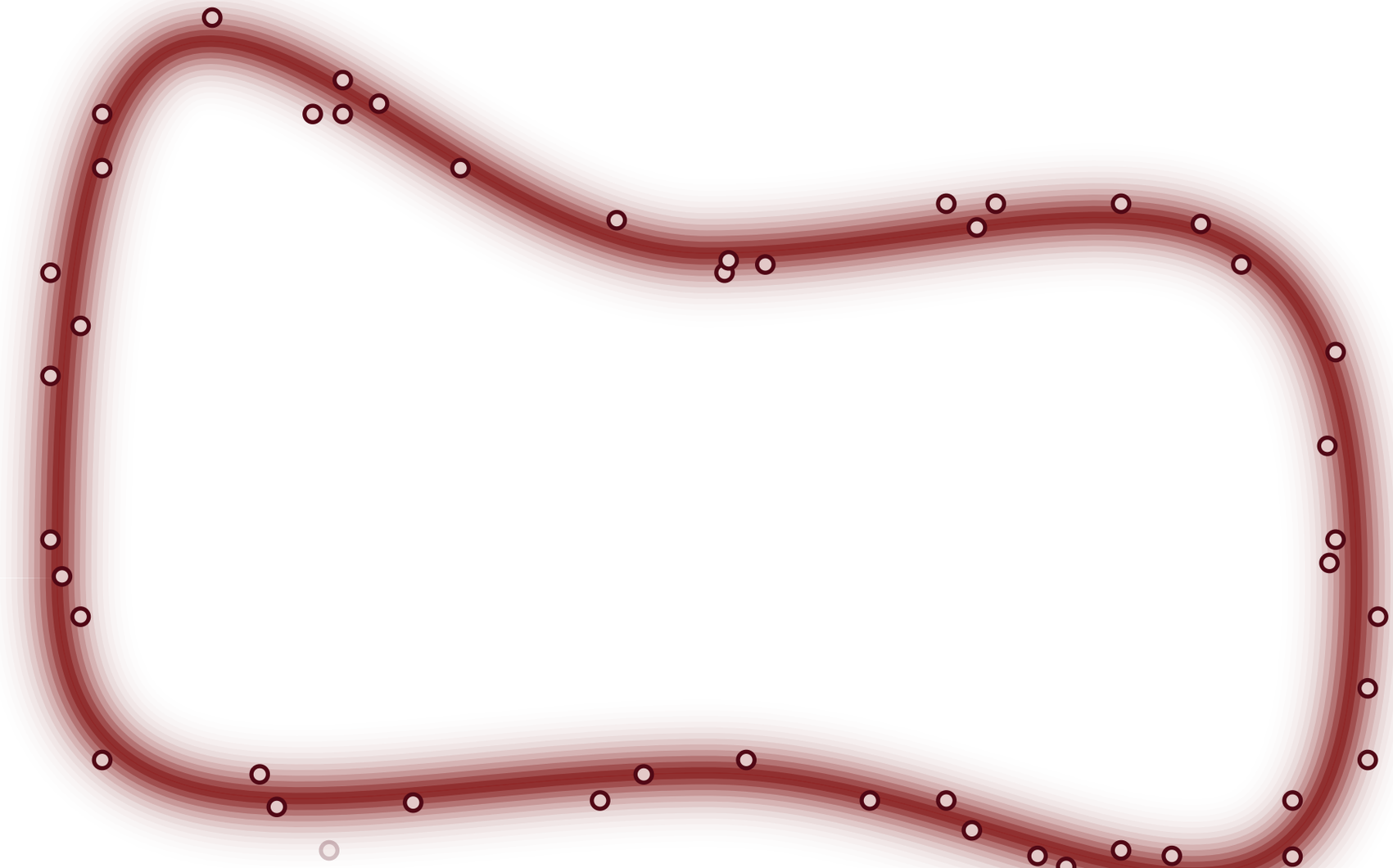
We can then exploit the subsequent *Markov chains* as a generic scheme for finding and exploring typical sets.



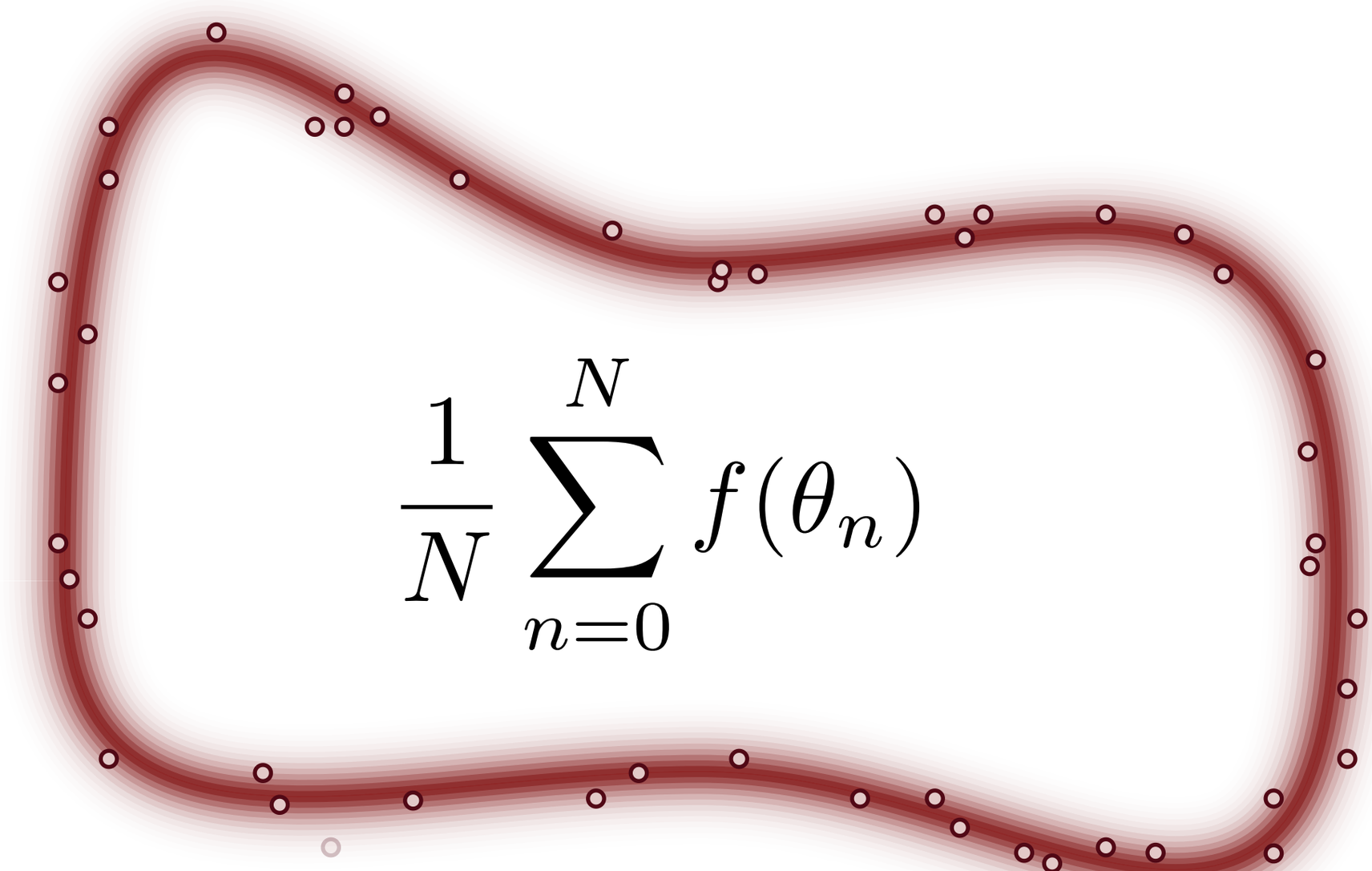
We can then exploit the subsequent *Markov chains* as a generic scheme for finding and exploring typical sets.



Each Markov chain defines an asymptotically consistent *Markov Chain Monte Carlo estimator*.

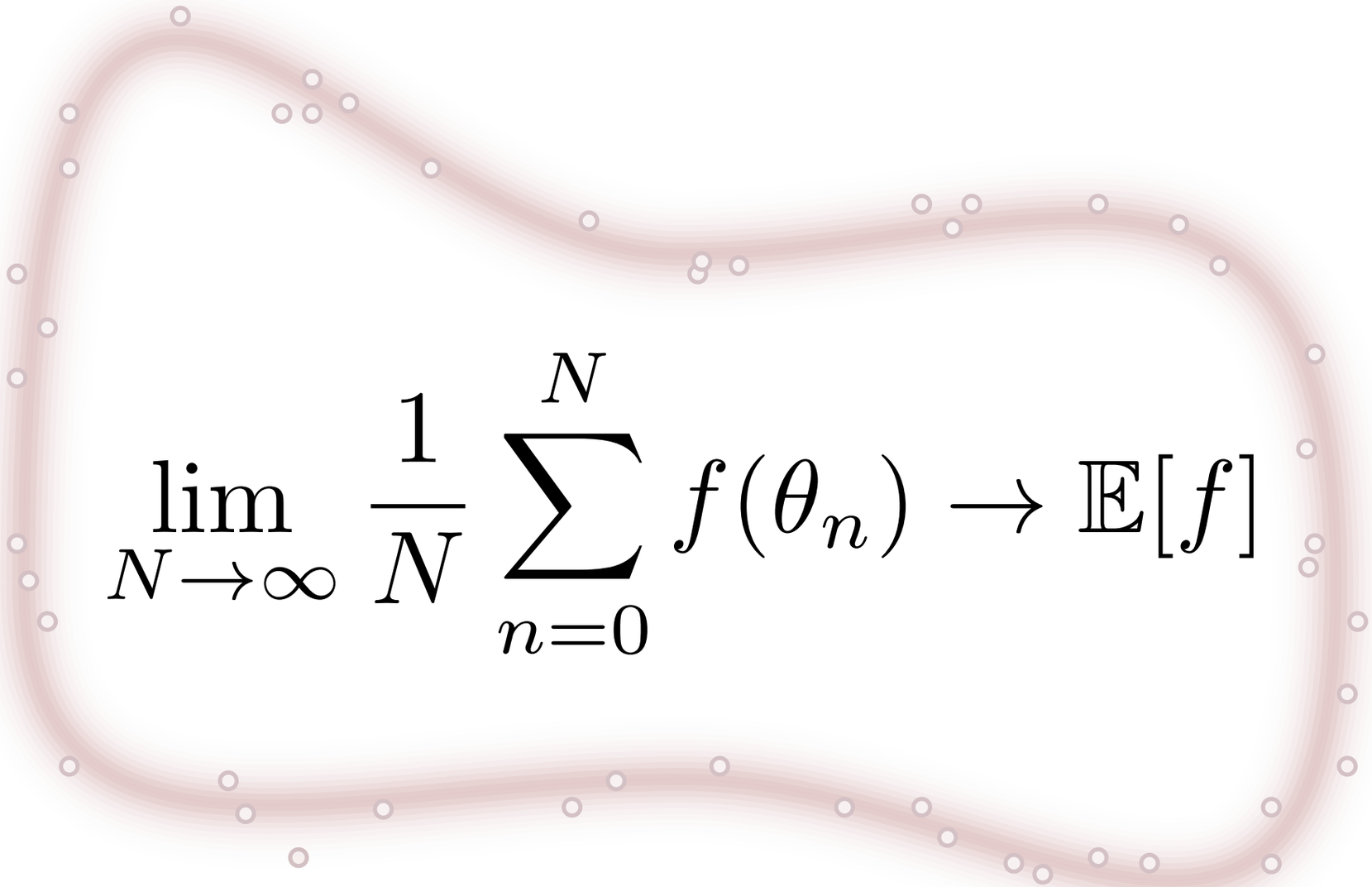


Each Markov chain defines an asymptotically consistent *Markov Chain Monte Carlo estimator*.



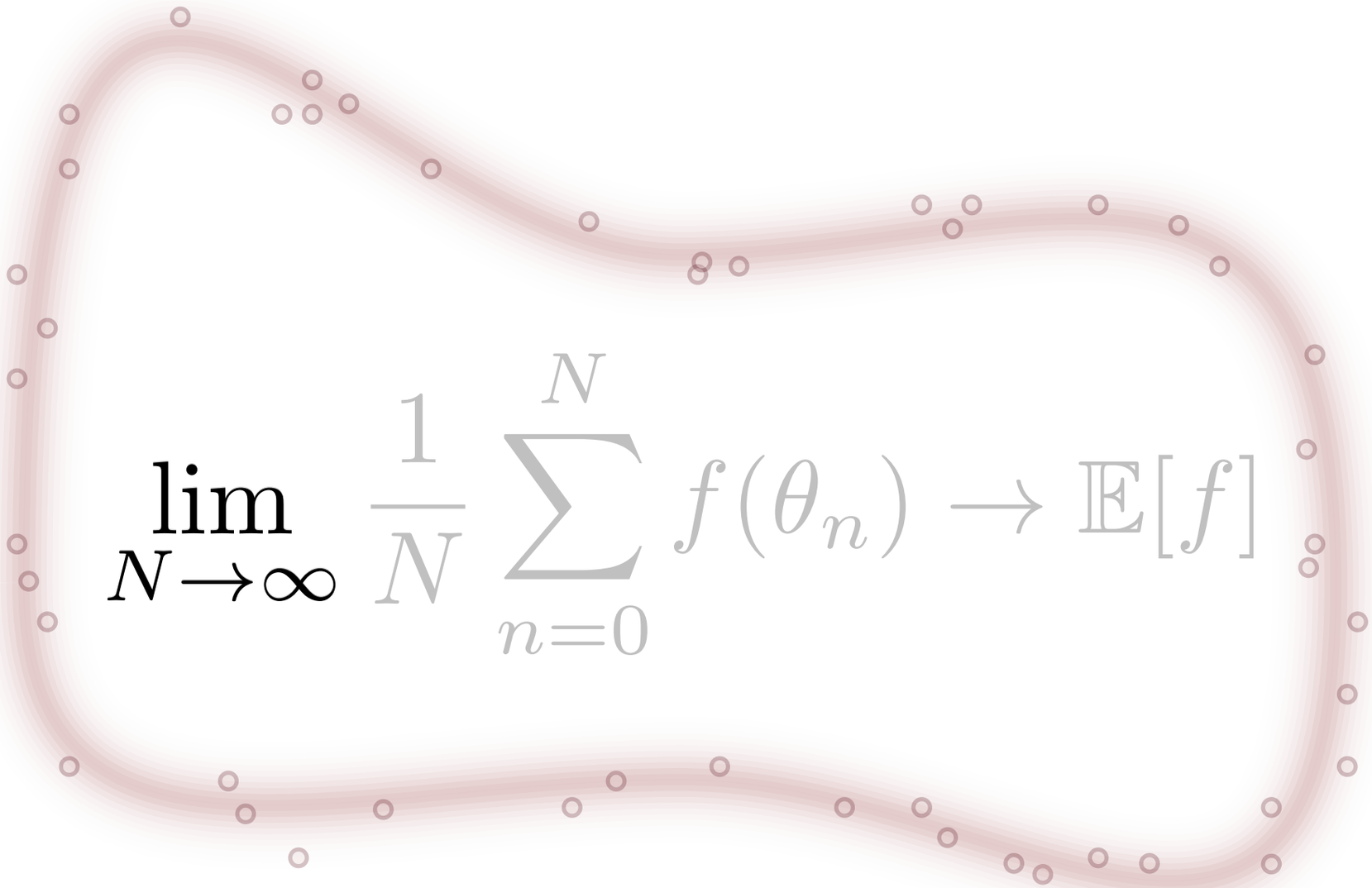
$$\frac{1}{N} \sum_{n=0}^{N} f(\theta_n)$$

Each Markov chain defines an asymptotically consistent *Markov Chain Monte Carlo estimator*.



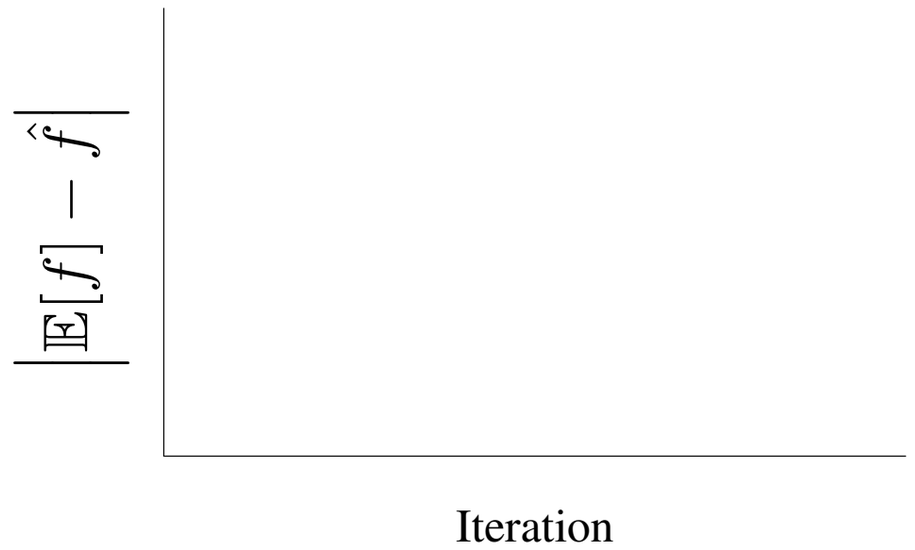
$$\lim_{N \to \infty} \frac{1}{N} \sum_{n=0}^{N} f(\theta_n) \to \mathbb{E}[f]$$

Unfortunately this asymptotic consistency makes no guarantees on the *finite time* behavior of the estimators.

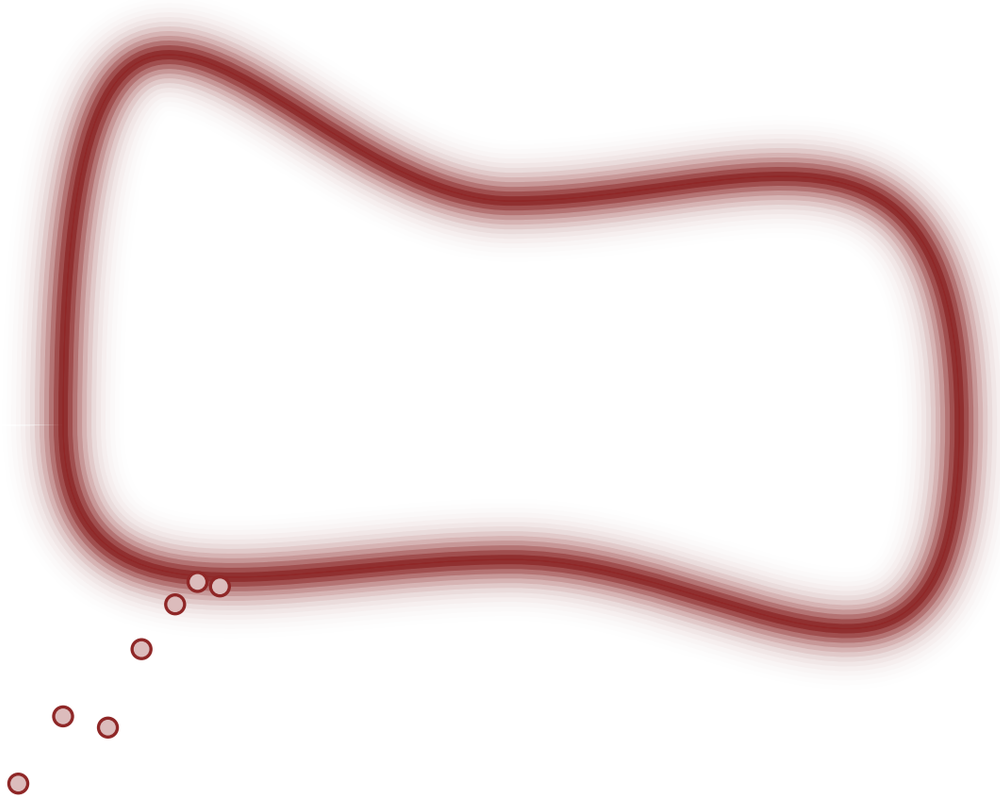


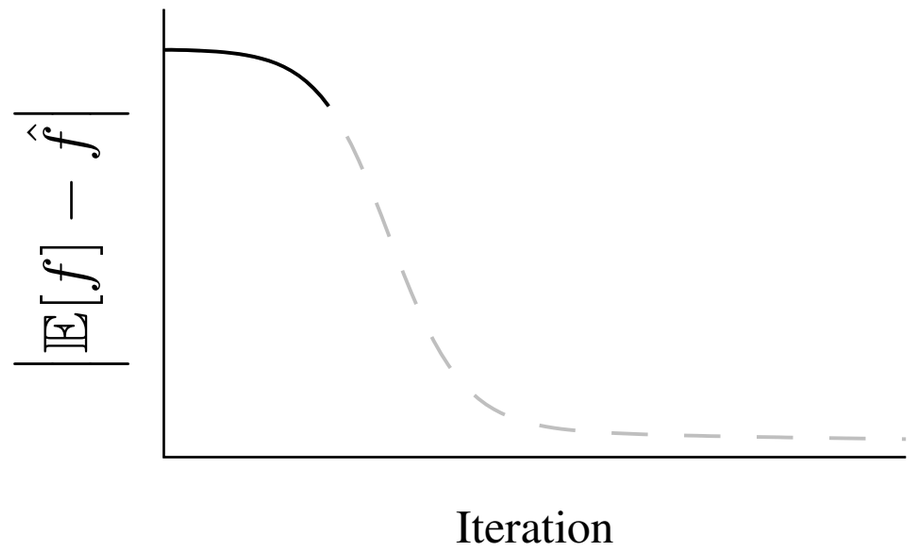
Under ideal conditions, MCMC estimators converge to the true expectations in a very practical progression.

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

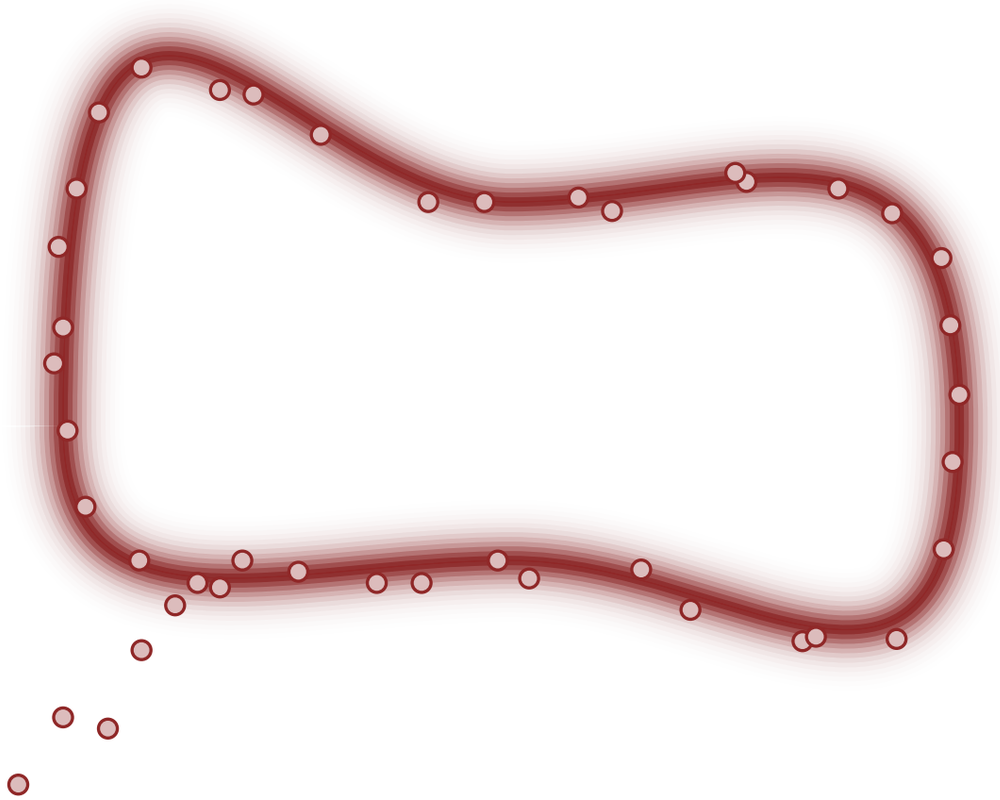


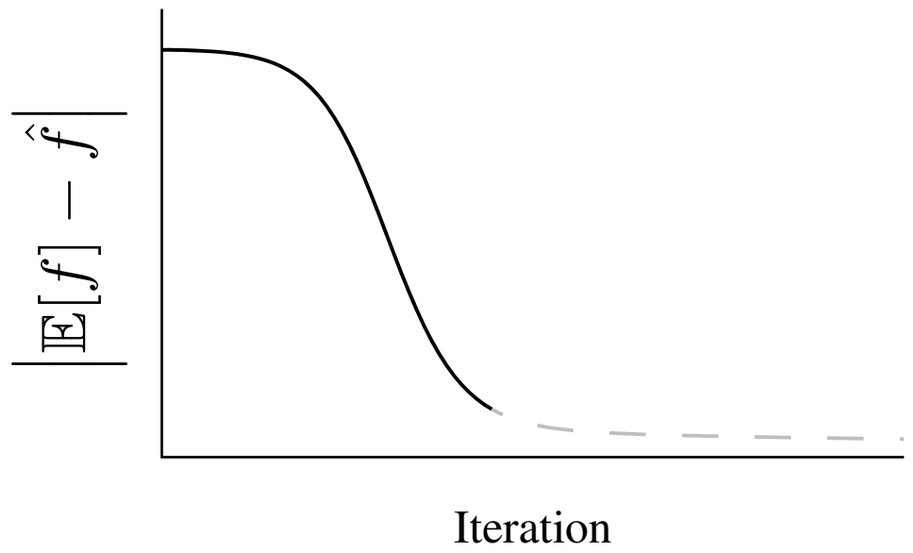
Under ideal conditions MCMC estimators converge to the true expectations in a very practically useful progression.



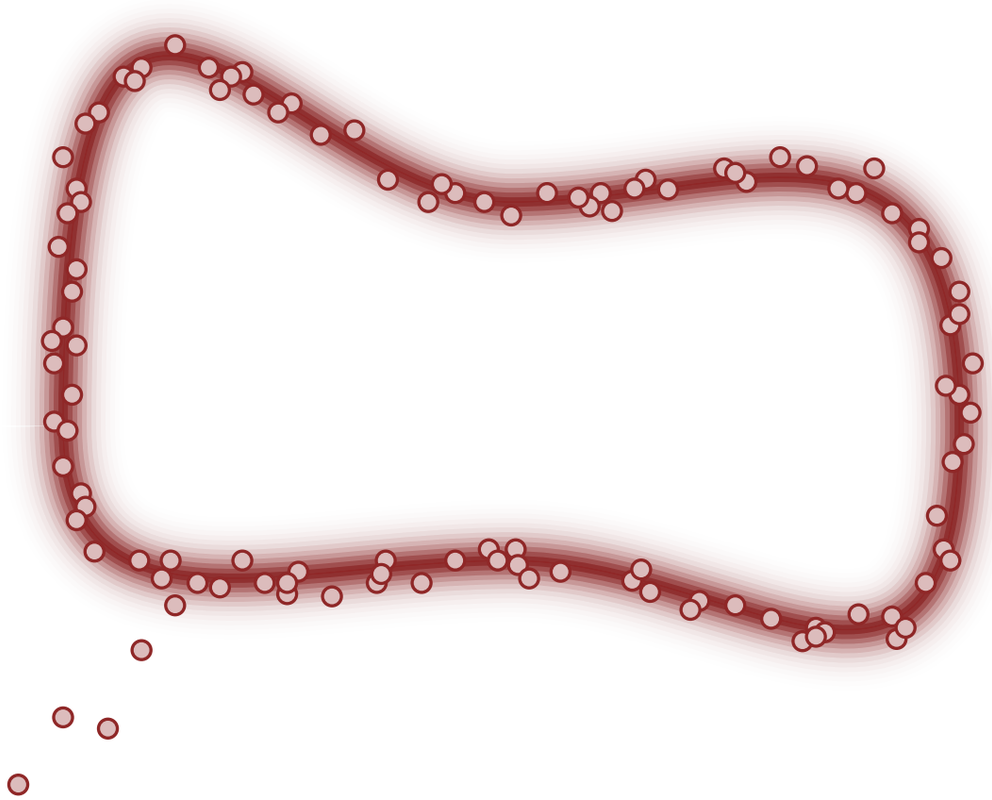


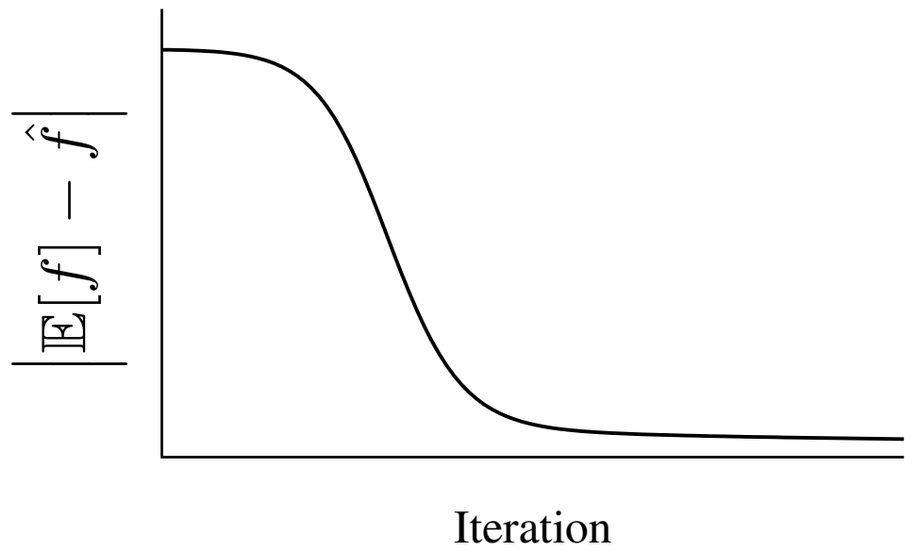
Under ideal conditions MCMC estimators converge to the true expectations in a very practically useful progression.



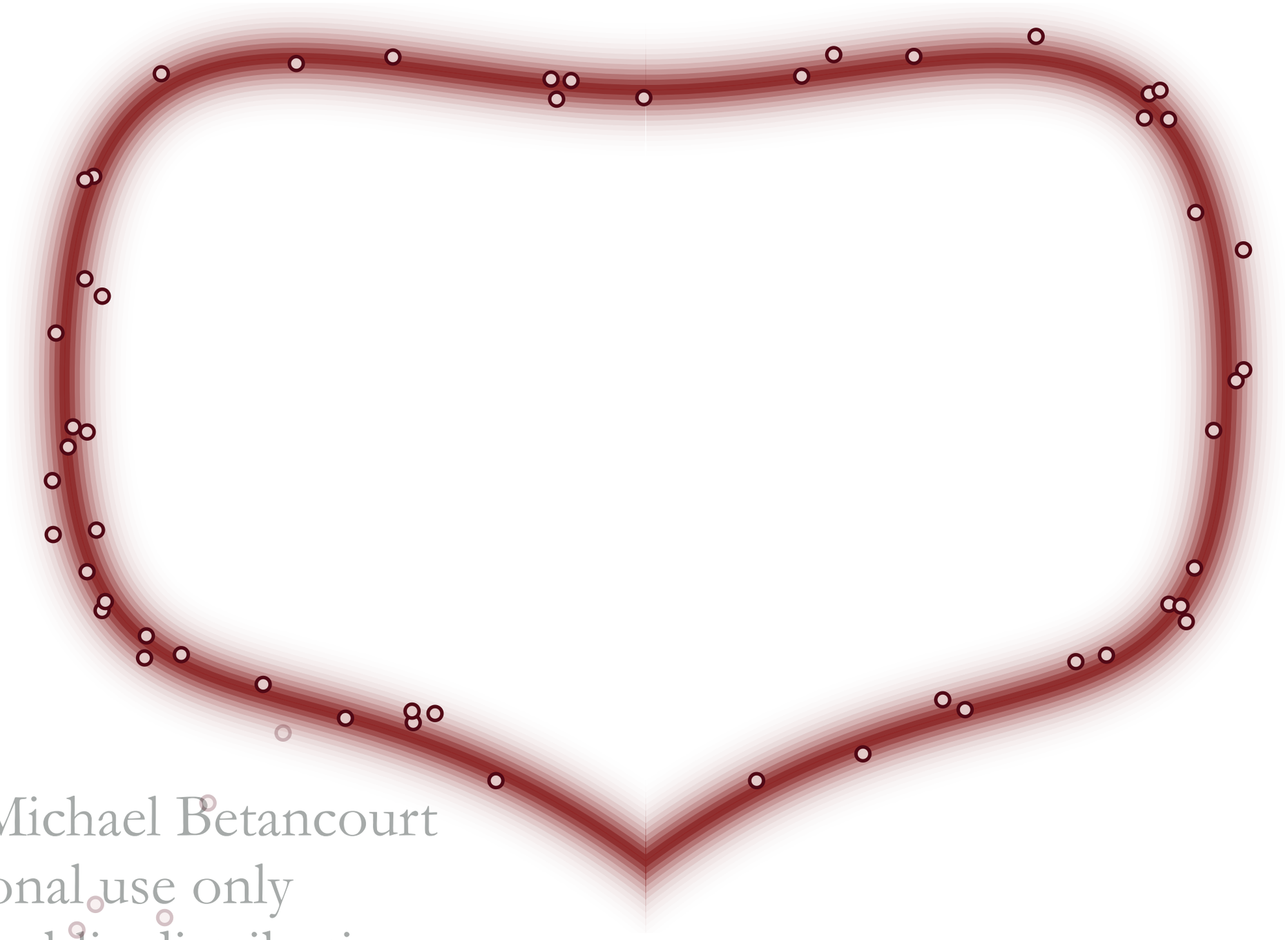


Under ideal conditions MCMC estimators converge to the true expectations in a very practically useful progression.

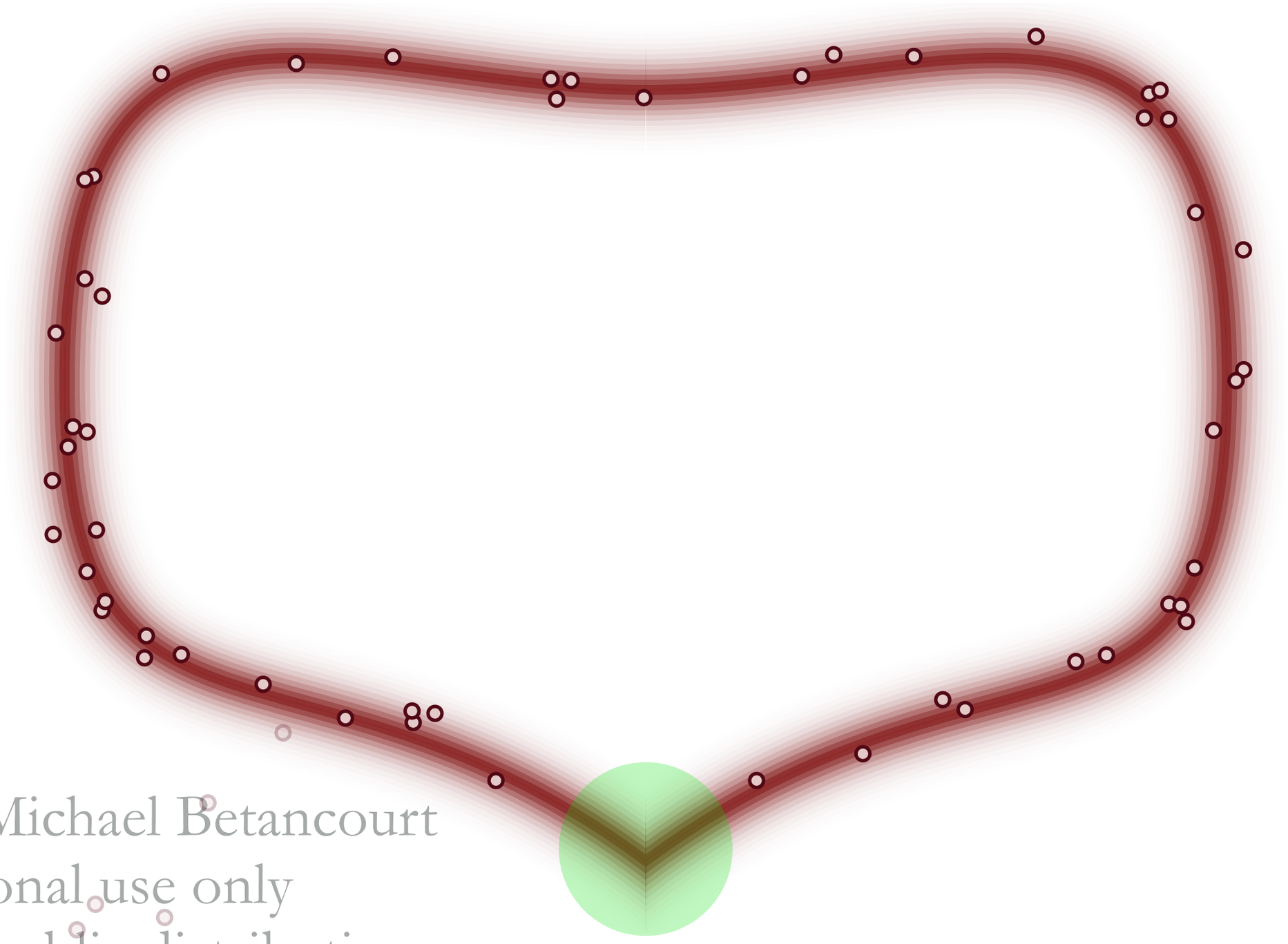




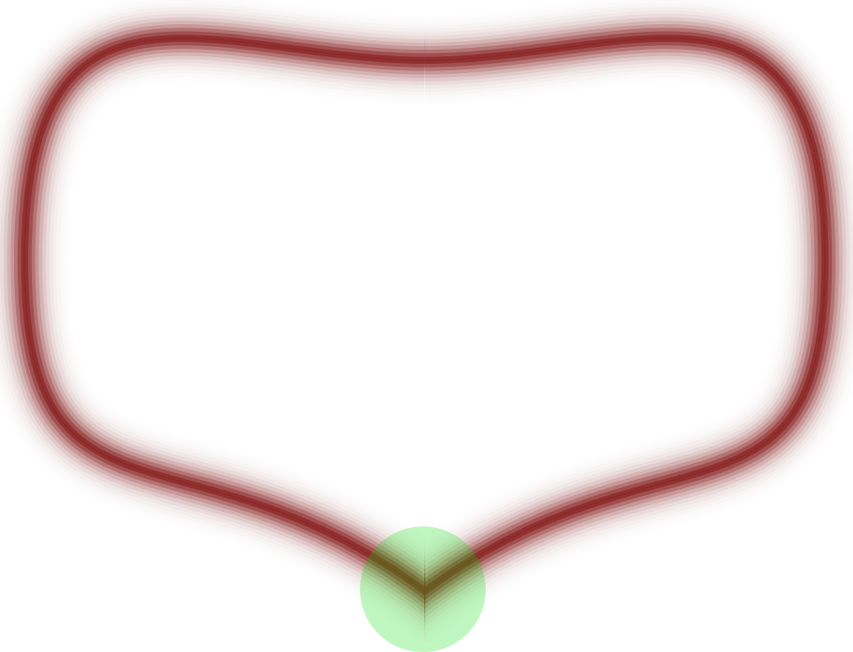
There are pathological interactions between the transition and target, however, that can spoil these ideal conditions.



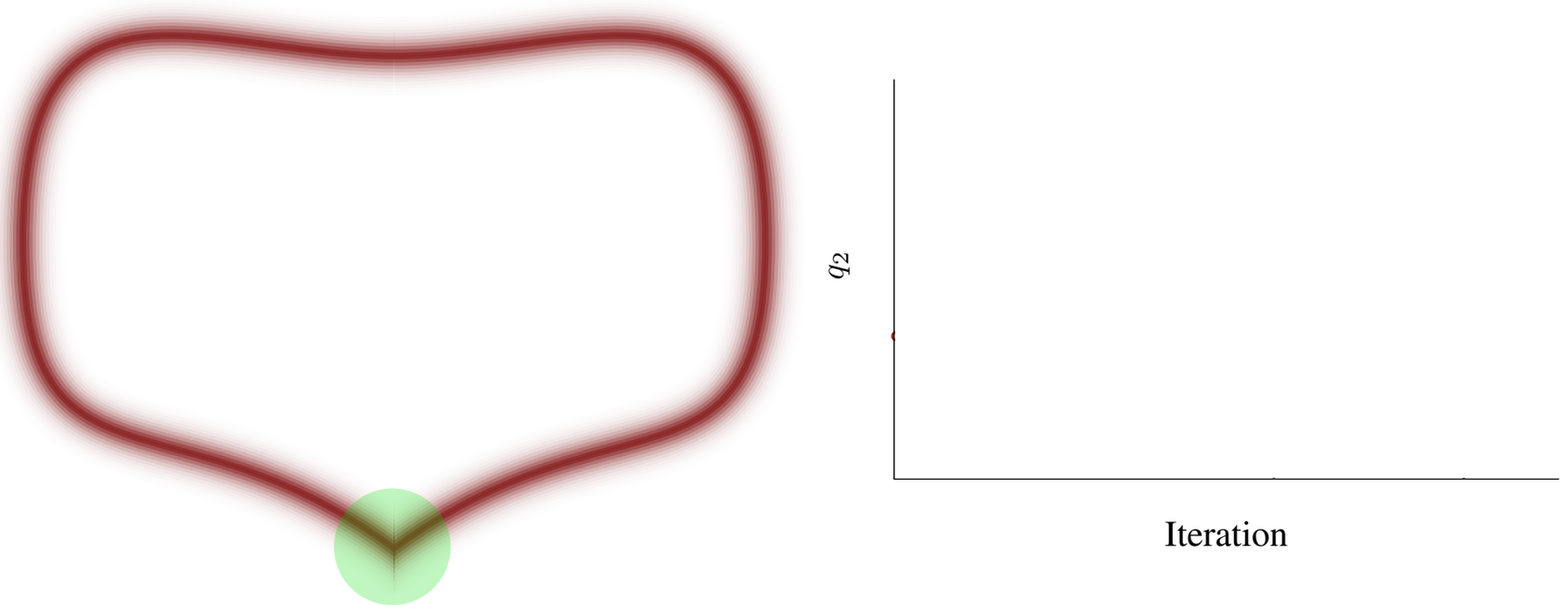
There are pathological interactions between the transition and target, however, that can spoil these ideal conditions.



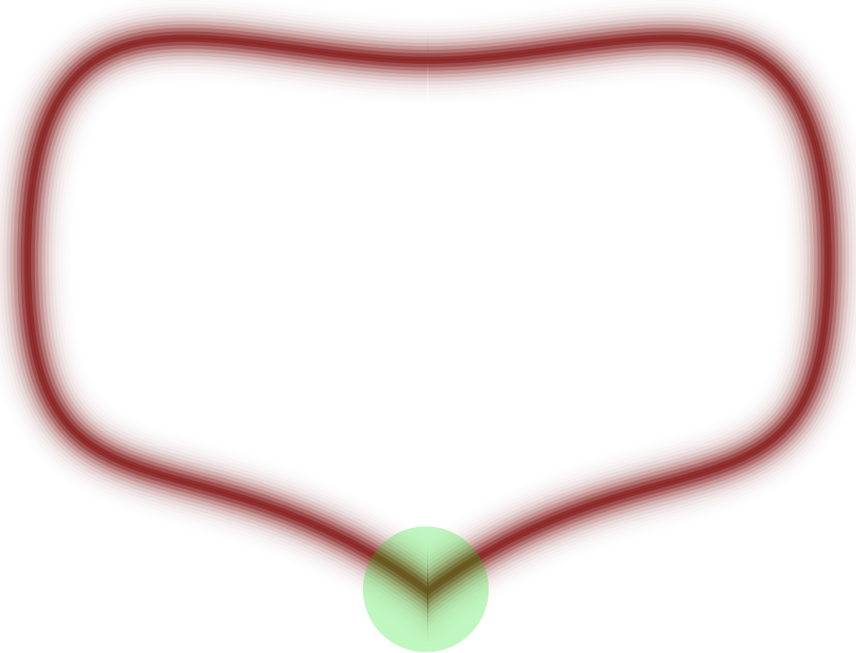
There are pathological interactions between the transition and target, however, that can spoil these ideal conditions.

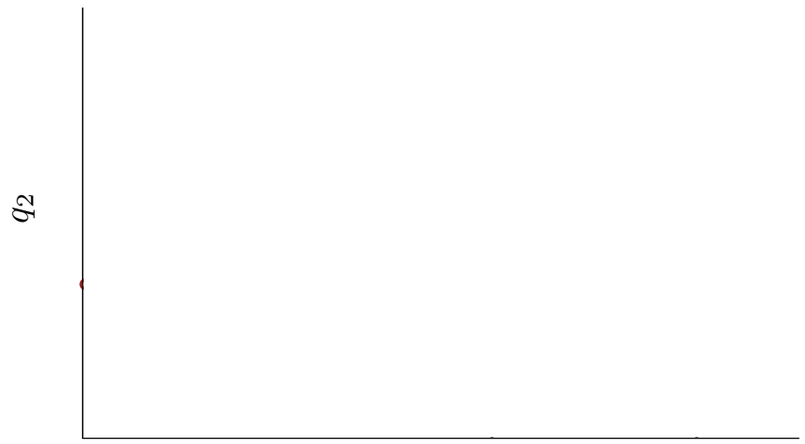


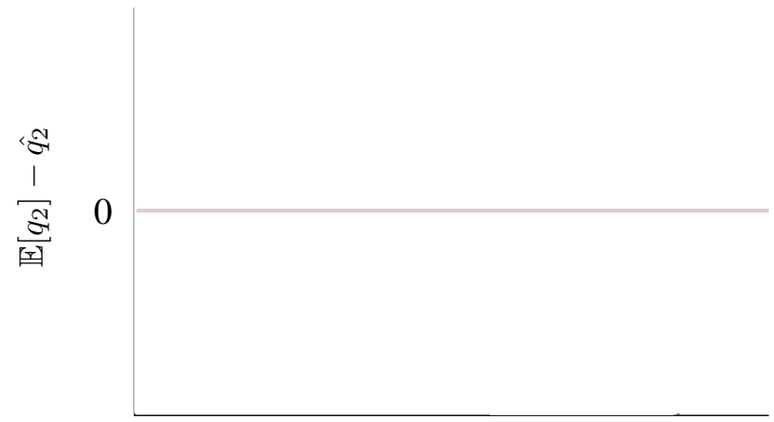
There are pathological interactions between the transition and target, however, that can spoil these ideal conditions.



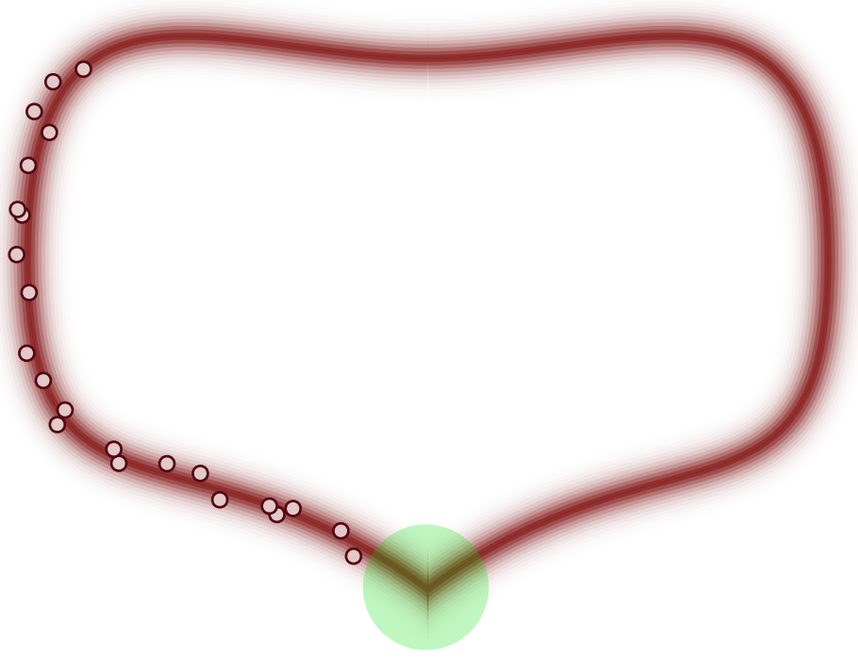
There are pathological interactions between the transition and target, however, that can spoil these ideal conditions.

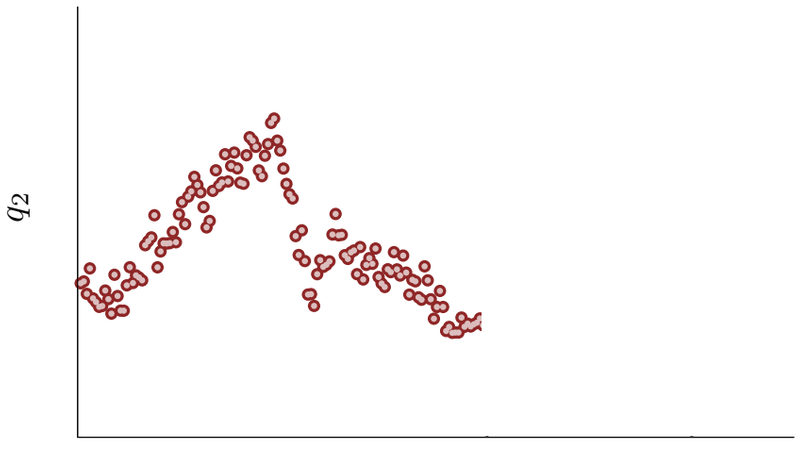


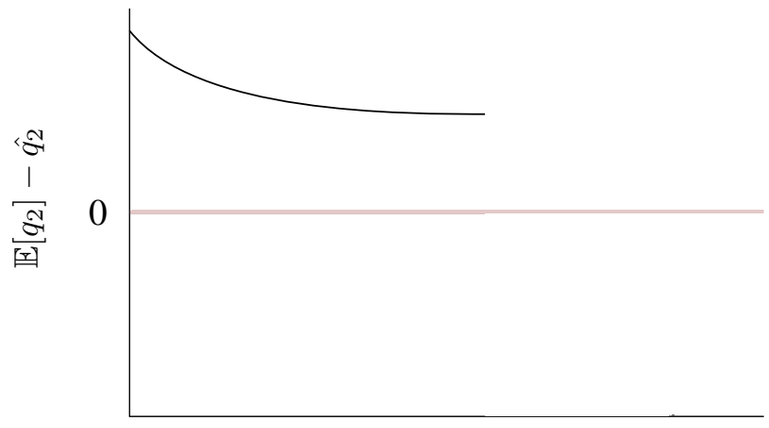




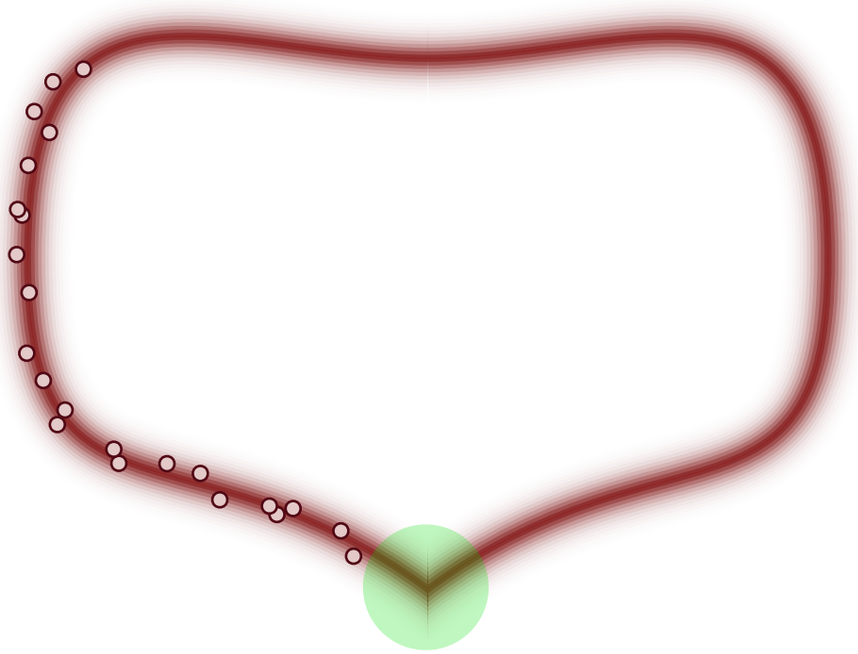
There are pathological interactions between the transition and target, however, that can spoil these ideal conditions.

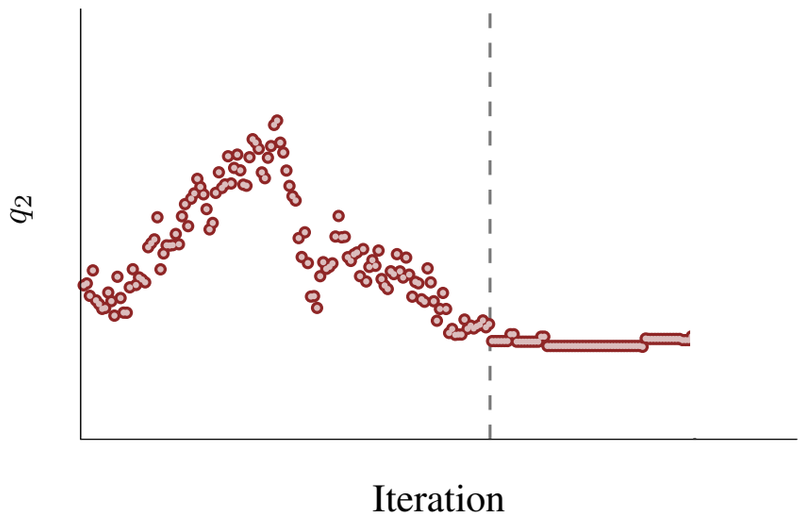


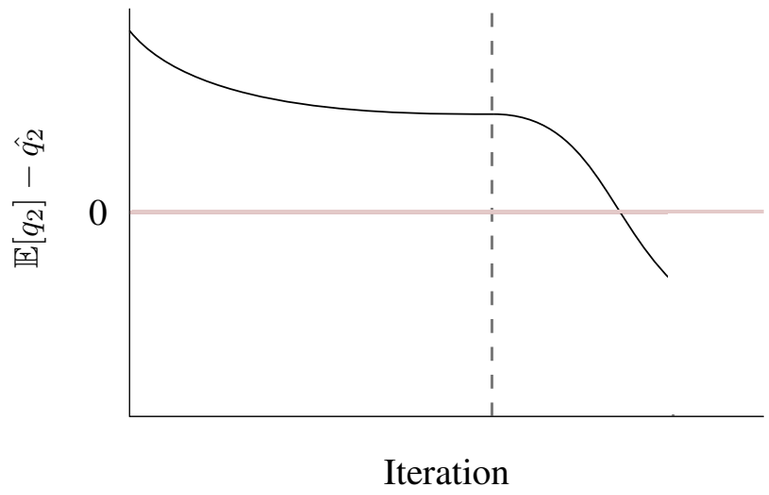




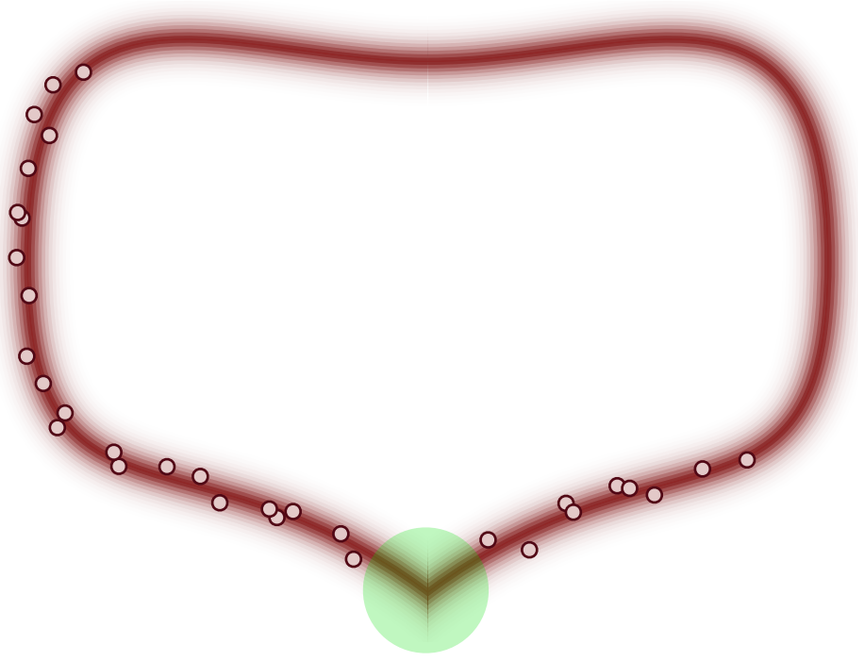
There are pathological interactions between the transition and target, however, that can spoil these ideal conditions.

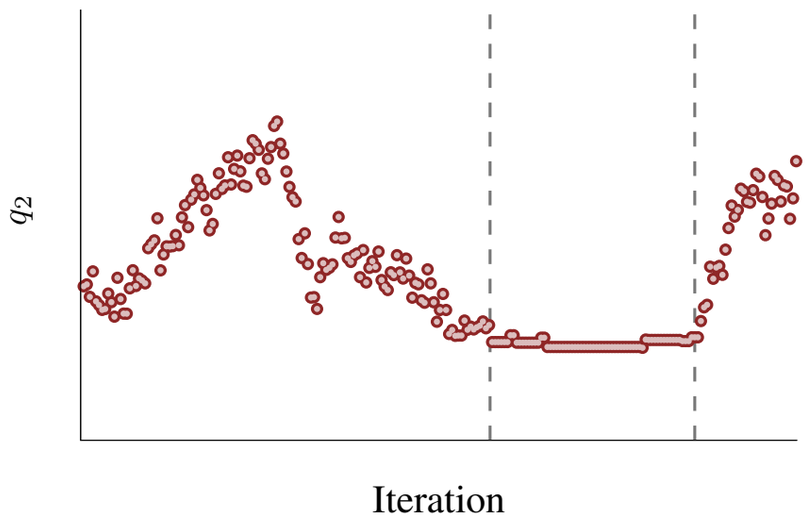


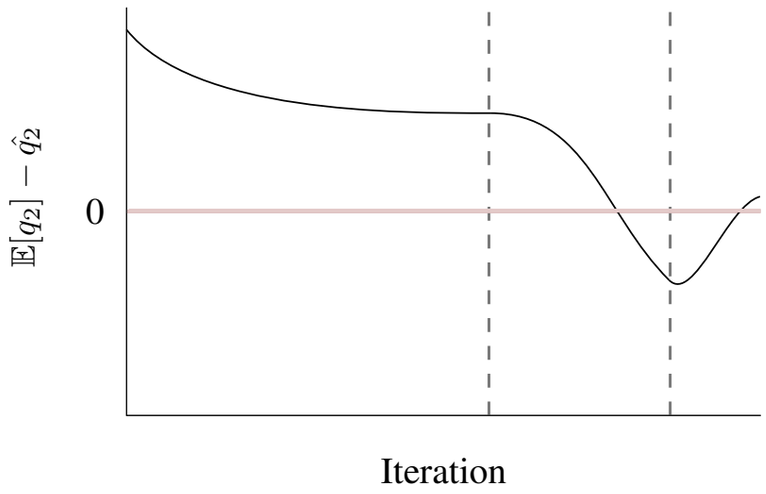




There are pathological interactions between the transition and target, however, that can spoil these ideal conditions.







To understand finite time behavior we need to understand how the *N-step distribution* behaves.

$$\delta(\theta_1 - \theta_0)$$

To understand finite time behavior we need to understand how the *N-step distribution* behaves.

$$T\delta_{\theta_0} = \int \mathrm{d}\theta_1 \, T(\theta_2 \mid \theta_1) \delta(\theta_1 - \theta_0)$$

To understand finite time behavior we need to understand how the *N-step distribution* behaves.

$$T\delta_{\theta_0} = \int \mathrm{d}\theta_1 \, T(\theta_2 \mid \theta_1) \delta(\theta_1 - \theta_0)$$

$$T^2\delta_{\theta_0} = T(T\delta_{\theta_0})$$

To understand finite time behavior we need to understand how the *N-step distribution* behaves.

$$T \delta_{\theta_0} = \int \mathrm{d}\theta_1 \, T(\theta_2 \mid \theta_1) \delta(\theta_1 - \theta_0)$$

$$T^2 \delta_{\theta_0} = T(T \delta_{\theta_0})$$

$$= \int \mathrm{d}\theta_2 \, T(\theta_3 \mid \theta_2) \int \mathrm{d}\theta_1 \, T(\theta_2 \mid \theta_1) \, \delta(\theta_1 - \theta_0)$$

To understand finite time behavior we need to understand how the *N-step distribution* behaves.

$$T \delta_{\theta_0} = \int \mathrm{d}\theta_1 \, T(\theta_2 \mid \theta_1) \delta(\theta_1 - \theta_0)$$

$$T^2 \delta_{\theta_0} = T(T \delta_{\theta_0})$$

$$= \int \mathrm{d}\theta_2 \, T(\theta_3 \mid \theta_2) \int \mathrm{d}\theta_1 \, T(\theta_2 \mid \theta_1) \, \delta(\theta_1 - \theta_0)$$

$$T^N \delta_{\theta_0} = T(T^{N-1} \delta_{\theta_0})$$

To understand finite time behavior we need to understand how the *N-step distribution* behaves.

$$T\delta_{\theta_0} = \int \mathrm{d}\theta_1 \, T(\theta_2 \mid \theta_1) \delta(\theta_1 - \theta_0)$$

$$T^2\delta_{\theta_0} = T(T\delta_{\theta_0})$$

$$= \int \mathrm{d}\theta_2 \, T(\theta_3 \mid \theta_2) \int \mathrm{d}\theta_1 \, T(\theta_2 \mid \theta_1) \, \delta(\theta_1 - \theta_0)$$

$$T^N\delta_{\theta_0} = T(T^{N-1}\delta_{\theta_0}) = \underbrace{T \ldots T}_{N \, \mathrm{times}} \delta_{\theta_0}$$

In particular we want to understand how the N-step distribution *converges* towards the target distribution.

$$||T^N \delta_{\theta_0} - \pi||$$

In particular we want to understand how the N-step distribution *converges* towards the target distribution.

$$||T^N \delta_{\theta_0} - \pi|| \leq f(\theta_0, N)$$

Geometric ergodicity occurs when the convergence proceeds *geometrically* with the number of transitions.

$$||T^N \delta_{\theta_0} - \pi||_{\mathrm{TV}} \leq C(\theta_0) \rho^N$$

Geometric ergodicity is particularly important because it guarantees a central limit theorem for MCMC estimators.

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

Geometric ergodicity is particularly important because it guarantees a central limit theorem for MCMC estimators.

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

$$\hat{f} \sim \mathcal{N}(\mathbb{E}[f], \text{MCMC-SE}[f])$$

Geometric ergodicity is particularly important because it guarantees a central limit theorem for MCMC estimators.

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

$$\hat{f} \sim \mathcal{N}(\mathbb{E}[f], \text{MCMC-SE}[f])$$

$$\text{MCMC-SE}[f] = \sqrt{\frac{\text{Var}[f]}{\text{ESS}[f]}}$$

At least under the same circumstances as the Monte Carlo central limit theorem.

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

$$\hat{f} \sim \mathcal{N}(\mathbb{E}[f], \text{MCMC-SE}[f])$$

$$\text{MCMC-SE}[f] = \sqrt{\frac{\text{Var}[f]}{\text{ESS}[f]}}$$

In this context the *effective sample size* quantifies how correlations in the Markov chain influence an estimator.

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

$$\hat{f} \sim \mathcal{N}(\mathbb{E}[f], \text{MCMC-SE}[f])$$

$$\text{MCMC-SE}[f] = \sqrt{\frac{\text{Var}[f]}{\text{ESS}[f]}}$$


ess Bar


FRISCHE Bi
Neff
drück

In this context the *effective sample size* quantifies how correlations in the Markov chain influence an estimator.

$$\rho_l[f]$$

In this context the *effective sample size* quantifies how correlations in the Markov chain influence an estimator.

$$\sum_{l=-\infty}^{\infty} \rho_l[f]$$

In this context the *effective sample size* quantifies how correlations in the Markov chain influence an estimator.

$$\frac{N}{\sum_{l=-\infty}^{\infty} \rho_l[f]}$$

In this context the *effective sample size* quantifies how correlations in the Markov chain influence an estimator.

$$\mathrm{ESS}[f] = \frac{N}{\sum_{l=-\infty}^{\infty} \rho_l[f]}$$

In this context the *effective sample size* quantifies how correlations in the Markov chain influence an estimator.

$$\mathrm{ESS}[f] = \frac{N}{\sum_{l=-\infty}^{\infty} \rho_l[f]}$$

$$\rho_{-l} = \rho_l$$

In this context the *effective sample size* quantifies how correlations in the Markov chain influence an estimator.

$$\mathrm{ESS}[f] = \frac{N}{\sum_{l=-\infty}^{\infty} \rho_l[f]}$$

$$\rho_{-l} = \rho_l, \, \rho_0 = 1$$

In this context the *effective sample size* quantifies how correlations in the Markov chain influence an estimator.

$$\mathrm{ESS}[f] = \frac{N}{\sum_{l=-\infty}^{\infty} \rho_l[f]}$$

$$\rho_{-l} = \rho_l,\ \rho_0 = 1$$

$$\mathrm{ESS}[f] = \frac{N}{1 + 2\sum_{l=1}^{\infty} \rho_l[f]}$$

The more positive the autocorrelation sum the lower the effective sample size; the more negative the higher.

$$\mathrm{ESS}[f] = \frac{N}{1 + 2 \sum_{l=1}^{\infty} \rho_l[f]}$$

The autocorrelations quantify how states in the Markov chain separated by a *lag* are correlated with each other.

$$\mu_f = \mathbb{E}[f]$$

The autocorrelations quantify how states in the Markov chain separated by a *lag* are correlated with each other.

$$\mu_f = \mathbb{E}[f]$$

$$(f(x) - \mu_f)$$

The autocorrelations quantify how states in the Markov chain separated by a *lag* are correlated with each other.

$$\mu_f = \mathbb{E}[f]$$

$$(f(T^l x) - \mu_f)(f(x) - \mu_f)$$

The autocorrelations quantify how states in the Markov chain separated by a *lag* are correlated with each other.

$$\mu_f = \mathbb{E}[f]$$

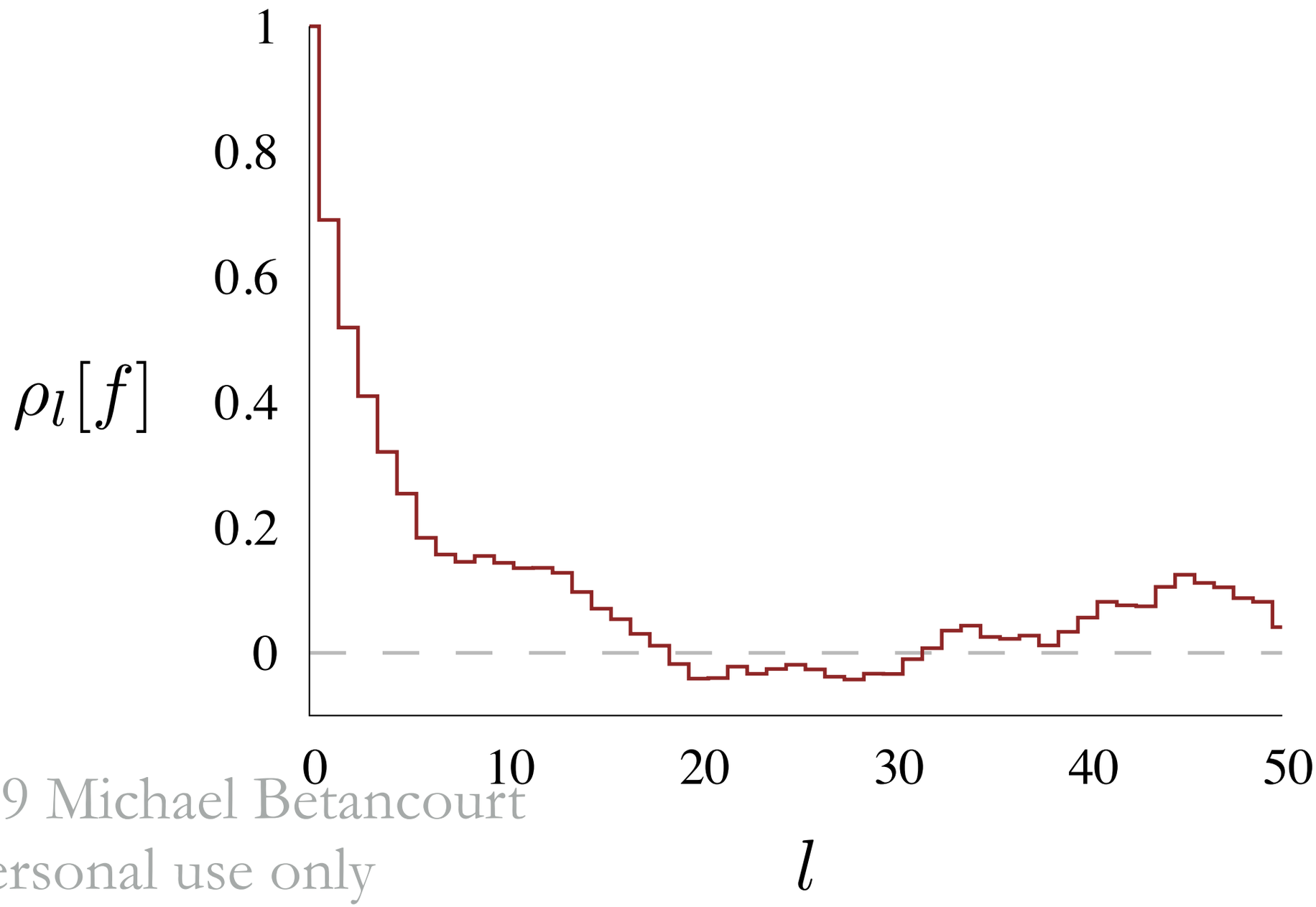
$$\frac{\mathbb{E}[(f(T^l x) - \mu_f)(f(x) - \mu_f)]}{\mathrm{Var}[f]}$$

The autocorrelations quantify how states in the Markov chain separated by a *lag* are correlated with each other.

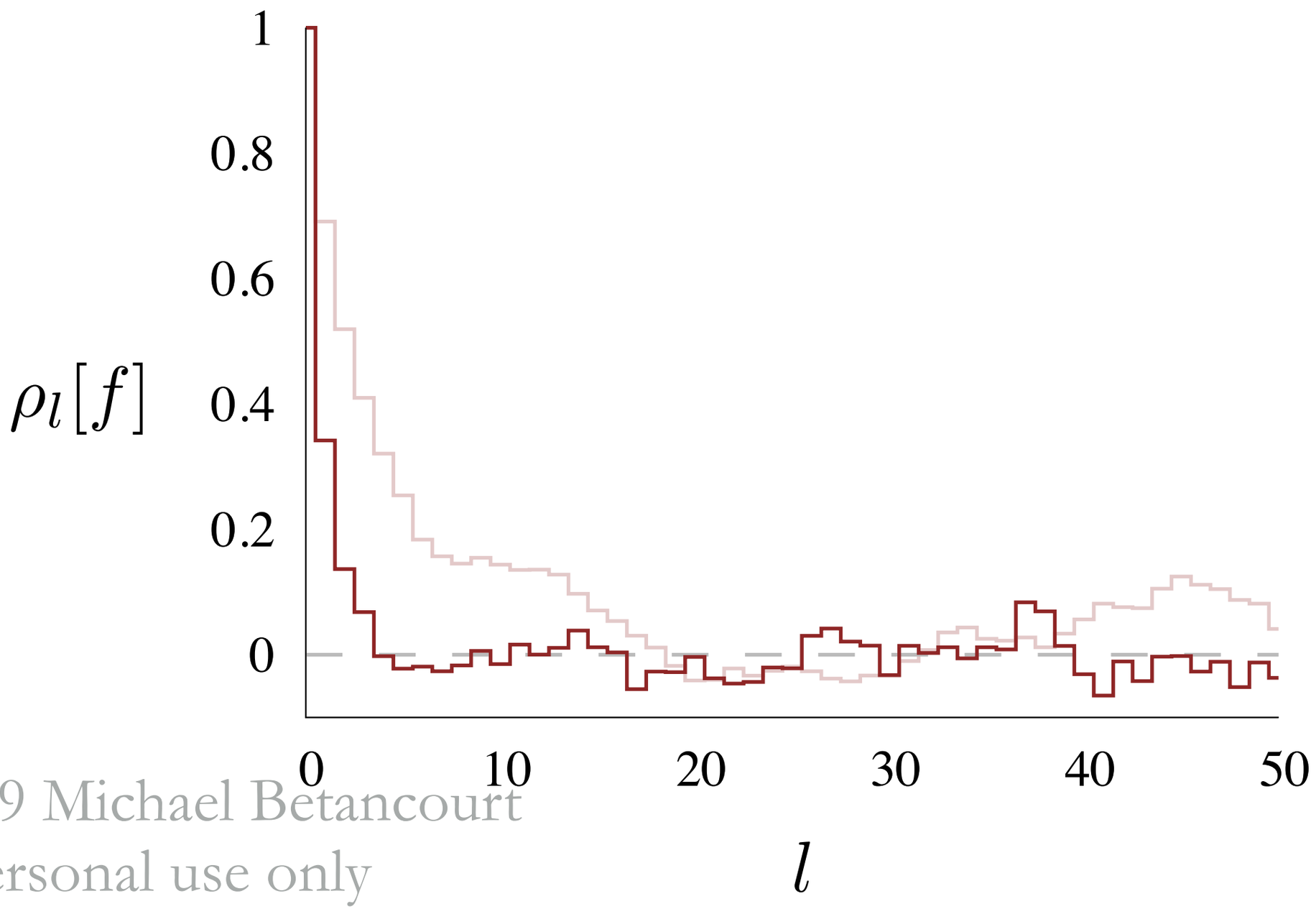
$$\mu_f = \mathbb{E}[f]$$

$$\rho_l[f] = \frac{\mathbb{E}[(f(T^l x) - \mu_f)(f(x) - \mu_f)]}{\mathrm{Var}[f]}$$

The autocorrelations quantify how states in the Markov chain separated by a *lag* are correlated with each other.



The autocorrelations quantify how states in the Markov chain separated by a *lag* are correlated with each other.



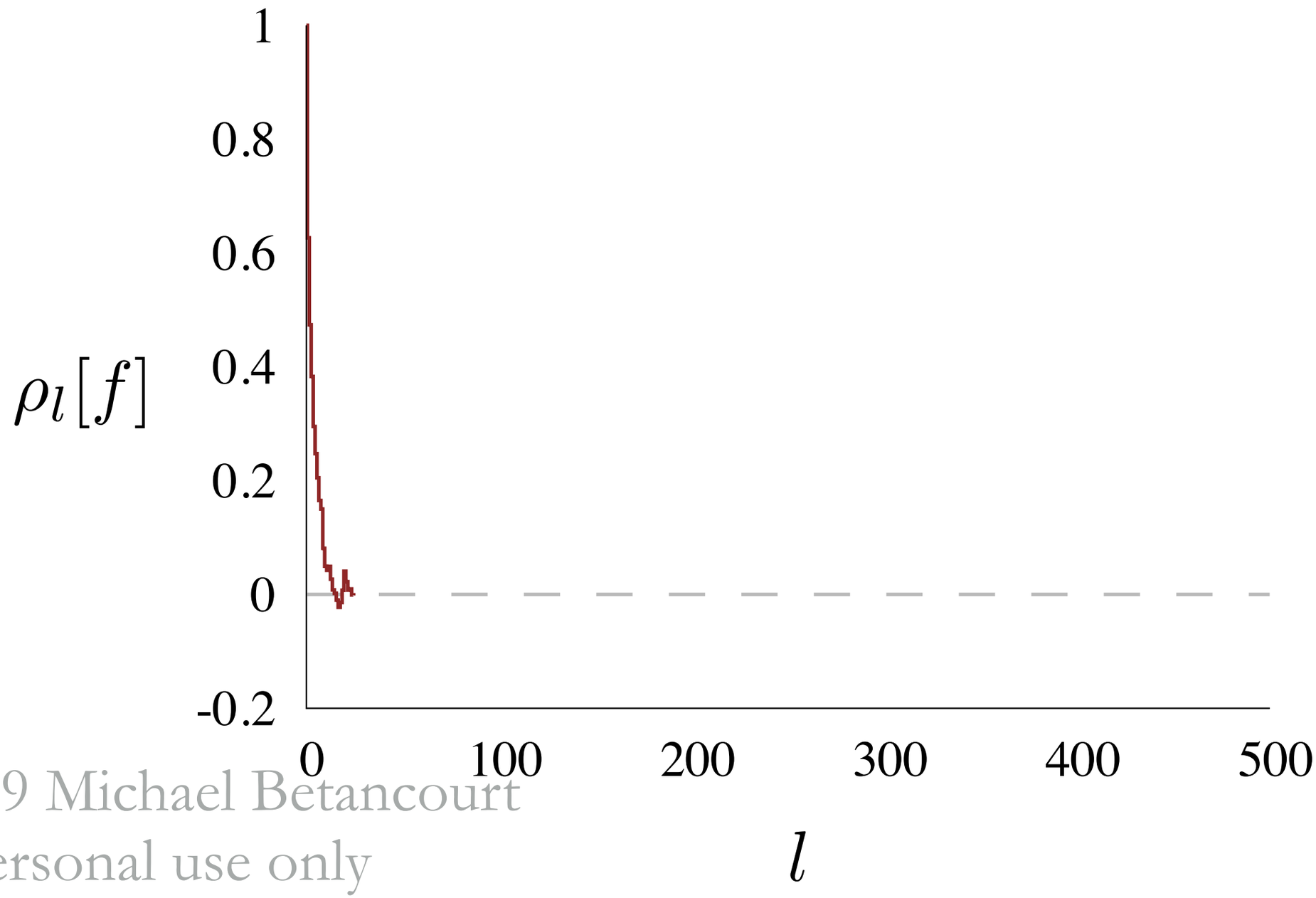
In practice we have to estimate each autocorrelation using the history of a given Markov chain.

$$\rho_l[f] = \frac{\mathbb{E}[(f(T^l x) - \mu_f)(f(x) - \mu_f)]}{\mathrm{Var}[f]}$$

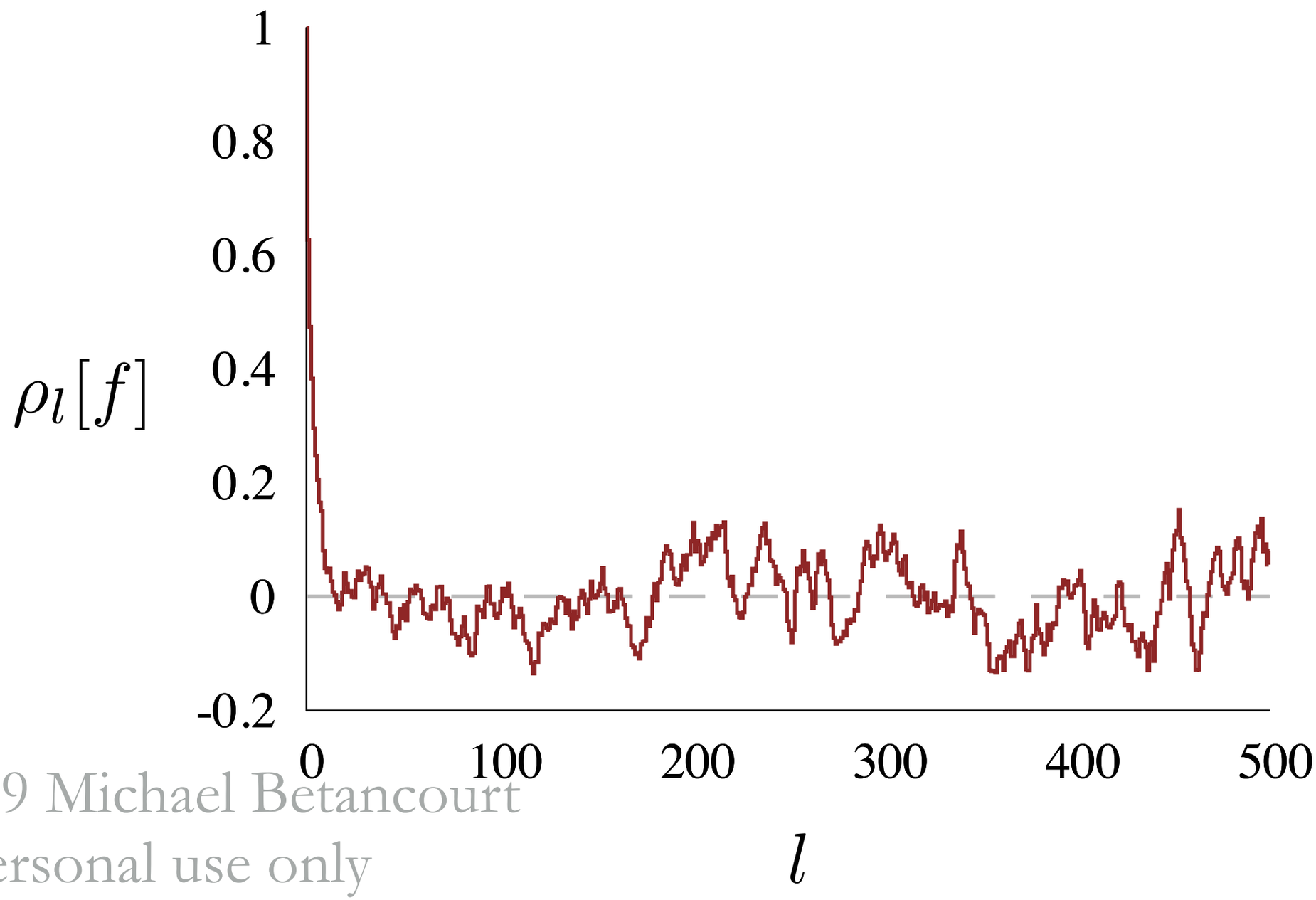
In practice we have to estimate each autocorrelation using the history of a given Markov chain.

$$\hat{\rho}_l[f] = \frac{\frac{1}{N} \sum_{n=0}^{N-l} (f(x_{n+l}) - \hat{\mu}_f)(f(x_n) - \hat{\mu}_f)}{\widehat{\mathrm{Var}}[f]}$$

In order to accurately estimate autocorrelations we have to make sure that we run sufficiently long chains.



In order to accurately estimate autocorrelations we have to make sure that we run sufficiently long chains.



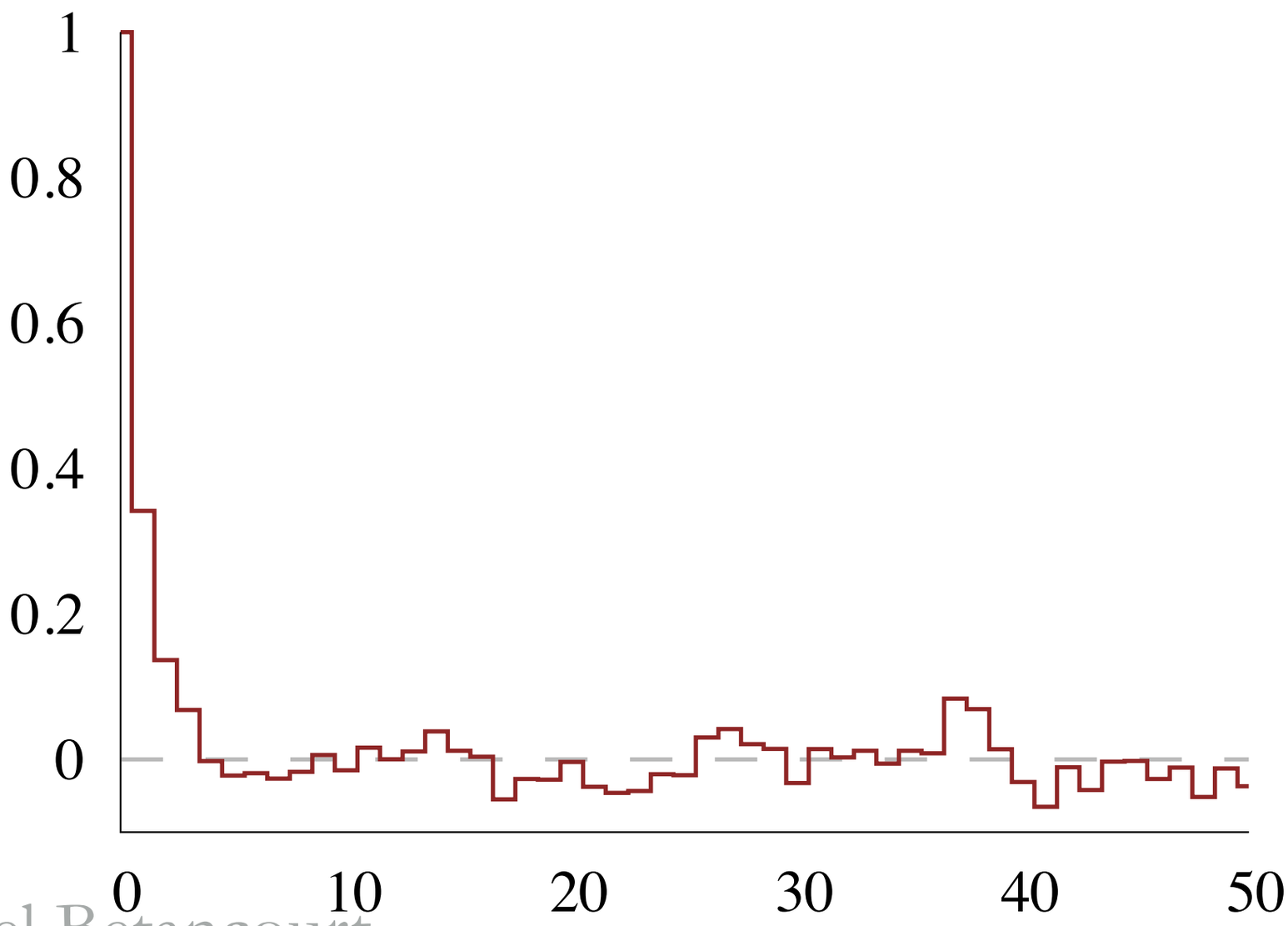
Once we've estimated the autocorrelations we can estimate the effective sample sizes.

$$\widehat{\mathrm{ESS}}[f] = \frac{N}{1 + 2 \sum_{l=1}^{L} \hat{\rho}_l[f]}$$

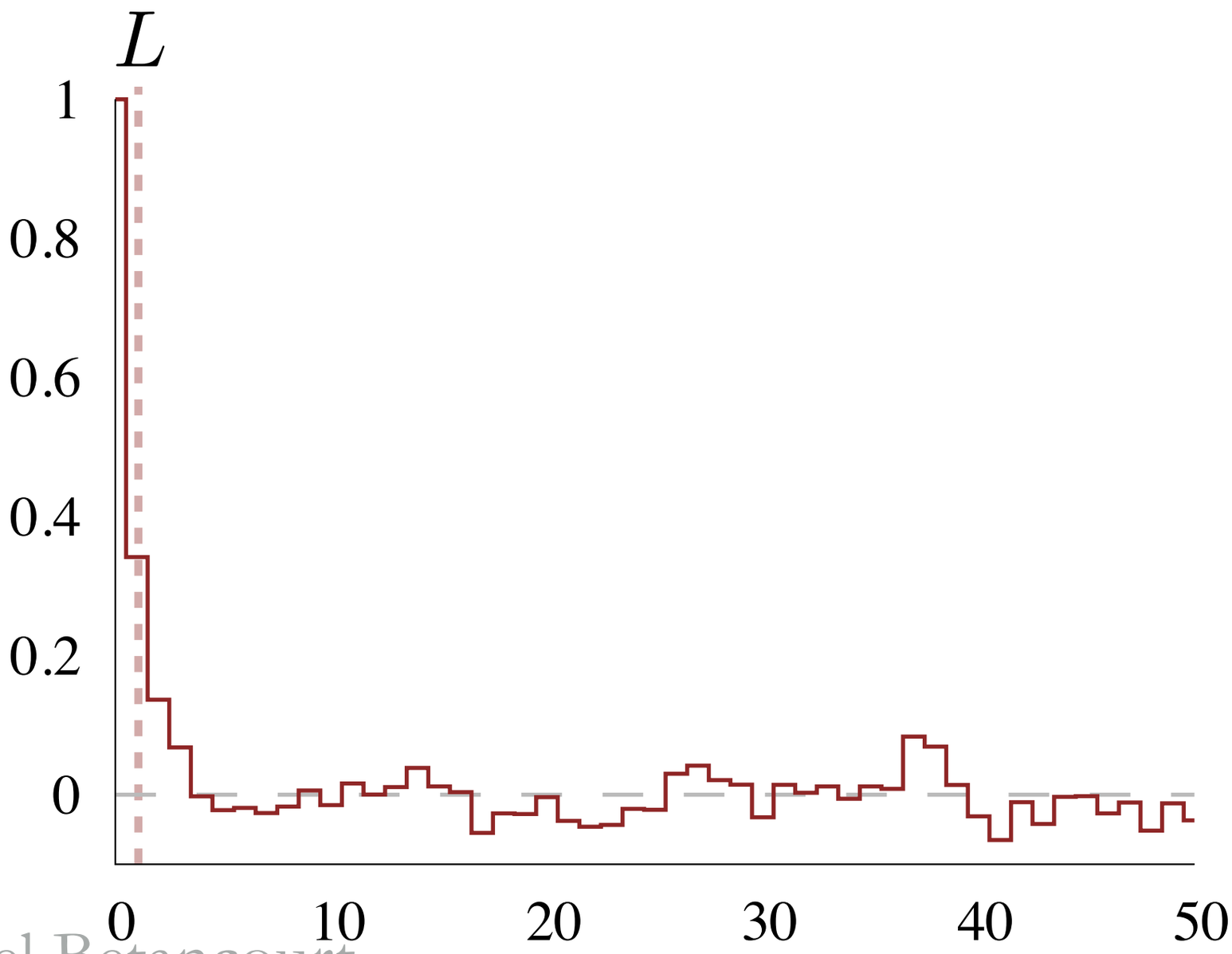
Once we've estimated the autocorrelations we can estimate the effective sample sizes.

$$\widehat{\text{ESS}}[f] = \frac{N}{1 + 2 \sum_{l=1}^{L} \hat{\rho}_l[f]}$$

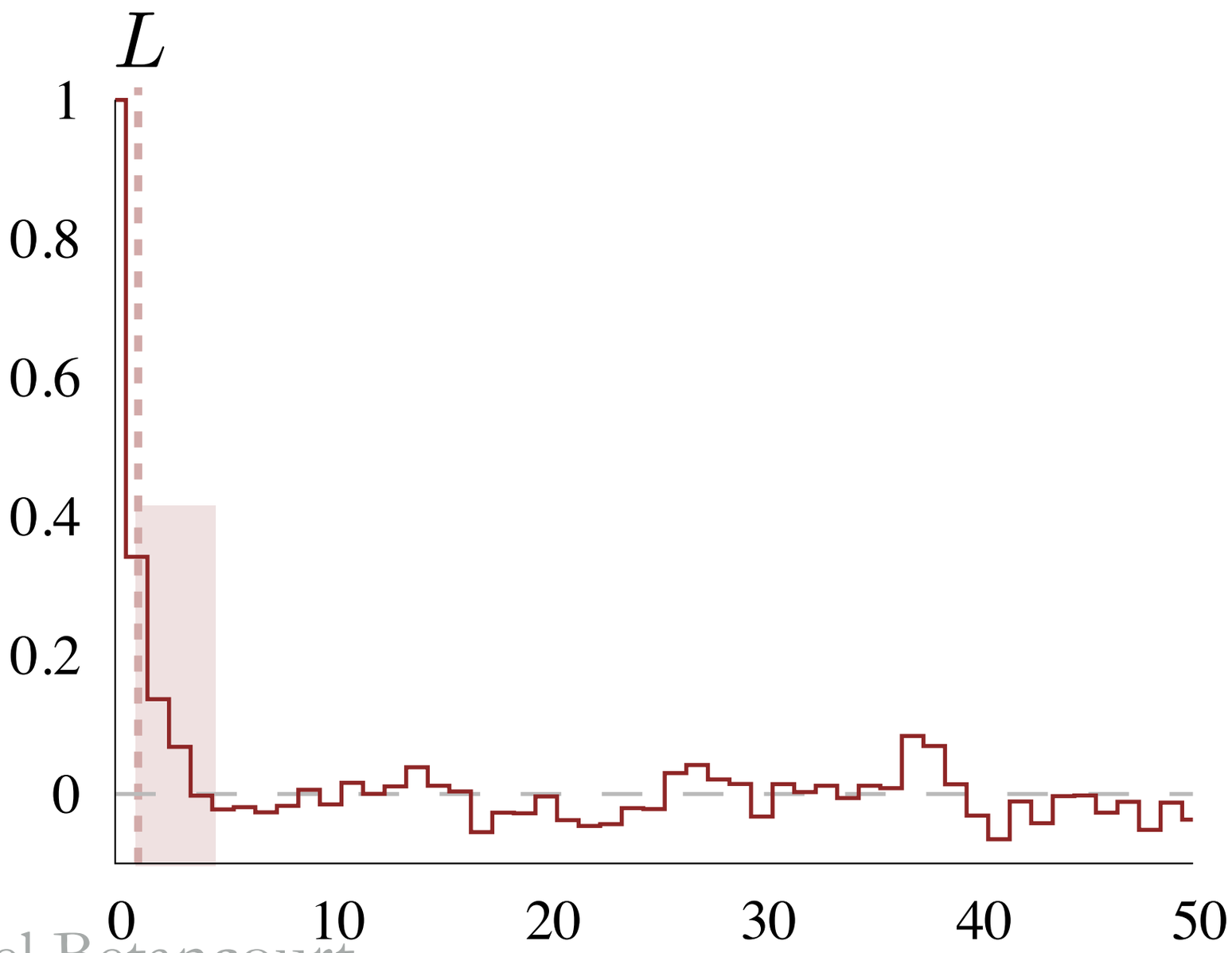
The accuracy of the effective sample size estimator is *very* sensitive to the lag threshold.



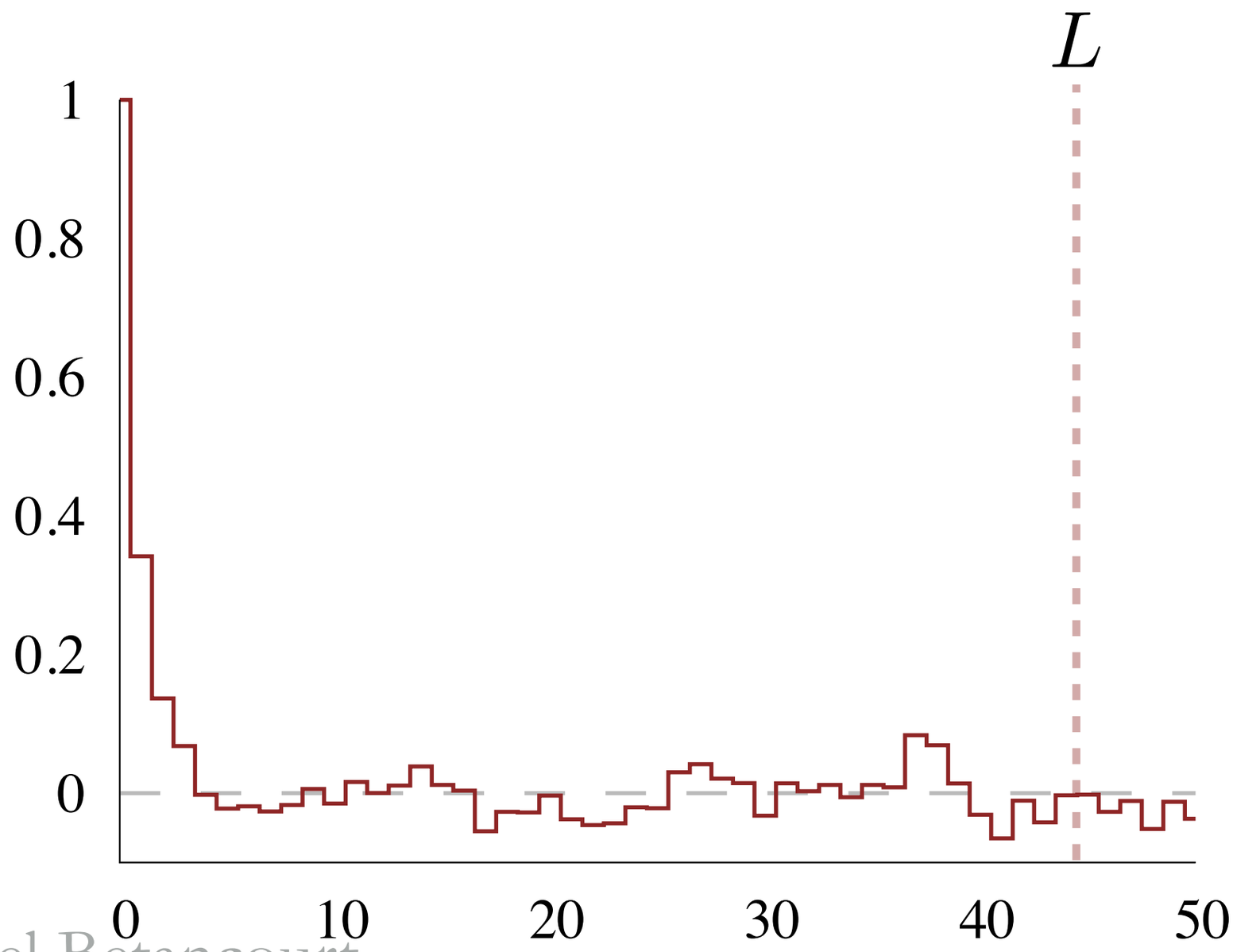
The accuracy of the effective sample size estimator is *very* sensitive to the lag threshold.



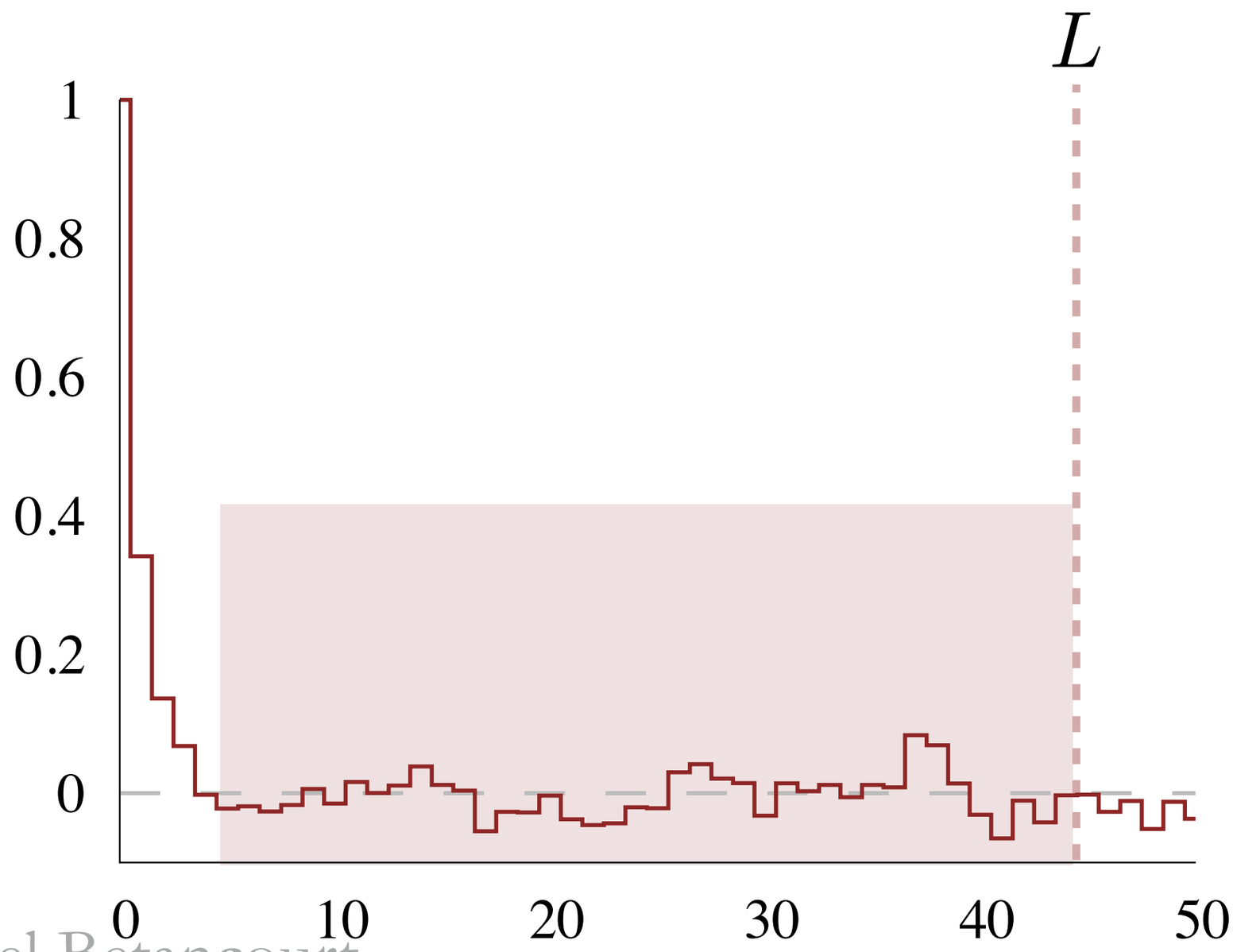
The accuracy of the effective sample size estimator is *very* sensitive to the lag threshold.



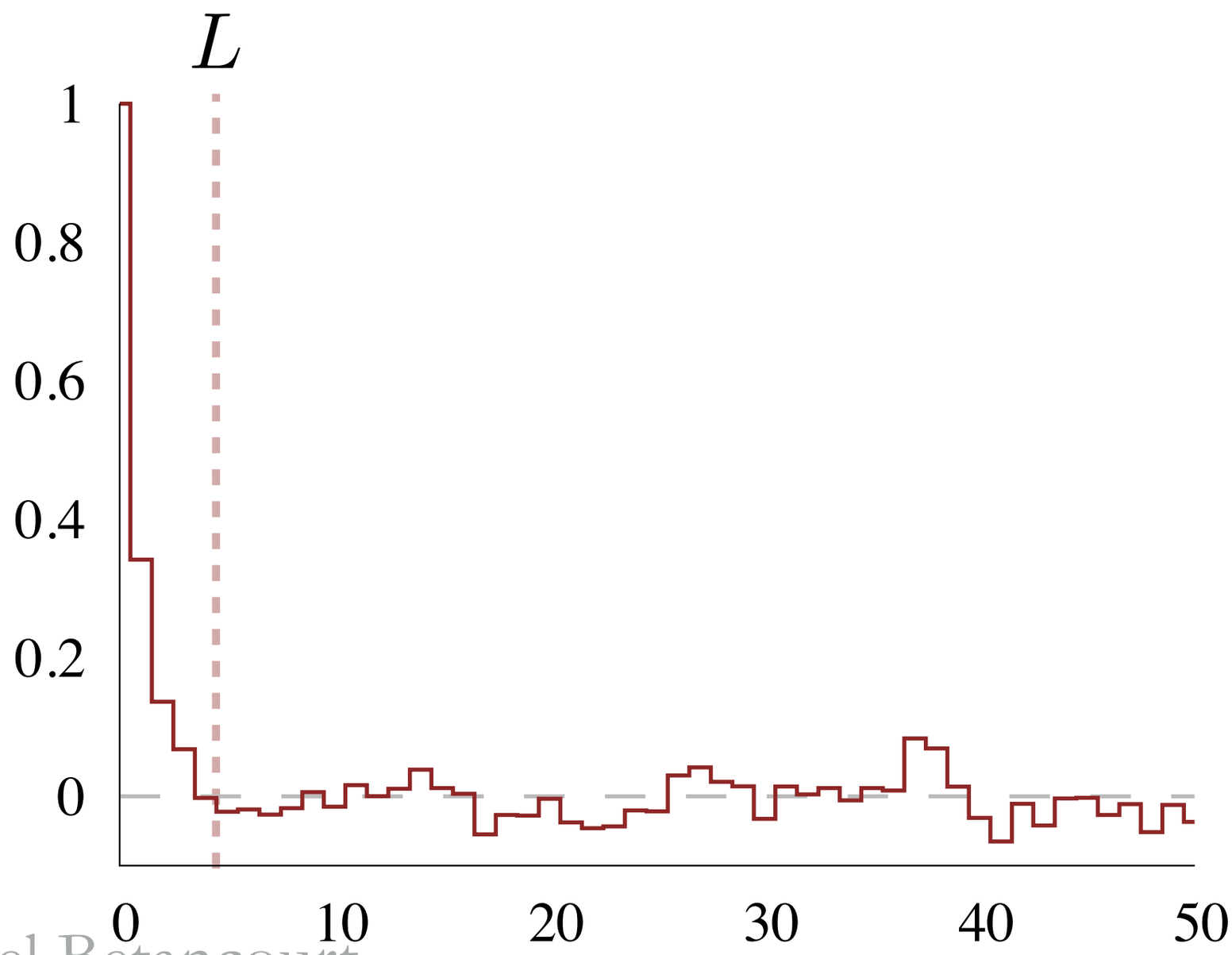
The accuracy of the effective sample size estimator is *very* sensitive to the lag threshold.



The accuracy of the effective sample size estimator is *very* sensitive to the lag threshold.



The accuracy of the effective sample size estimator is *very* sensitive to the lag threshold.



Once we've estimated the effective sample sizes we can estimate the standard errors of our MCMC estimators.

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

$$\hat{f} \sim \mathcal{N}(\mathbb{E}[f], \text{MCMC-SE}[f])$$

$$\text{MCMC-SE}[f] = \sqrt{\frac{\text{Var}[f]}{\text{ESS}[f]}}$$

Once we've estimated the effective sample sizes we can estimate the standard errors of our MCMC estimators.

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

$$\hat{f} \sim \mathcal{N}(\mathbb{E}[f], \widehat{\text{MCMC-SE}}[f])$$

$$\widehat{\text{MCMC-SE}}[f] = \sqrt{\frac{\widehat{\text{Var}}[f]}{\widehat{\text{ESS}}[f]}}$$

All of these estimations depend on the true expectations being well-defined, and we can't that take for granted!

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

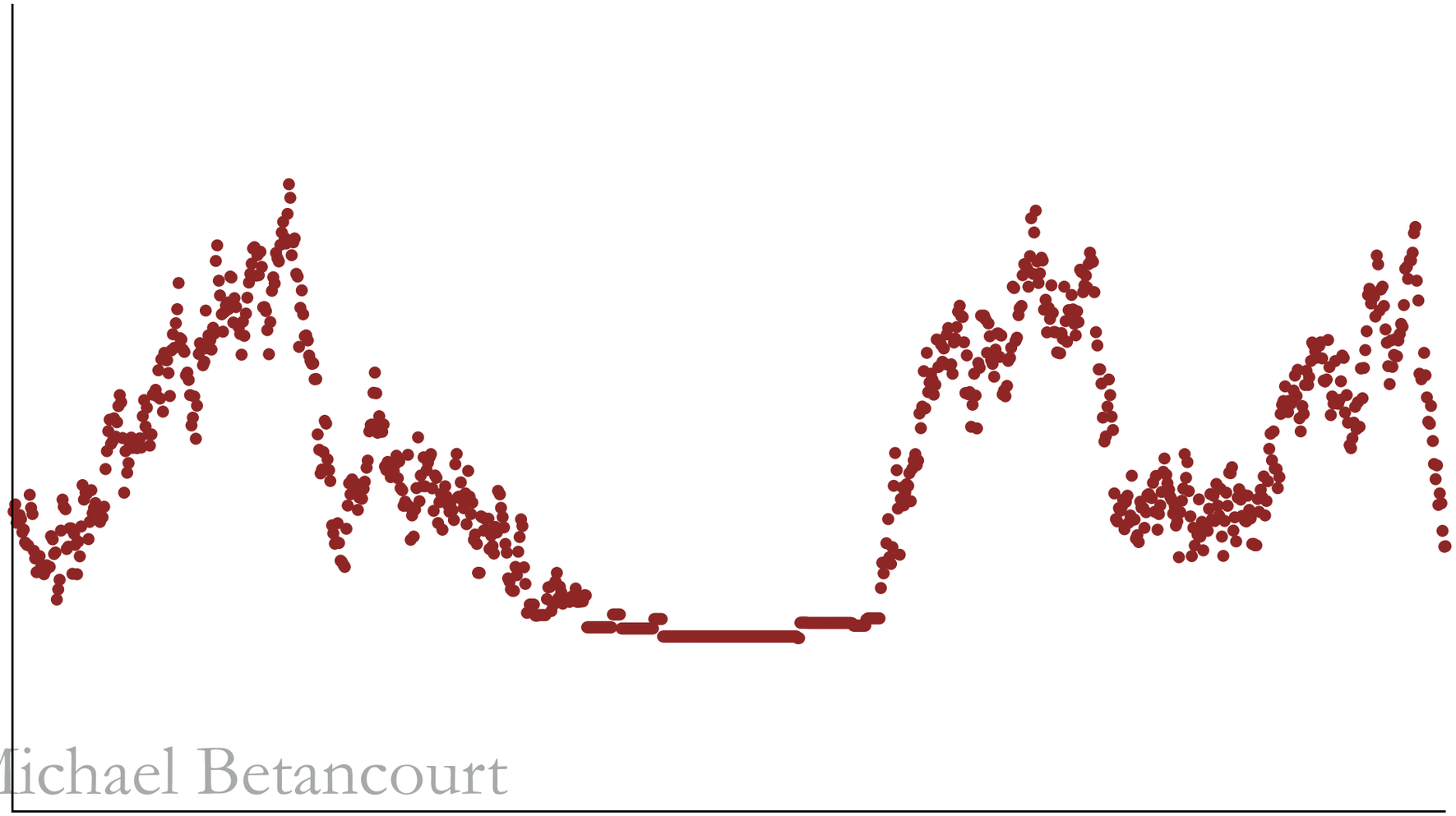
All of these estimations depend on the true expectations being well-defined, and we can't that take for granted!

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

$$\hat{f} \sim \mathcal{N}(\mathbb{E}[f], \widehat{\text{MCMC-SE}}[f])$$

$$\widehat{\text{MCMC-SE}}[f] = \sqrt{\frac{\widehat{\text{Var}}[f]}{\widehat{\text{ESS}}[f]}}$$

Diagnosing Inadequate Convergence



Ideally we would be able to work out convergence properties analytically in a given circumstance.

$$||T^N \delta_{\theta_0} - \pi||_{\mathrm{TV}} \leq C(\theta_0) \rho^N$$

Ideally we would be able to work out convergence properties analytically in a given circumstance.

$$||T^N \delta_{\theta_0} - \pi||_{\mathrm{TV}} \leq C(\theta_0) \rho^N$$

Such an analysis, however, is impractically complex even in seemingly simple problems.

$$\pi_S(\theta \mid \tilde{y}) = \int \mathrm{d}\theta' \, \pi_S(\theta' \mid \tilde{y}) \, T(\theta \mid \theta')$$

If we have geometric ergodicity, however, then we can *estimate* convergence empirically.

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

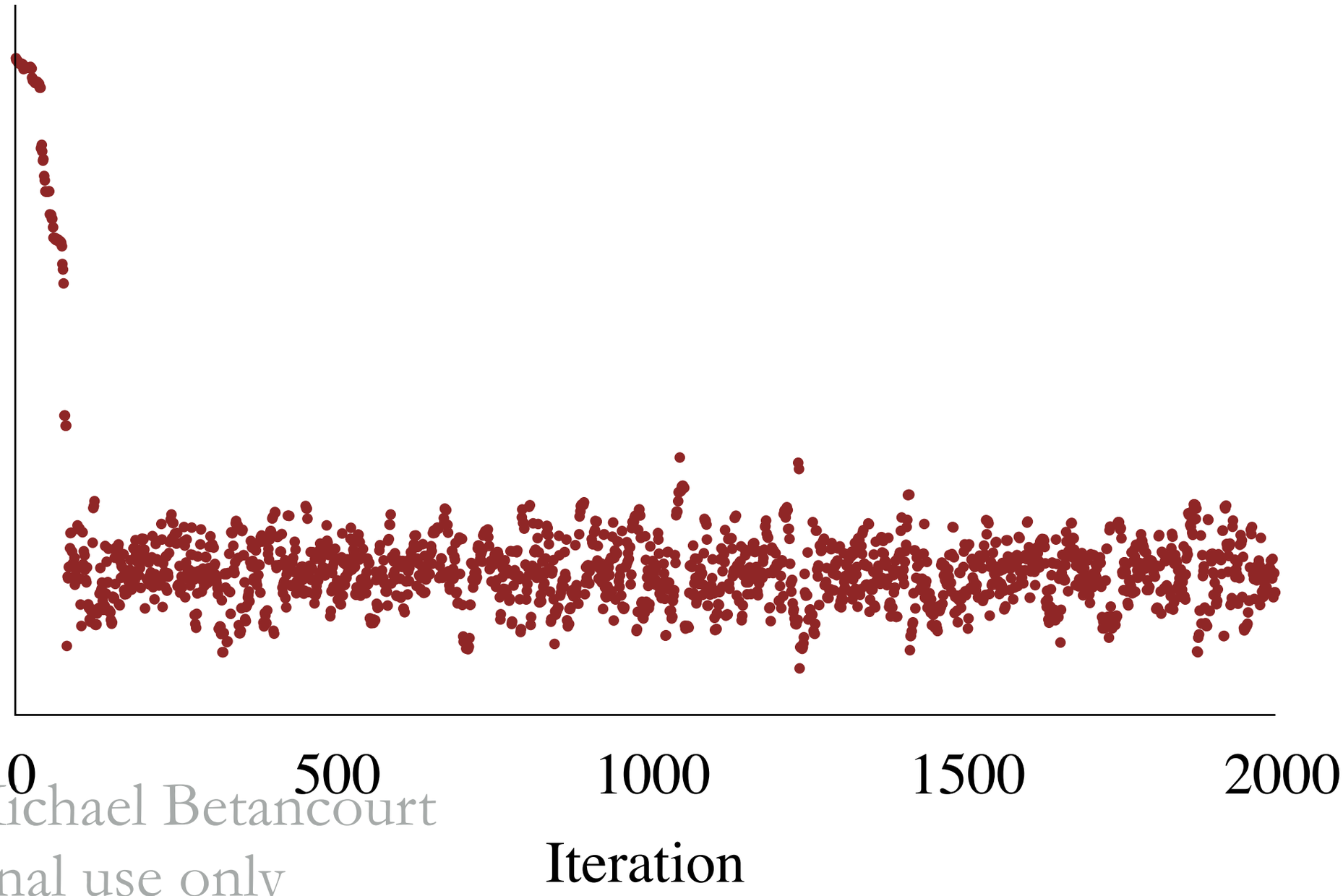
$$\hat{f} \sim \mathcal{N}(\mathbb{E}[f], \widehat{\text{MCMC-SE}}[f])$$

$$\widehat{\text{MCMC-SE}}[f] = \sqrt{\frac{\widehat{\text{Var}}[f]}{\widehat{\text{ESS}}[f]}}$$

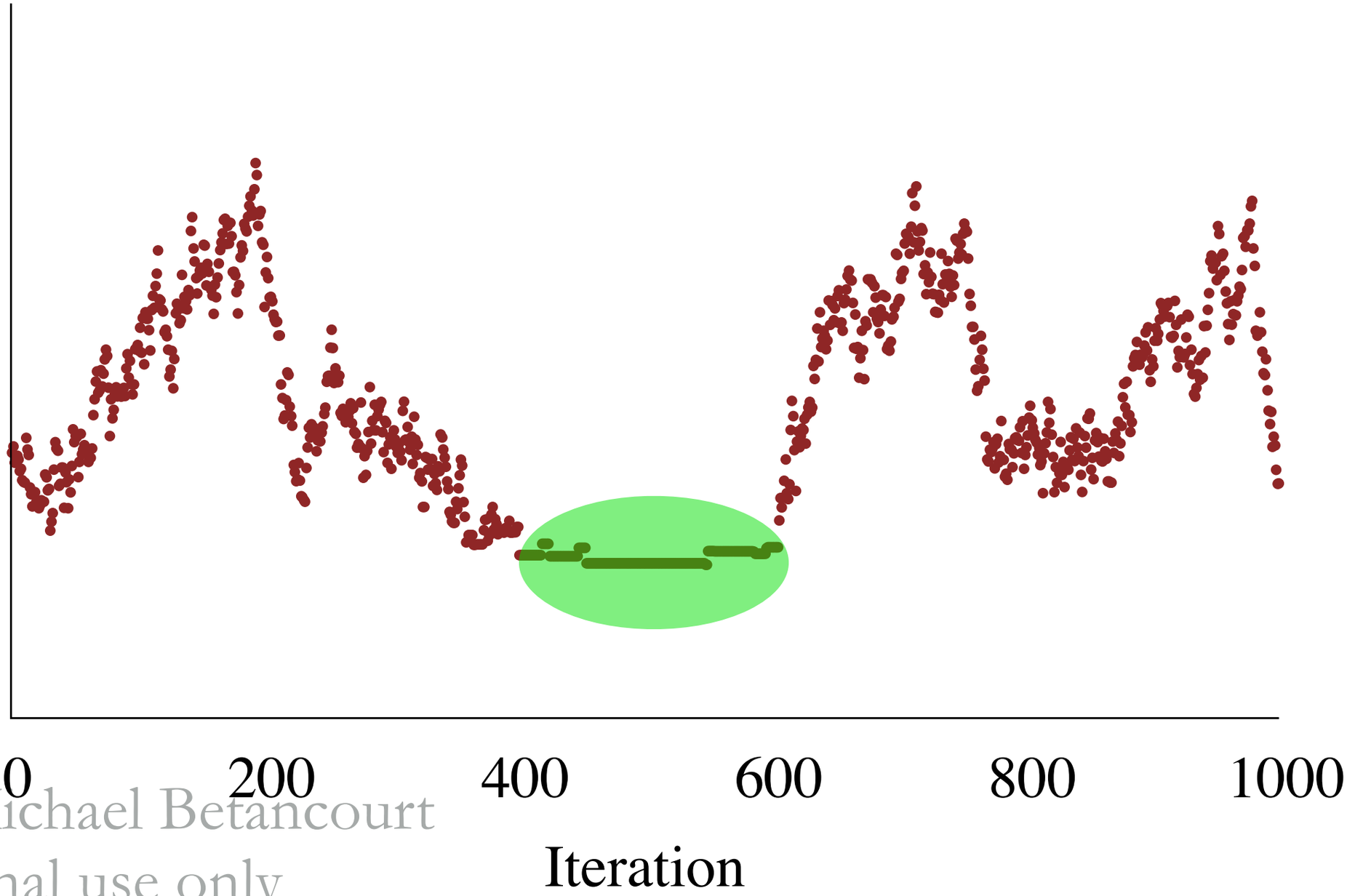
Often our best strategy is to empirically identify *obstructions* to geometric ergodicity.

$$\hat{f} \sim \mathcal{N}(\mathbb{E}[f], \text{MCMC-SE}[f])$$

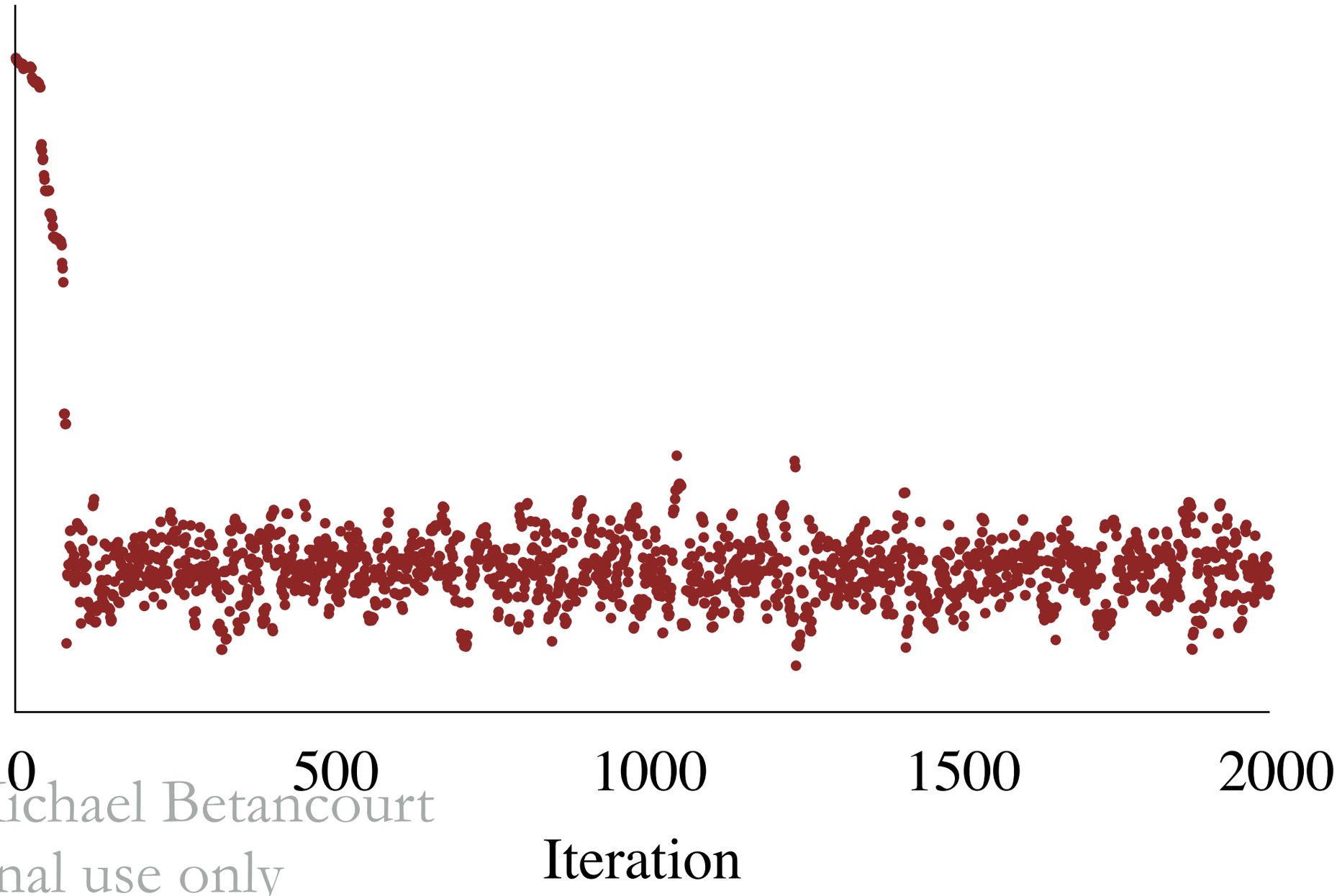
Visual diagnostics from *trace plots* is one immediate option for identify obstructive behavior.



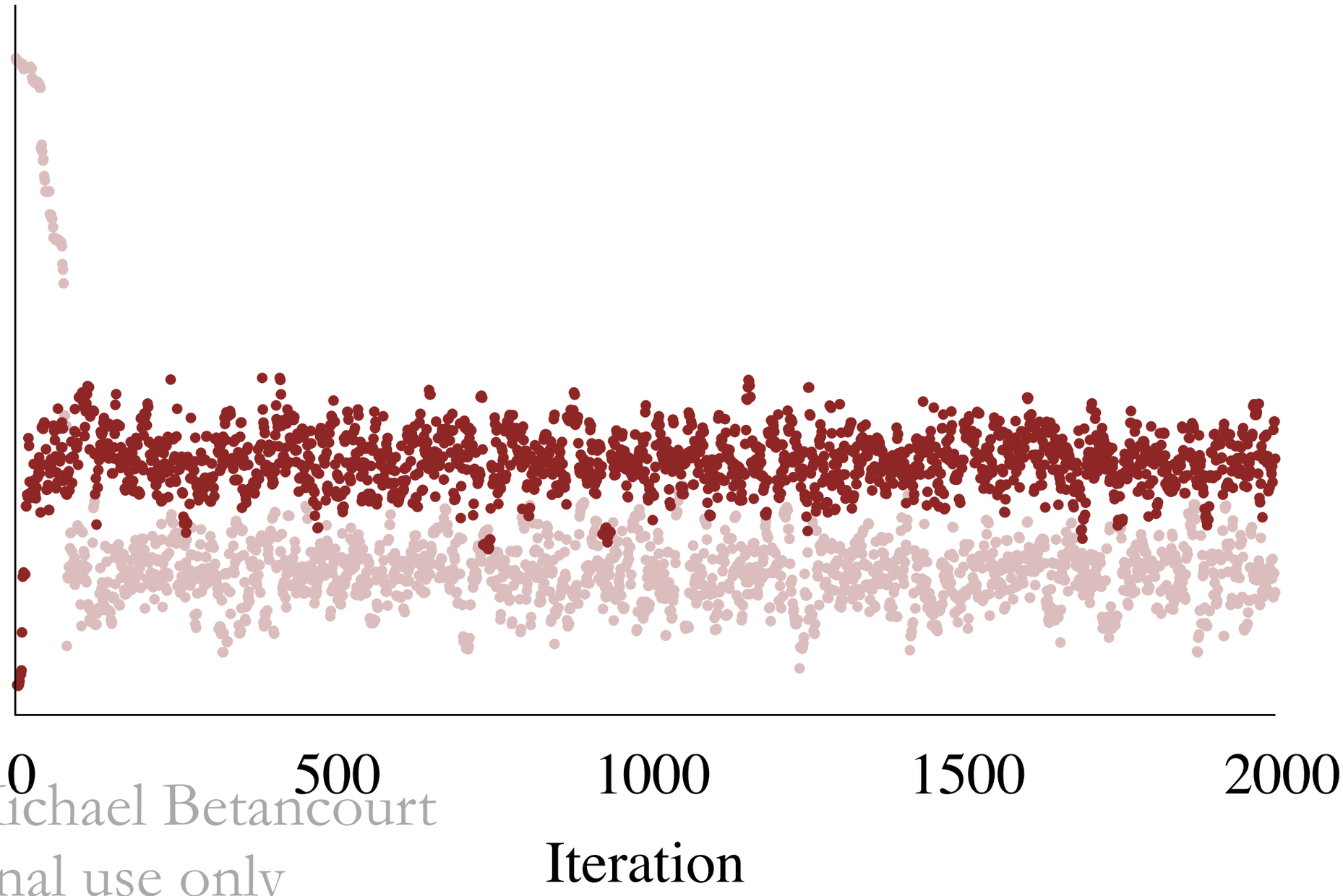
For example, we might identify regions of high curvature where Markov chain stick.



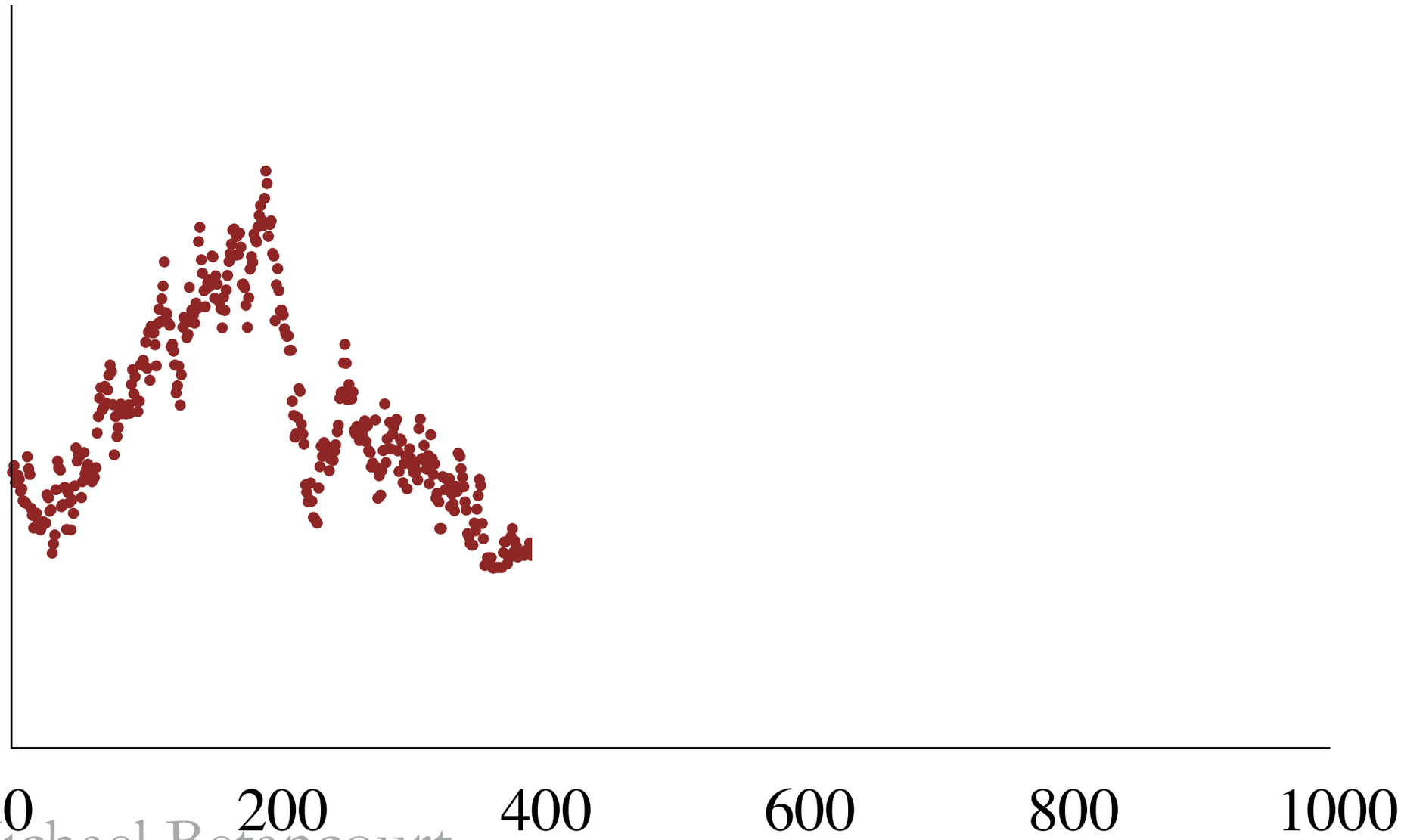
Unfortunately visual diagnostics can be misleading.



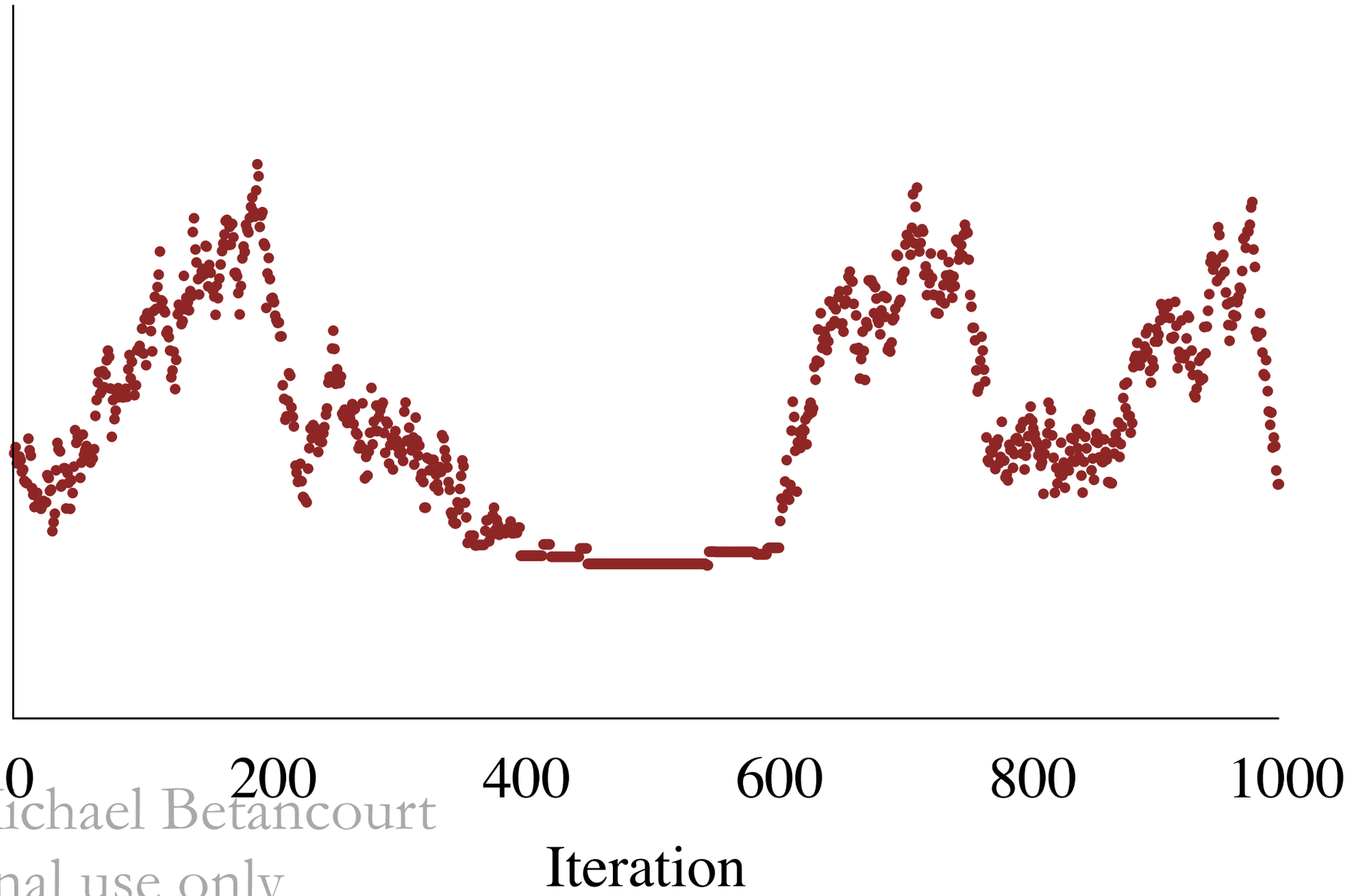
Unfortunately visual diagnostics can be misleading.



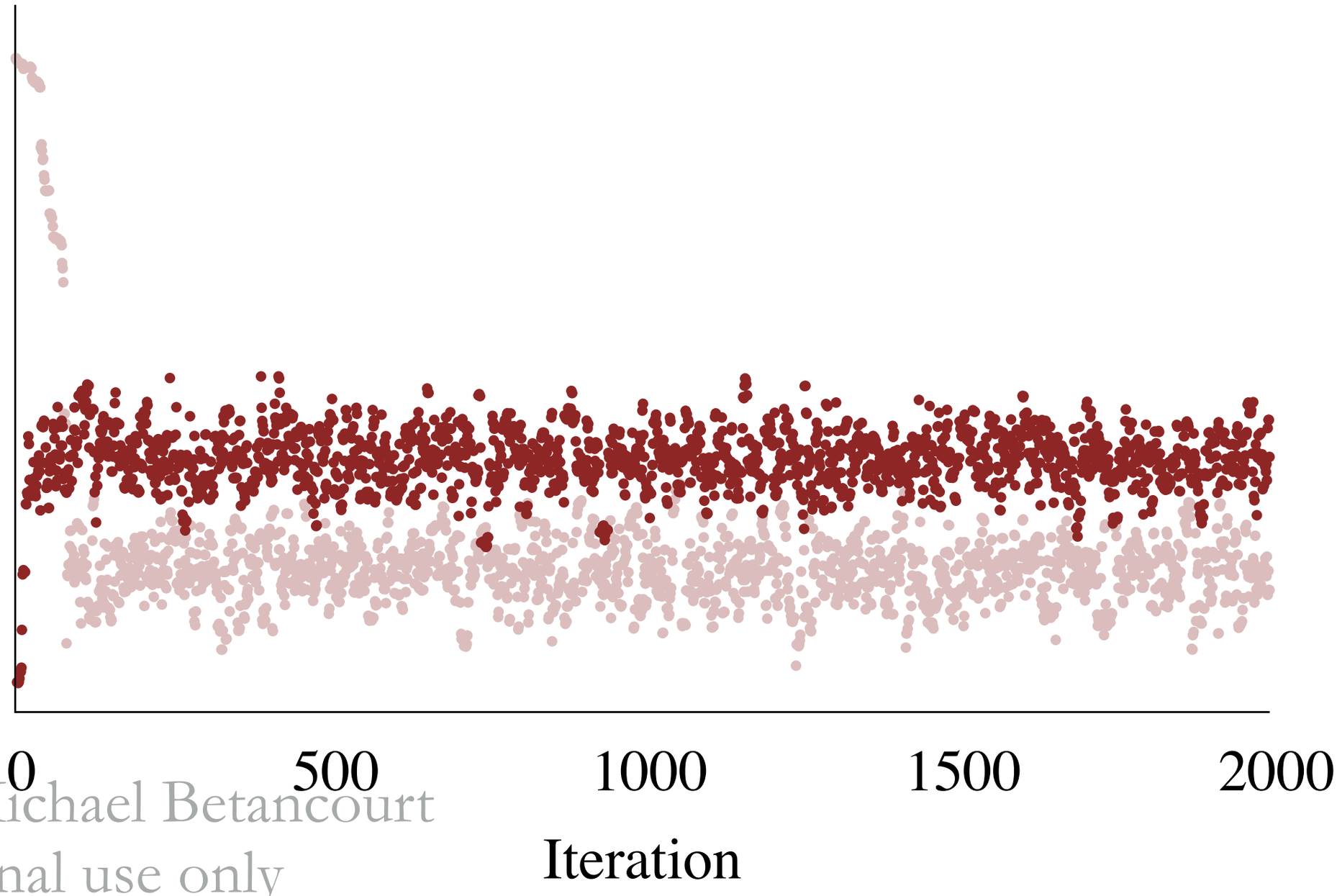
Unfortunately visual diagnostics can be misleading.



Unfortunately visual diagnostics can be misleading.



A more robust strategy runs *multiple* chains from *diffuse* initial states and compares the expectations to each other.



The potential scale reduction factor, or R-*hat*, is similar to an analysis of variance between multiple chains.

$$\hat{B}[f]$$

The potential scale reduction factor, or R-*hat*, is similar to an analysis of variance between multiple chains.

$$\hat{B}[f]$$

$$\hat{W}[f]$$

The potential scale reduction factor, or R-*hat*, is similar to an analysis of variance between multiple chains.

$$\hat{B}[f]$$

$$\overline{\hat{W}[f]}$$

The potential scale reduction factor, or *R-hat*, is similar to an analysis of variance between multiple chains.

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

The potential scale reduction factor, or *R-hat*, is similar to an analysis of variance between multiple chains.

Improved Rhat

In practice one should first verify that there are no identifiable obstructions to geometric ergodicity.

```
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 diagnostic failures then we can proceed to computing MCMC estimators.

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

$$\hat{f} \sim \mathcal{N}(\mathbb{E}[f], \widehat{\text{MCMC-SE}}[f])$$

$$\widehat{\text{MCMC-SE}}[f] = \sqrt{\frac{\widehat{\text{Var}}[f]}{\widehat{\text{ESS}}[f]}}$$

If there are no diagnostic failures then we can proceed to computing MCMC estimators.

```
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

As well as the corresponding effective sample sizes.

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

$$\hat{f} \sim \mathcal{N}(\mathbb{E}[f], \widehat{\text{MCMC-SE}}[f])$$

$$\widehat{\text{MCMC-SE}}[f] = \sqrt{\frac{\widehat{\text{Var}}[f]}{\widehat{\text{ESS}}[f]}}$$

As well as the corresponding effective sample sizes.

```
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

And finally the MCMC estimator standard errors.

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

$$\hat{f} \sim \mathcal{N}(\mathbb{E}[f], \widehat{\text{MCMC-SE}}[f])$$

$$\widehat{\text{MCMC-SE}}[f] = \sqrt{\frac{\widehat{\text{Var}}[f]}{\widehat{\text{ESS}}[f]}}$$

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

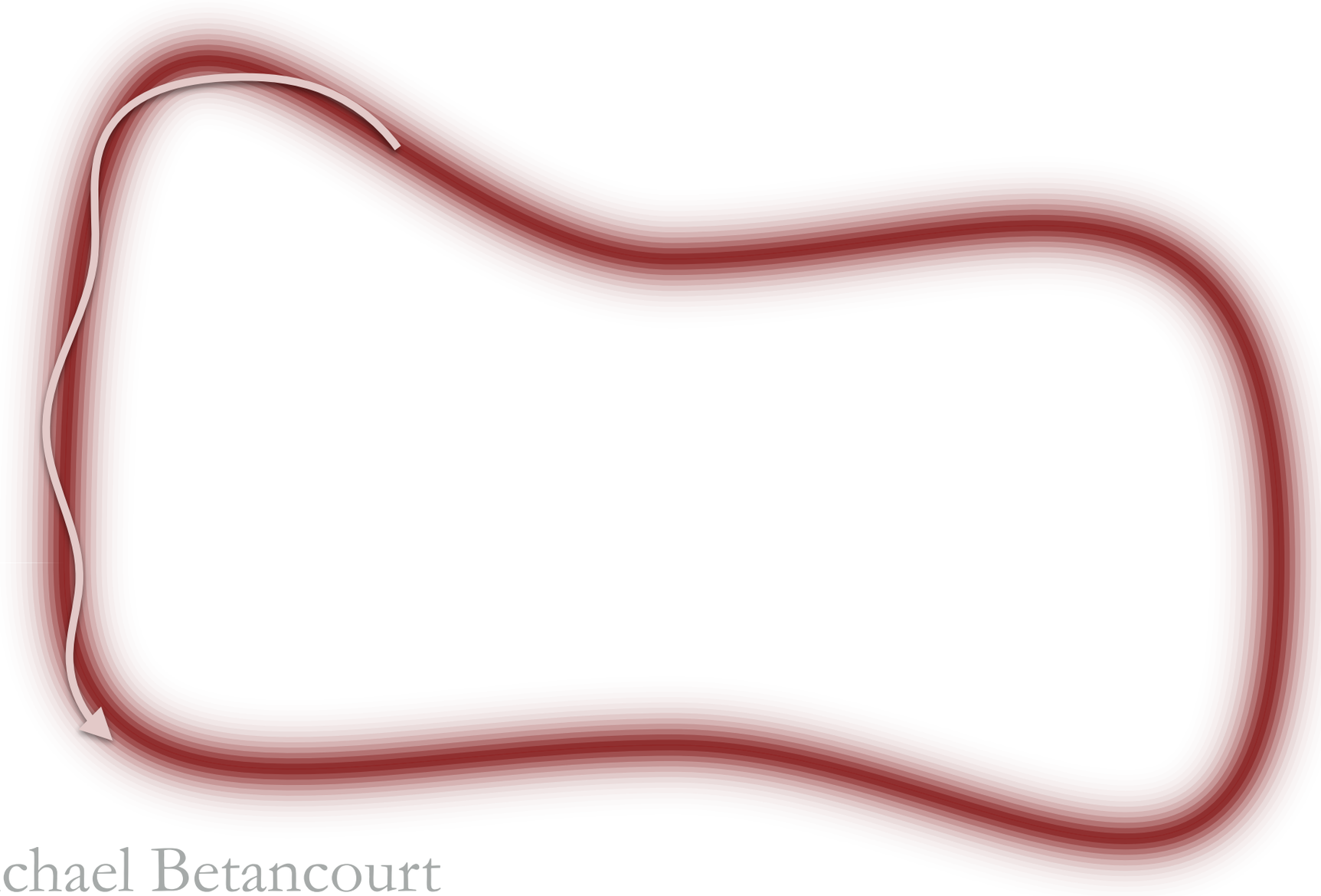
This is not the only approach -- various *preasymptotic* methods attempt to bound the convergence directly.

$$|| \, T^N \delta_{\theta_0} - \pi || \leq f(\theta_0, N)$$

This is not the only approach -- various *preasymptotic* methods attempt to bound the convergence directly.

$$|| \, T^N \delta_{\theta_0} - \pi || \leq \hat{f}(\theta_0, N)$$

Hamiltonian Monte Carlo



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

$$\pi_S(\theta \mid \tilde{y}) = \int \mathrm{d}\theta' \, \pi_S(\theta' \mid \tilde{y}) \, T(\theta \mid \theta')$$

To simplify notation let's generalize from a posterior distribution to an arbitrary target distribution.

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

How exactly can we design a Markov transition that preserves a given target distribution?

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

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

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

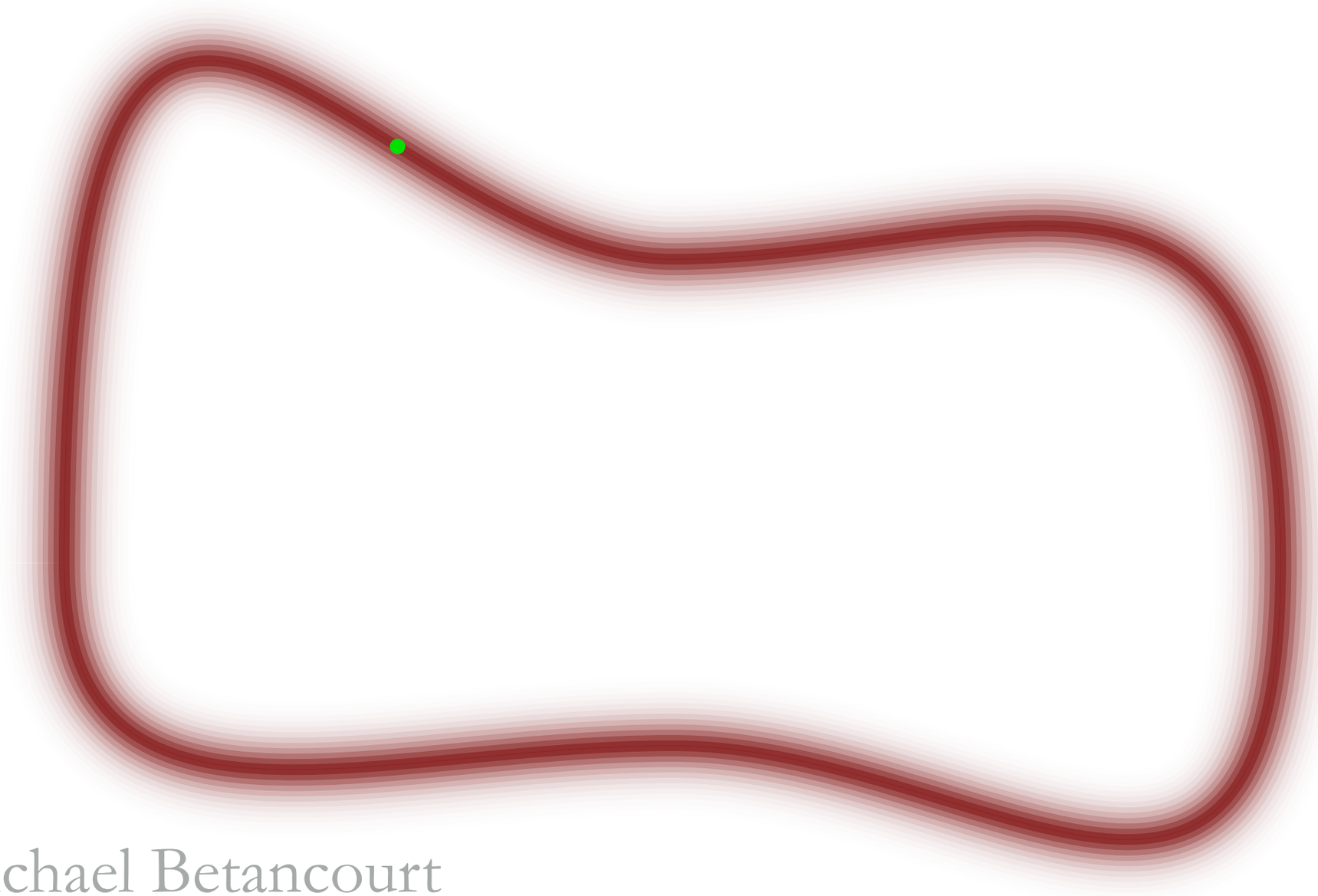
One way to construct a chain is Random Walk Metropolis which explores the posterior with a "guided" diffusion.

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

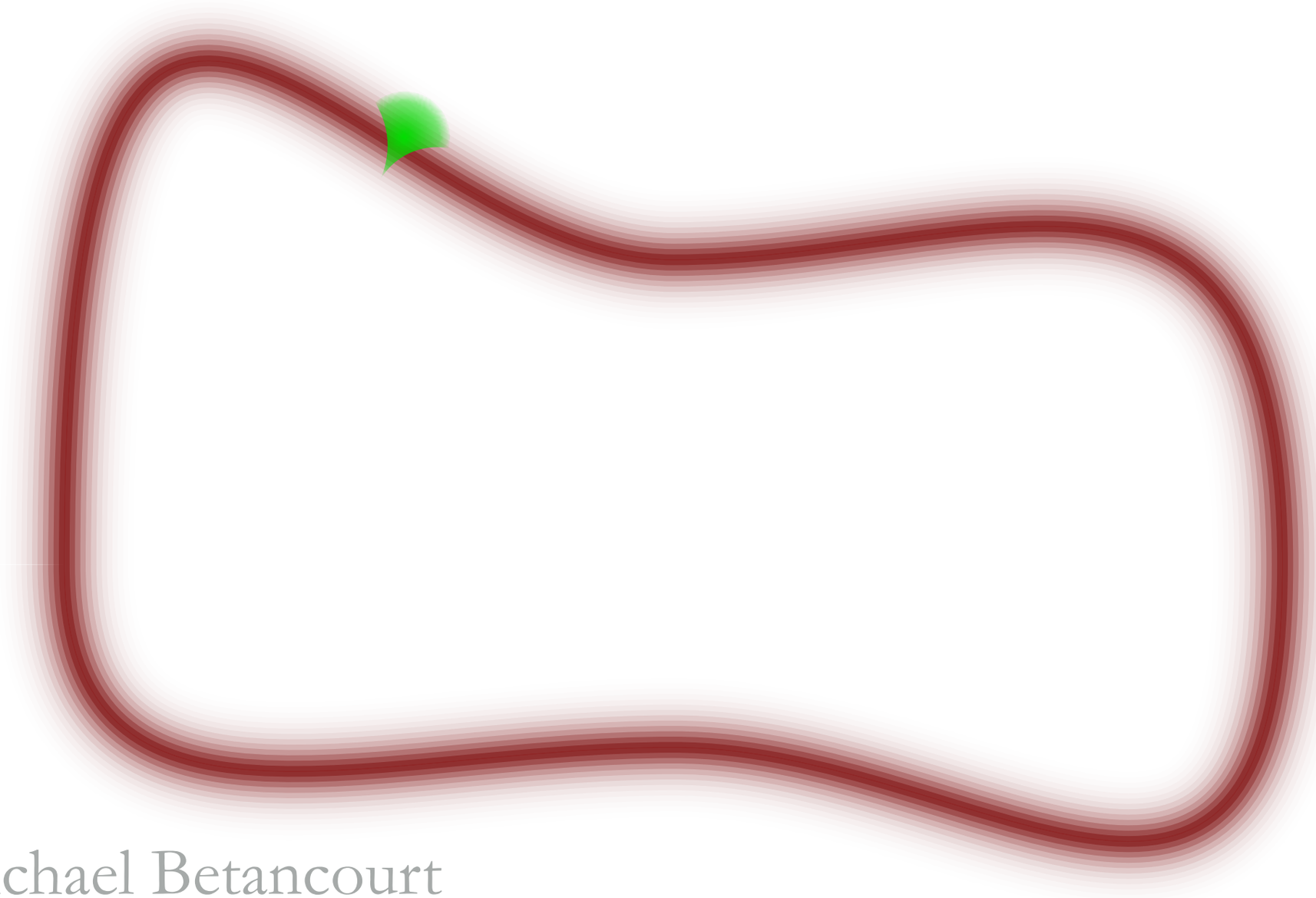
One way to construct a chain is Random Walk Metropolis which explores the posterior with a "guided" diffusion.

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

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.