Black-box Variational Inference for Probabilistic Programs

Hongseok Yang
University of Oxford

Joint work with Raphael Monat (ENS Lyon) and Yee Whye Teh (U. of Oxford)

Inference engine for probabilistic programs

- Should be generic.
- Typically based on an Monte-Carlo algo., not on a variational inference algo.
- Recently generic black-box variational inference algorithms* were proposed.
- How to adapt these algos. for PPs?

* Waingate&Weber'13, Ranganath et al.'s AISTATS14, Kucukelbir et al.'s NIPS'15, van de Meent et al.'s AISTATS'16, Li & Turner's NIPS'16, etc

Inference engine for probabilistic programs

- Should be generic.
- Typically based on an Monte-Carlo algo., not on a variational inference algo.
- Recently generic black-box variational inference algorithms* were proposed.
- How to adapt these algos. for PPs?

* Waingate&Weber'13, Ranganath et al.'s AISTATS14, Kucukelbir et al.'s NIPS'15, van de Meent et al.'s AISTATS'16, Li & Turner's NIPS'16, etc

Objective 1: Explain black-box variational inference by Ranganath et al.

Objective 2: Present some preliminary results.

Review of probabilistic programming

```
x=sample(beta(3,2));

if (sample(flip(x))) {

  y=sample(normal(x*x,1));

} else {

  y=sample(normal(5*x,1));

}
obs(normal(y,1),3);
```

[x:0, y:0, w:1]

```
x=sample(beta(3,2));

if (sample(flip(x))) {

  y=sample(normal(x*x,1));

} else {

  y=sample(normal(5*x,1));

}
obs(normal(y,1),3);
```

```
x=sample(beta(3,2));

if (sample(flip(x))) {

  y=sample(normal(x*x,1));

} else {

  y=sample(normal(5*x,1));

}
obs(normal(y,1),3);
```

[x:0, y:0, w:1]

[x:0.8, y:0, w:1]

```
x=sample(beta(3,2));

if (sample(flip(x))) {

  y=sample(normal(x*x,1));

} else {

  y=sample(normal(5*x,1));

}
obs(normal(y,1),3);
```

[x:0, y:0, w:1]

[x:0.8, y:0, w:1]

[x:0.8, y:0, w:1]

```
x=sample(beta(3,2));

if (sample(flip(x))) {

  y=sample(normal(x*x,1));

} else {

  y=sample(normal(5*x,1));

}
obs(normal(y,1),3);
```

[x:0, y:0, w:1]

[x:0.8, y:0, w:1]

[x:0.8, y:0, w:1]

[x:0.8, <u>y:0.3</u>, w:1]

```
x=sample(beta(3,2));

if (sample(flip(x))) {

  y=sample(normal(x*x,1));

} else {

  y=sample(normal(5*x,1));

}
obs(normal(y,1),3);
```

[x:0, y:0, w:1]

[x:0.8, y:0, w:1]

[x:0.8, y:0, w:1]

[x:0.8, y:0.3, w:1]

[x:0.8, y:0.3, w:1]

```
x=sample(beta(3,2));

if (sample(flip(x))) {

  y=sample(normal(x*x,1));

} else {

  y=sample(normal(5*x,1));

}
obs(normal(y,1),3);
```

[x:0, y:0, w:1]

[x:0.8, y:0, w:1]

[x:0.8, y:0, w:1]

[x:0.8, y:0.3, w:1]

[x:0.8, y:0.3, w:1]

[x:0.8, y:0.3, <u>w:0.01..</u>]

```
x=sample(beta(3,2));

if (sample(flip(x))) {

  y=sample(normal(x*x,1));

} else {

  y=sample(normal(5*x,1));

}
obs(normal(y,1),3);
```

[x:0, y:0, w:1]

[x:0.8, y:0, w:1]

[x:0.8, y:0, w:1]

[x:0.8, y:0.3, w:1]

[x:0.8, y:0.3, w:1]

[x:0.8, y:0.3, w:0.01..]

$$p(\tau) \times p(d=3 \mid \tau)$$

```
x=sample(beta(3,2));

if (sample(flip(x))) {

  y=sample(normal(x*x,1));

} else {

  y=sample(normal(5*x,1));

}
obs(normal(y,1),3);
```

[x:0, y:0, w:1]

[x:0.8, y:0, w:1]

[x:0.8, y:0, w:1]

[x:0.8, y:0.3, w:1]

[x:0.8, y:0.3, w:1]

[x:0.8, y:0.3, w:0.01..]

$$p(\tau) \times p(d=3 \mid \tau)$$

```
x=sample(beta(3,2));

if (sample(flip(x))) {

  y=sample(normal(x*x,1));

} else {

  y=sample(normal(5*x,1));

}
obs(normal(y,1),3);
```

$$p(\tau) \times p(d=3 \mid \tau)$$

```
x=sample(beta(3,2));

if (sample(flip(x))) {

  y=sample(normal(x*x,1));

} else {

  y=sample(normal(5*x,1));

}
obs(normal(y,1),3);
```

[x:0, y:0, w:1]

[x:<u>0.6</u>, y:0, w:1]

$$p(\tau) \times p(d=3 \mid \tau)$$

```
x=sample(beta(3,2));

if (sample(flip(x))) {

  y=sample(normal(x*x,1));

} else {

  y=sample(normal(5*x,1));

}
obs(normal(y,1),3);
```

[x:0, y:0, w:1]

[x:0.6, y:0, w:1]

[x:0.6, y:0, w:1]

$$p(\tau) \times p(d=3 \mid \tau)$$

```
x=sample(beta(3,2));

if (sample(flip(x))) {

  y=sample(normal(x*x,1));

} else {

  y=sample(normal(5*x,1));

}
obs(normal(y,1),3);
```

[x:0, y:0, w:1]

[x:0.6, y:0, w:1]

[x:0.6, y:0, w:1]

[x:0.6, <u>y:3.5</u>, w:1]

$$p(\tau) \times p(d{=}3 \mid \tau)$$

```
x=sample(beta(3,2));

if (sample(flip(x))) {

  y=sample(normal(x*x,1));

} else {

  y=sample(normal(5*x,1));

}
obs(normal(y,1),3);
```

[x:0, y:0, w:1]

[x:0.6, y:0, w:1]

[x:0.6, y:0, w:1]

[x:0.6, y:3.5, w:1]

[x:0.6, y:3.5, w:1]

[x:0.6, y:3.5, <u>w:0.35..</u>]

$$p(\tau) \times p(d=3 \mid \tau)$$

```
x=sample(beta(3,2));

if (sample(flip(x))) {

  y=sample(normal(x*x,1));

} else {

  y=sample(normal(5*x,1));

}
obs(normal(y,1),3);
```

[x:0, y:0, w:1]

[x:0.6, y:0, w:1]

[x:0.6, y:0, w:1]

[x:0.6, y:3.5, w:1]

[x:0.6, y:3.5, w:1]

[x:0.6, y:3.5, w:0.35..]

$$p(\tau) \times p(d=3 \mid \tau)$$

```
x=sample(beta(3,2));
if (sample(flip(x))) {
  y=sample(normal(x*x,1));
} else {
  y=sample(normal(5*x,1));
}
obs(normal(y,1),3);
```

[x:0, y:0, w:1]

[x:0.6, y:0, w:1]

[x:0.6, y:0, w:1]

[x:0.6, y:3.5, w:1]

[x:0.6, y:3.5, w:1]

[x:0.6, y:3.5, w:0.35..]

$$p(\tau) \times p(d=3 \mid \tau)$$

```
x=sample(beta(3,2));

if (sample(flip(x))) {

  y=sample(normal(x*x,1));

} else {

  y=sample(normal(5*x,1));

}
obs(normal(y,1),3);
```

[x:0, y:0, w:1]

[x:0.6, y:0, w:1]

[x:0.6, y:0, w:1]

[x:0.6, y:3.5, w:1]

[x:0.6, y:3.5, w:1]

[x:0.6, y:3.5, w:0.35..]

$$p(\tau \mid d=3) = \frac{p(\tau) \times p(d=3 \mid \tau)}{Z}$$

Importance sampling

$$p(\tau|d) = \frac{p(\tau) \times p(d|\tau)}{Z}$$

1. Run the prog. and generate weighted traces:

$$(\tau_1, w_1), \; \ldots \;, \; (\tau_N, w_N).$$

2. Approximate $p(\tau|d)$ by a discrete distr.:

$$p(\tau|d) \approx \frac{\sum_i [\tau = \tau_i] \times w_i}{\sum_i w_i}$$

3. The approximation converges to $p(\tau|d)$.

Review of black-box variational inference

Variational approximation

$$p(\tau|d) = \frac{p(\tau)\times p(d|\tau)}{Z}$$

1. Choose a family of distributions $\{q_{\boldsymbol{\theta}}(\tau)\}_{\boldsymbol{\theta}}$.
2. Solve exactly or approximately:

   $$\boldsymbol{\theta} = \text{argmin}_{\boldsymbol{\theta}}\, KL(q_{\boldsymbol{\theta}}(\tau) \,||\, p(\tau|d)).$$

3. Approximate $p(\tau|d)$ by $q_{\boldsymbol{\theta}}(\tau)$.

Variational approximation

$$p(\tau|d) = \frac{p(\tau) \times p(d|\tau)}{Z}$$

1. Choose a family of distributions $\{q_{\theta}(\tau)\}_{\theta}$.
2. Solve exactly or approximately:

   $$\theta = \text{argmin}_{\theta} \, KL(q_{\theta}(\tau) \,||\, p(\tau|d)).$$

3. Approximate $p(\tau|d)$ by $q_{\theta}(\tau)$.

Variational approximation

$$p(\tau|d) = \frac{p(\tau)\times p(d|\tau)}{Z}$$

1. Choose a family of distributions $\{q_\theta(\tau)\}_\theta$.
2. Solve exactly or approximately:

$$\theta = \text{argmin}_\theta \, KL(q_\theta(\tau) \,||\, p(\tau|d)).$$

3. Approximate $p(\tau|d)$ by $q_\theta(\tau)$.

$\nabla_\theta KL(q_\theta(\tau) \,||\, p(\tau|d)) = 0$

Solve fixpoint equations.

Black-box Variational approximation

$$p(\tau|d) = \frac{p(\tau) \times p(d|\tau)}{Z}$$

1. Choose a family of distributions $\{q_\theta(\tau)\}_\theta$.
2. Solve exactly or approximately:

$$\theta = \mathrm{argmin}_\theta \, KL(q_\theta(\tau) \,||\, p(\tau|d)).$$

3. Approximate $p(\tau|d)$ by $q_\theta(\tau)$.

Gradient descent.
Use estimated gradient.

Minimising KL divergence

$\text{argmin}_{\theta}\, \text{KL}(q_{\theta}(\tau) \,||\, p(\tau|d))$

$= \text{argmin}_{\theta}\, \mathbb{E}_{q_{\theta}(\tau)}[\log(q_{\theta}(\tau) \,/\, p(\tau|d))]$

Minimising KL divergence

$\text{argmin}_{\boldsymbol{\theta}}\ \text{KL}(q_{\boldsymbol{\theta}}(\tau)\ ||\ p(\tau|d))$

$= \text{argmin}_{\boldsymbol{\theta}}\ \mathbb{E}_{q_{\boldsymbol{\theta}}(\tau)}[\log(q_{\boldsymbol{\theta}}(\tau)\ /\ p(\tau|d))]$

$= \text{argmin}_{\boldsymbol{\theta}}\ \mathbb{E}_{q_{\boldsymbol{\theta}}(\tau)}[\log(q_{\boldsymbol{\theta}}(\tau)Z\ /\ p(\tau)p(d|\tau))]$

Minimising KL divergence

$\text{argmin}_\theta \, \text{KL}(q_\theta(\tau) \,||\, p(\tau|d))$

$= \text{argmin}_\theta \, \mathbb{E}_{q_\theta(\tau)}[\log(q_\theta(\tau) \,/\, p(\tau|d))]$

$= \text{argmin}_\theta \, \mathbb{E}_{q_\theta(\tau)}[\log(q_\theta(\tau)Z \,/\, p(\tau)p(d|\tau))]$

$= \text{argmin}_\theta \, \mathbb{E}_{q_\theta(\tau)}[\log(q_\theta(\tau) \,/\, p(\tau)p(d|\tau)) + \log Z]$

Minimising KL divergence

$\mathrm{argmin}_\theta \, KL(q_\theta(\tau) \,||\, p(\tau|d))$

$= \mathrm{argmin}_\theta \, \mathbb{E}_{q_\theta(\tau)}[\log(q_\theta(\tau) \,/\, p(\tau|d))]$

$= \mathrm{argmin}_\theta \, \mathbb{E}_{q_\theta(\tau)}[\log(q_\theta(\tau)Z \,/\, p(\tau)p(d|\tau))]$

$= \mathrm{argmin}_\theta \, \mathbb{E}_{q_\theta(\tau)}[\log(q_\theta(\tau) \,/\, p(\tau)p(d|\tau)) + \log Z]$

$= \mathrm{argmin}_\theta \, \mathbb{E}_{q_\theta(\tau)}[\log(q_\theta(\tau) \,/\, p(\tau)p(d|\tau))].$

No need to know the normalisation const Z.

Key observation

$$\nabla_{\boldsymbol{\theta}}\ \mathbb{E}_{q_{\boldsymbol{\theta}}(\tau)}[\log(q_{\boldsymbol{\theta}}(\tau)\ /\ p(\tau)p(d|\tau))]$$

$$= \mathbb{E}_{q_{\boldsymbol{\theta}}(\tau)}[\log(q_{\boldsymbol{\theta}}(\tau)\ /\ p(\tau)p(d|\tau)) \times \nabla_{\boldsymbol{\theta}} \log(q_{\boldsymbol{\theta}}(\tau))].$$

Black-box variational inference

$$\boxed{\mathbb{E}_{q_{\theta}(\tau)}[\log(q_{\theta}(\tau) \,/\, p(\tau)p(d|\tau)) \times \nabla_{\theta} \log(q_{\theta}(\tau))]}$$

Repeat until θ doesn't change much:

1. Sample $\tau_1, \ldots, \tau_N$ from q_θ and let:

$$g \longleftarrow \frac{\sum_i \log(q_{\theta}(\tau_i) \,/\, p(\tau)p(d|\tau)) \times \nabla_{\theta} \log(q_{\theta}(\tau_i))}{N}$$

2. $\theta \longleftarrow \theta - (\eta \times g)$.

```
x=sample(beta(3,2));
if (sample(flip(x))) {
  y=sample(normal(x*x,1));
} else {
  y=sample(normal(5*x,1));
}
obs(normal(y,1),3);
```

```
x=sample(beta(3,2));
if (sample(flip(x))) {
  y=sample(normal(x*x,1));
} else {
  y=sample(normal(5*x,1));
}
obs(normal(y,1),3);
```

```
x = sample(beta(θ1,θ2));

if (sample(flip(θ3))) {

  y = sample(normal(θ4,1));

} else {

  y = sample(normal(θ5,1));

}
```

```
x=sample(beta(3,2));
if (sample(flip(x))) {
  y=sample(normal(x*x,1));
} else {
  y=sample(normal(5*x,1));
}
obs(normal(y,1),3);
```

```
x = sample(beta(θ1,θ2));

if (sample(flip(θ3))) {

  y = sample(normal(θ4,1));

} else {

  y = sample(normal(θ5,1));

}
```

```
x=sample(beta(3,2));
if (sample(flip(x))) {
  y=sample(normal(x*x,1));
} else {
  y=sample(normal(5*x,1));
}
obs(normal(y,1),3);
```

```
x = sample(beta(θ1,θ2));

if (sample(flip(θ3))) {

  y = sample(normal(θ4,1));

} else {

  y = sample(normal(θ5,1));

}
```

```
x=sample(beta(3,2));
if (sample(flip(x))) {
  y=sample(normal(x*x,1));
} else {
  y=sample(normal(5*x,1));
}
obs(normal(y,1),3);
```

```
x = sample(beta(1,1));

if (sample(flip(0.5))) {

  y = sample(normal(0,1));

} else {

  y = sample(normal(0,1));

}
```

```
x=sample(beta(3,2));
if (sample(flip(x))) {
  y=sample(normal(x*x,1));
} else {
  y=sample(normal(5*x,1));
}
obs(normal(y,1),3);
```

[x:0, y:0]

```
x = sample(beta(1,1));

if (sample(flip(0.5))) {

  y = sample(normal(0,1));

} else {

  y = sample(normal(0,1));

}
```

```
x=sample(beta(3,2));
if (sample(flip(x))) {
  y=sample(normal(x*x,1));
} else {
  y=sample(normal(5*x,1));
}
obs(normal(y,1),3);
```

```
x = sample(beta(1,1));

if (sample(flip(0.5))) {

  y = sample(normal(0,1));

} else {

  y = sample(normal(0,1));

}
```

[x:0, y:0]

[x:0.8, y:0]

```
x=sample(beta(3,2));
if (sample(flip(x))) {
  y=sample(normal(x*x,1));
} else {
  y=sample(normal(5*x,1));
}
obs(normal(y,1),3);
```

[x:0, y:0]

```
x = sample(beta(1,1));

if (sample(flip(0.5))) {

  y = sample(normal(0,1));

} else {

  y = sample(normal(0,1));

}
```

[x:0.8, y:0]

[x:0.8, y:0]

```
x=sample(beta(3,2));
if (sample(flip(x))) {
  y=sample(normal(x*x,1));
} else {
  y=sample(normal(5*x,1));
}
obs(normal(y,1),3);
```

[x:0, y:0]

```
x = sample(beta(1,1));

if (sample(flip(0.5))) {

  y = sample(normal(0,1));

} else {

  y = sample(normal(0,1));

}
```

[x:0.8, y:0]

[x:0.8, y:0]

[x:0.8, y:0.5]

```
x=sample(beta(3,2));
if (sample(flip(x))) {
  y=sample(normal(x*x,1));
} else {
  y=sample(normal(5*x,1));
}
obs(normal(y,1),3);
```

$$g \longleftarrow \log(q_\theta(\tau) \,/\, p(\tau)p(d|\tau)) \times \nabla_\theta \log(q_\theta(\tau))$$

$$\theta \longleftarrow \theta - \eta \times g$$

```
x = sample(beta(1,1));

if (sample(flip(0.5))) {

  y = sample(normal(0,1));

} else {

  y = sample(normal(0,1));

}
```

[x:0, y:0]

[x:0.8, y:0]

[x:0.8, y:0]

[x:0.8, y:0.5]

```
x=sample(beta(3,2));
if (sample(flip(x))) {
  y=sample(normal(x*x,1));
} else {
  y=sample(normal(5*x,1));
}
obs(normal(y,1),3);
```

$g \leftarrow$

$\quad \log(q_\theta(\tau) / p(\tau)p(d|\tau))$

$\quad \times \nabla_\theta \log(q_\theta(\tau))$

$\theta \leftarrow \theta - \eta \times g$

```
x = sample(beta(1,1));

if (sample(flip(0.5))) {

  y = sample(normal(0,1));

} else {

  y = sample(normal(0,1));

}
```

[x:0, y:0]

[x:0.8, y:0]

[x:0.8, y:0]

[x:0.8, y:0.5]

```
x=sample(beta(3,2));
if (sample(flip(x))) {
  y=sample(normal(x*x,1));
} else {
  y=sample(normal(5*x,1));
}
obs(normal(y,1),3);
```

$g \leftarrow$

$\quad \log(q_\theta(\tau) \,/\, p(\tau)p(d|\tau))$

$\quad \times \nabla_\theta \log(q_\theta(\tau))$

$\theta \leftarrow \theta - \eta \times g$

```
x = sample(beta(1,1));

if (sample(flip(0.5))) {

  y = sample(normal(0,1));

} else {

  y = sample(normal(0,1));

}
```

[x:0, y:0]

[x:0.8, y:0]

[x:0.8, y:0]

[x:0.8, y:0.5]

```
x=sample(beta(3,2));
if (sample(flip(x))) {
  y=sample(normal(x*x,1));
} else {
  y=sample(normal(5*x,1));
}
obs(normal(y,1),3);
```

$g \leftarrow$

$\quad \log(q_\theta(\tau) \,/\, p(\tau)p(d|\tau))$

$\quad \times \nabla_\theta \log(q_\theta(\tau))$

$\theta \leftarrow \theta - \eta \times g$

```
x = sample(beta(1,1));

if (sample(flip(0.5))) {

  y = sample(normal(0,1));

} else {

  y = sample(normal(0,1));

}
```

[x:0, y:0]

[x:0.8, y:0]

[x:0.8, y:0]

[x:0.8, y:0.5]

```
x=sample(beta(3,2));
if (sample(flip(x))) {
  y=sample(normal(x*x,1));
} else {
  y=sample(normal(5*x,1));
}
obs(normal(y,1),3);
```

$g \leftarrow$

$\log(q_\theta(\tau) \,/\, p(\tau)p(d|\tau))$

$\times \nabla_\theta \log(q_\theta(\tau))$

$\theta \leftarrow \theta - \eta \times g$

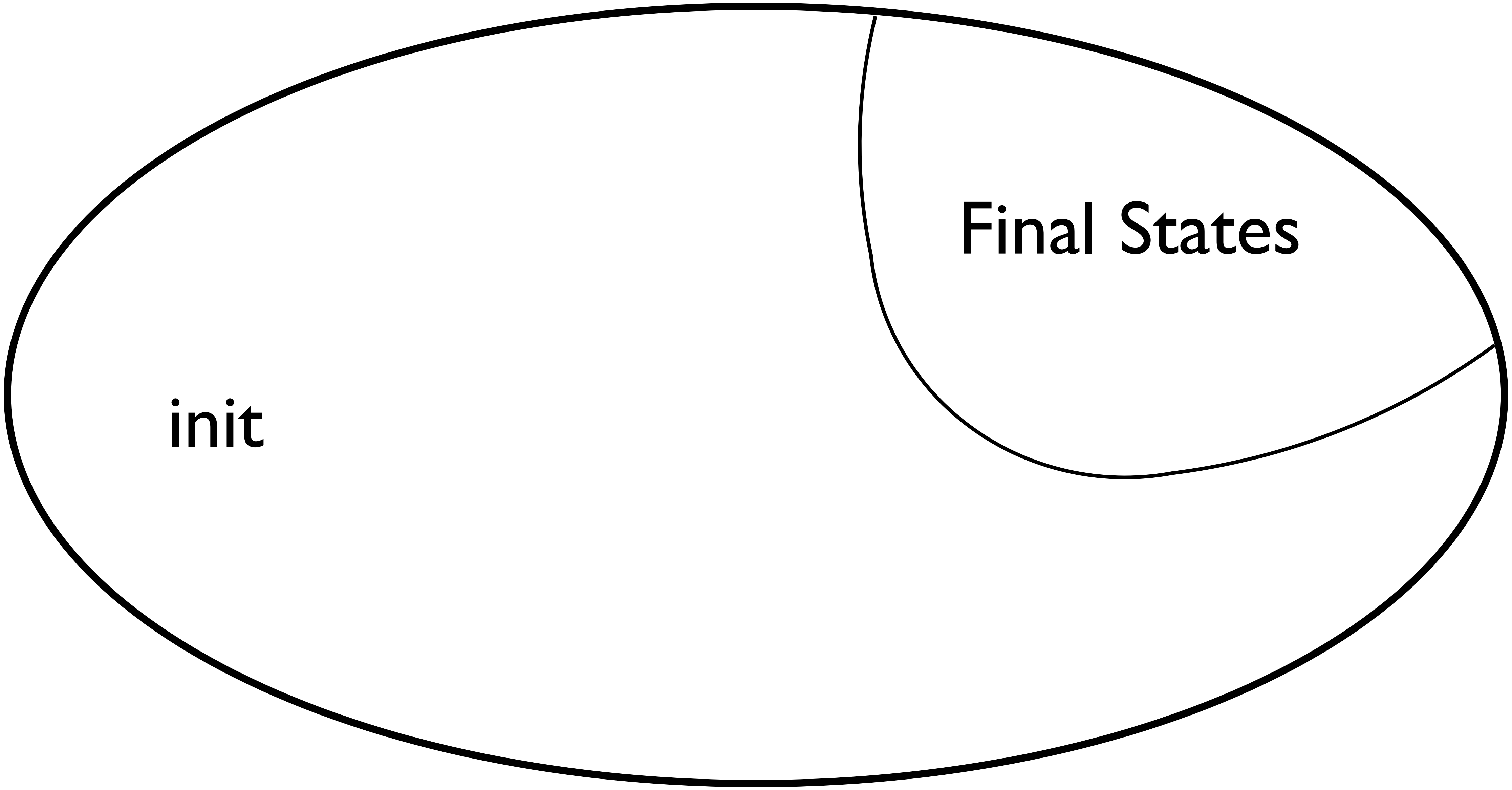
```
x = sample(beta(0.9,2));

if (sample(flip(0.6))) {

  y = sample(normal(0,1));

} else {

  y = sample(normal(2,1));

}
```

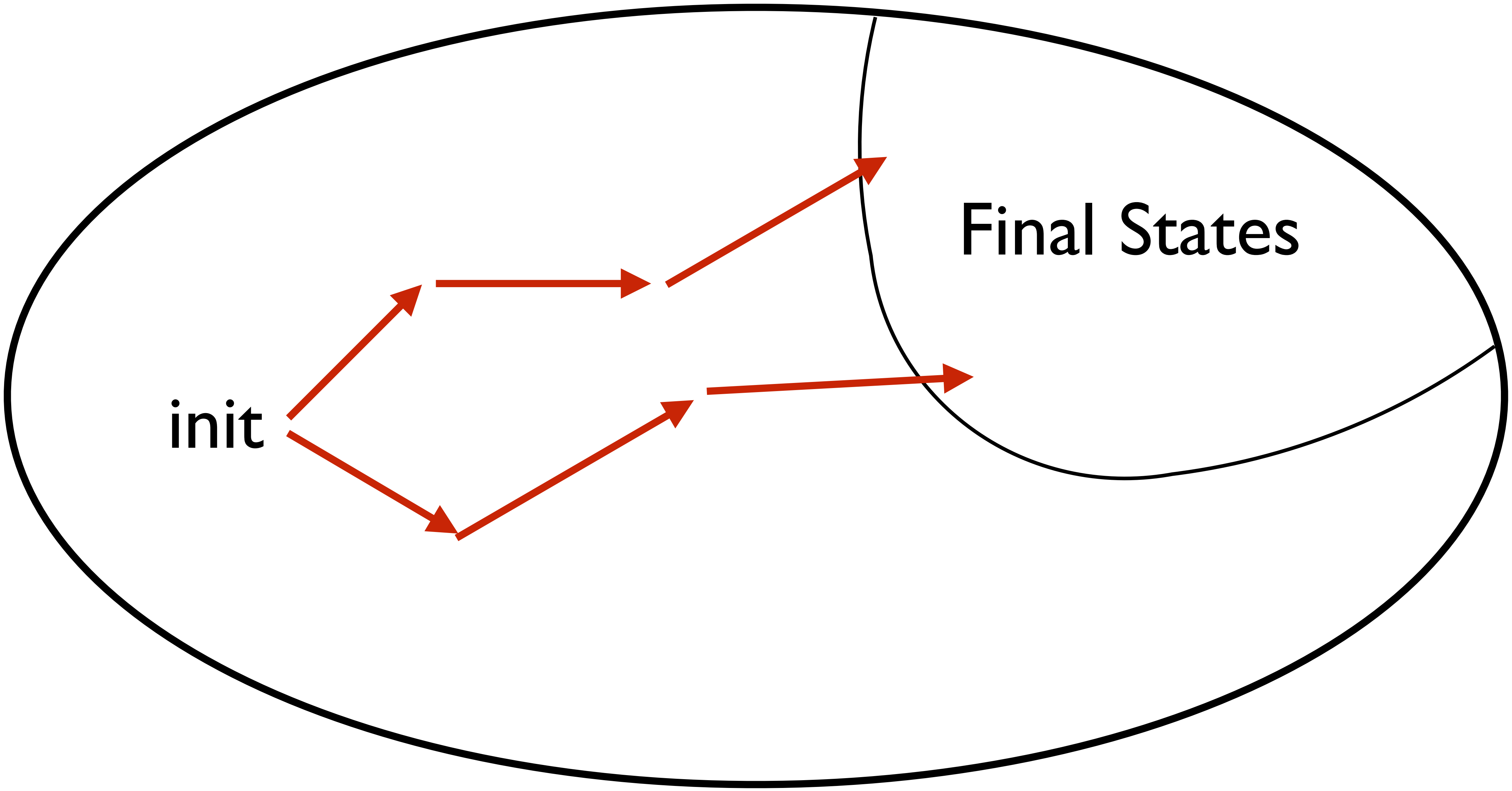
[x:0, y:0]

[x:0.8, y:0]

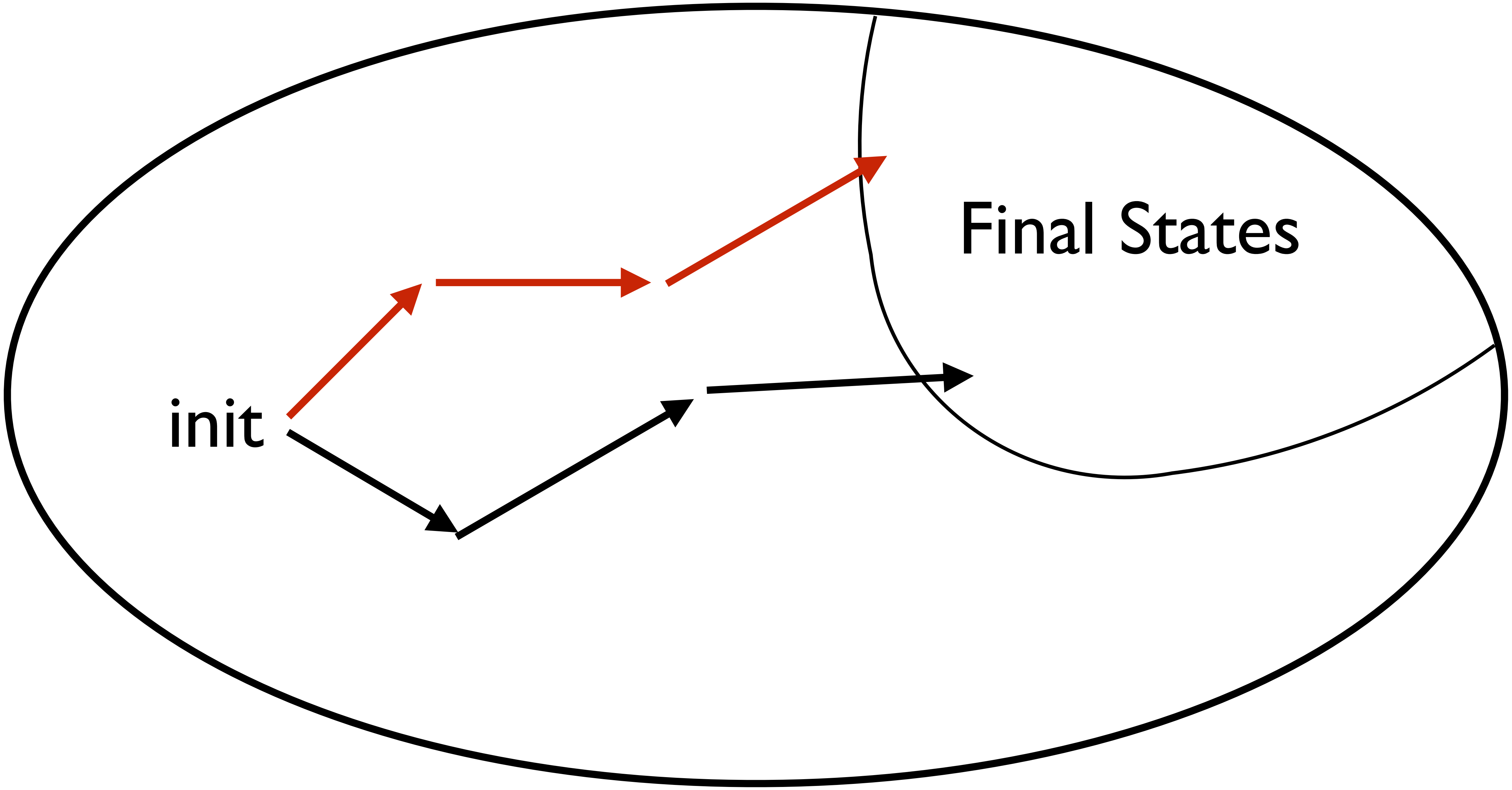
[x:0.8, y:0]

[x:0.8, y:0.5]


Final States
init



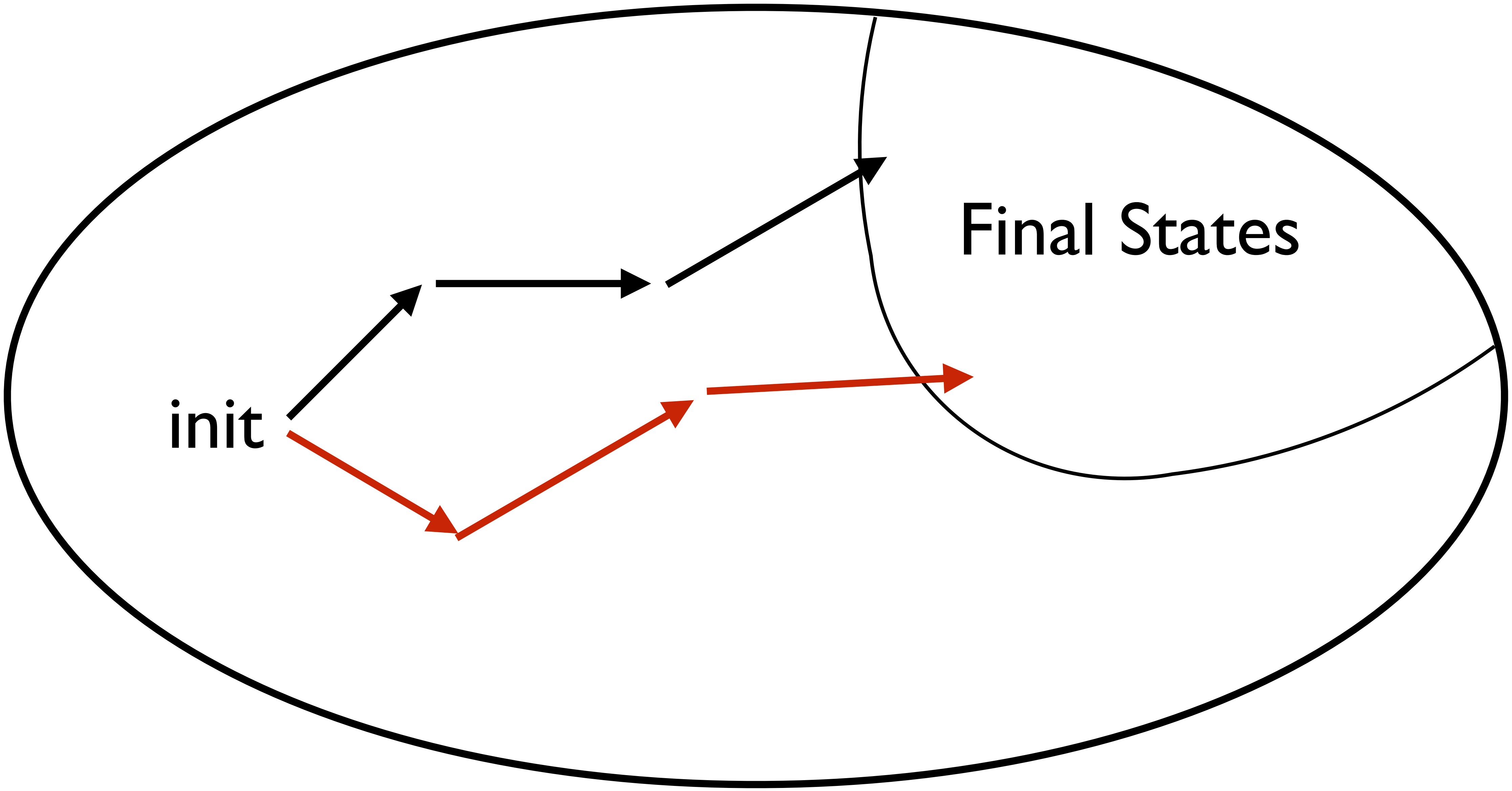
1. Sample traces by running approx. prog.



1. Sample traces by running approx. prog.

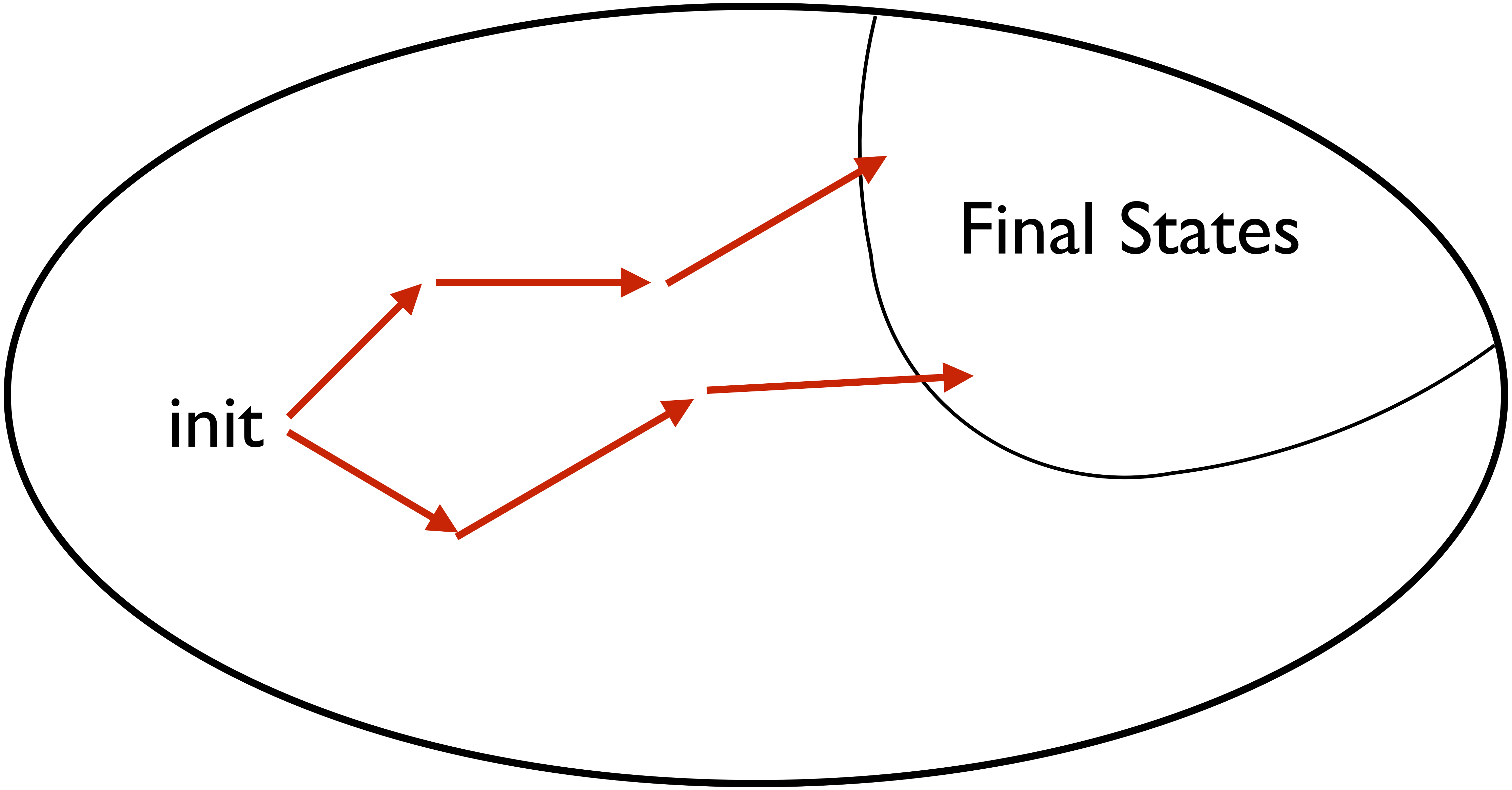
2. For each sampled τ_i, compute:

$$\log(q_\theta(\tau_i) / p(\tau_i)p(d|\tau_i)) \times \nabla_\theta \log(q_\theta(\tau_i))$$



1. Sample traces by running approx. prog.
2. For each sampled τ_i, compute:

$$\log(q_\theta(\tau_i) \;/\; p(\tau_i)p(d|\tau_i)) \times \nabla_\theta \log(q_\theta(\tau_i))$$



1. Sample traces by running approx. prog.
2. For each sampled τ_i, compute:

$$\log(q_\theta(\tau_i) \,/\, p(\tau_i)p(d|\tau_i)) \times \nabla_\theta \log(q_\theta(\tau_i))$$

3. Use their average as the estimated grad.

High variance

- Estimator of the gradient has high variance.
- Many techniques exist.
- Our approach is to exploit the structure:

$$q_{\theta}(\tau) = \text{valid}(\tau) \times \prod_{0<i<|\tau|} f_{\theta}(\tau_i, \tau_{i+1})$$

$$p(\tau)p(d|\tau) = \text{valid}(\tau) \times \prod_{0<i<|\tau|} f(\tau_i, \tau_{i+1})g(\tau_{i+1})$$

Our preliminary results

[Theorem 1] Under some condition,

$$\mathrm{argmin}_\theta \, KL(q_\theta(\tau) \,||\, p(\tau|d))$$

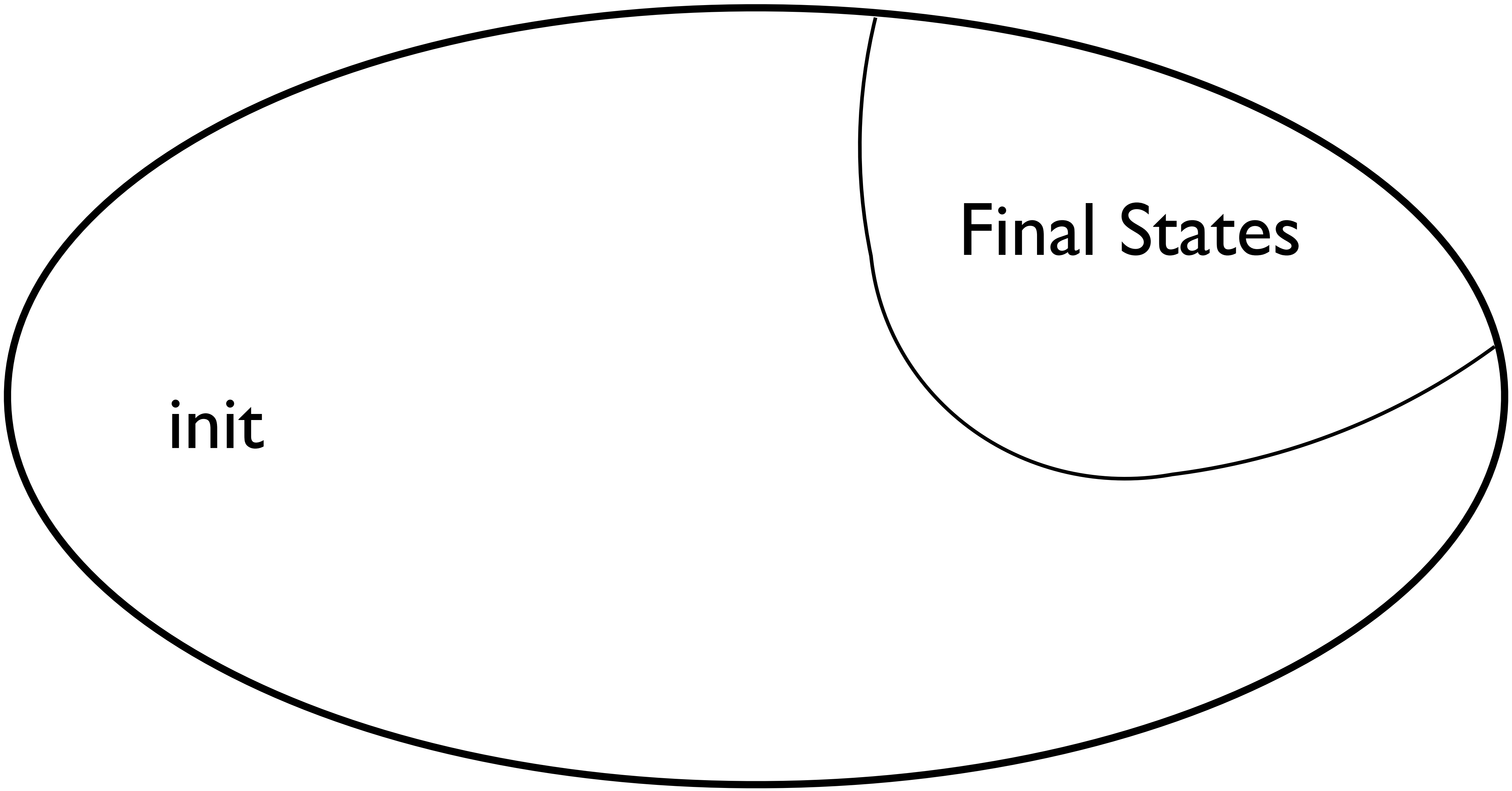
$$= \mathrm{argmin}_\theta \, \mathbb{E}_{q\theta(\tau)}[\log(q_\theta(\tau) \,/\, p(\tau)p(d|\tau))]$$

$$= \mathrm{argmin}_\theta \, \mathbb{E}_{q\theta(\tau)}[\textstyle\sum_{0<i<|\tau|} k_\theta(\tau_i)]$$

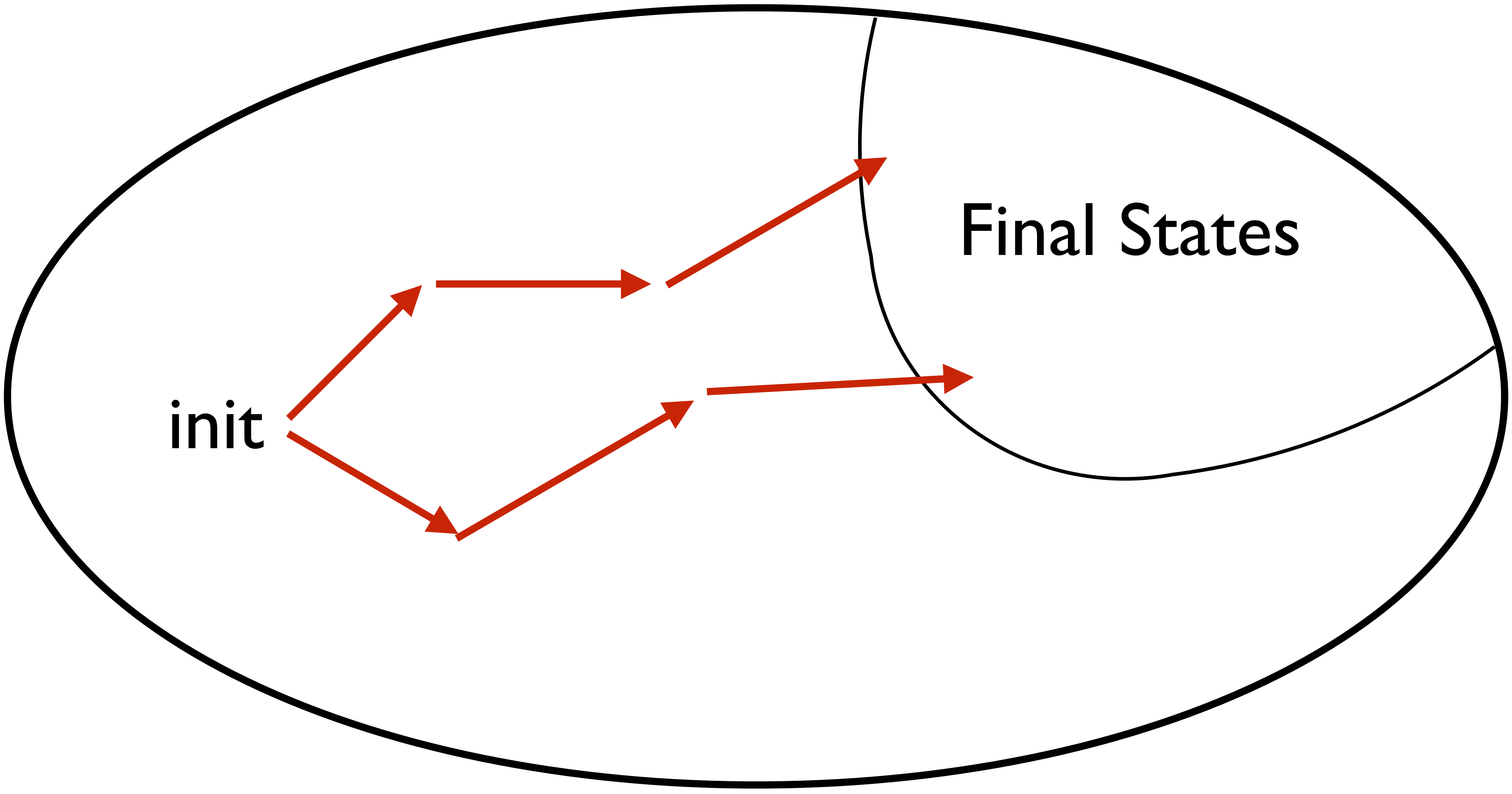
where

$$k_\theta(s) = \mathbb{E}_{f\theta(s,s')}[\log(f_\theta(s,s') \,/\, f(s,s')g(s'))].$$

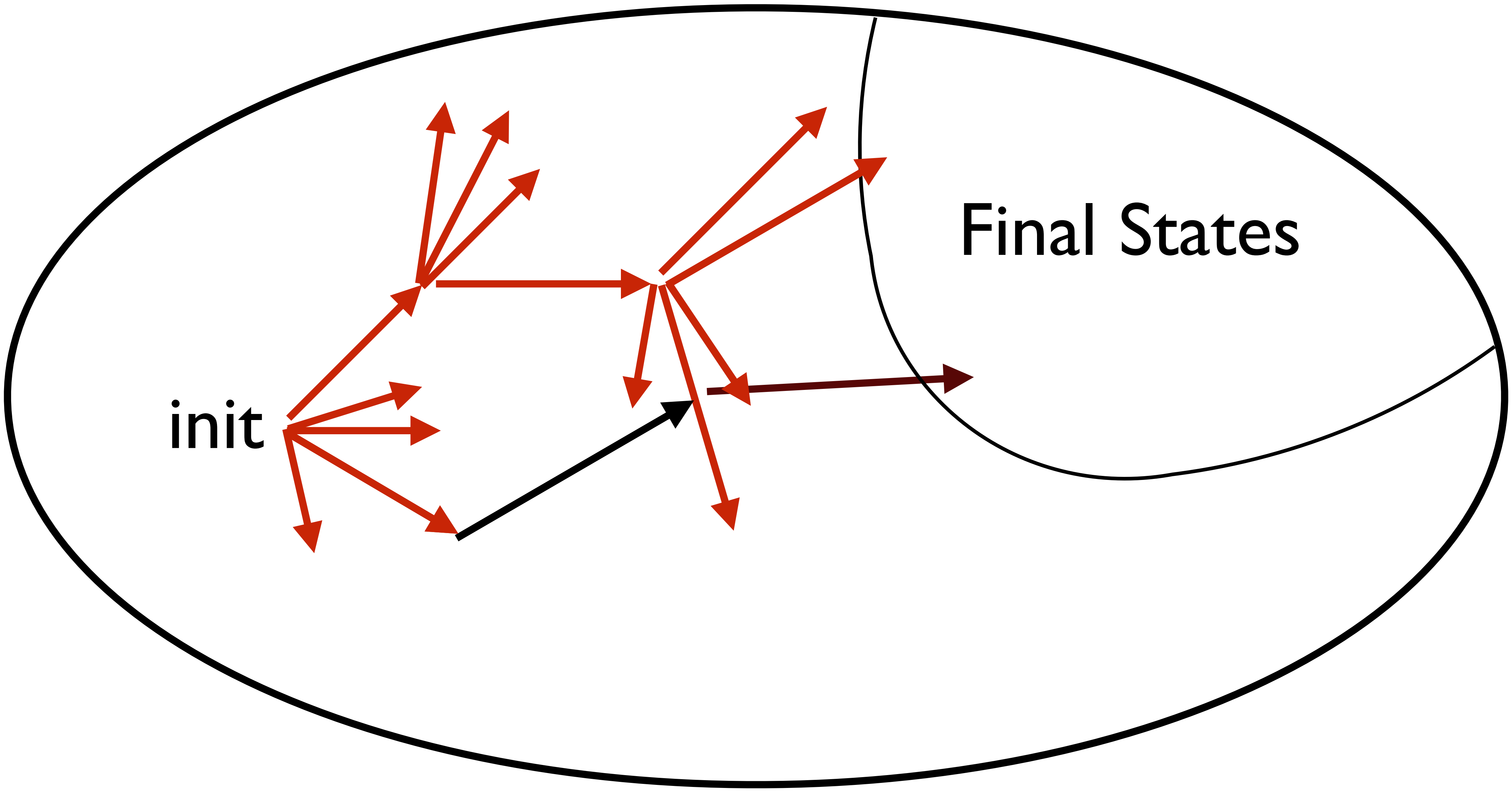
Execute & Compute & Average



Execute & Compute & Average



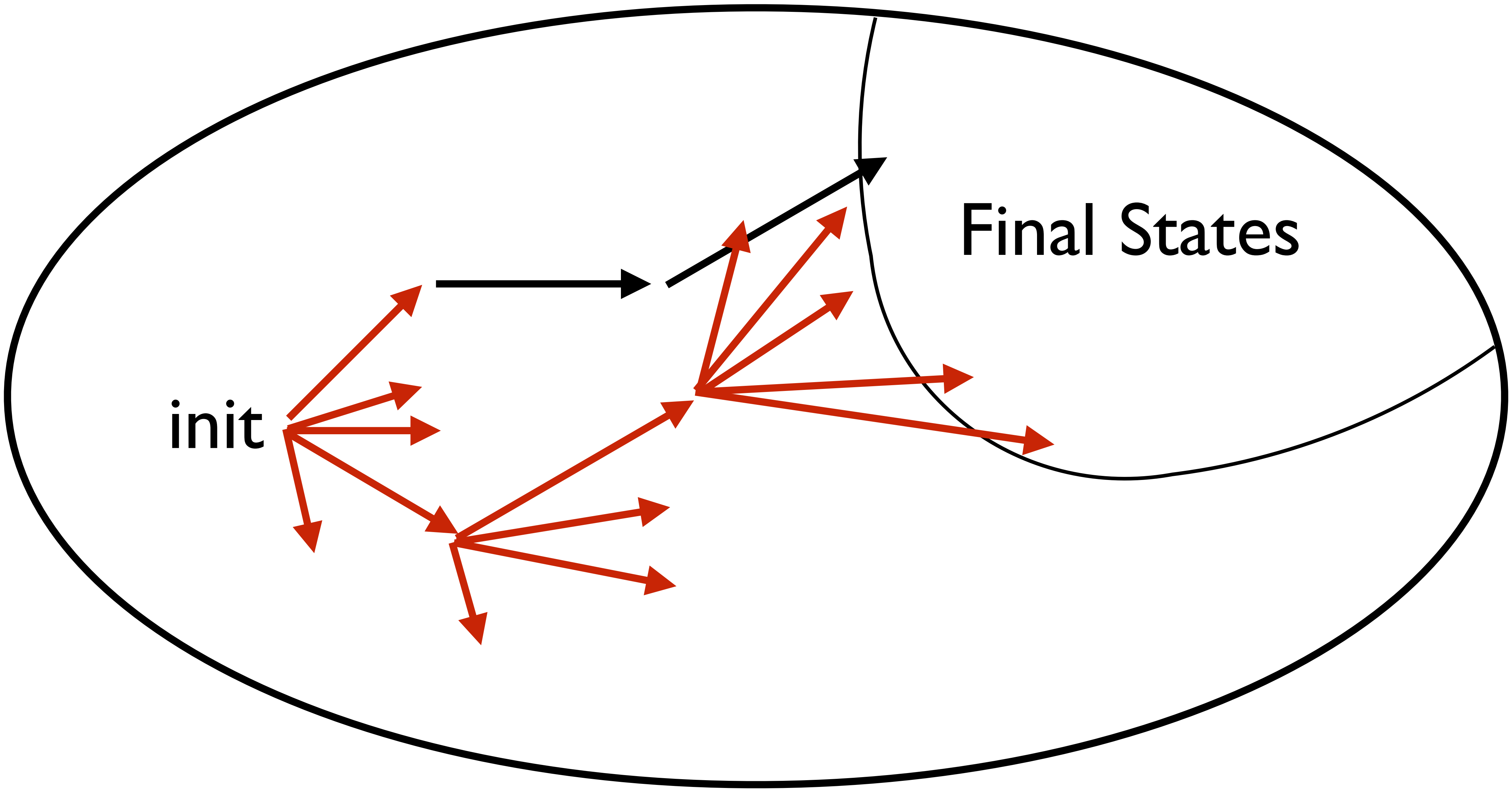
Execute & Compute & Average



$$\sum_{0<i<|\tau|} k_\theta(\tau_i)\nabla_\theta \log(q_\theta(\tau)) \; + \; \sum_{0<i<|\tau|} \nabla_\theta k_\theta(\tau_i)$$

where $\quad k_\theta(s) = \mathbb{E}_{f\theta(s,s')}[\log(f_\theta(s,s') \,/\, f(s,s')g(s'))]$

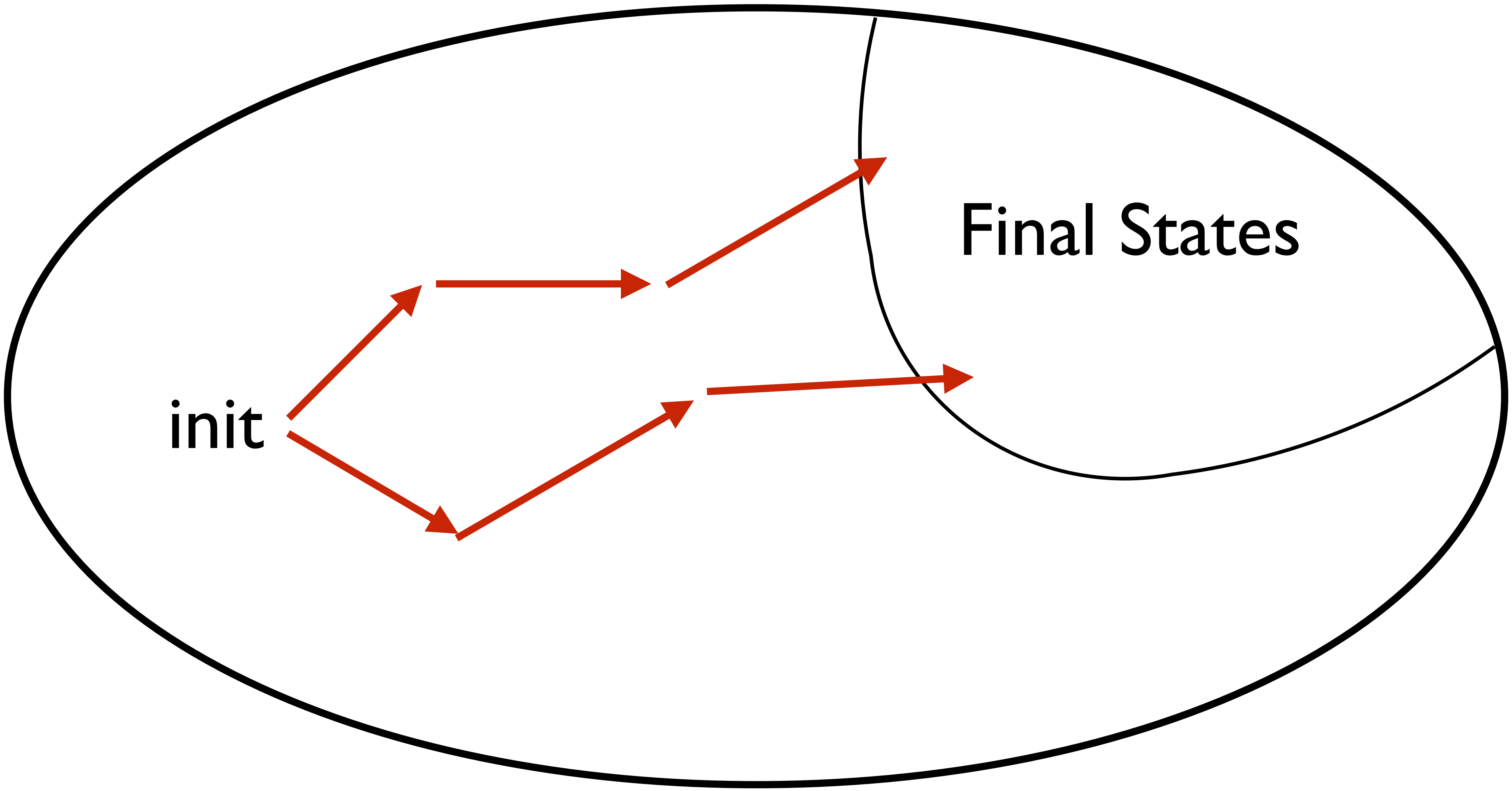
Execute & Compute & Average



$$\sum_{0<i<|\tau|} k_\theta(\tau_i)\nabla_\theta \log(q_\theta(\tau)) \;+\; \sum_{0<i<|\tau|} \nabla_\theta k_\theta(\tau_i)$$

where $k_\theta(s) = \mathbb{E}_{f\theta(s,s')}[\log(f_\theta(s,s') \,/\, f(s,s')g(s'))]$

Execute & Compute & Average



$$\sum_{0<i<|\tau|} k_\theta(\tau_i)\nabla_\theta \log(q_\theta(\tau)) \;+\; \sum_{0<i<|\tau|} \nabla_\theta k_\theta(\tau_i)$$

$$\text{where} \quad k_\theta(s) = \mathbb{E}_{f\theta(s,s')}[\log(f_\theta(s,s') \,/\, f(s,s')g(s'))]$$

[Theorem 2] Under some condition,

$$\mathrm{argmin}_\theta \, KL(q_\theta(\tau) \,||\, p(\tau|d))$$

$$= \mathrm{argmin}_\theta \, \mathbb{E}_{q\theta(\tau)}[\log(q_\theta(\tau) \,/\, p(\tau)p(d|\tau))]$$

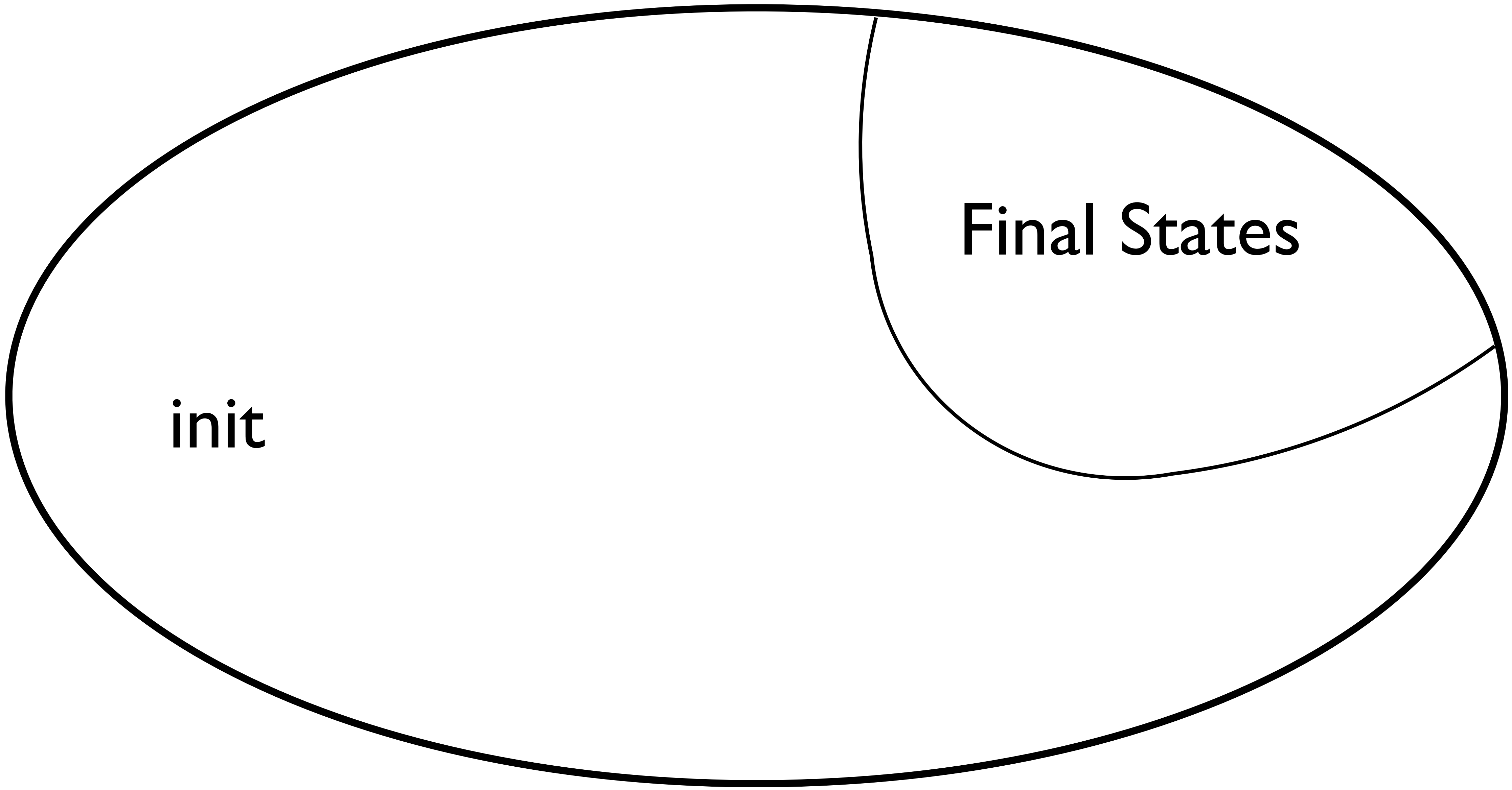
$$= \mathrm{argmin}_\theta \, N \, \mathbb{E}_{B\theta(s)}[k_\theta(s)]$$

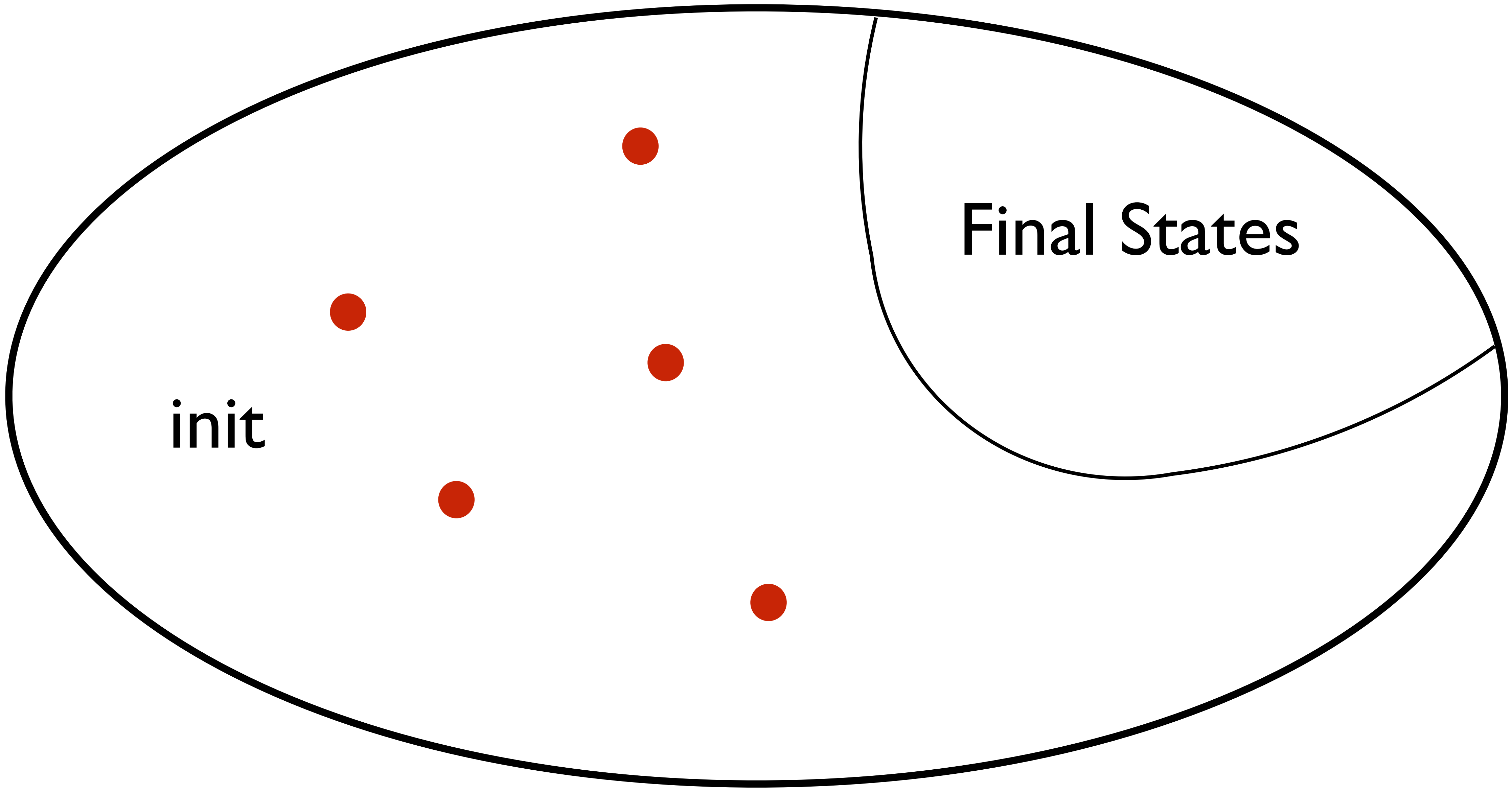
where

$$k_\theta(s) = \mathbb{E}_{f\theta(s,s')}[\log(f_\theta(s,s') \,/\, f(s,s')g(s'))]$$

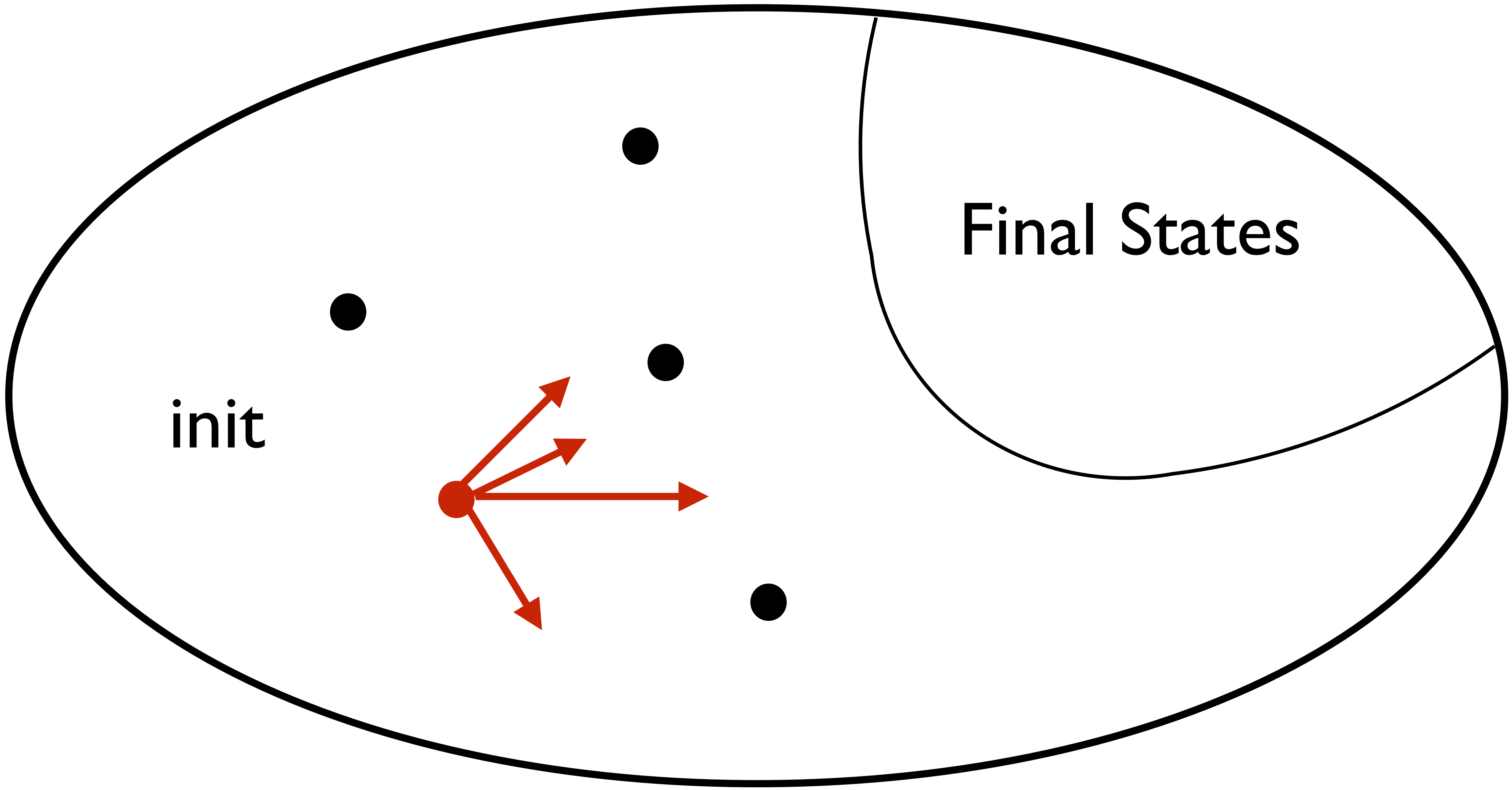
$$A_\theta(s) = [s{=}init] + \int A_\theta(s_0) f_\theta(s_0,s) \, ds_0$$

$$N = \int A_\theta(s) ds \quad \text{and} \quad B_\theta(s) = A_\theta(s) \,/\, N$$

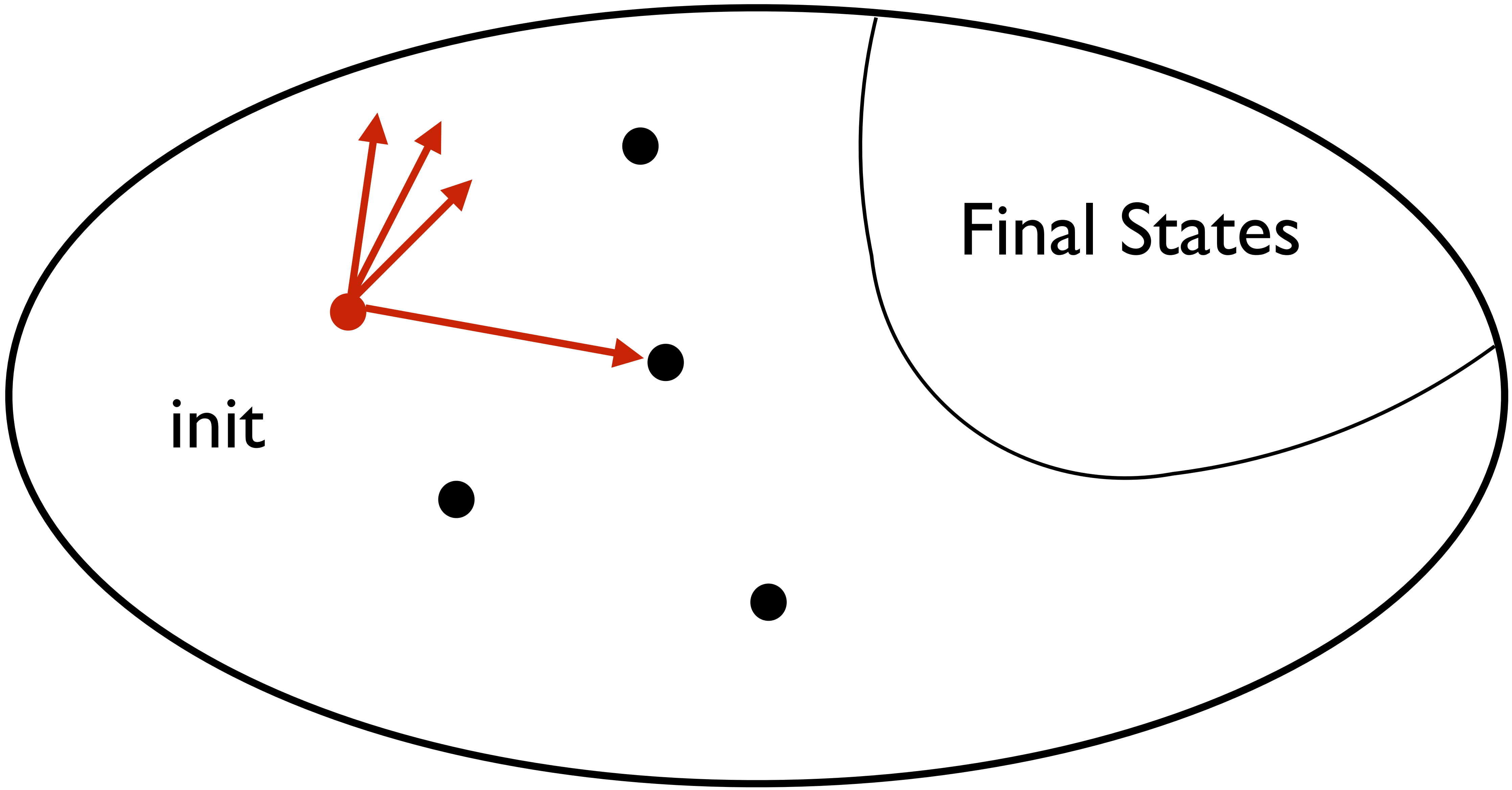

Final States
init



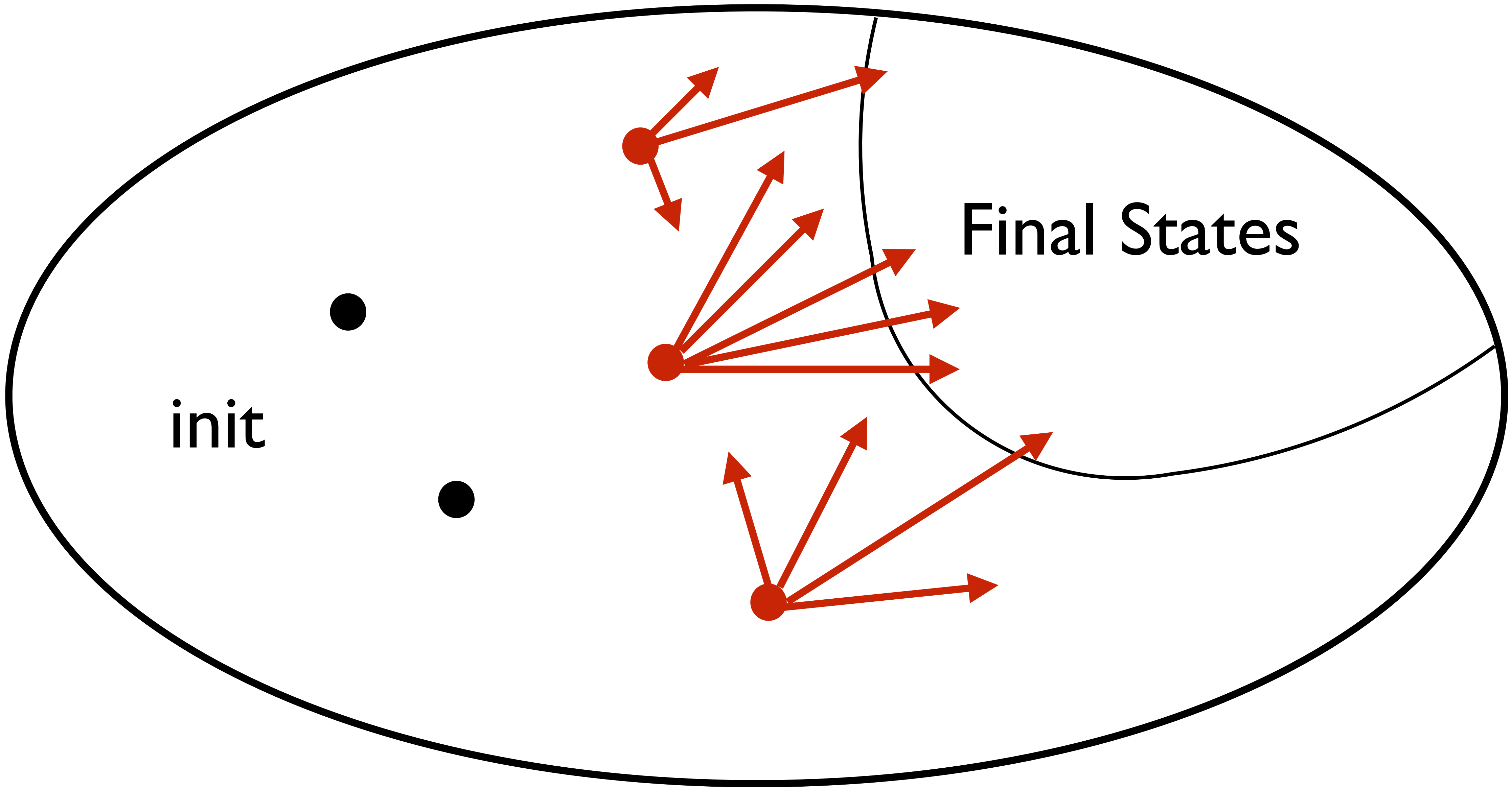
1. Sample states from $B_\theta(s)$.



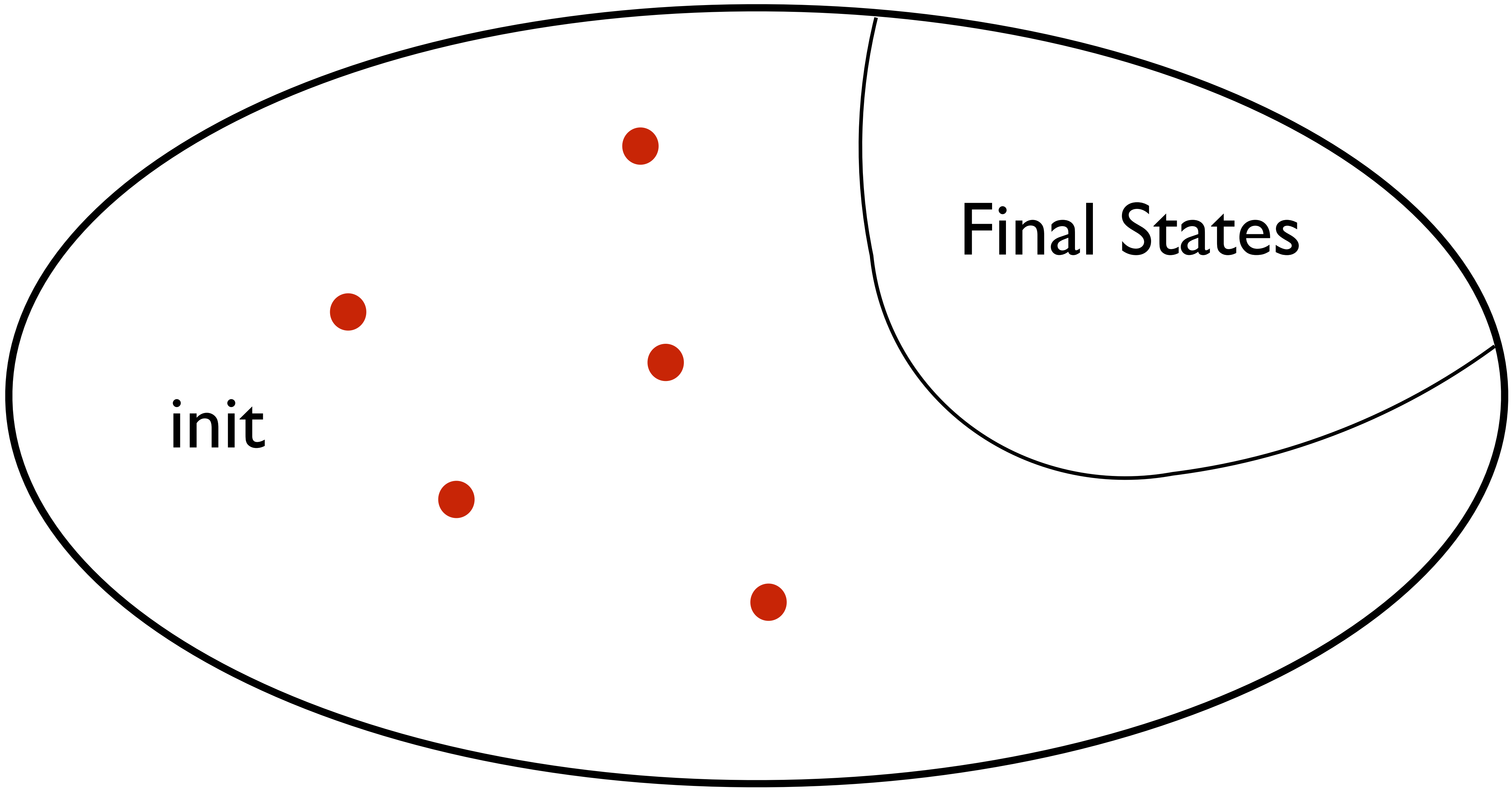
1. Sample states from $B_\theta(s)$.
2. For each sample s_i, compute:
   $N \times (\nabla_\theta k_\theta(s_i) + k_\theta(s_i)\nabla_\theta \log(B_\theta(s_i)))$



1. Sample states from $B_\theta(s)$.
2. For each sample s_i, compute:
   $N \times (\nabla_\theta k_\theta(s_i) + k_\theta(s_i)\nabla_\theta \log(B_\theta(s_i)))$



1. Sample states from $B_\theta(s)$.
2. For each sample s_i, compute:
   $N \times (\nabla_\theta k_\theta(s_i) + k_\theta(s_i)\nabla_\theta \log(B_\theta(s_i)))$



1. Sample states from $B_\theta(s)$.
2. For each sample s_i, compute:
   $N \times (\nabla_\theta k_\theta(s_i) + k_\theta(s_i)\nabla_\theta \log(B_\theta(s_i)))$
3. Use their average as the estimator.

Do they work?

- Sorry. We don't know yet.
- Implementation is ongoing. I hope that next time we will be able to answer.