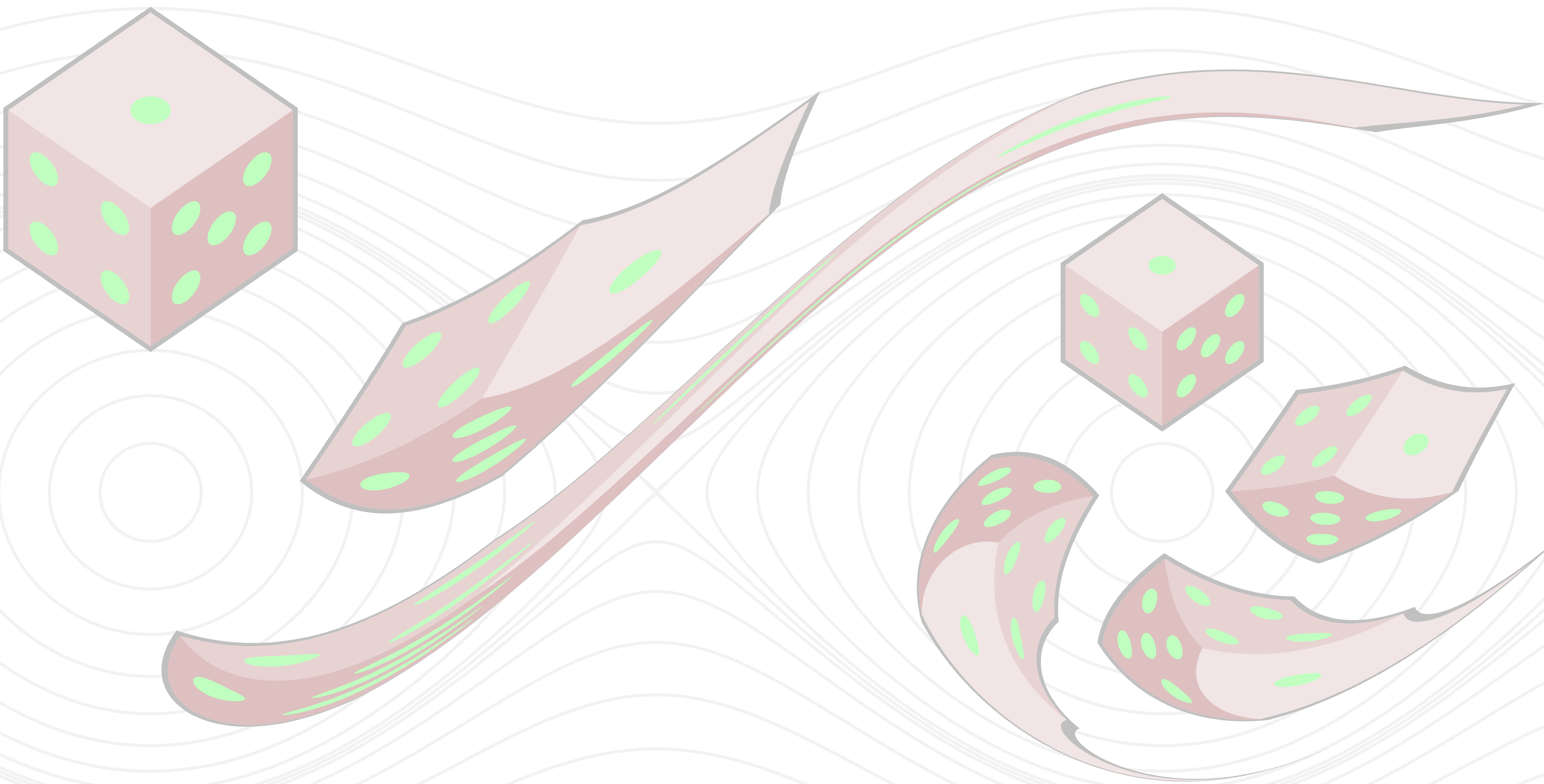
Scalable Bayesian Computation



Michael Betancourt @betanalpha
Applied Statistics Center
Columbia University

DS@HEP,
Fermilab
May 11, 2017

Statistical models are quantified by collections of data generating processes, or *likelihoods*.

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

Models are readily built by thinking *generatively*: don't try to "correct" the data but rather model its variation.

$$\mathcal{D}$$

Models are readily built by thinking *generatively*: don't try to "correct" the data but rather model its variation.

$$\mathcal{D} \rightarrow f(\mathcal{D}) \approx \theta$$

Models are readily built by thinking *generatively*: don't try to "correct" the data but rather model its variation.

$$\cancel{\mathcal{D} \to f(\mathcal{D}) \approx \theta}$$

Models are readily built by thinking *generatively*: don't try to "correct" the data but rather model its variation.

$$\cancel{\mathcal{D} \to f(\mathcal{D}) \approx \theta}$$

$$\pi(\phi \mid \theta)$$

Models are readily built by thinking *generatively*: don't try to "correct" the data but rather model its variation.

$$\cancel{\mathcal{D} \to f(\mathcal{D}) \approx \theta}$$

$$\pi(\psi \mid \phi)\pi(\phi \mid \theta)$$

Models are readily built by thinking *generatively*: don't try to "correct" the data but rather model its variation.

$$\cancel{\mathcal{D} \to f(\mathcal{D}) \approx \theta}$$

$$\pi(\mathcal{D} \mid \psi)\pi(\psi \mid \phi)\pi(\phi \mid \theta)$$

Models are readily built by thinking *generatively*: don't try to "correct" the data but rather model its variation.

$$\cancel{\mathcal{D} \to f(\mathcal{D}) \approx \theta}$$

$$\int \mathrm{d}\psi \, \mathrm{d}\phi \, \pi(\mathcal{D} \mid \psi) \pi(\psi \mid \phi) \pi(\phi \mid \theta)$$

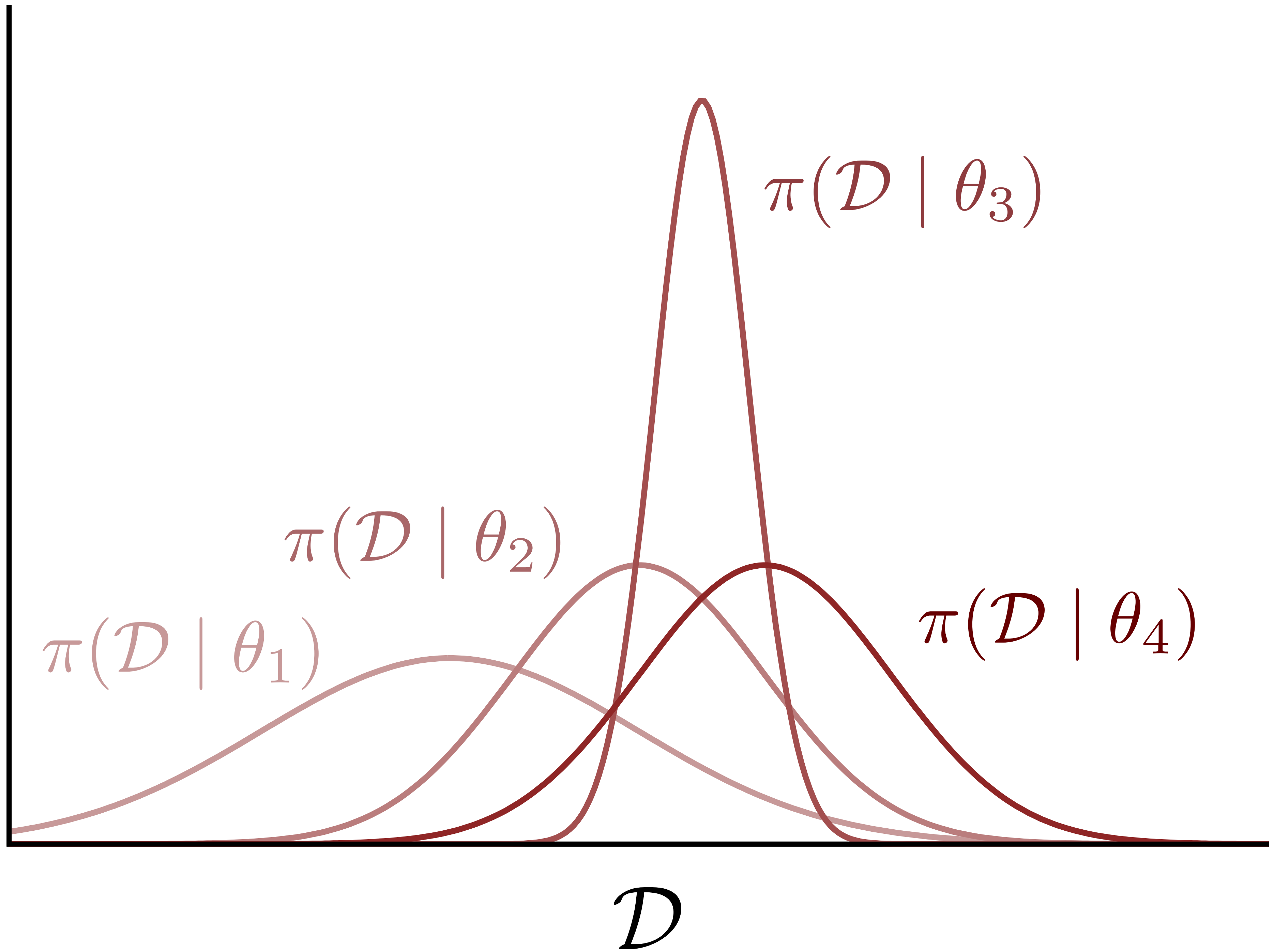
Models are readily built by thinking *generatively*: don't try to "correct" the data but rather model its variation.

$$\cancel{\mathcal{D} \to f(\mathcal{D}) \approx \theta}$$

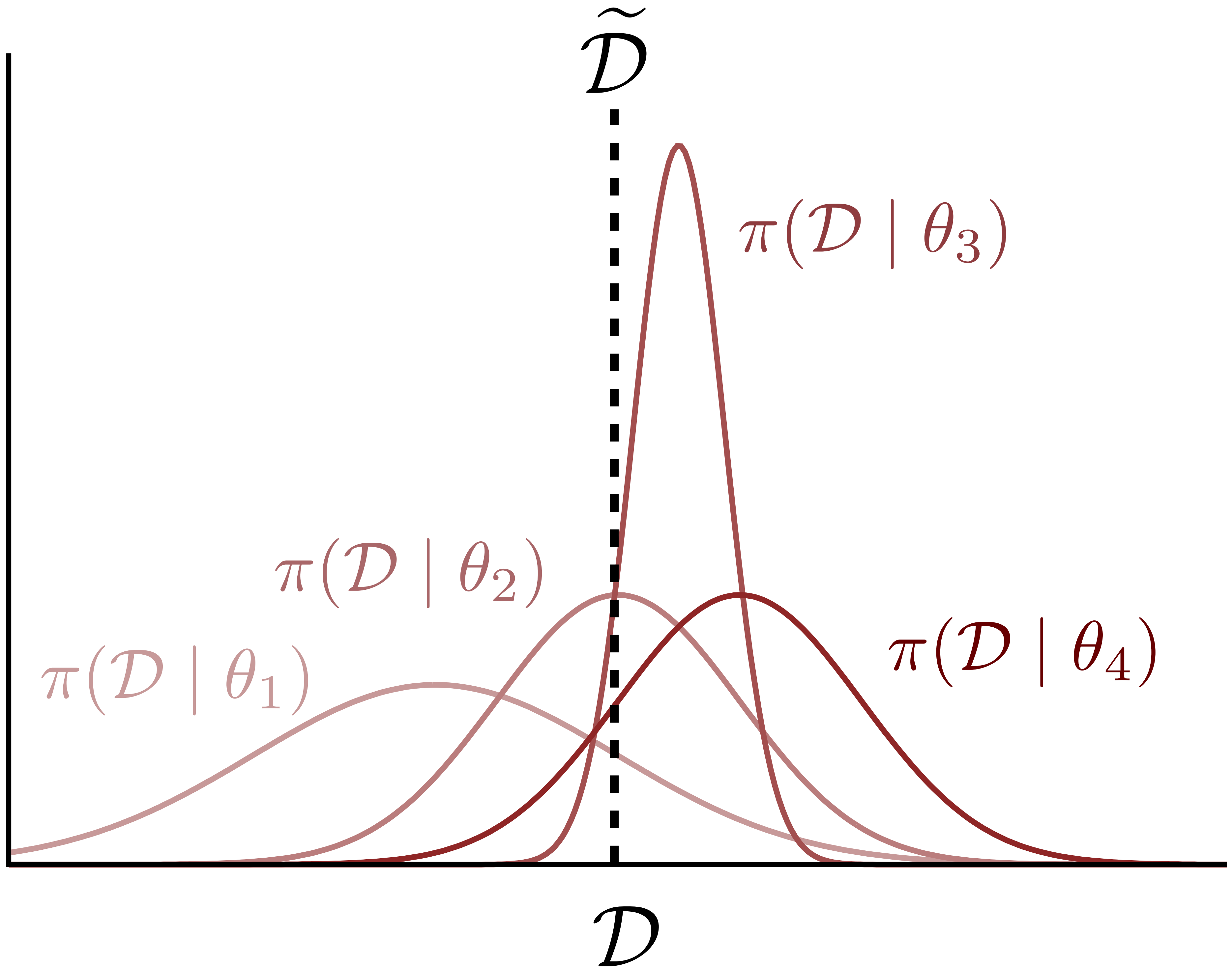
$$\int \mathrm{d}\psi \, \mathrm{d}\phi \, \pi(\mathcal{D} \mid \psi) \pi(\psi \mid \phi) \pi(\phi \mid \theta)$$

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

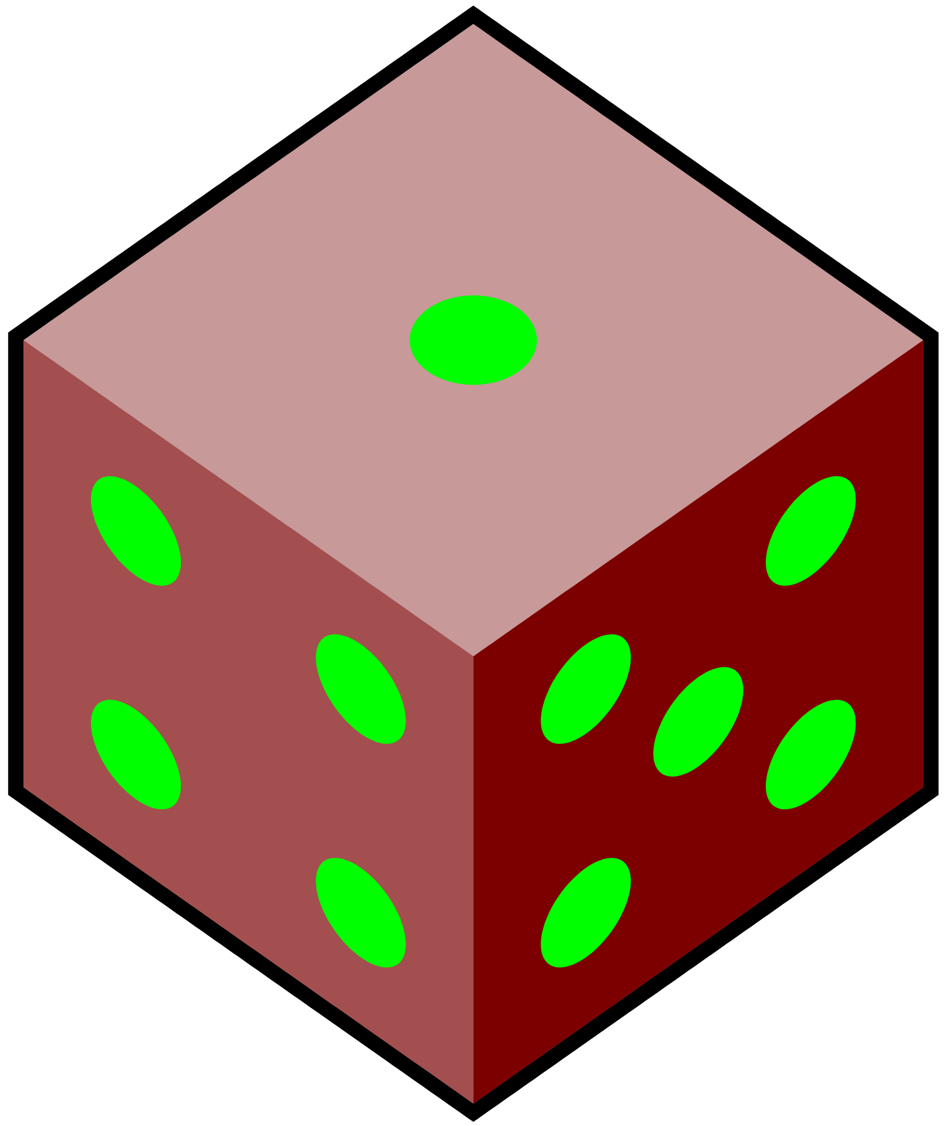
Inference identifies the model configurations that yield data distributions *consistent* with a given measurement.



For any given measurement, however, there will be many consistent configurations -- *uncertainty is fundamental*.



Exactly how we define consistency, however, depends on how we define probability itself.



In *frequentist statistics*, probability is defined in terms of frequencies, and hence can be applied to only the data.

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

In *frequentist statistics*, probability is defined in terms of frequencies, and hence can be applied to only the data.

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

In *frequentist statistics*, probability is defined in terms of frequencies, and hence can be applied to only the data.

$$\pi(\mathcal{D} \mid \boldsymbol{\theta})$$

Frequentist methods compute expectations with respect to the data to identify estimators that work well *on average*.

$$\hat{\theta}(\mathcal{D})$$

Frequentist methods compute expectations with respect to the data to identify estimators that work well *on average*.

$$\hat{\theta}(\mathcal{D})$$

$$\mathcal{L}(\hat{\theta}(\mathcal{D}), \theta)$$

Frequentist methods compute expectations with respect to the data to identify estimators that work well *on average*.

$$\hat{\theta}(\mathcal{D})$$

$$\mathcal{L}(\theta) = \int \mathcal{L}(\hat{\theta}(\mathcal{D})\,, \theta) \pi(\mathcal{D} \mid \theta) \, \mathrm{d}\mathcal{D}$$

Bayesian methods generalize the frequentist perspective, modeling the data *and* the parameters with probabilities.

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

Bayesian methods generalize the frequentist perspective, modeling the data *and* the parameters with probabilities.

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

Bayesian methods generalize the frequentist perspective, modeling the data *and* the parameters with probabilities.

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

The probability distribution encoding the consistency of model configurations is given by *Bayes' Theorem*.

$$\pi(\theta)$$

The probability distribution encoding the consistency of model configurations is given by *Bayes' Theorem*.

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

The probability distribution encoding the consistency of model configurations is given by *Bayes' Theorem*.

$$\pi(\theta \mid \mathcal{D}) \propto \pi(\mathcal{D} \mid \theta)\pi(\theta)$$

Importantly, in a Bayesian analysis *all* inferential queries are answered by posterior expectations.

$$\mathbb{E}[f] = \int \mathrm{d}\theta \, \pi(\theta \mid \mathcal{D}) \, f(\theta)$$

Consequently we have *very* different computational problems from these two inferential approaches.

Frequentist

Design estimator, specify loss function, verify acceptable expected loss

Bayesian

Specify prior, compute posterior expectations

Consequently we have *very* different computational problems from these two inferential approaches.

Frequentist	***Bayesian***
Design estimator, specify loss function, verify acceptable expected loss	Specify prior, compute posterior expectations

Asymptotics

Consequently we have *very* different computational problems from these two inferential approaches.

Frequentist	***Bayesian***
Design estimator, specify loss function, verify acceptable expected loss	Specify prior, compute posterior expectations

Asymptotics

Maximum likelihood

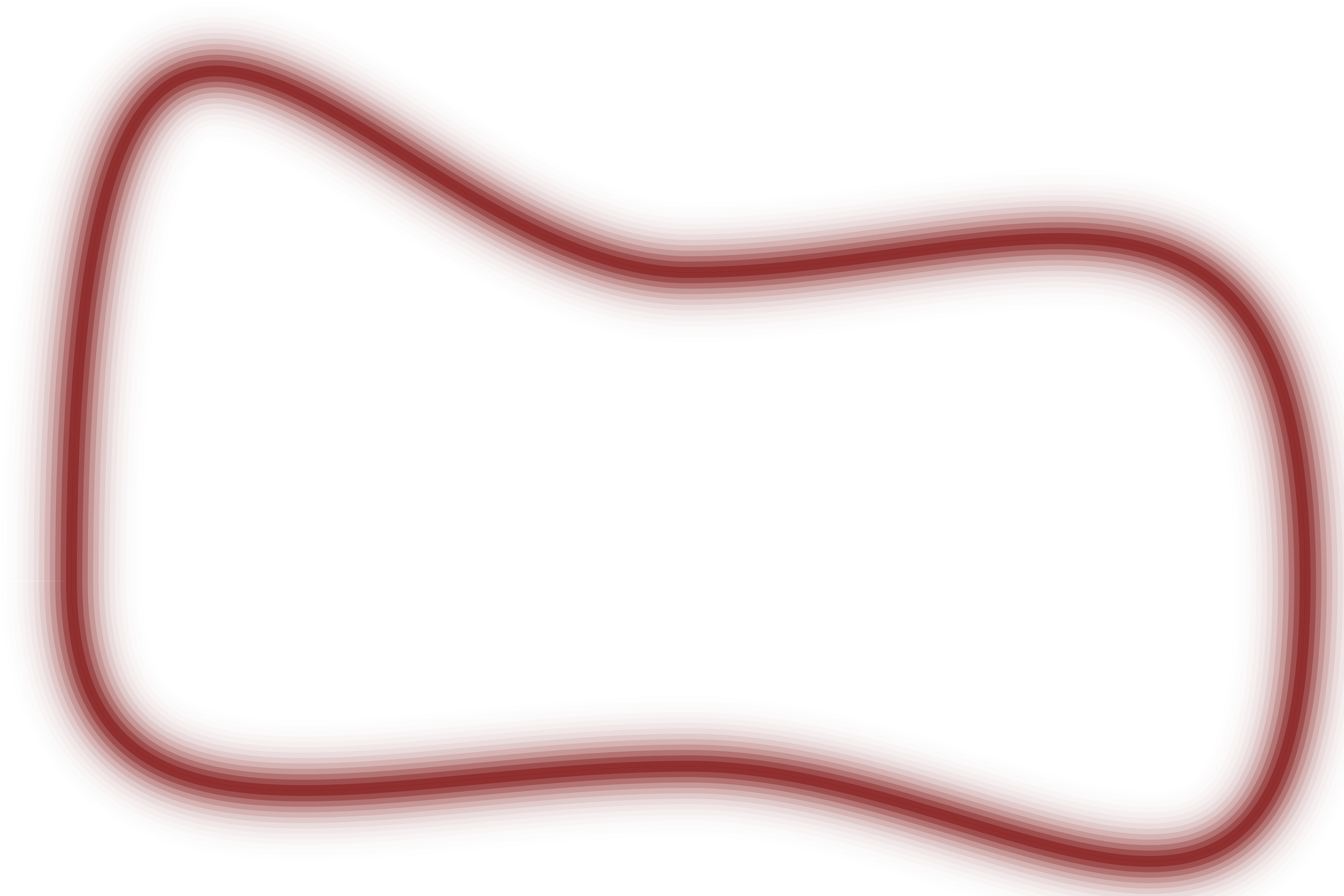
Consequently we have *very* different computational problems from these two inferential approaches.

Frequentist	***Bayesian***
Design estimator, specify loss function, verify acceptable expected loss	Specify prior, compute posterior expectations

Asymptotics

Maximum likelihood	Laplace approximation, etc

Why Is Bayesian Computation So Hard?



Bayesian computation is hard because high-dimensional integration is hard.

$$\mathbb{E}[f] = \int \mathrm{d}\theta \, \pi(\theta \mid \mathcal{D}) \, f(\theta)$$

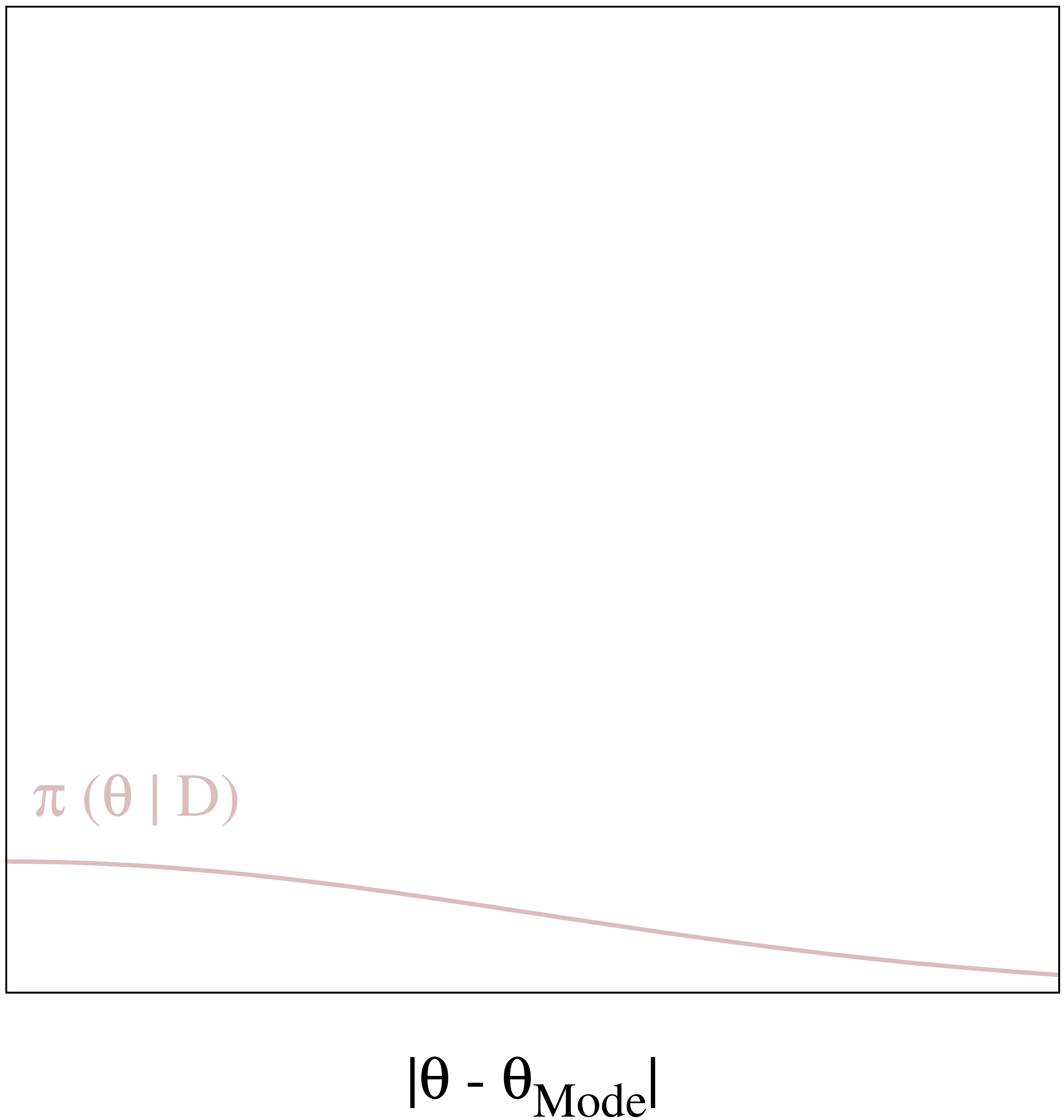
Bayesian computation is hard because high-dimensional integration is hard.

$$\mathbb{E}[f] = \int \mathrm{d}\theta \, \pi(\theta \mid \mathcal{D}) \, f(\theta)$$

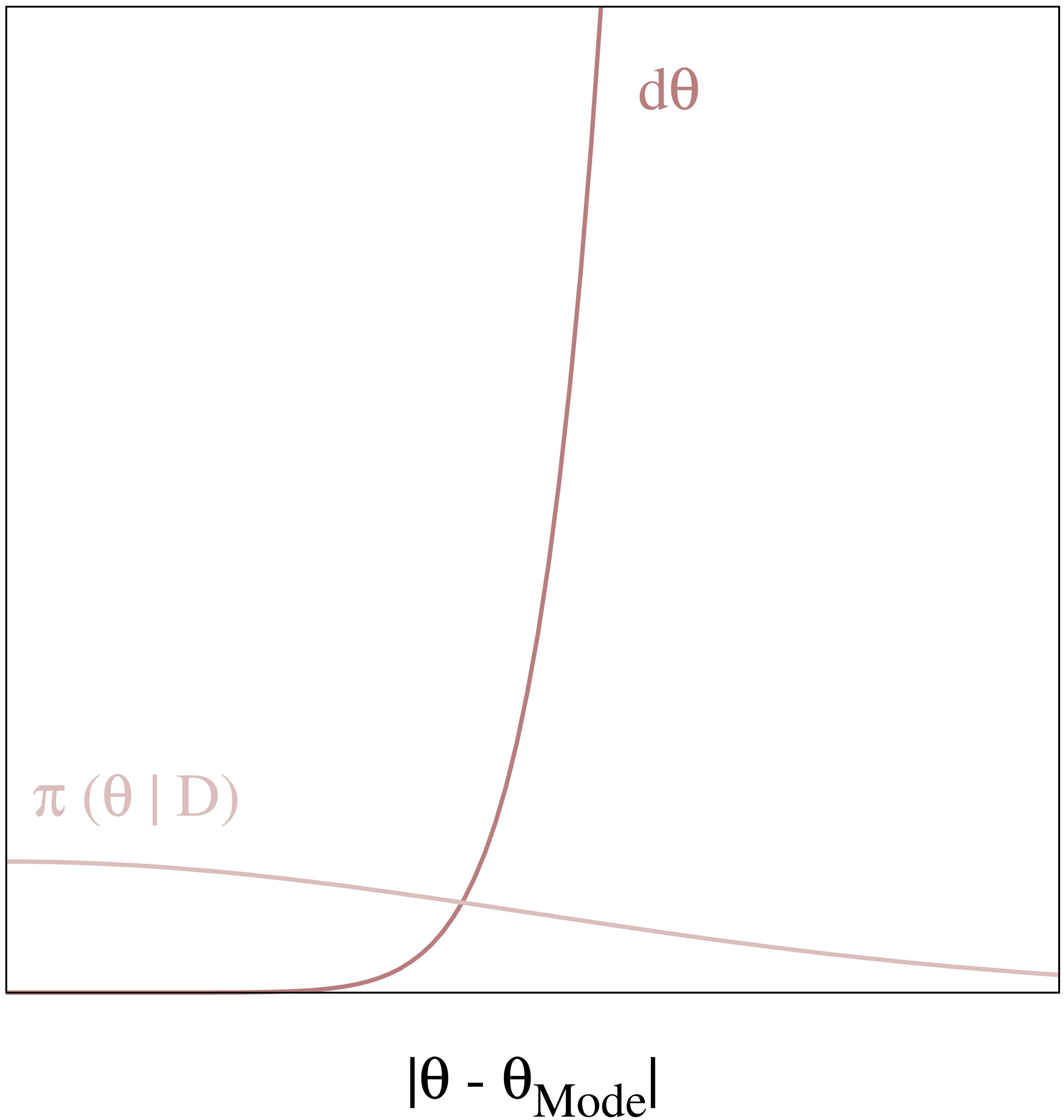
Bayesian computation is hard because high-dimensional integration is hard.

$$\mathbb{E}[f] = \int \mathrm{d}\theta \, \pi(\theta \mid \mathcal{D}) \, f(\theta)$$

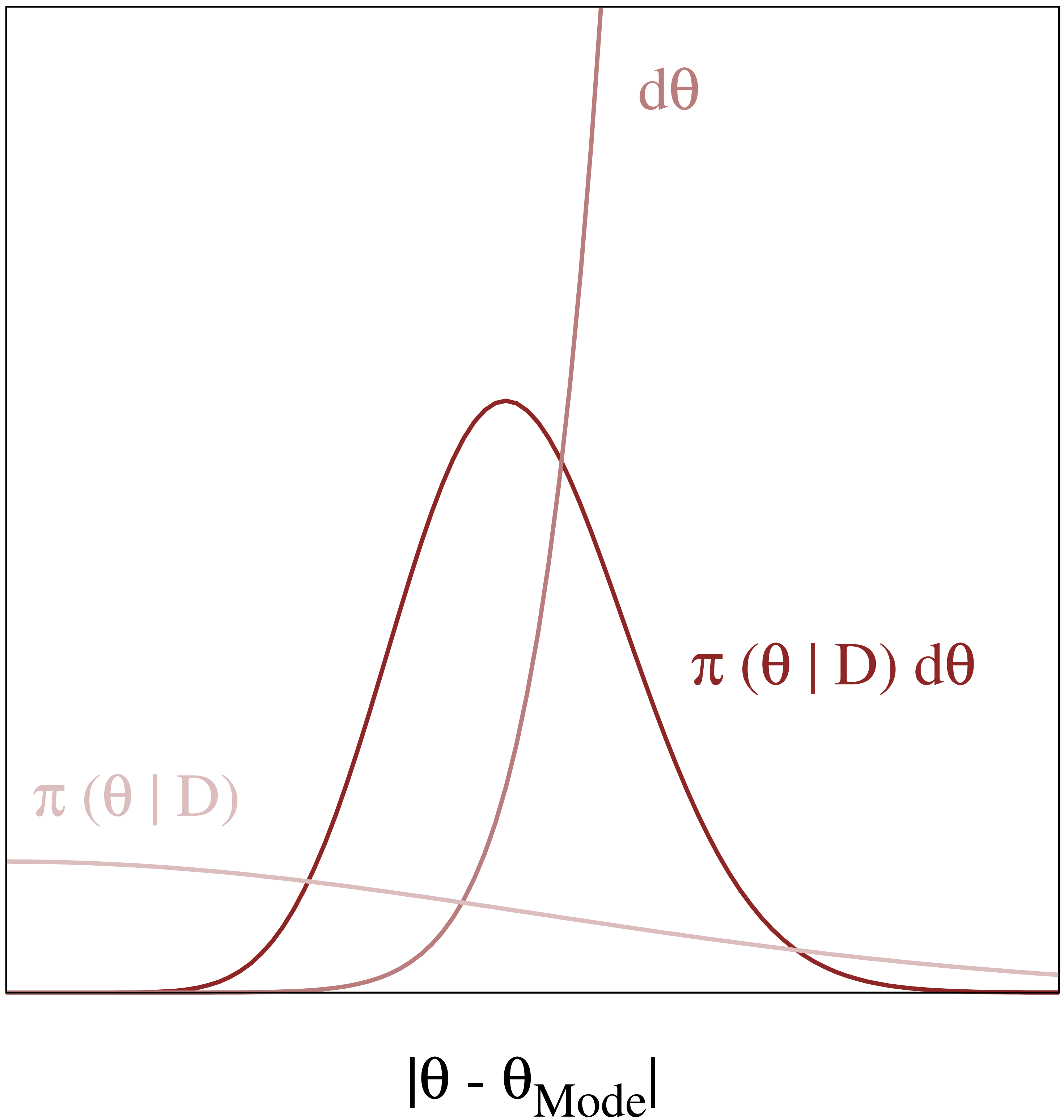
Computationally important regions are determined not by probability *density* but rather by probability *mass*.



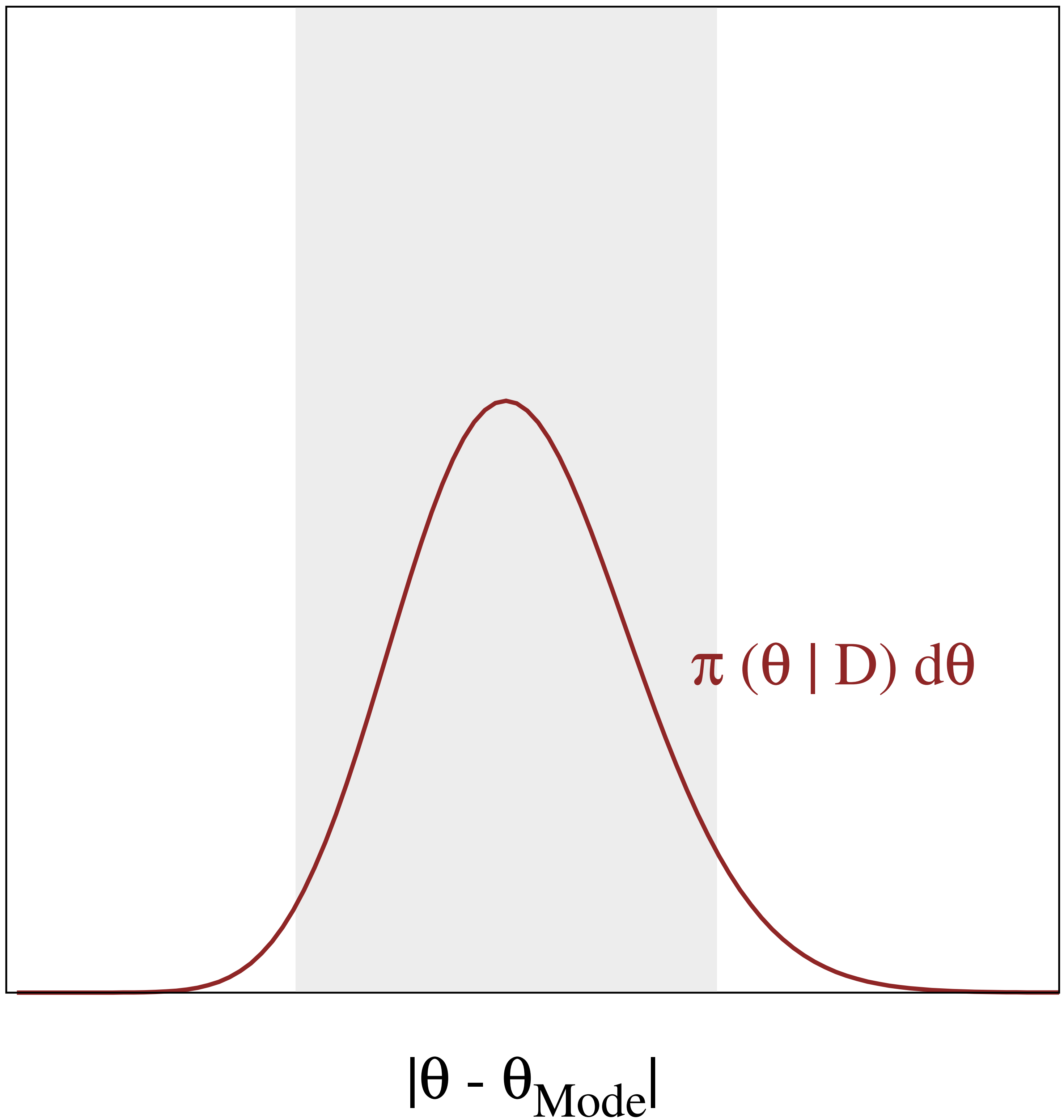
Computationally important regions are determined not by probability *density* but rather by probability *mass*.



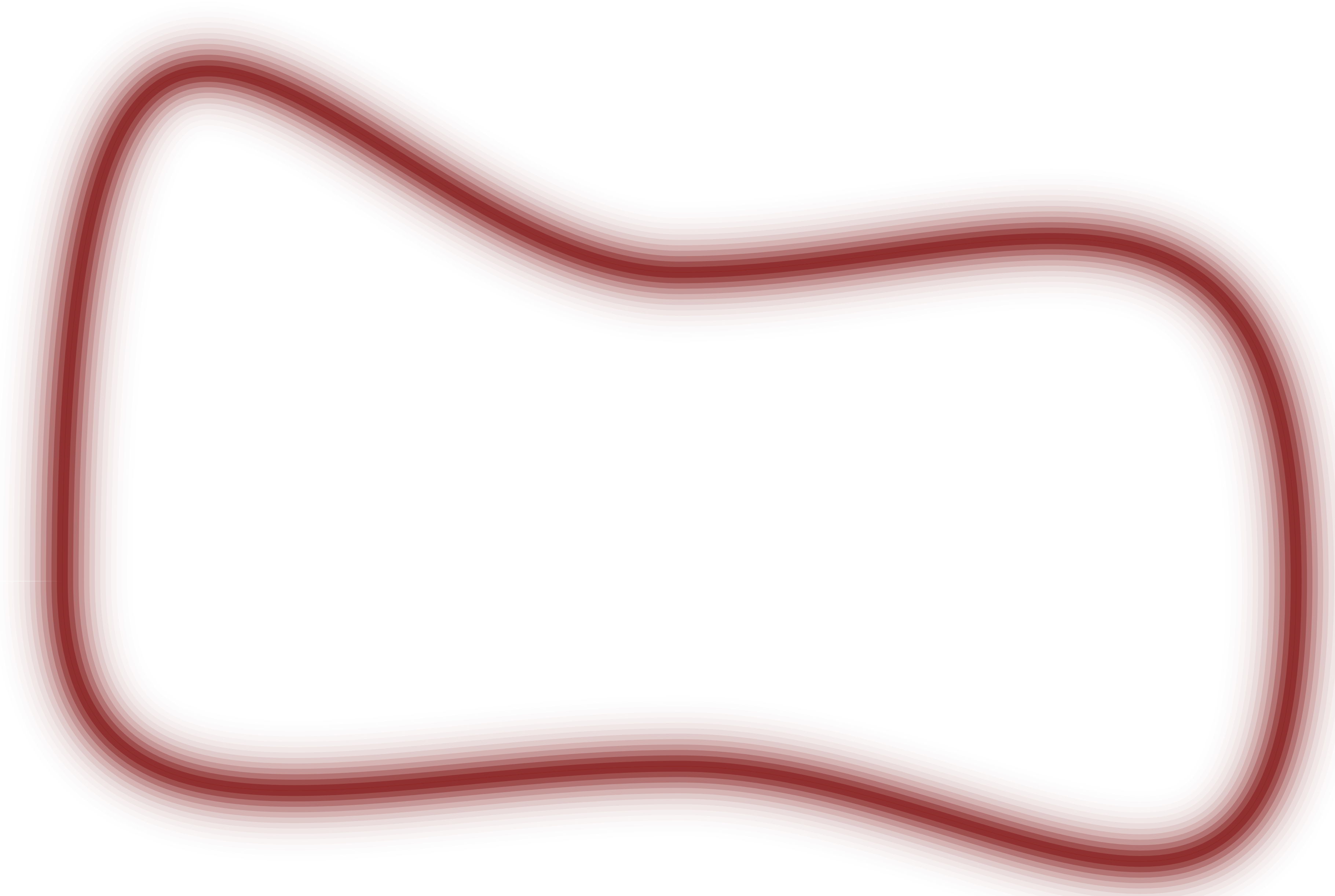
Computationally important regions are determined not by probability *density* but rather by probability *mass*.



As the dimensionality of the model increases, probability mass concentrates on a hypersurface called the *typical set*.

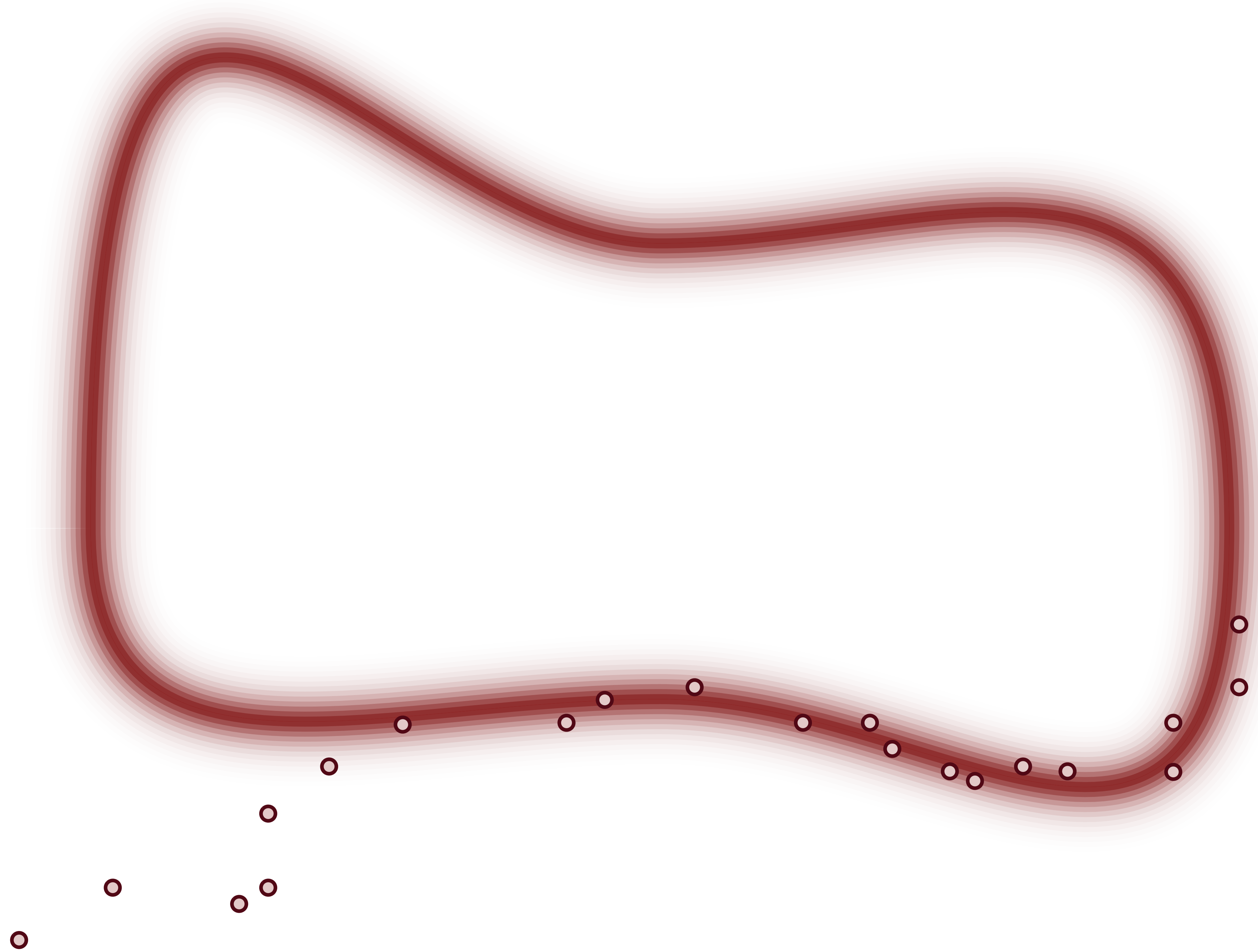


The concentration of probability mass into a singular typical set frustrates the accurate estimation of integrals.

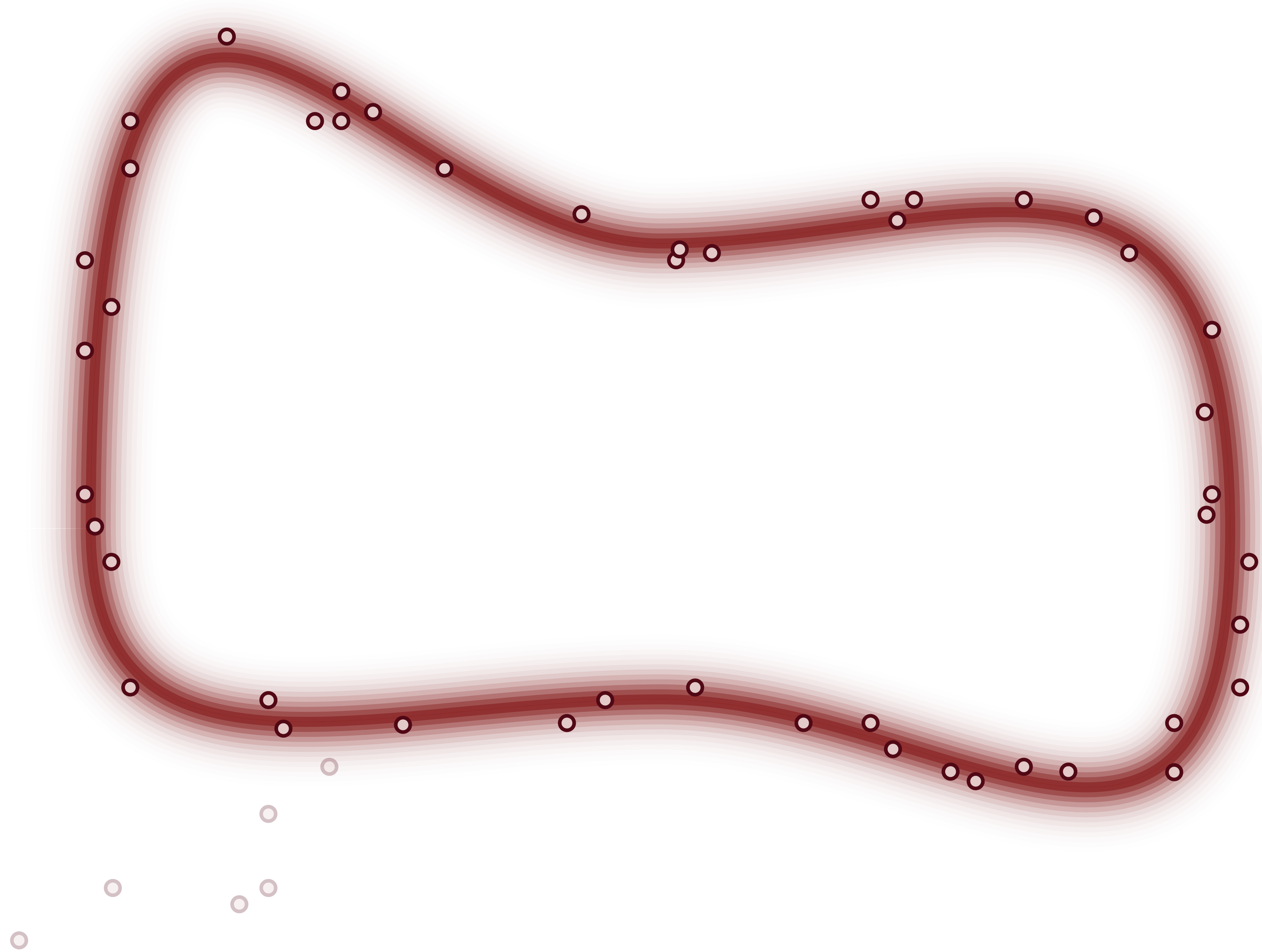


Markov chains, however, provide a particularly generic scheme for finding and then exploring this typical set.

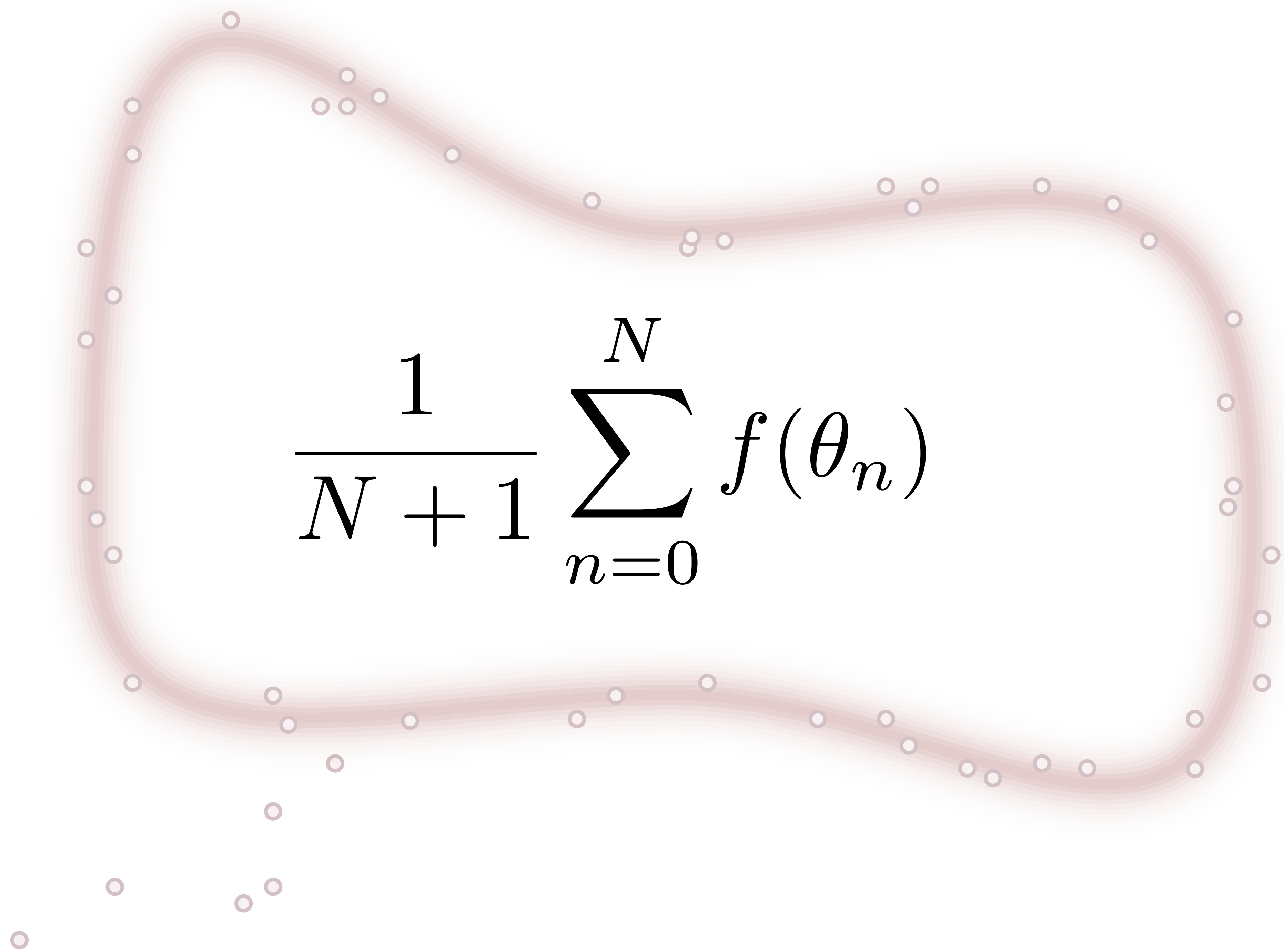
Markov chains, however, provide a particularly generic scheme for finding and then exploring this typical set.



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

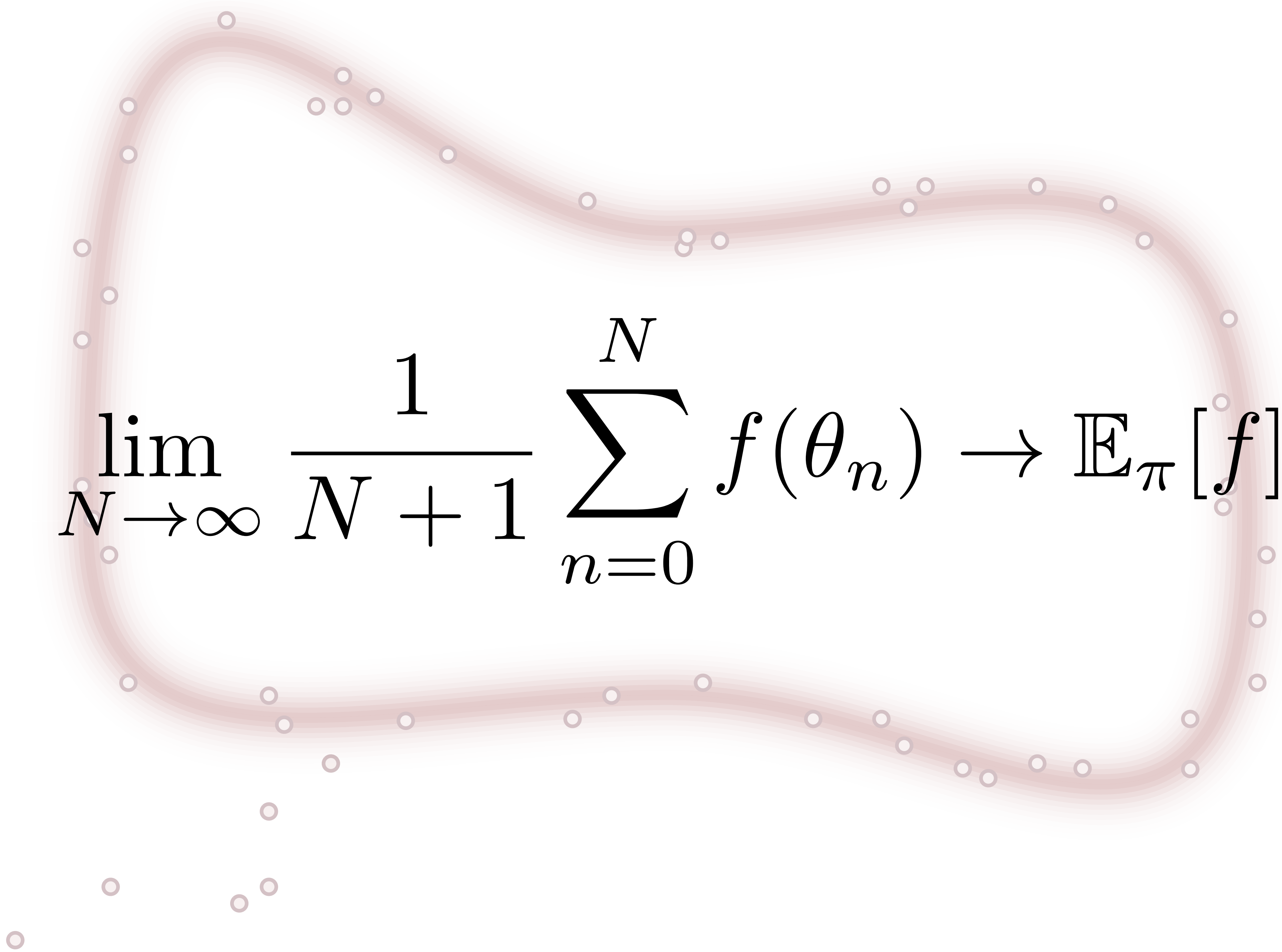


If run long enough the Markov chain defines consistent *Markov Chain Monte Carlo estimators*.



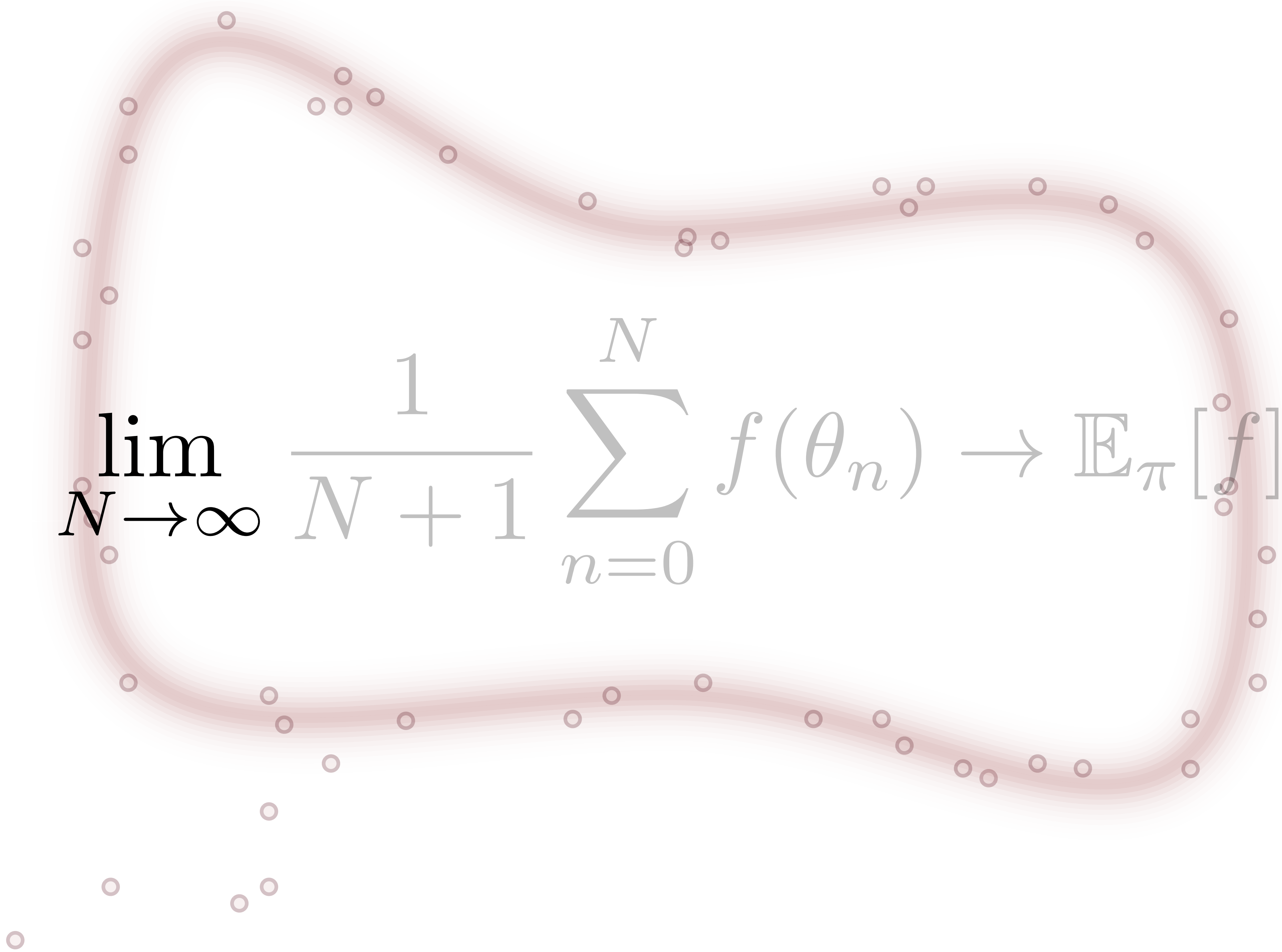
$$\frac{1}{N+1}\sum_{n=0}^{N} f(\theta_n)$$

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



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

But practical performance depends on how quickly the Markov chain can explore the typical set.

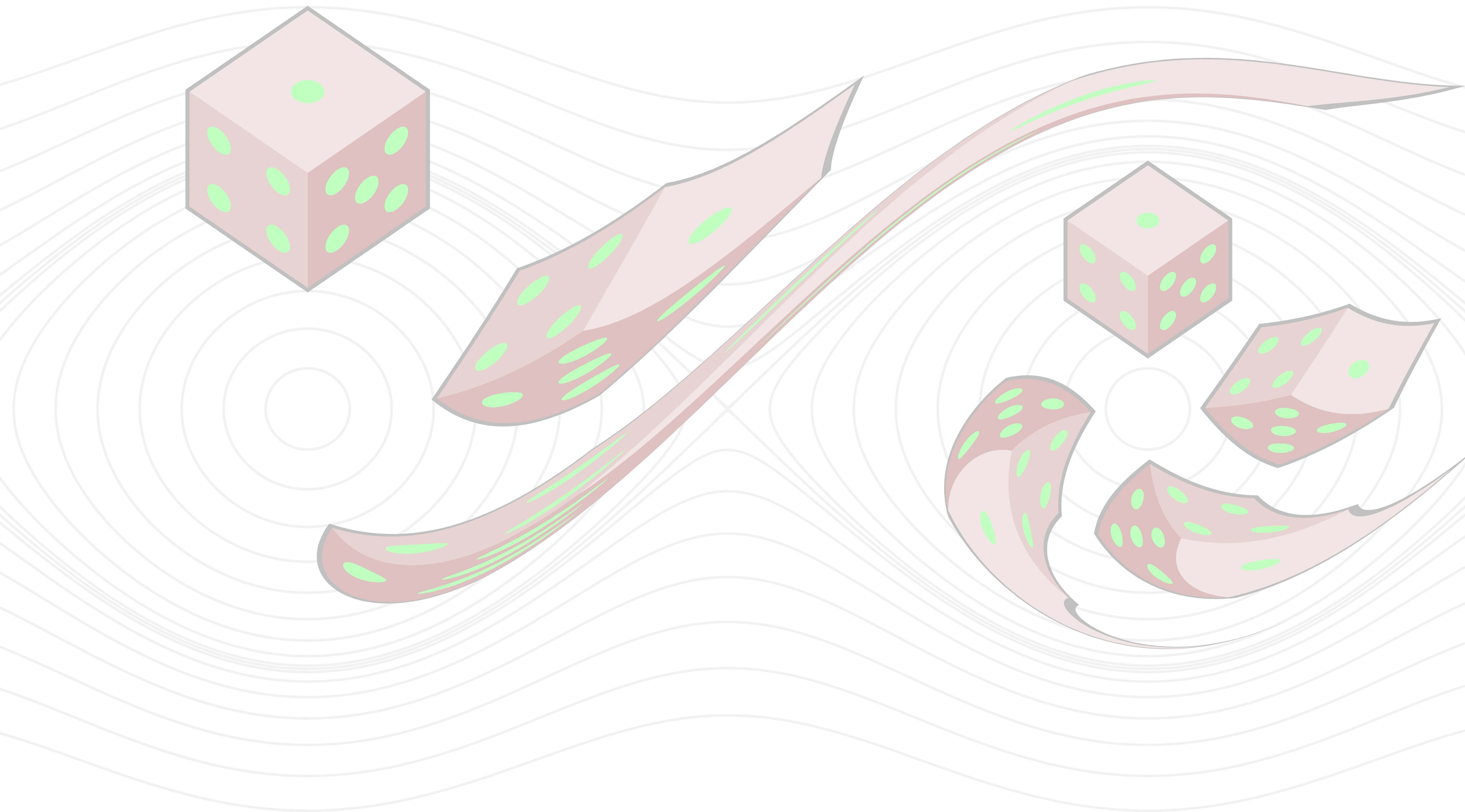


$$\lim_{N \to \infty} \frac{1}{N+1} \sum_{n=0}^{N} f(\theta_n) \to \mathbb{E}_{\pi}[f]$$

In order to scale Bayesian inference to high-dimensional problems we need *coherent* exploration of the typical set.

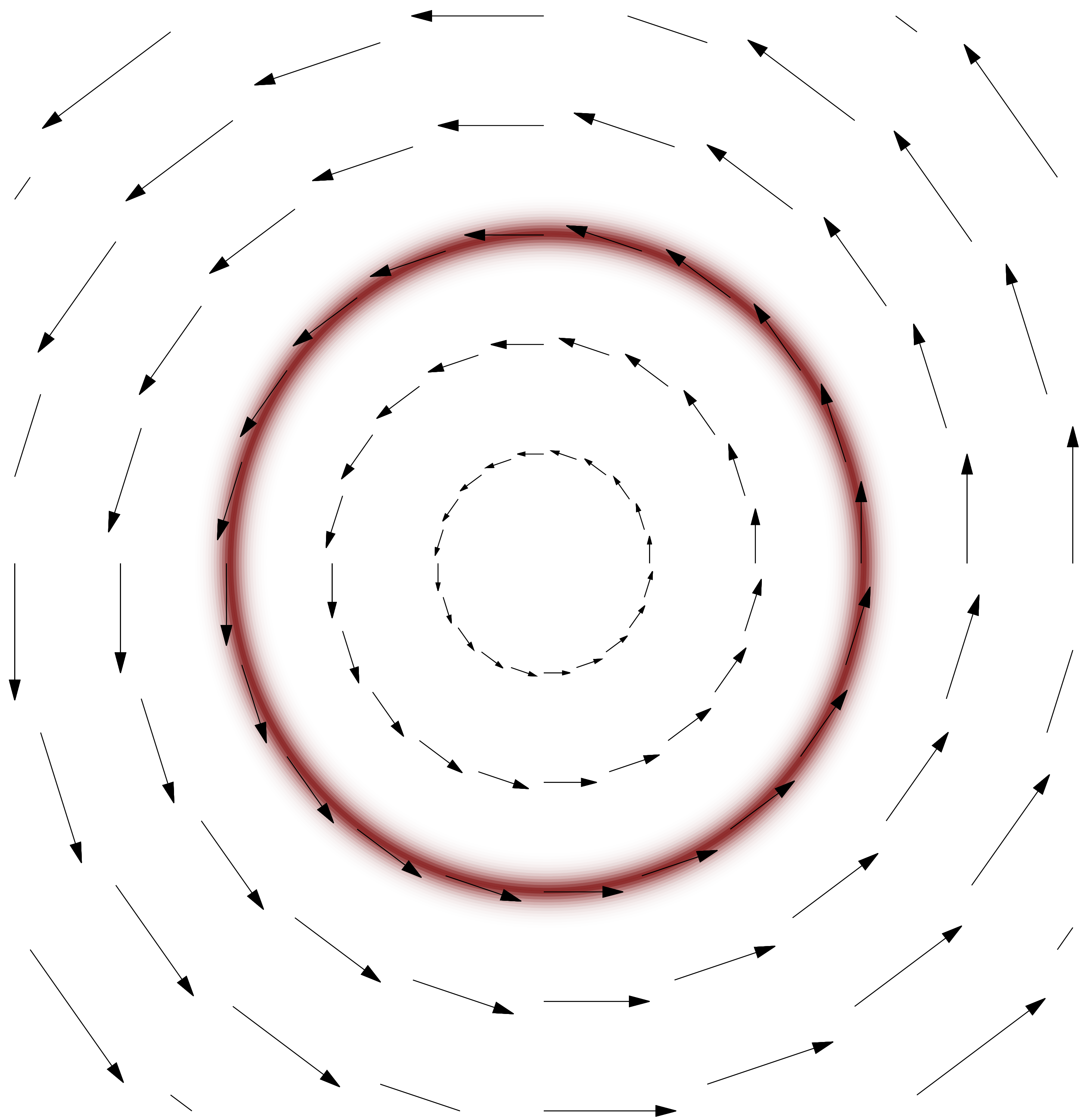
In order to scale Bayesian inference to high-dimensional problems we need *coherent* exploration of the typical set.

Hamiltonian Monte Carlo

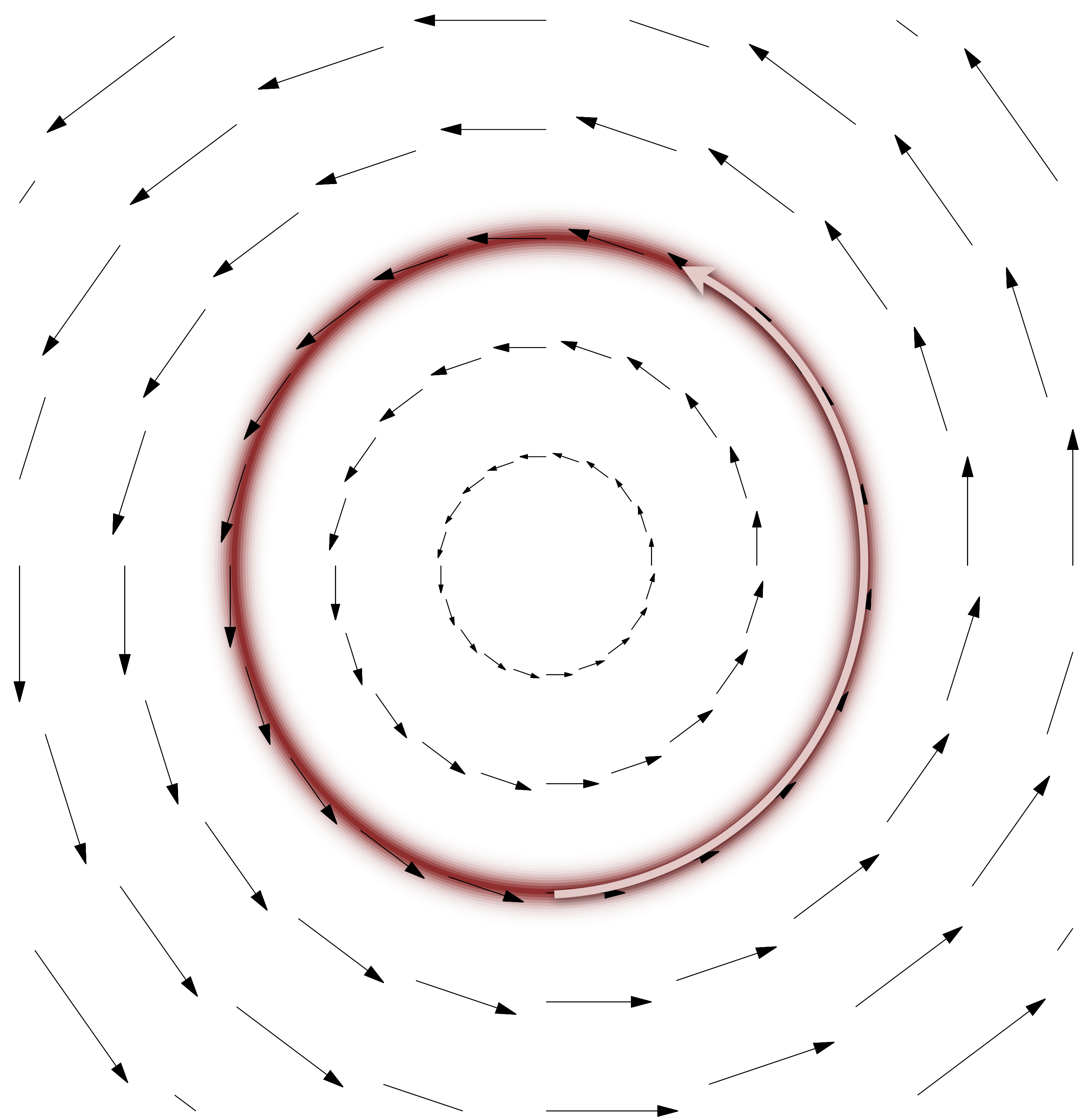


One way to construct the coherent exploration is to integrate along a *vector field* aligned with the typical set.

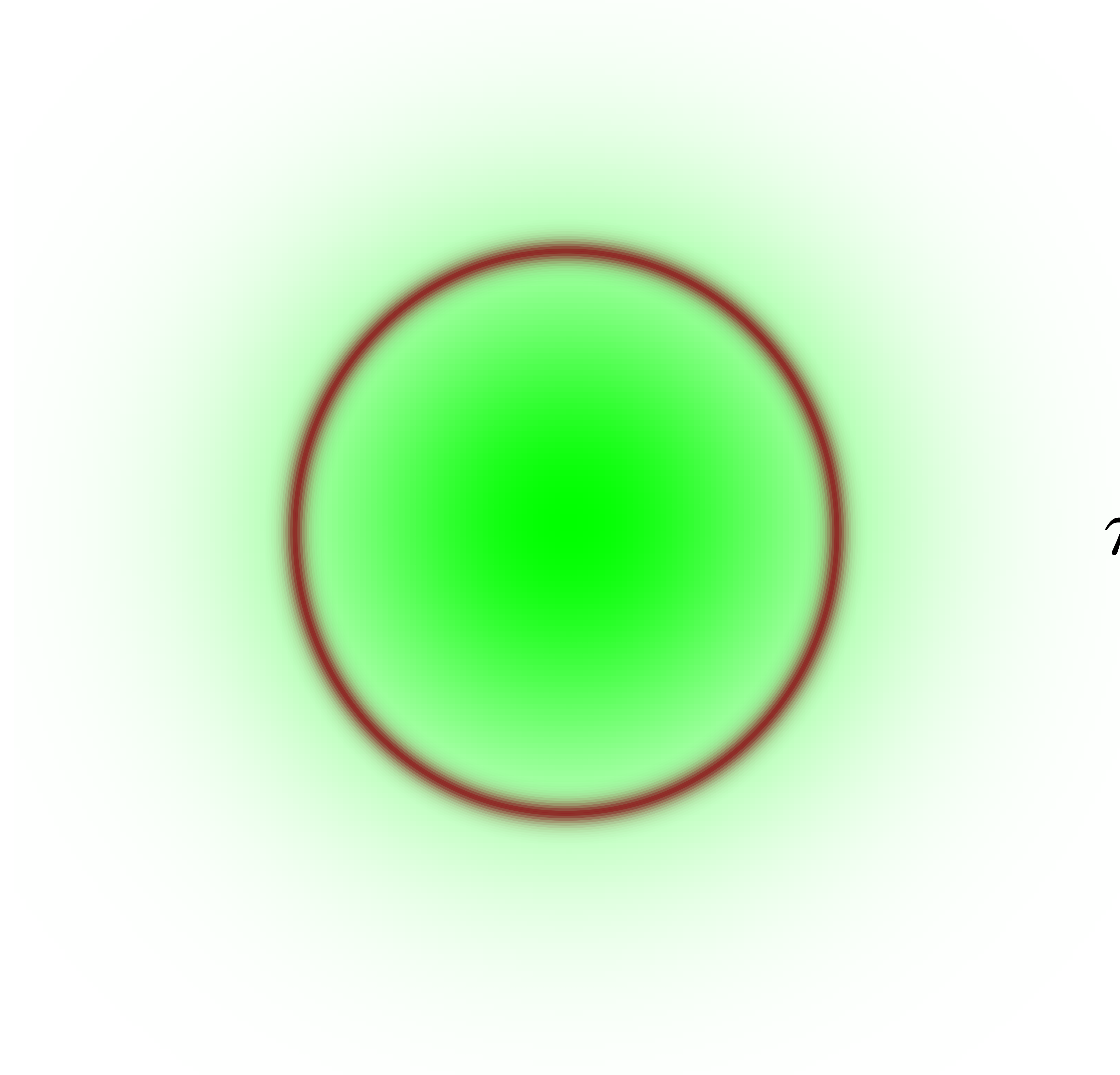
One way to construct the coherent exploration is to integrate along a *vector field* aligned with the typical set.



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

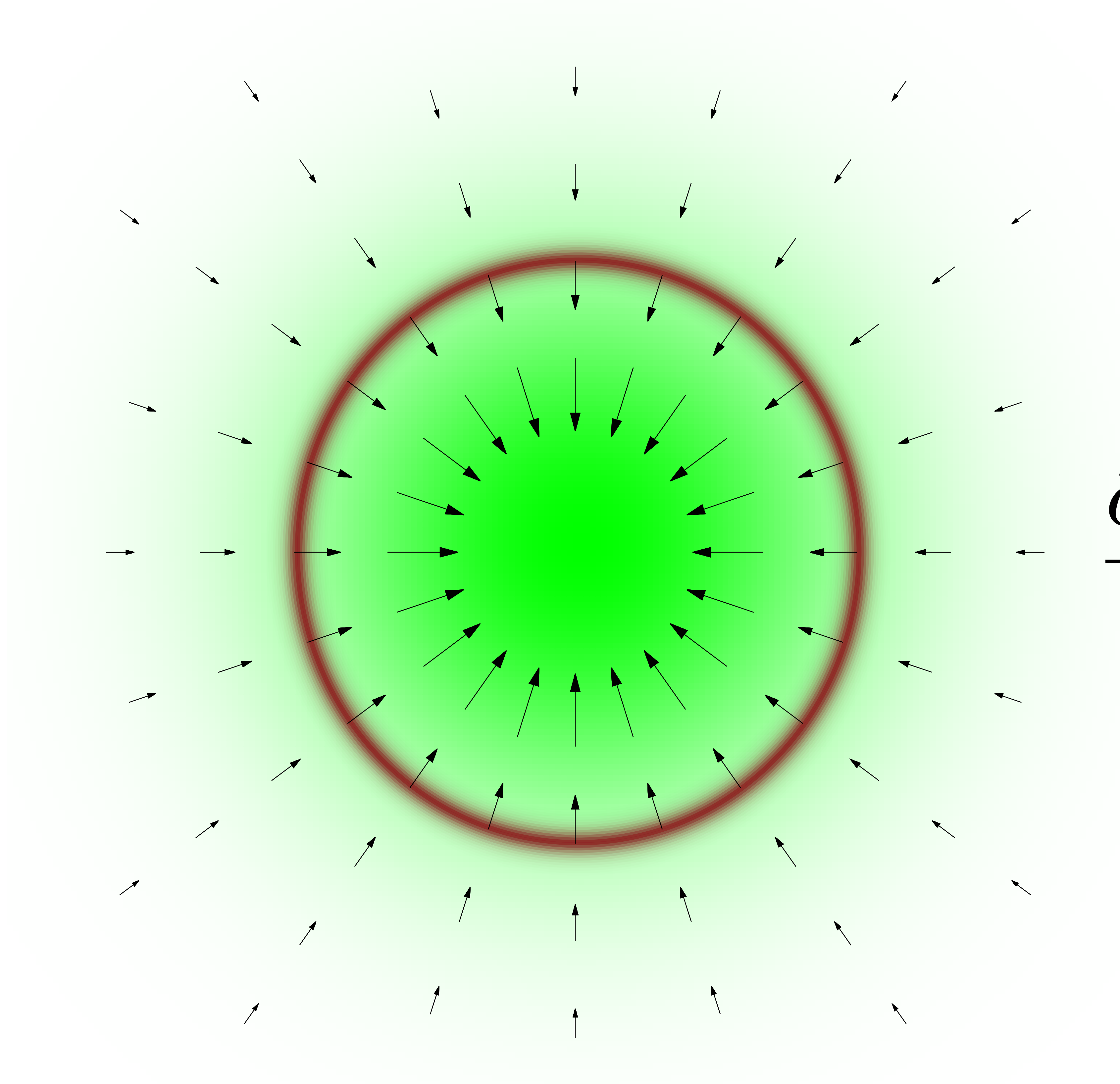


Creating such a vector field requires transforming available vector fields, such as the gradient.



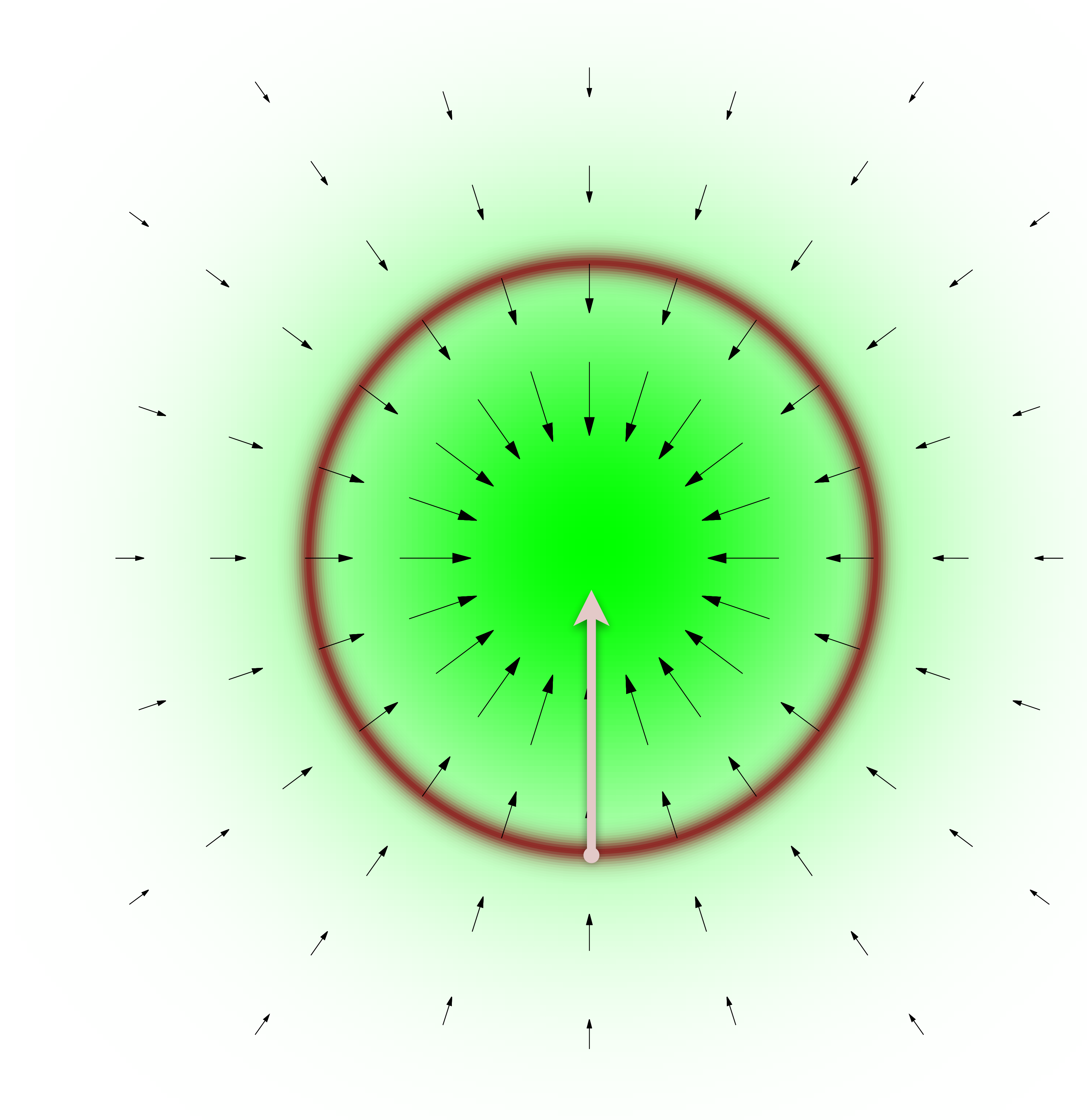
$$\pi(\theta \mid \mathcal{D})$$

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



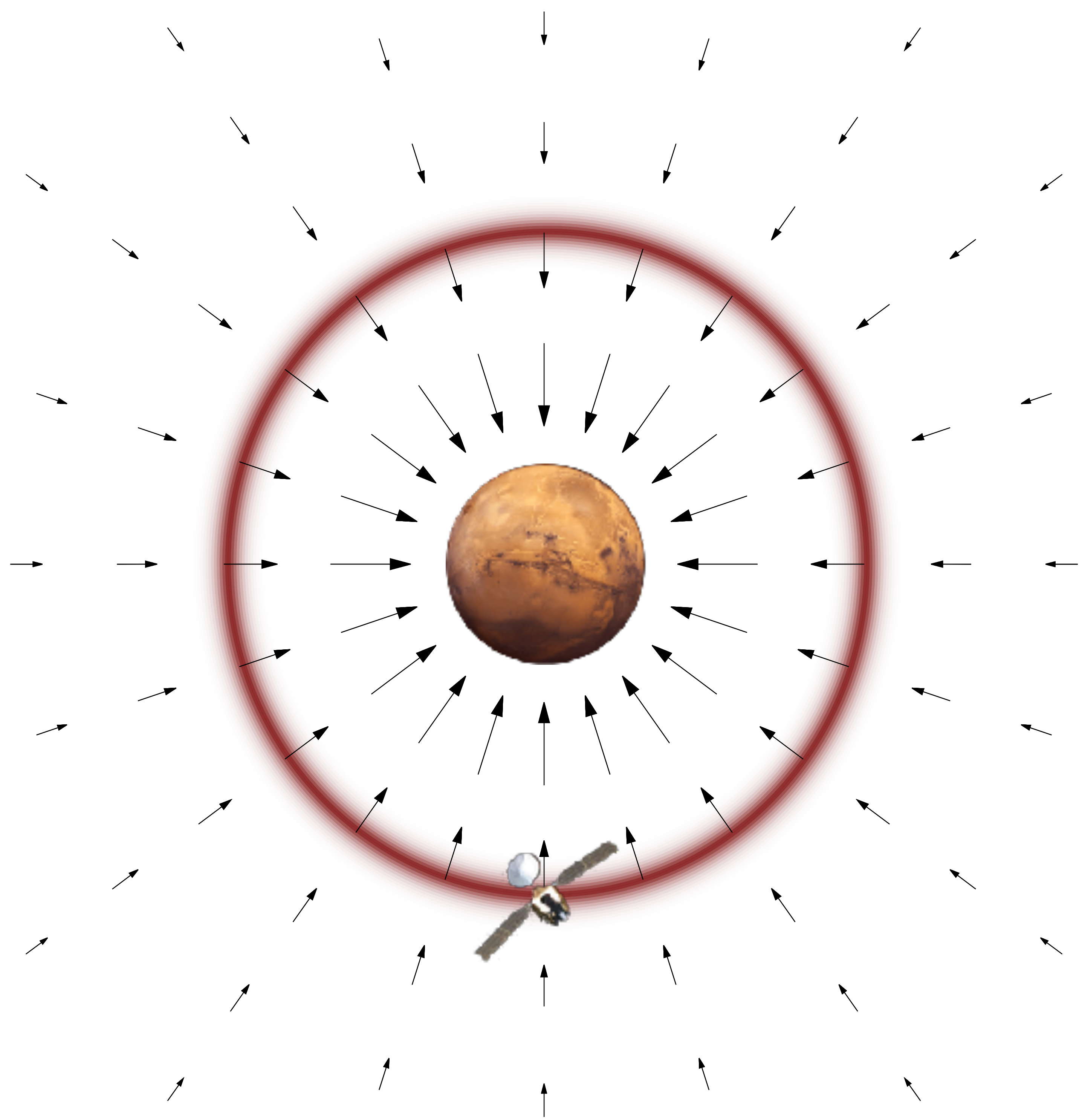
$$\frac{\partial \pi(\theta \mid \mathcal{D})}{\partial \theta}$$

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

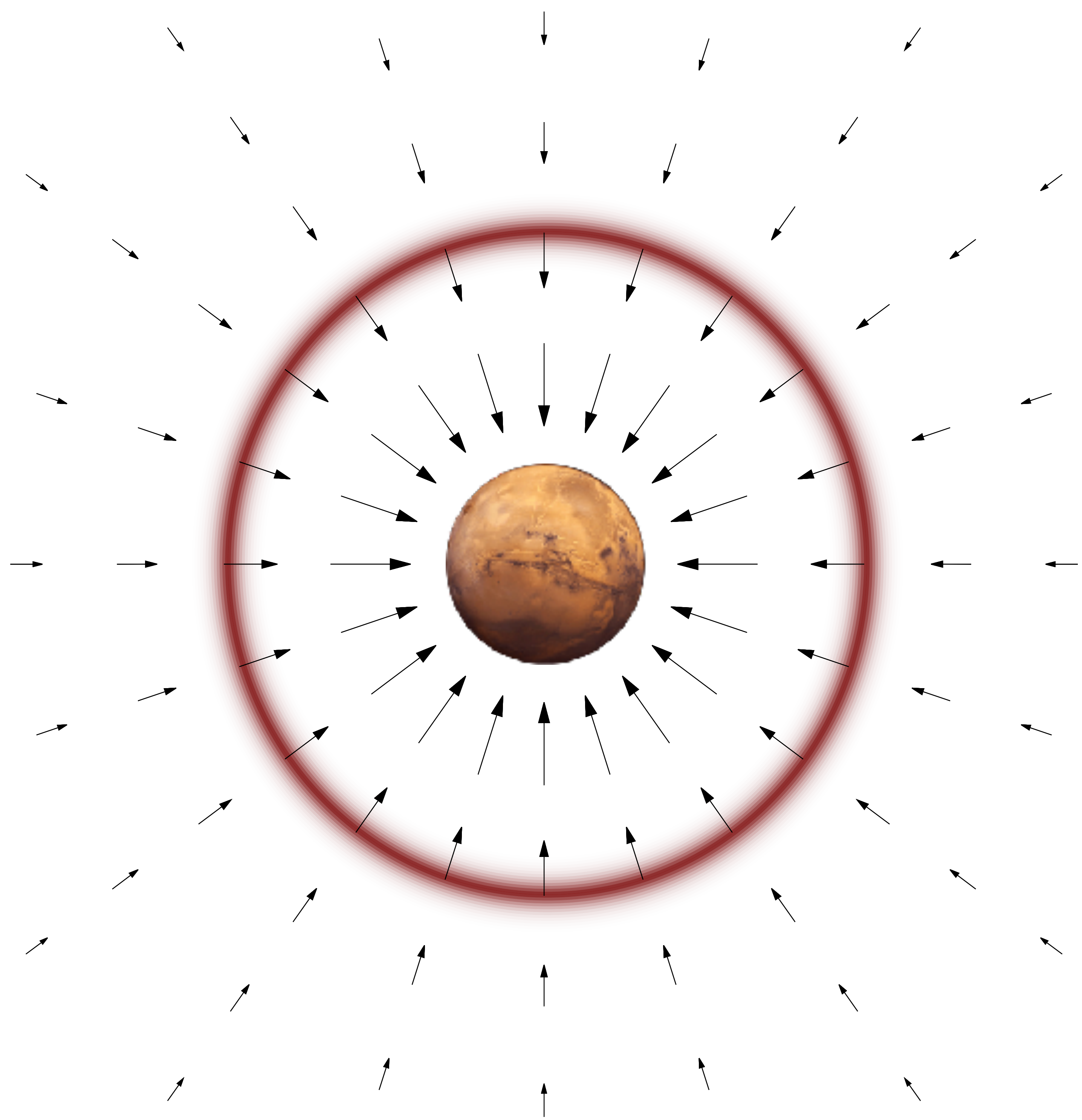


$$\frac{\partial \pi(\theta \mid \mathcal{D})}{\partial \theta}$$

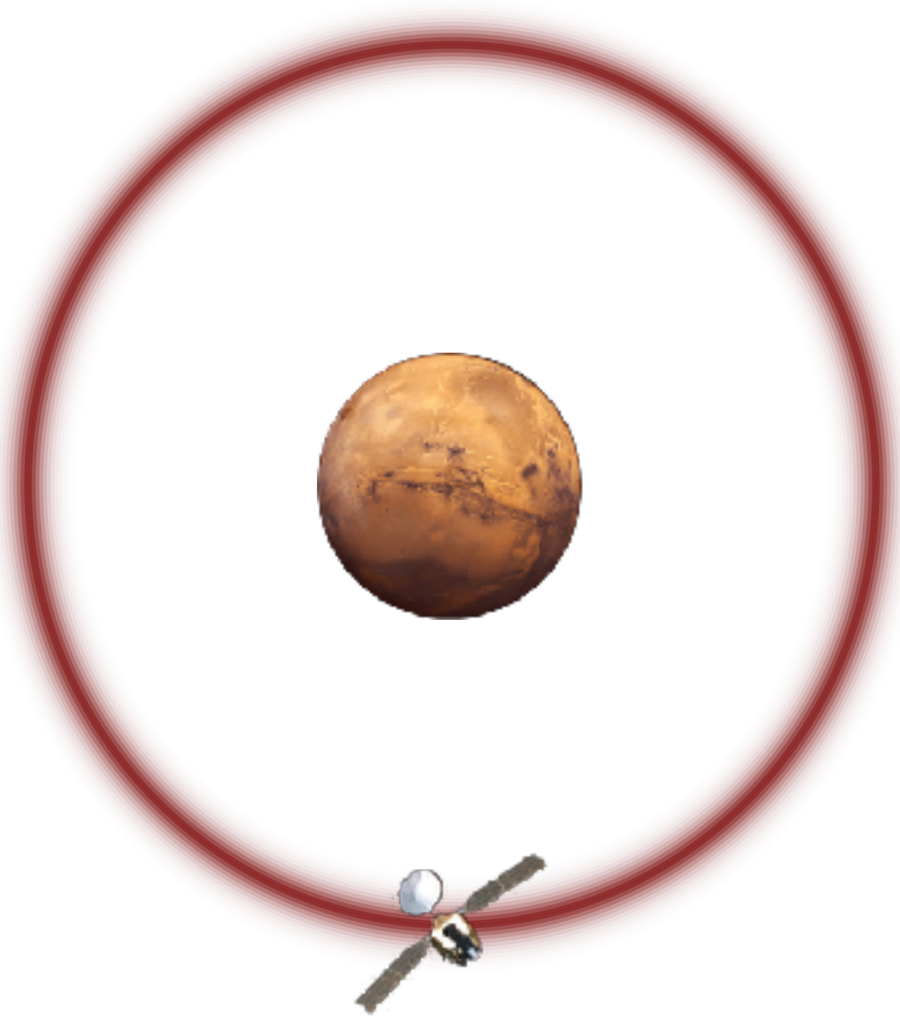
Differential geometry informs this transformation, although a physical analogy can be more intuitive.



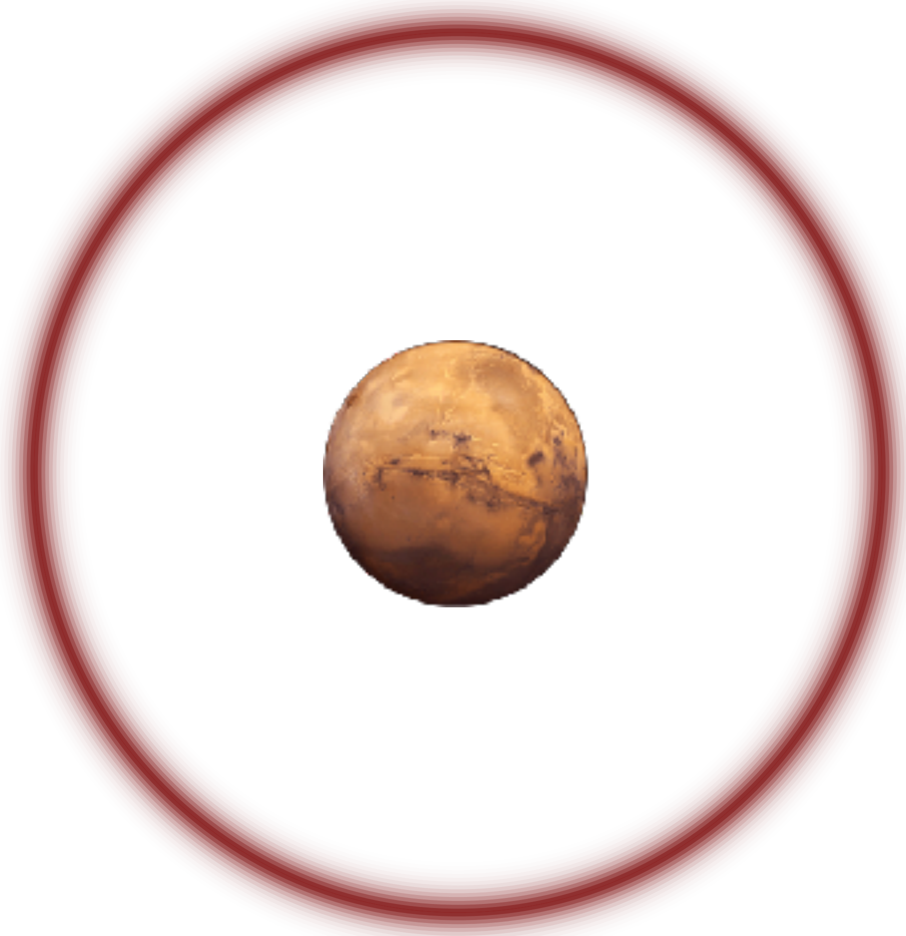
Differential geometry informs this transformation, although a physical analogy can be more intuitive.



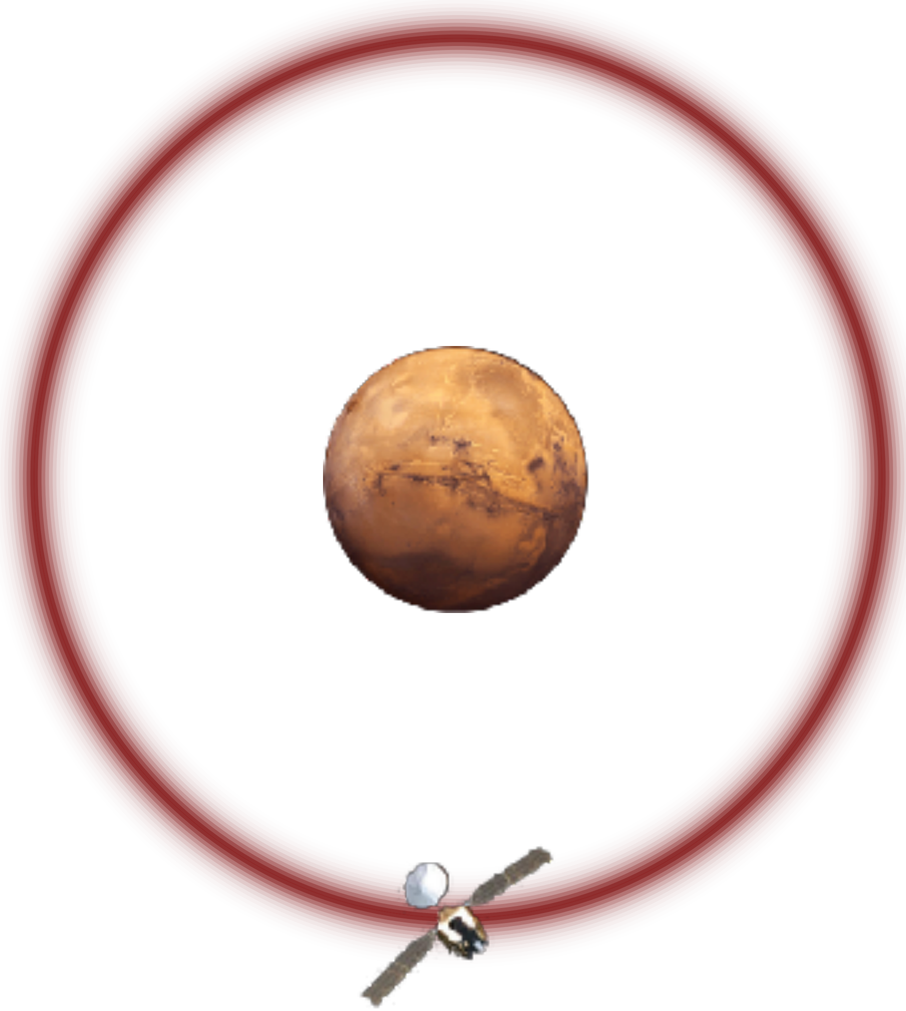
We need to add *momentum* in just the right way.
Too little and we still crash into the planet.



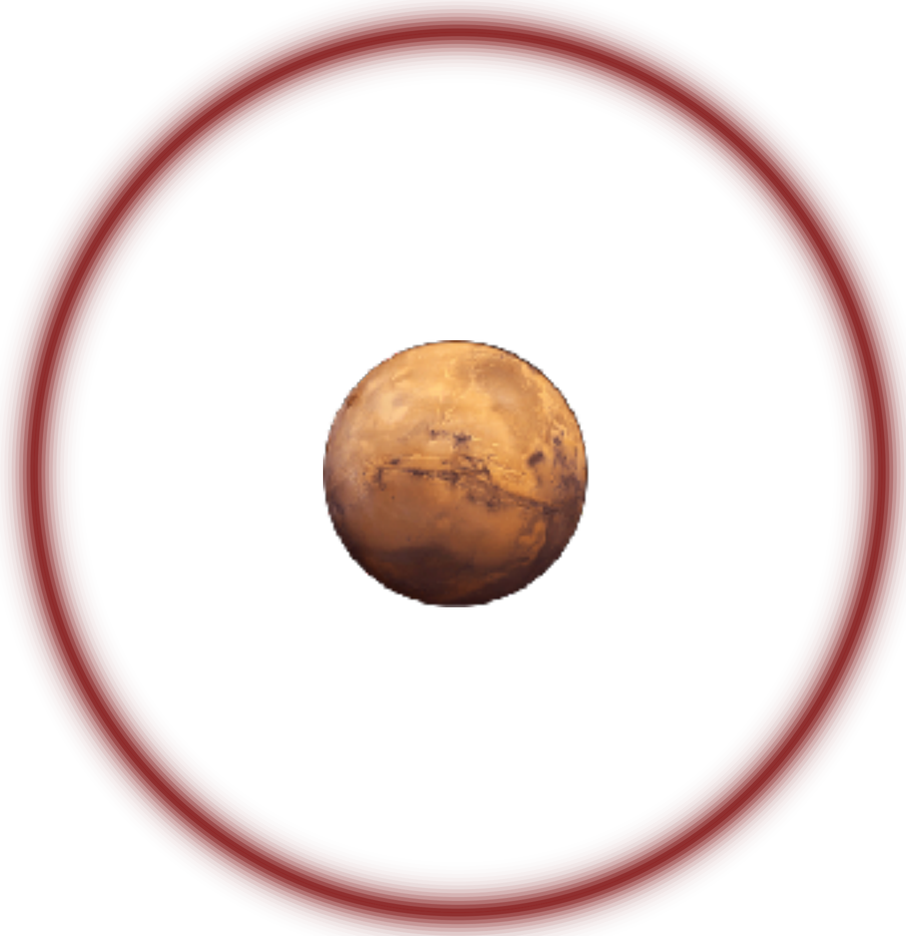
We need to add *momentum* in just the right way.
Too little and we still crash into the planet.



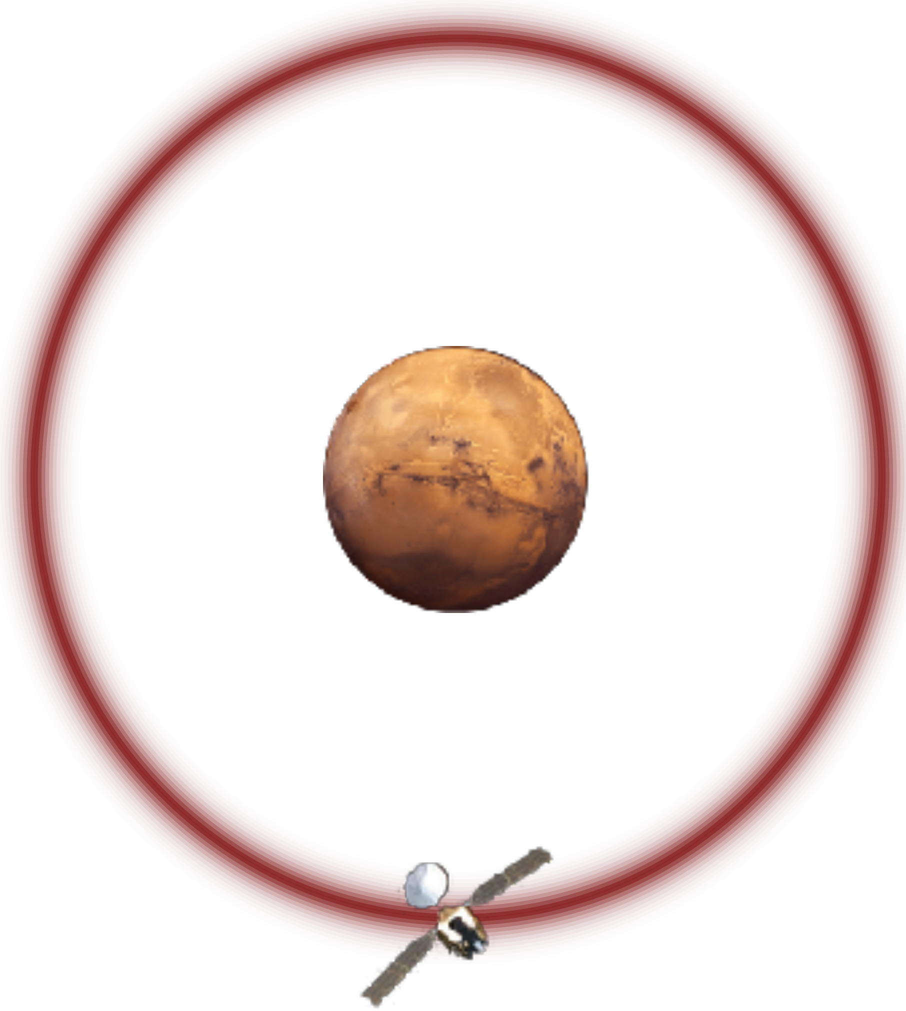
Too much and we fly off to infinity.



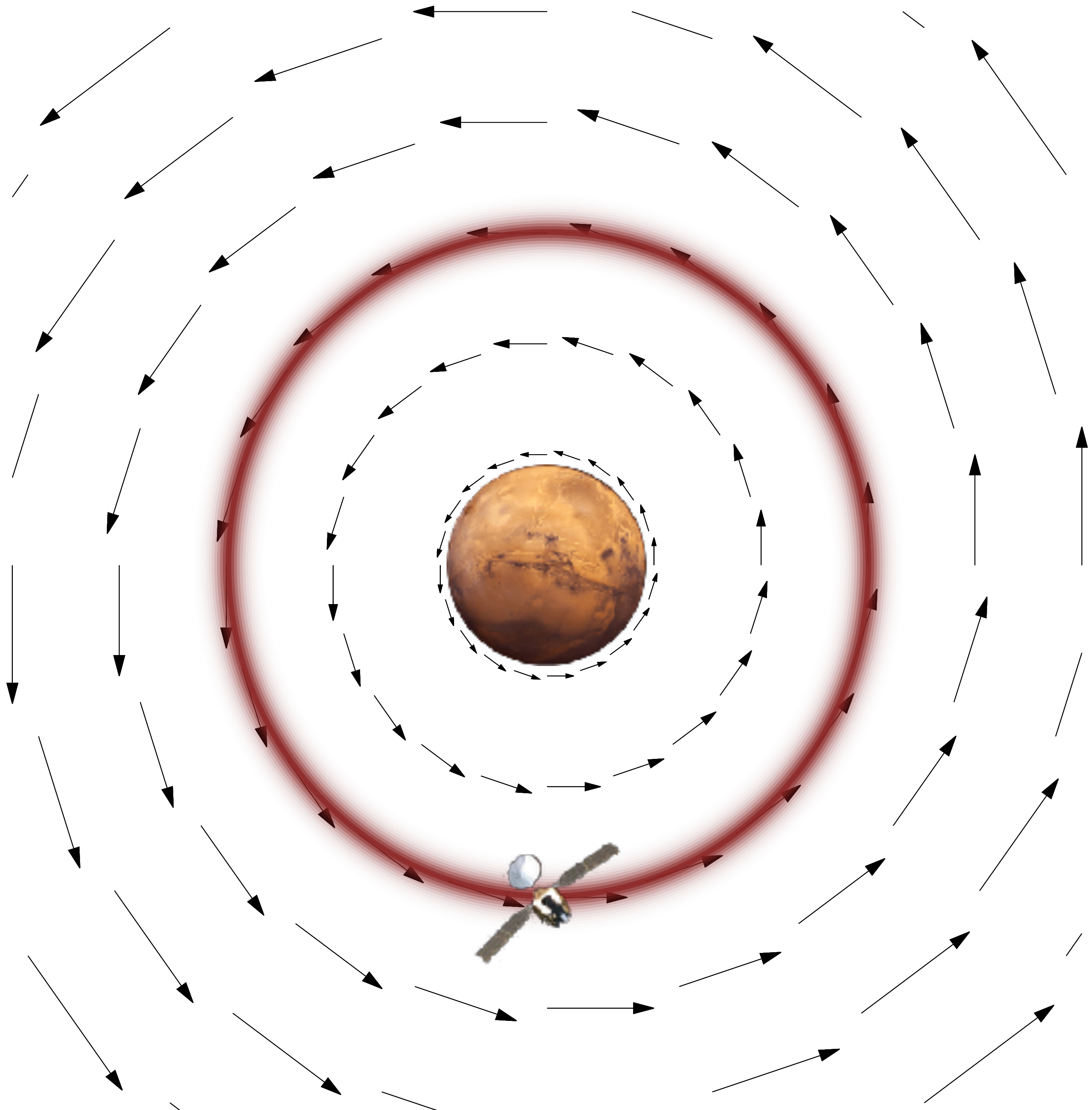
Too much and we fly off to infinity.



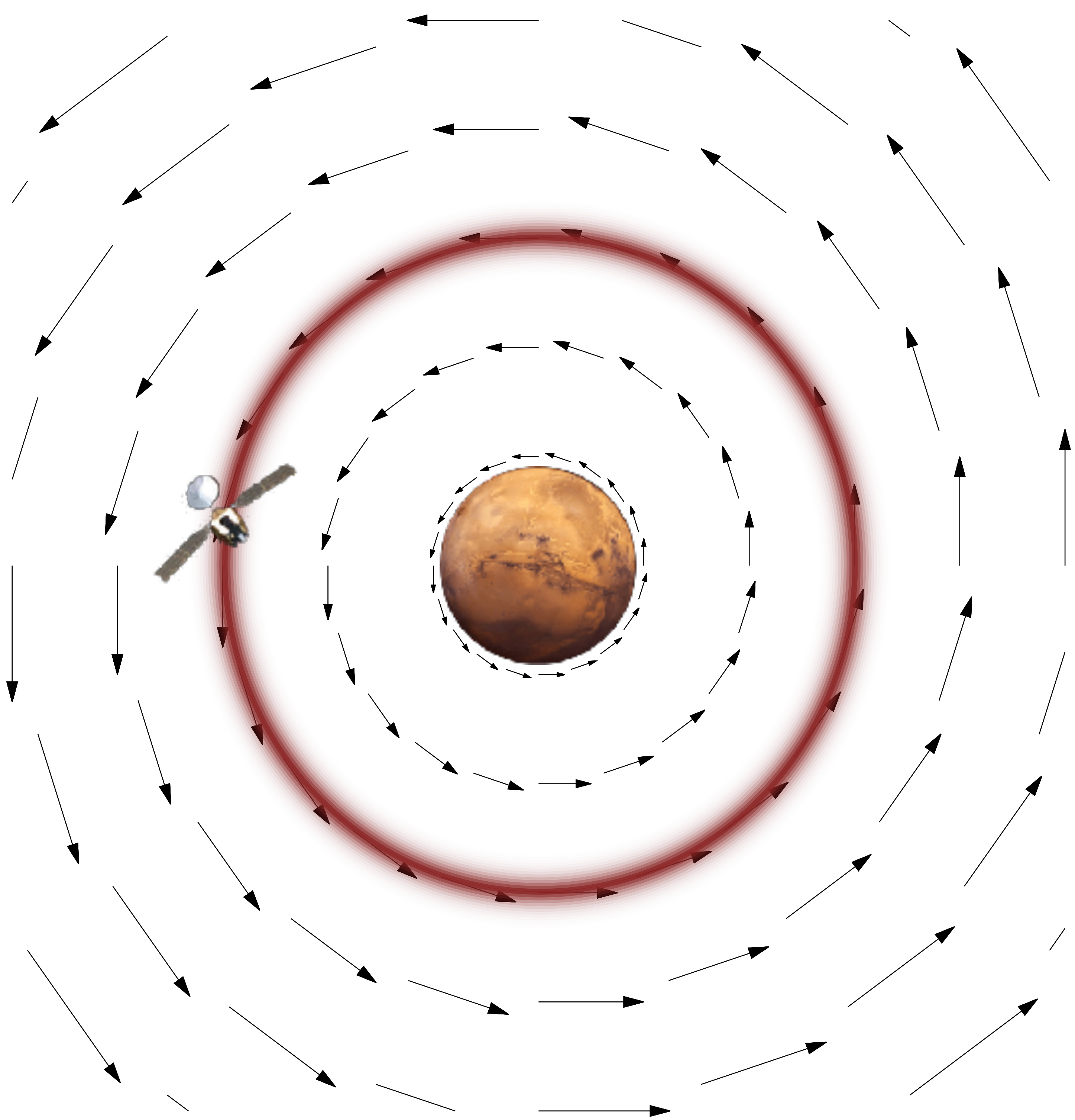
Just enough, however, aligns the gradients with the typical set and yields the desired orbital trajectory.



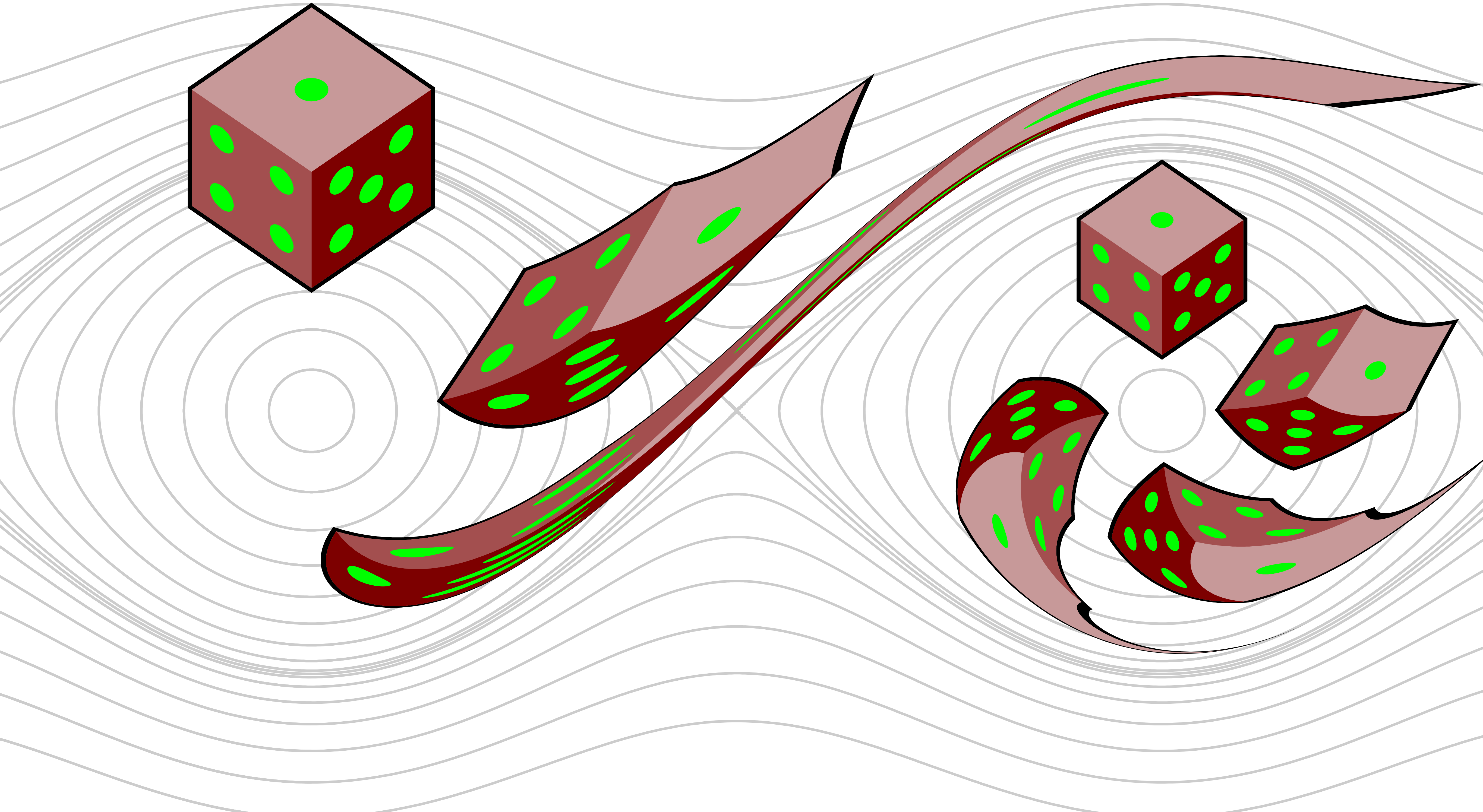
Just enough, however, aligns the gradients with the typical set and yields the desired orbital trajectory.



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



Stan is a high-performance C++ library for building models and fitting them with Hamiltonian Monte Carlo.



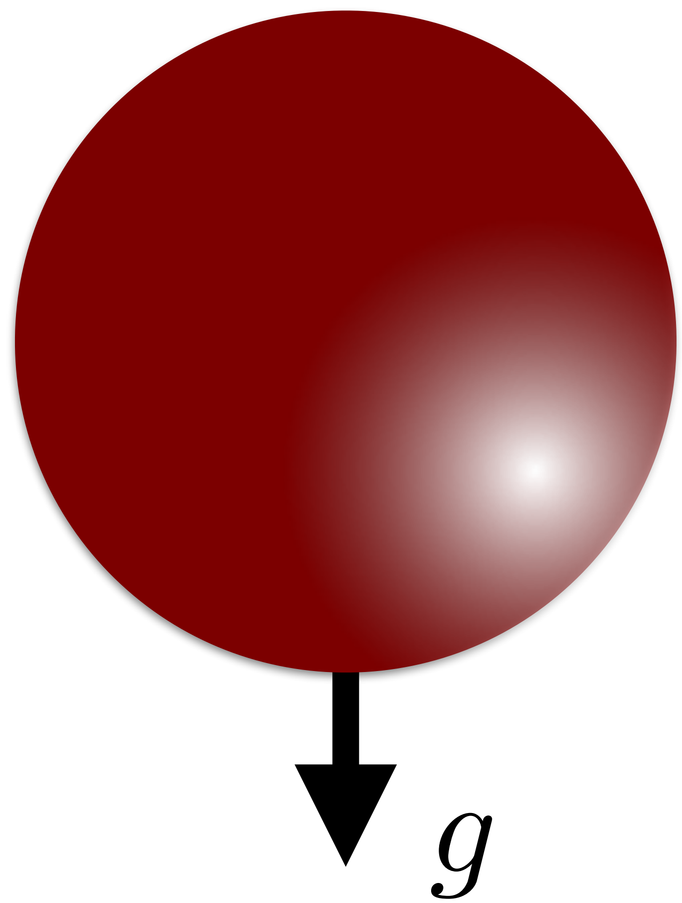
Conflicts of Interest

Institute of Prospective Technologies,

European Commission Joint Research Center

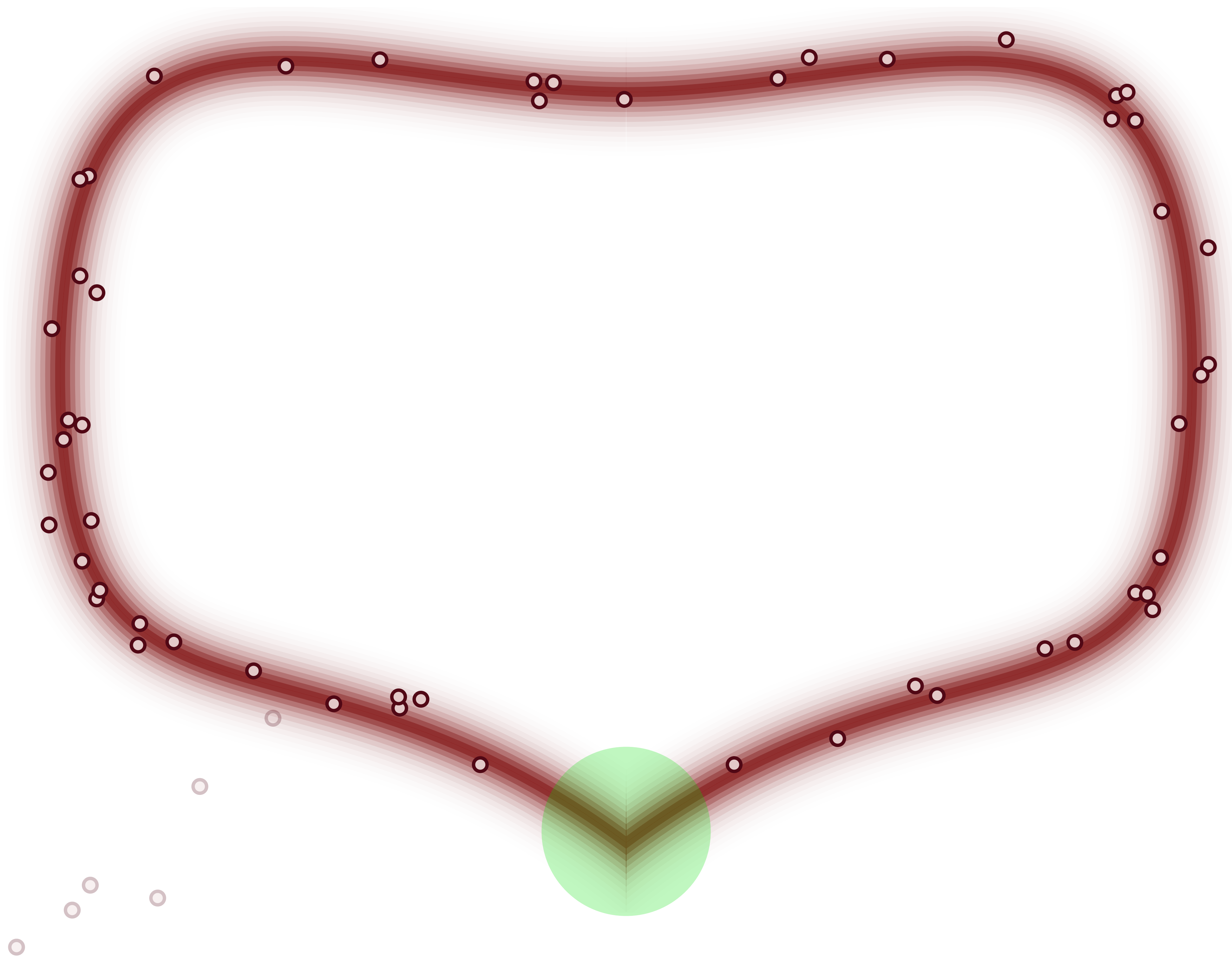
Stan Group

ADVERTISEMENT

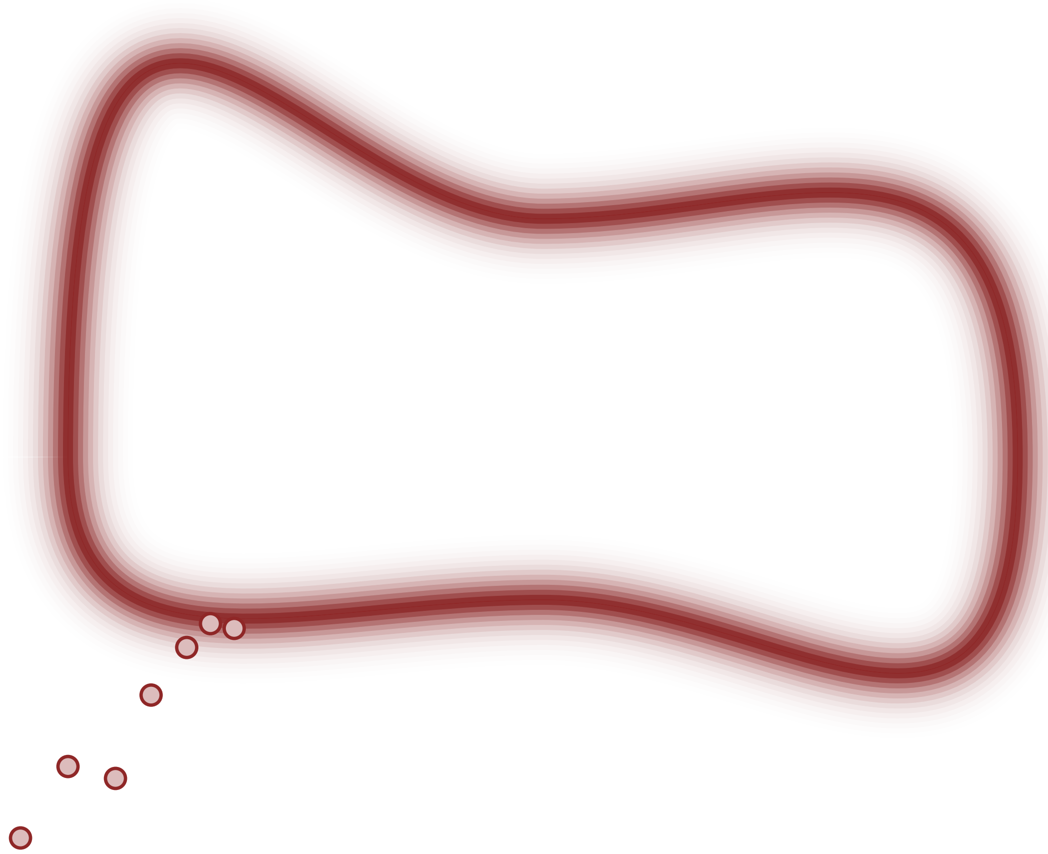


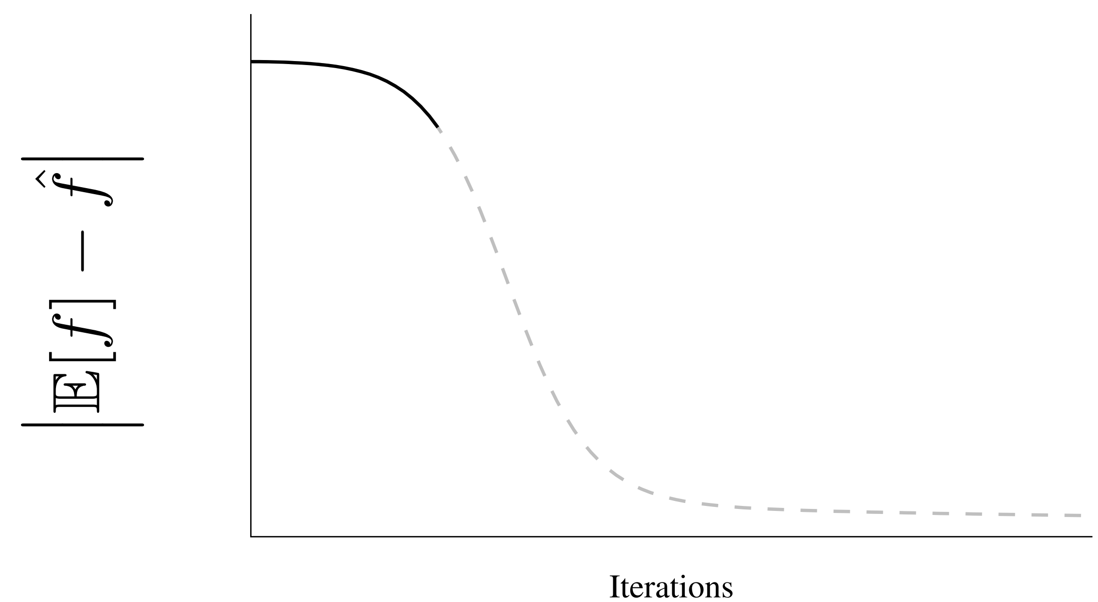
Stan Physics Workshop in late summer somewhere in the NY/NE area -- think undergrad labs but with full Bayesian analyses developed and fit in Stan. If interested see me!

Formalizing Fast Enough

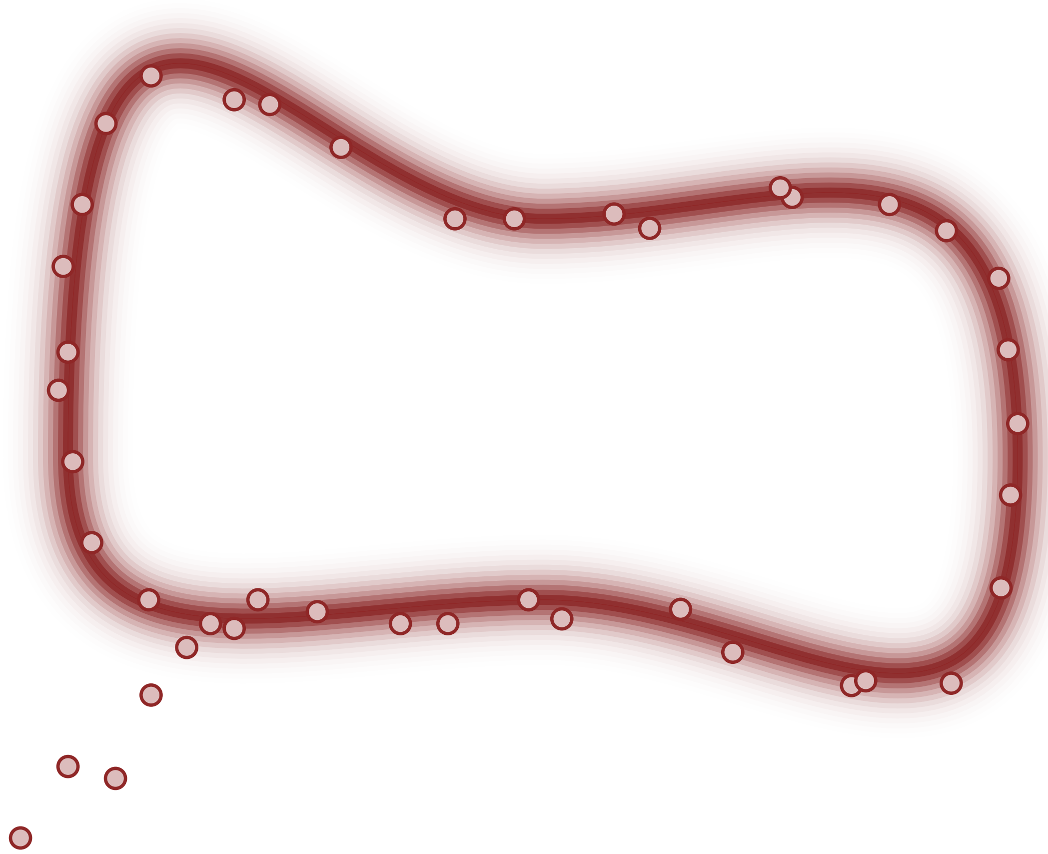


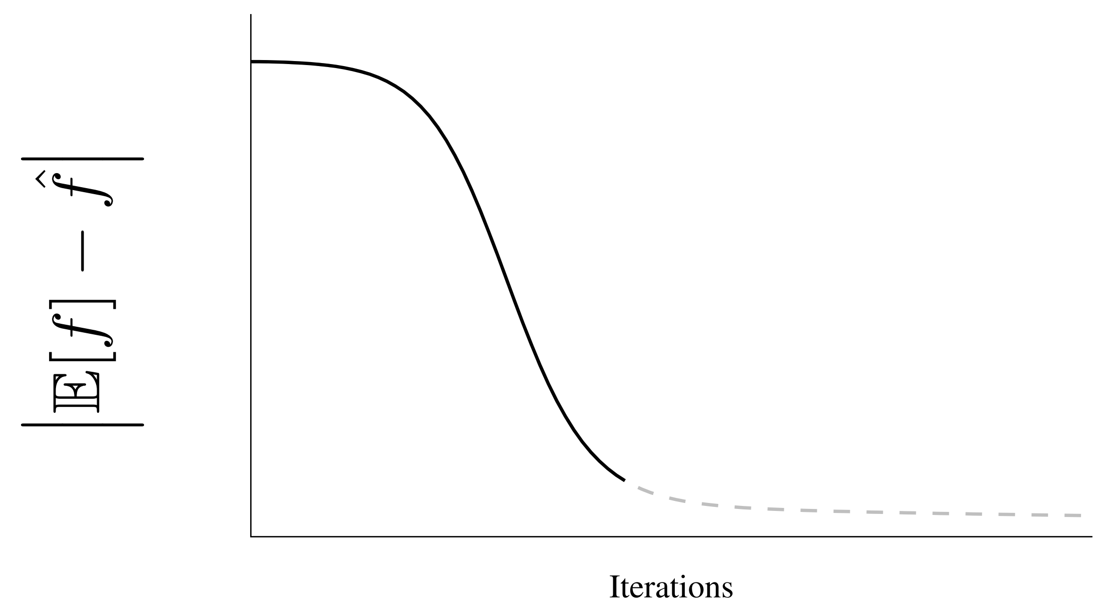
Under ideal conditions, MCMC estimators converge to the true expectations in a very practical progression.



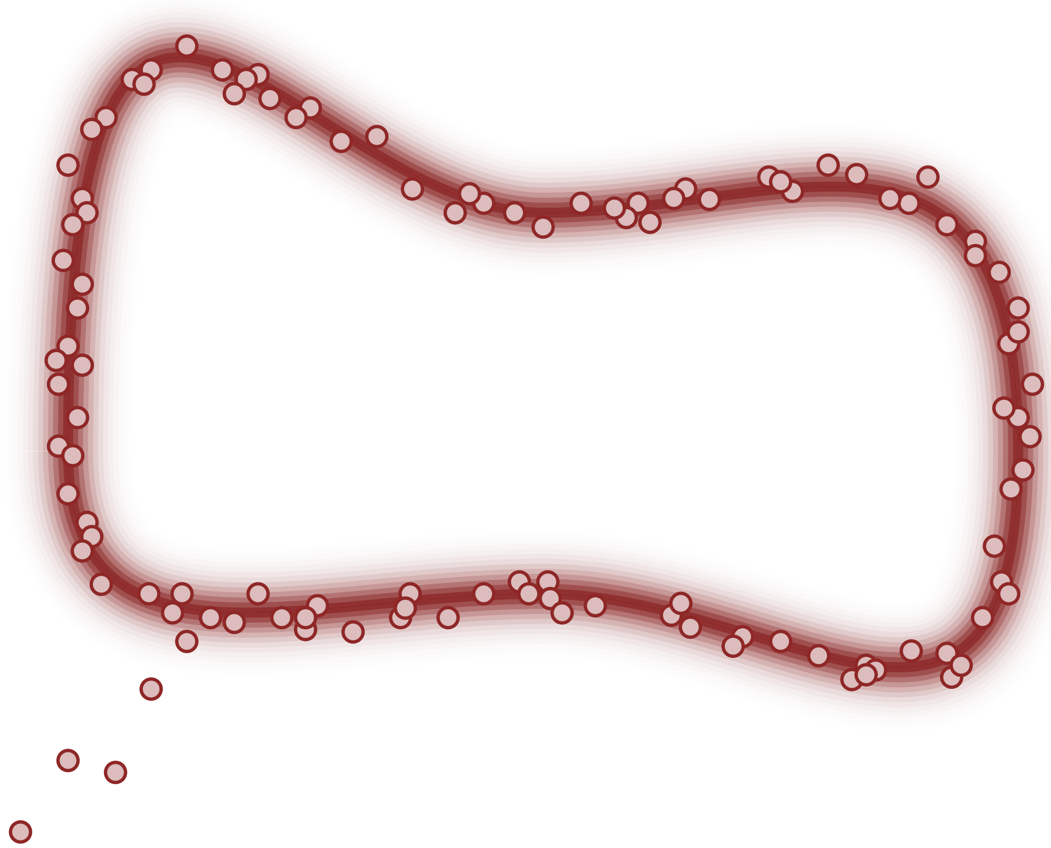


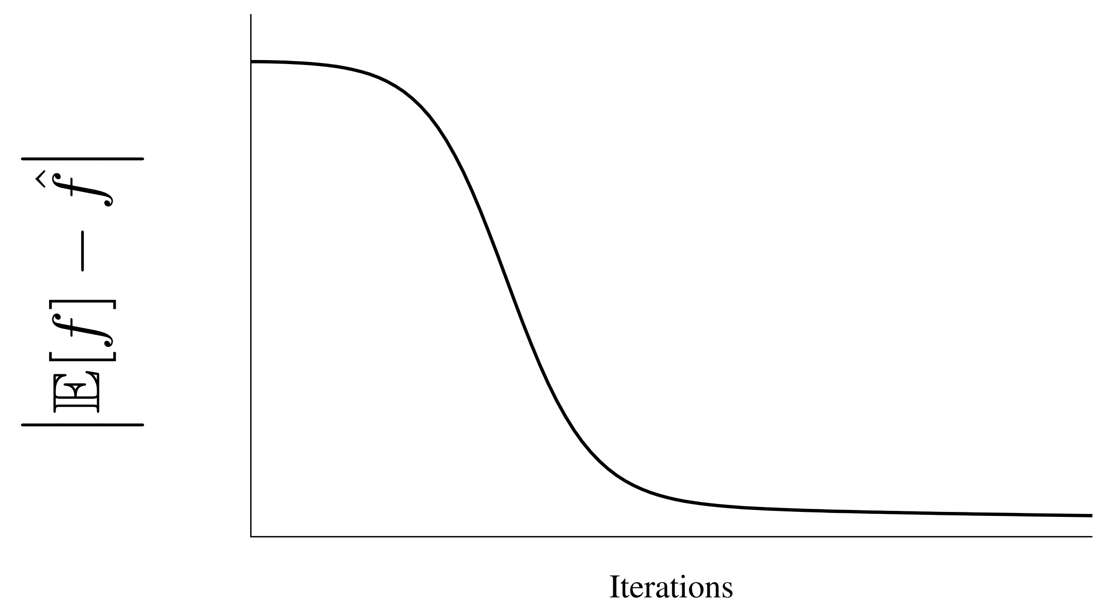
Under ideal conditions, MCMC estimators converge to the true expectations in a very practical progression.





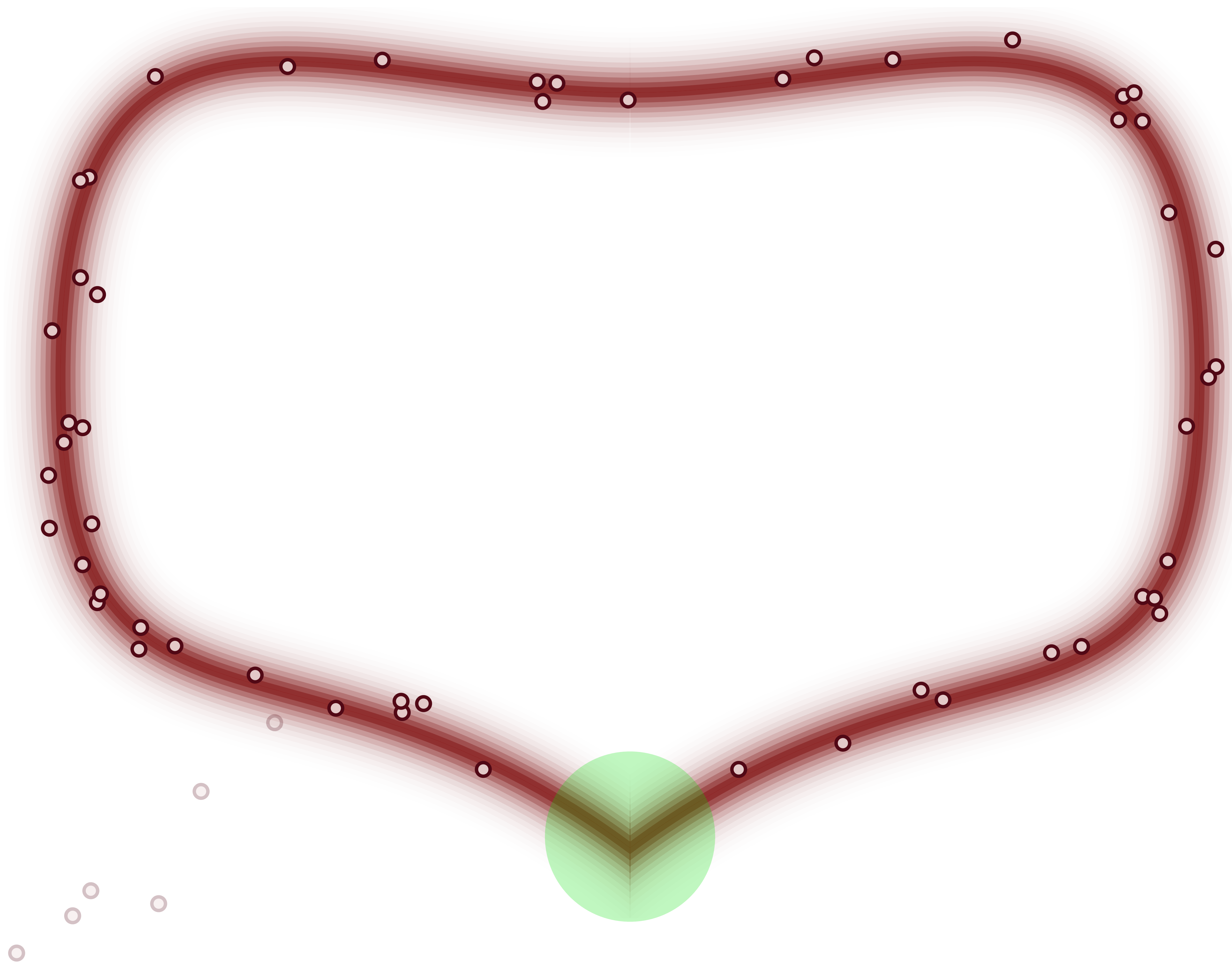
Under ideal conditions, MCMC estimators converge to the true expectations in a very practical progression.



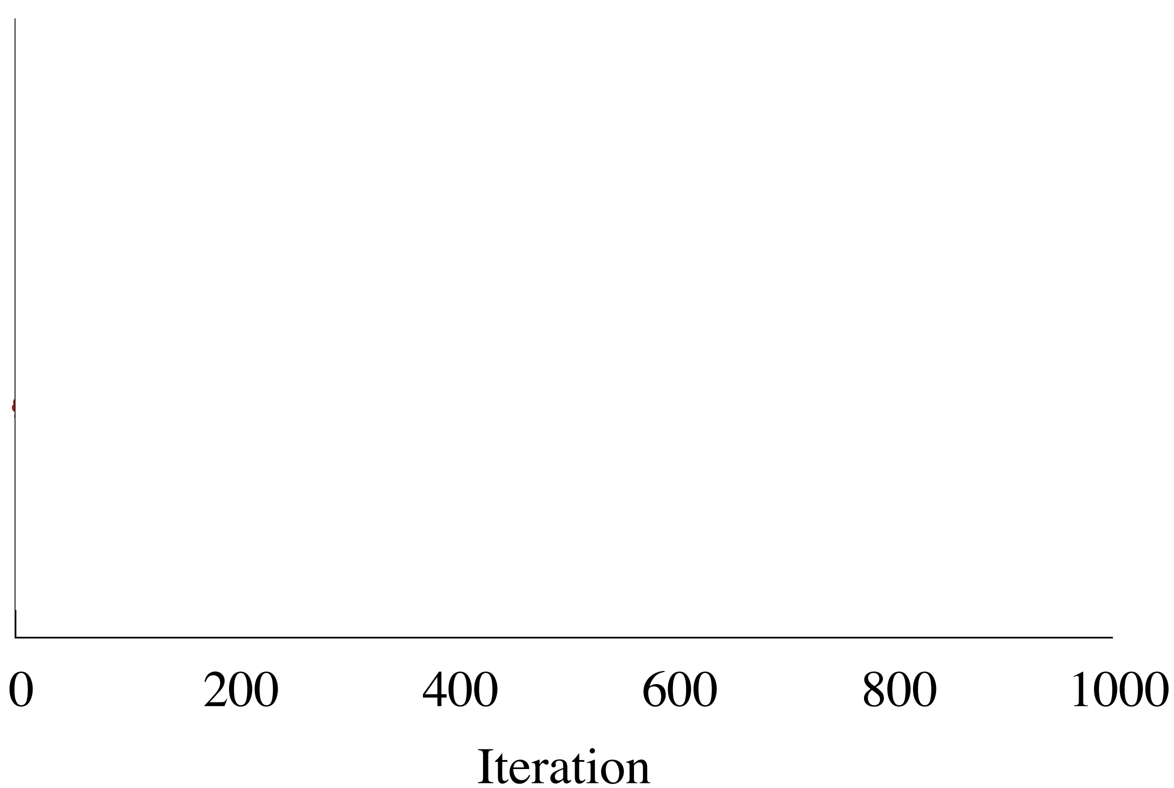


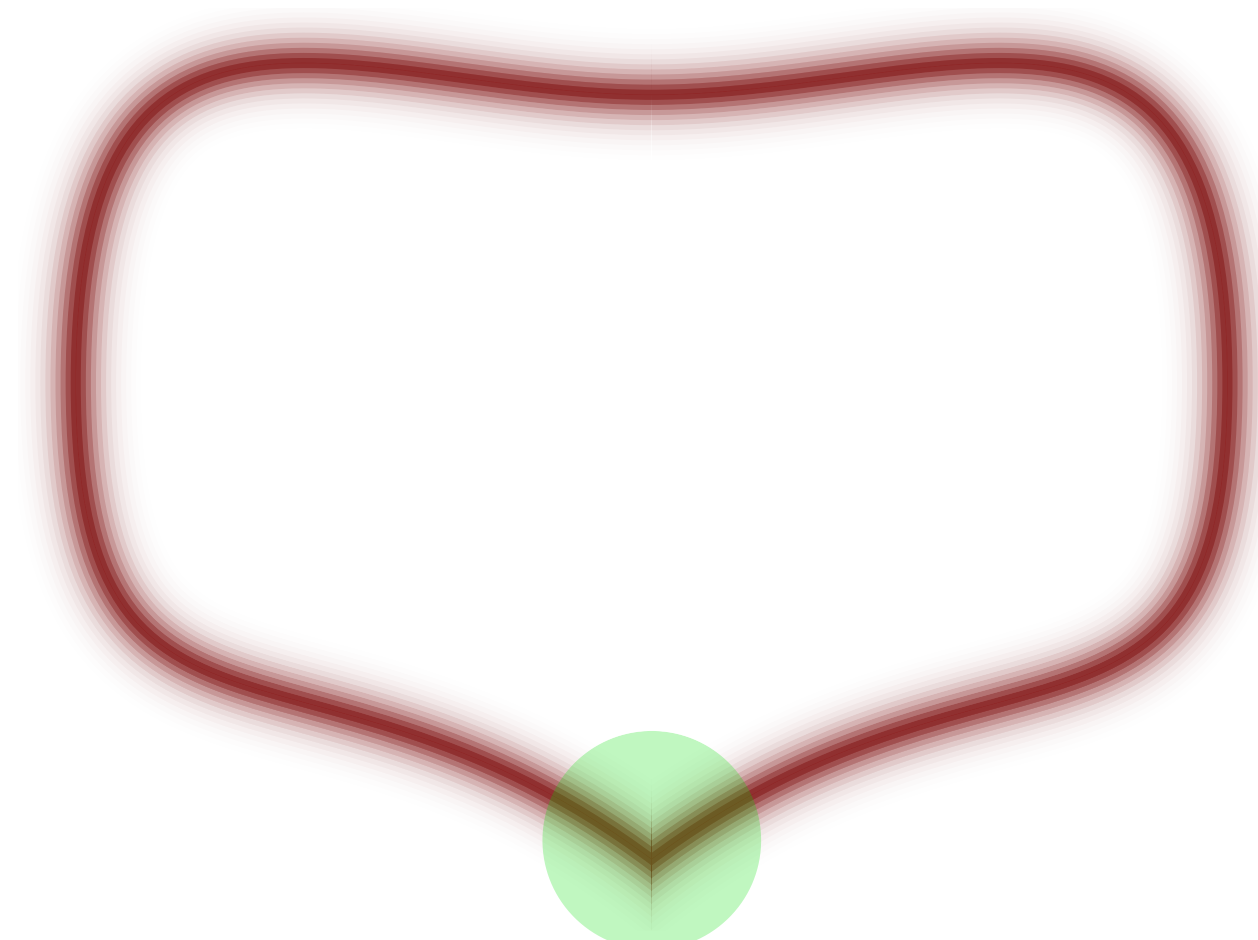
There are many pathological posterior geometries, however, that spoil these ideal conditions.

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

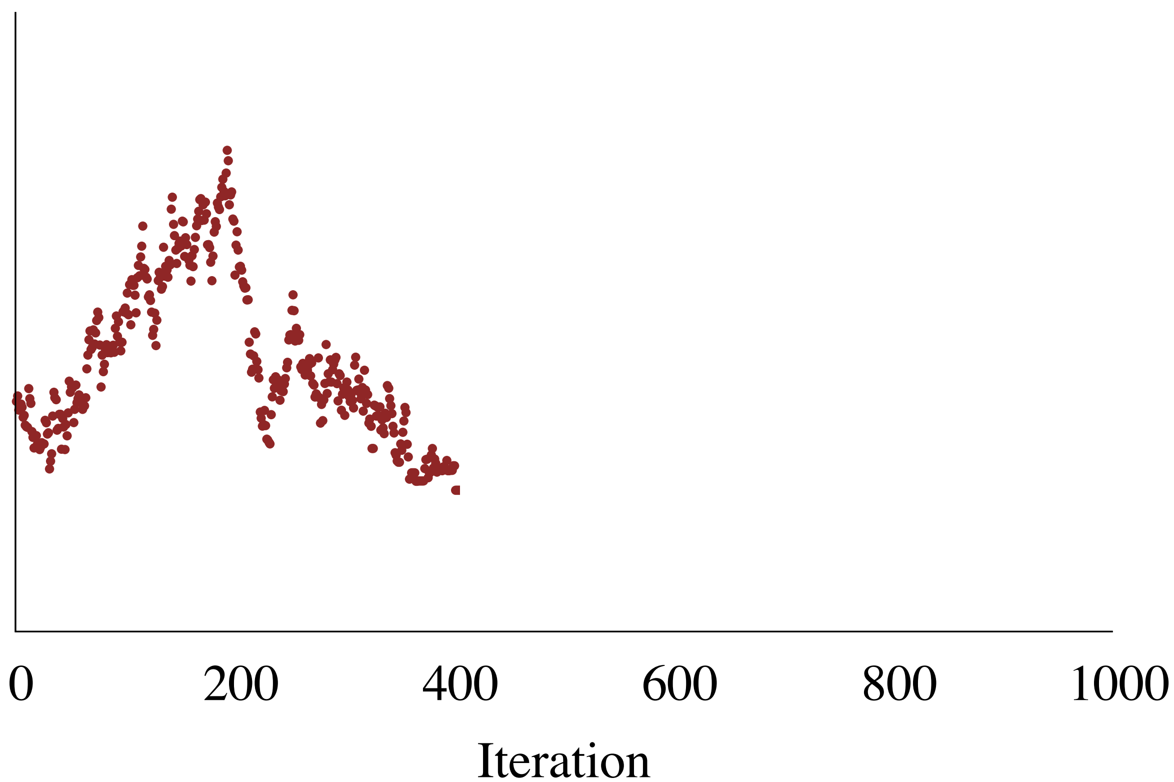


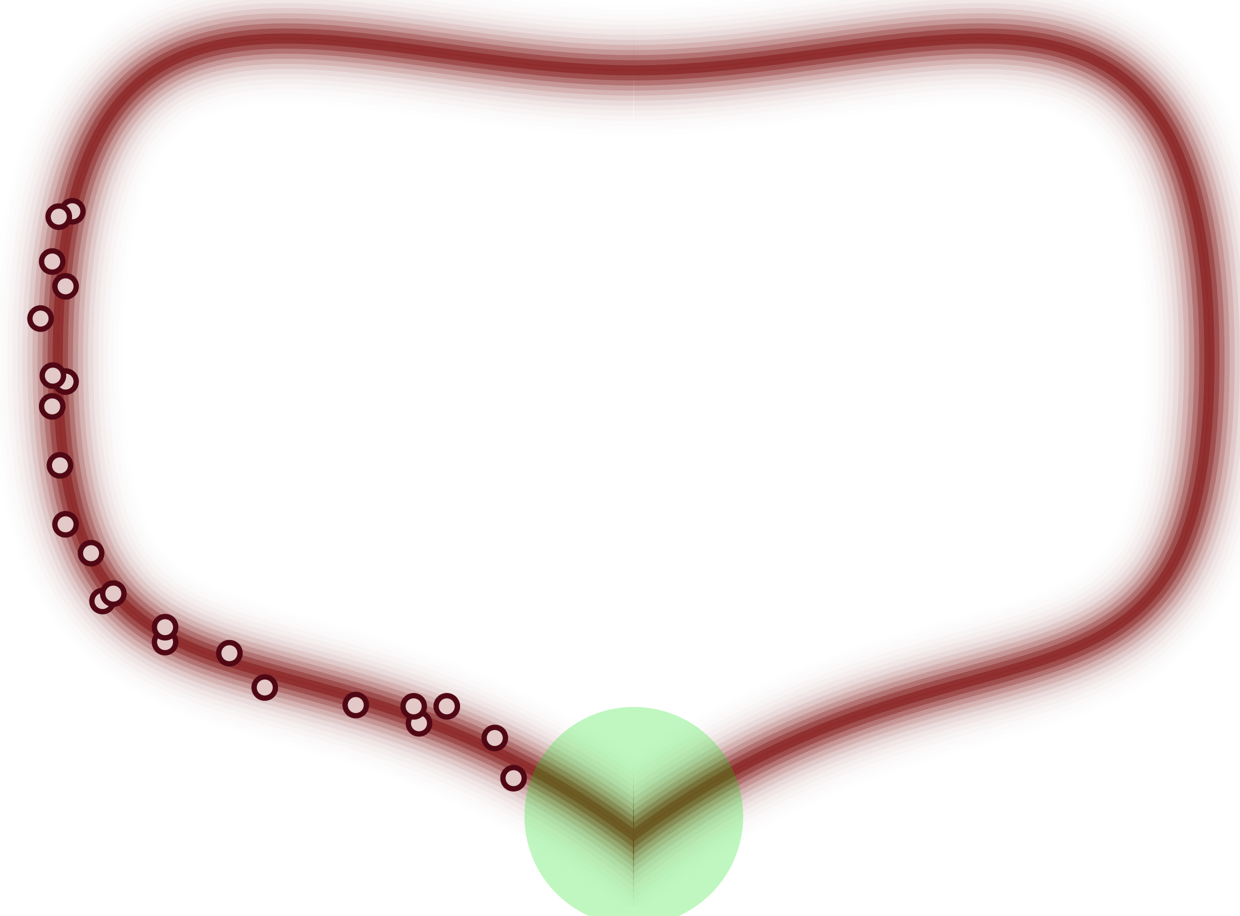
There are many pathological posterior geometries, however, that spoil these ideal conditions.



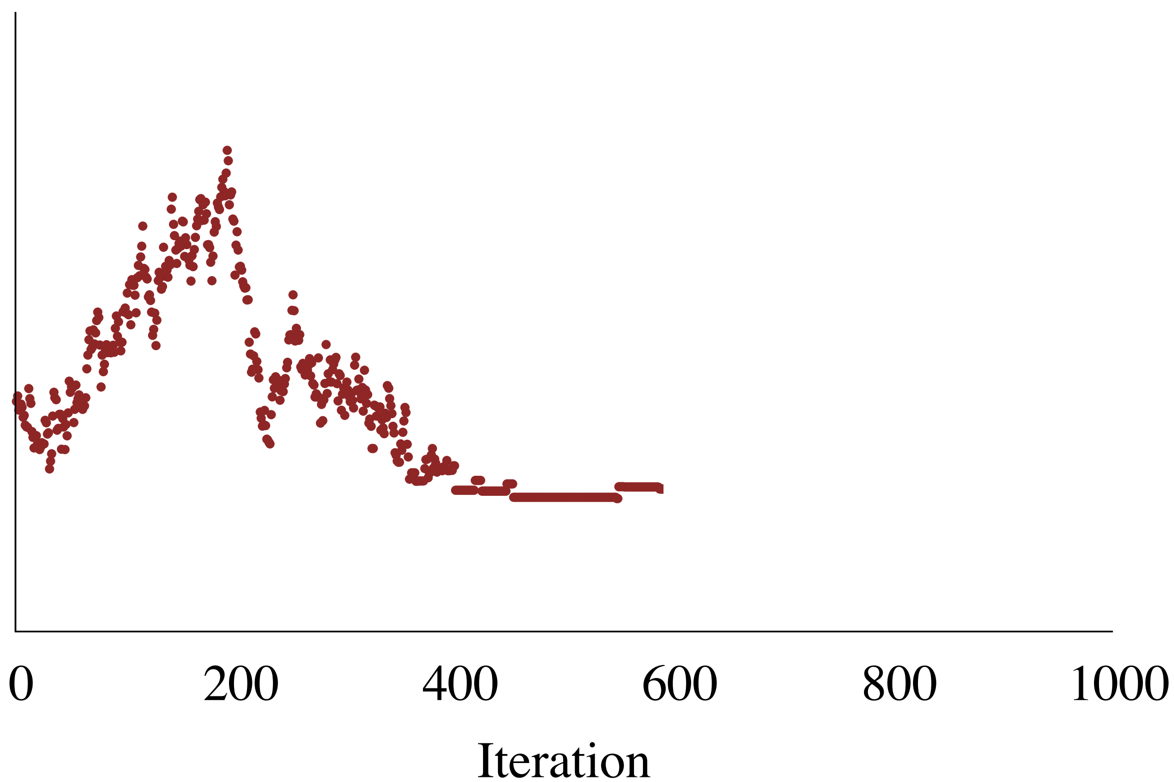


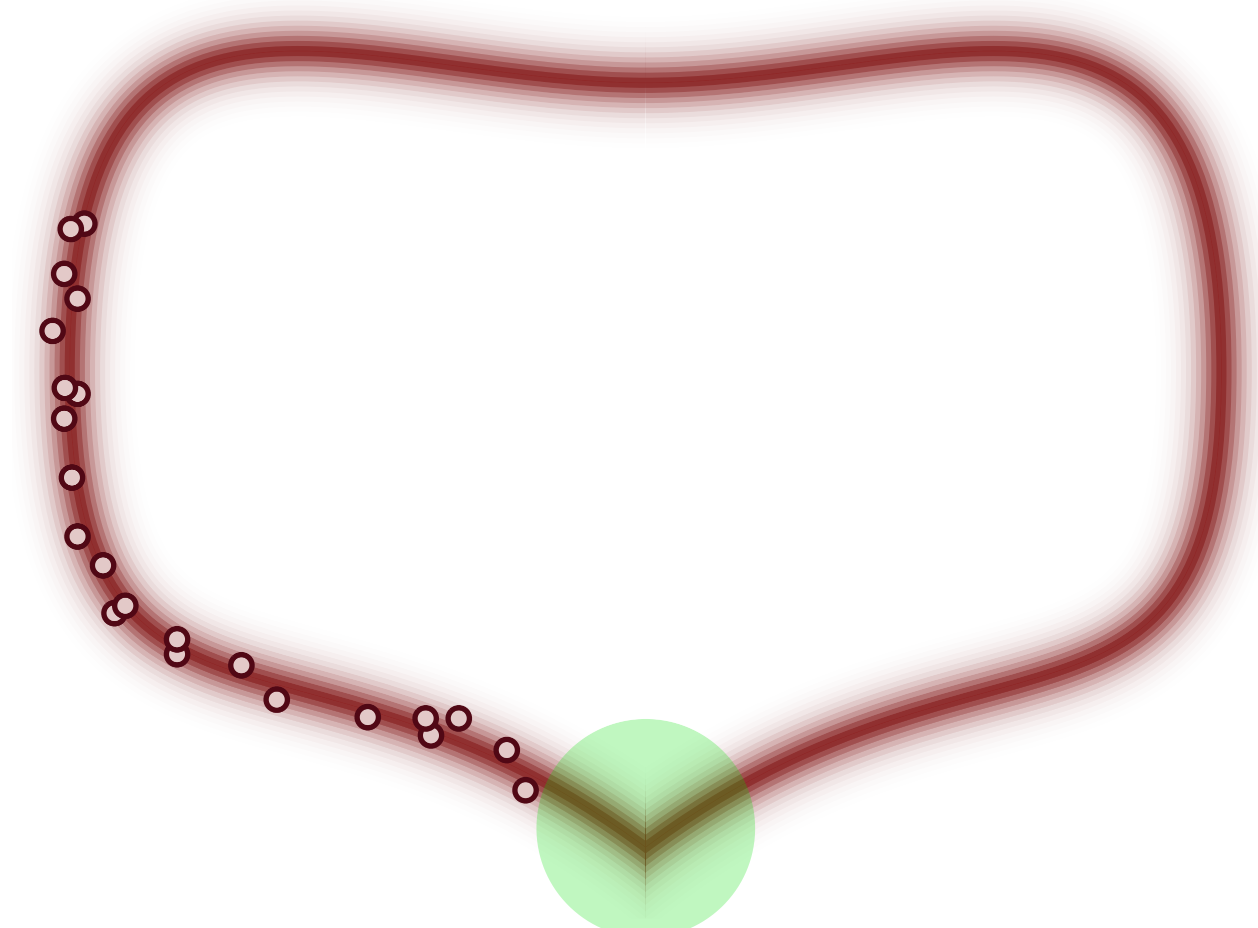
There are many pathological posterior geometries, however, that spoil these ideal conditions.



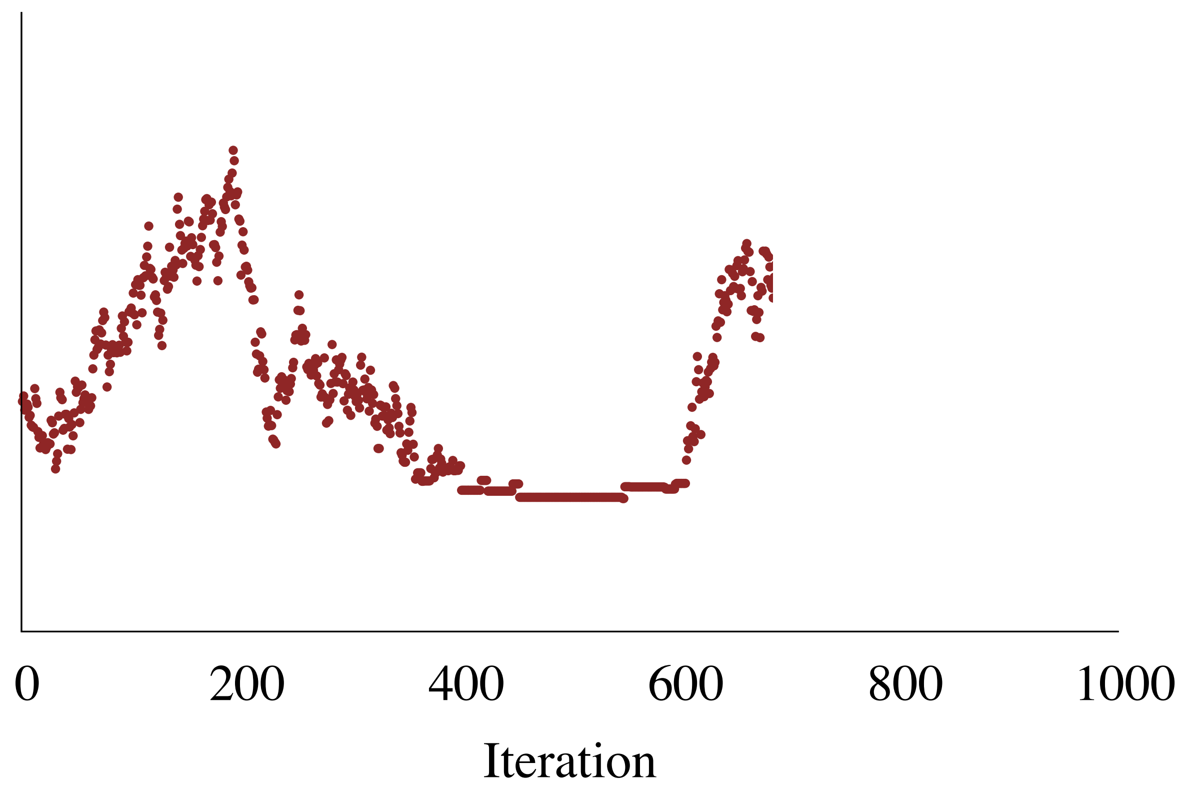


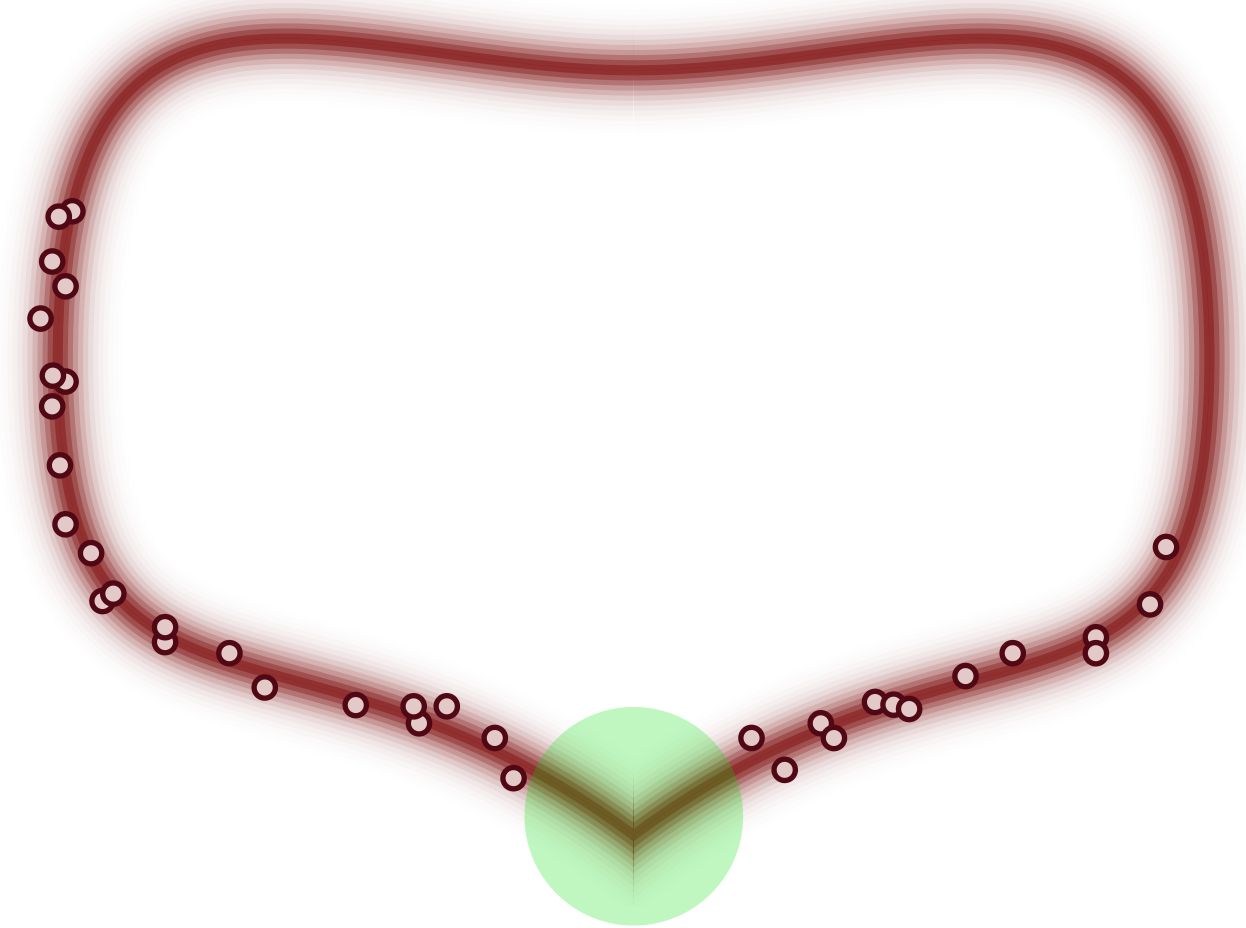
There are many pathological posterior geometries, however, that spoil these ideal conditions.





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

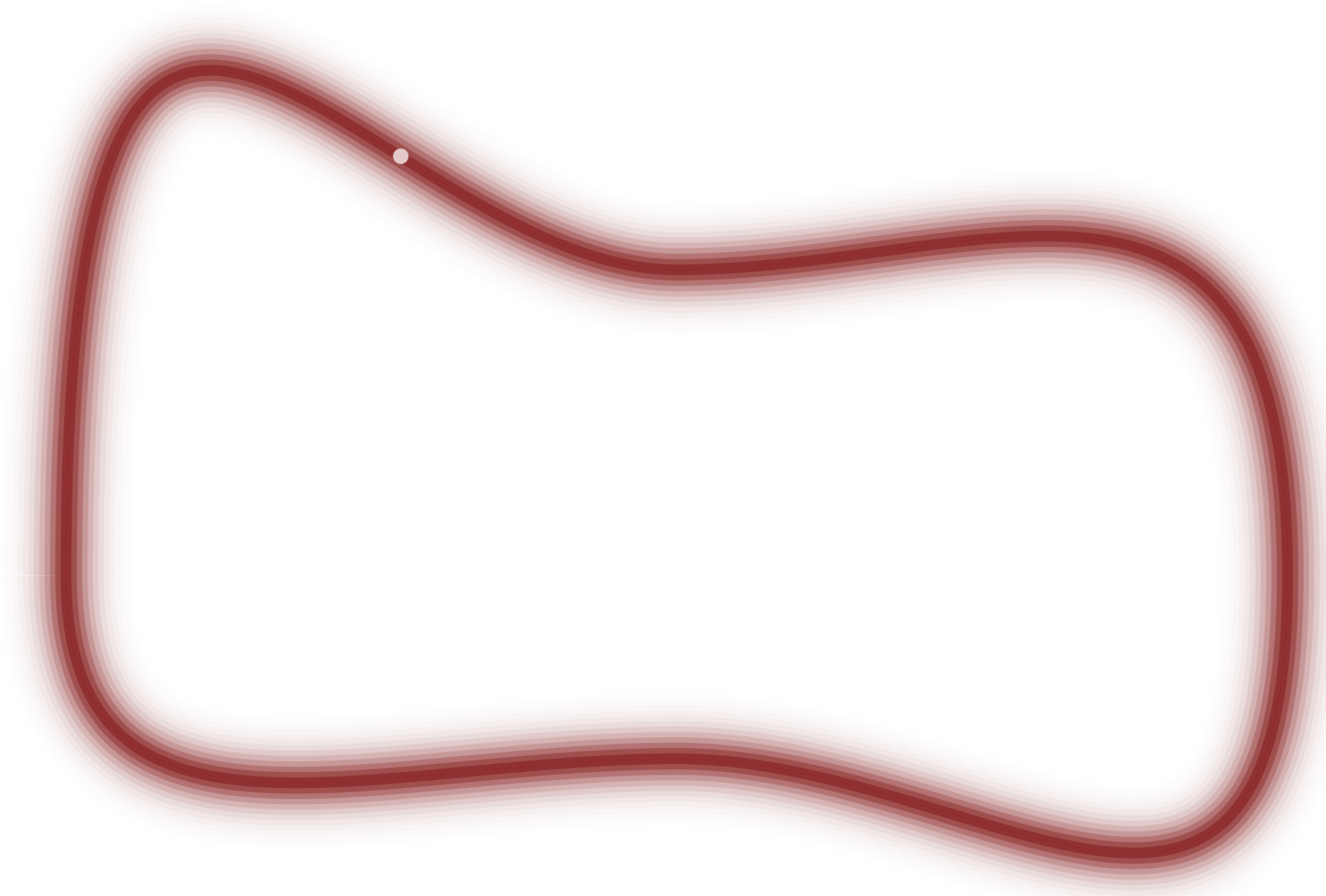




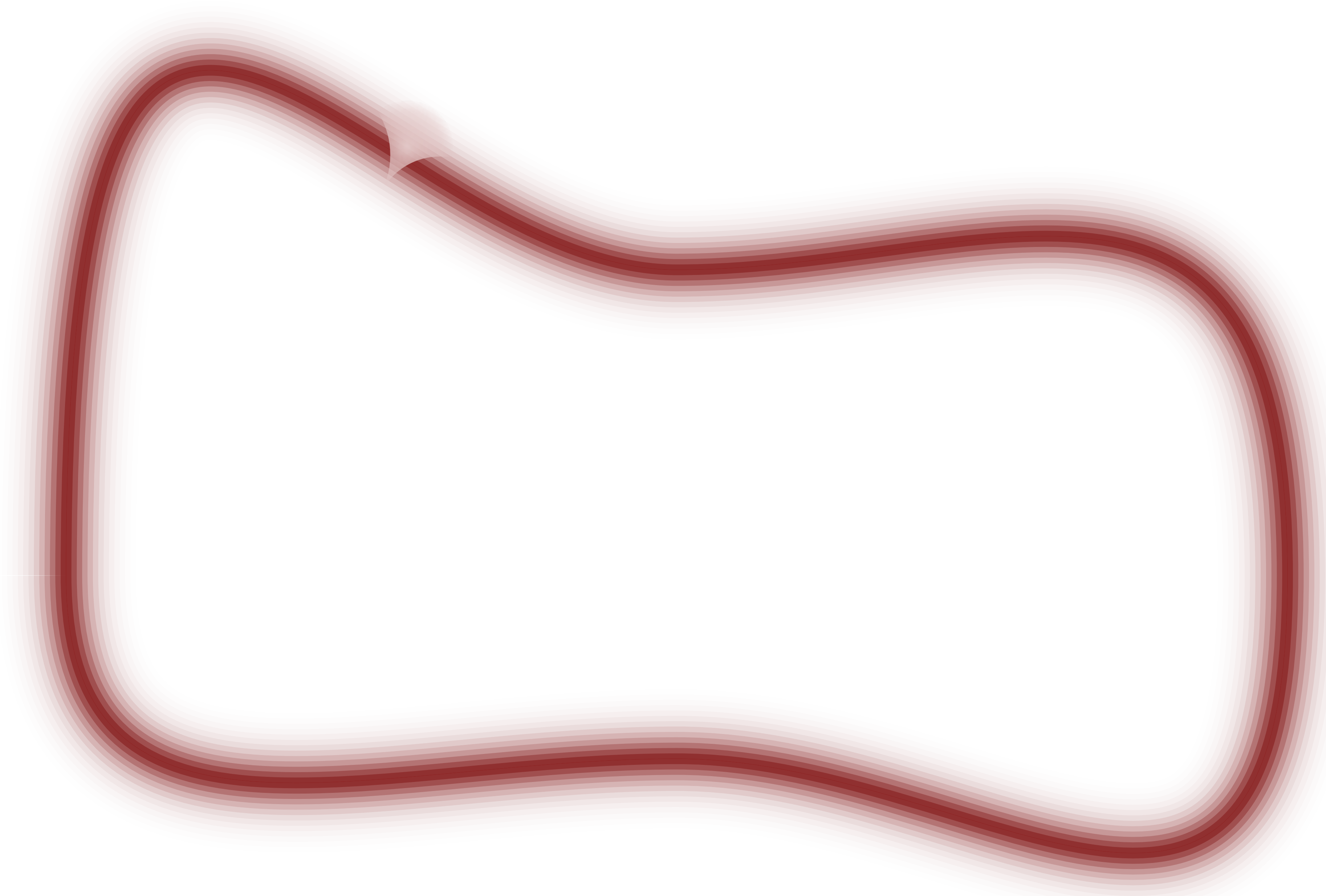
Geometric ergodicity ensures that there are no posterior pathologies obstructing accurate MCMC estimation.

$$\frac{1}{N} \sum_{n=1}^{N} f(q_n) \sim \mathcal{N}\left(\mathbb{E}_{\pi}[f] , \frac{\mathrm{Var}_{\pi}[f]}{\mathrm{ESS}} \right)$$

Unfortunately, common algorithms like Random Walk Metropolis are extremely fragile.



Unfortunately, common algorithms like
Random Walk Metropolis are extremely fragile.



Unfortunately, common algorithms like
Random Walk Metropolis are extremely fragile.