Computer Science

CSC696H: Advanced Topics in Probabilistic Graphical Models

No U-Turn Sampler

Hoffman, M. and Gelman, A. JMLR (2014)

Prof. Jason Pacheco

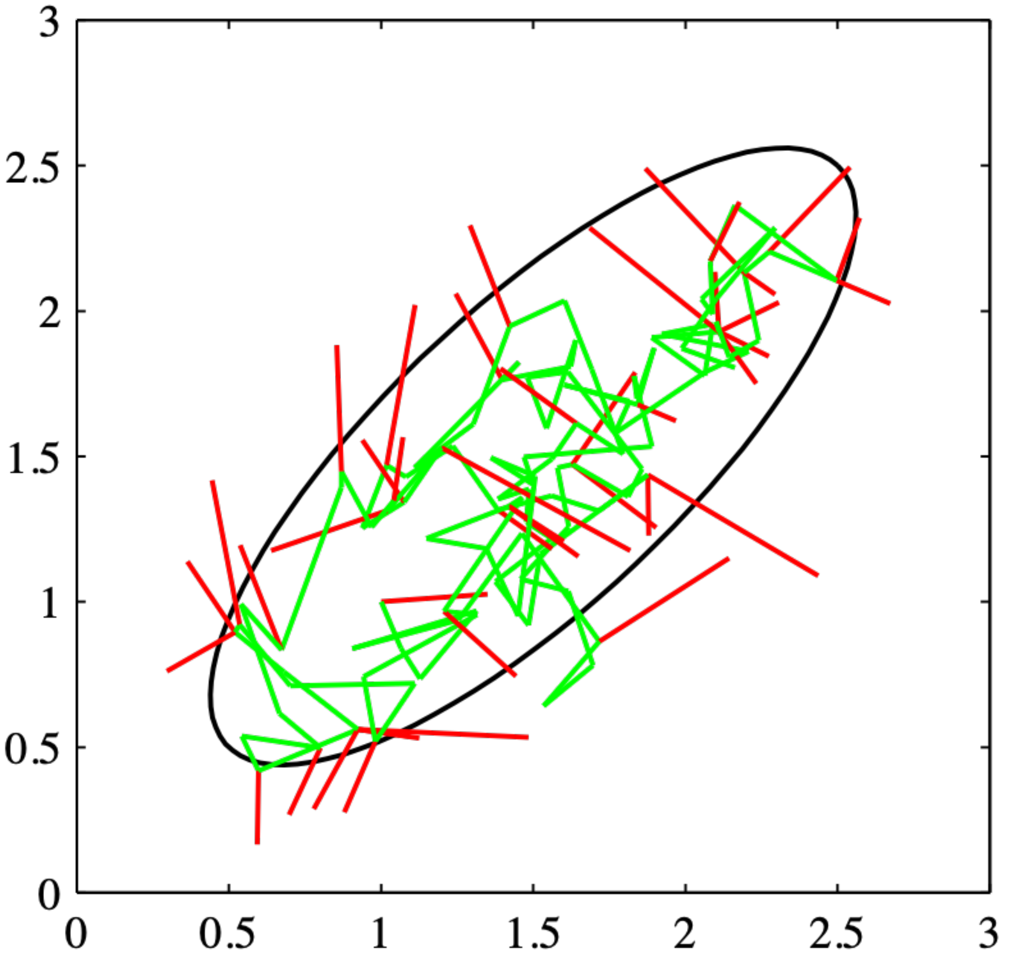
(Random Walk) Metropolis Algorithm

```
While not_bored
{
        Sample $q\left(z \middle| z^{(prev)}\right)$
        Accept with probability $A\left(z, z^{(prev)}\right) = \min\left(1, \frac{\tilde{p}(z)}{\tilde{p}\left(z^{(prev)}\right)}\right)$
        If accept, emit $z$, otherwise, emit $z^{(prev)}$.
}
```

Always emit one or the other

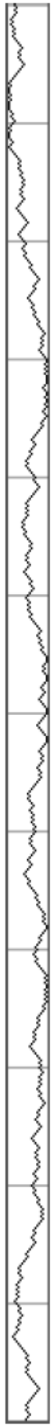
If things get better, always accept. If they get worse, sometimes accept.

(Random Walk) Metropolis Example



Green follows accepted proposals
Red are rejected moves.

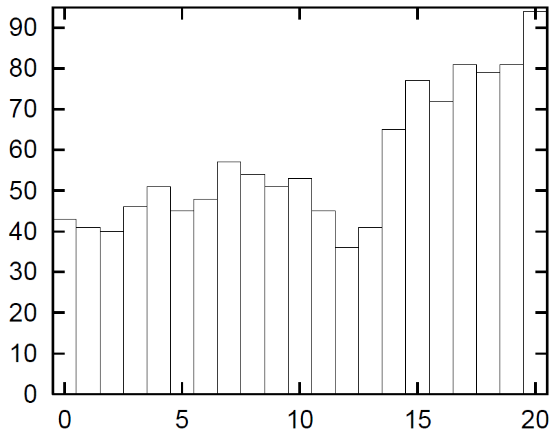
Example: Random Walk Dynamics



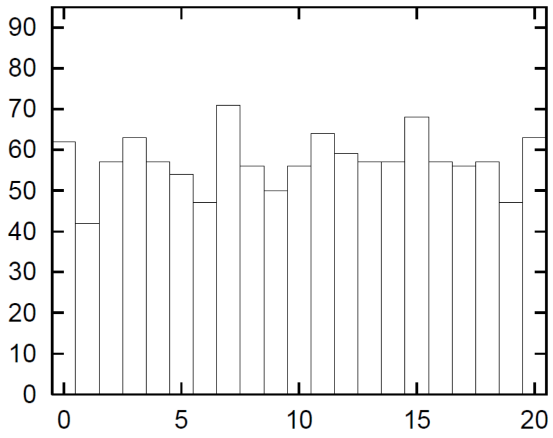
State evolution for t=1…600, horizontal bars denote intervals of 50

[Source: D. MacKay]

Metropolis


1200 iterations

Independent


1200 iterations

Target: $$p(x) = \begin{cases} \frac{1}{21} & x \in \{0, \ldots, 20\} \\ 0 & \text{otherwise} \end{cases}$$

Proposal: $$q(x' \mid x) = \begin{cases} \frac{1}{2} & x' = x \pm 1 \\ 0 & \text{otherwise} \end{cases}$$

From $x_0 = 10$ need ~400 steps to reach both end states (0 and 20). So, ~400 steps to generate 1 independent sample!

Very important to avoid random walk dynamics

Hamiltonian Monte Carlo (HMC)

Better at avoiding random walk behavior typically associated with Metropolis(-Hastings) and Gibbs samplers

Some Drawbacks…

- Per-iteration cost for D-dim RV is $\mathcal{O}(D^{5/4})$
- Contrast to random walk Metropolis $\mathcal{O}(D^2)$
- *Very Sensitive* to hyperparameters
  - Number of leapfrog steps L ← Requires costly tuning runs (NUTS focuses on this)
  - Stepsize ϵ ← Tuning this on-the-fly not too hard [Andrieu and Thomas (2008) + this paper]
- Requires gradient of (unnormalized) log-probability

HMC Recap

Canonical form of our target distribution (the one we want to sample):

$$p(\theta) = \frac{1}{Z} \exp\left(\mathcal{L}(\theta)\right)$$

where $\mathcal{L}(\theta)$ is the log-PDF

Introduce *momentum* to form $r \sim \mathcal{N}(0,1)$ Hamiltonian in canonical form:

$$p(\theta, r) = p(\theta)p(r) \propto \exp\left(\mathcal{L}(\theta) - \frac{1}{2} r^T r\right)$$

Intuition Fictitious Hamiltonian energy of D-dimensional "position" θ and r_d is momentum of d-th position dimension.

- Position-dependent potential energy: $-\mathcal{L}(\theta)$
- Kinetic energy: $-\frac{1}{2} r^T r$

HMC Recap

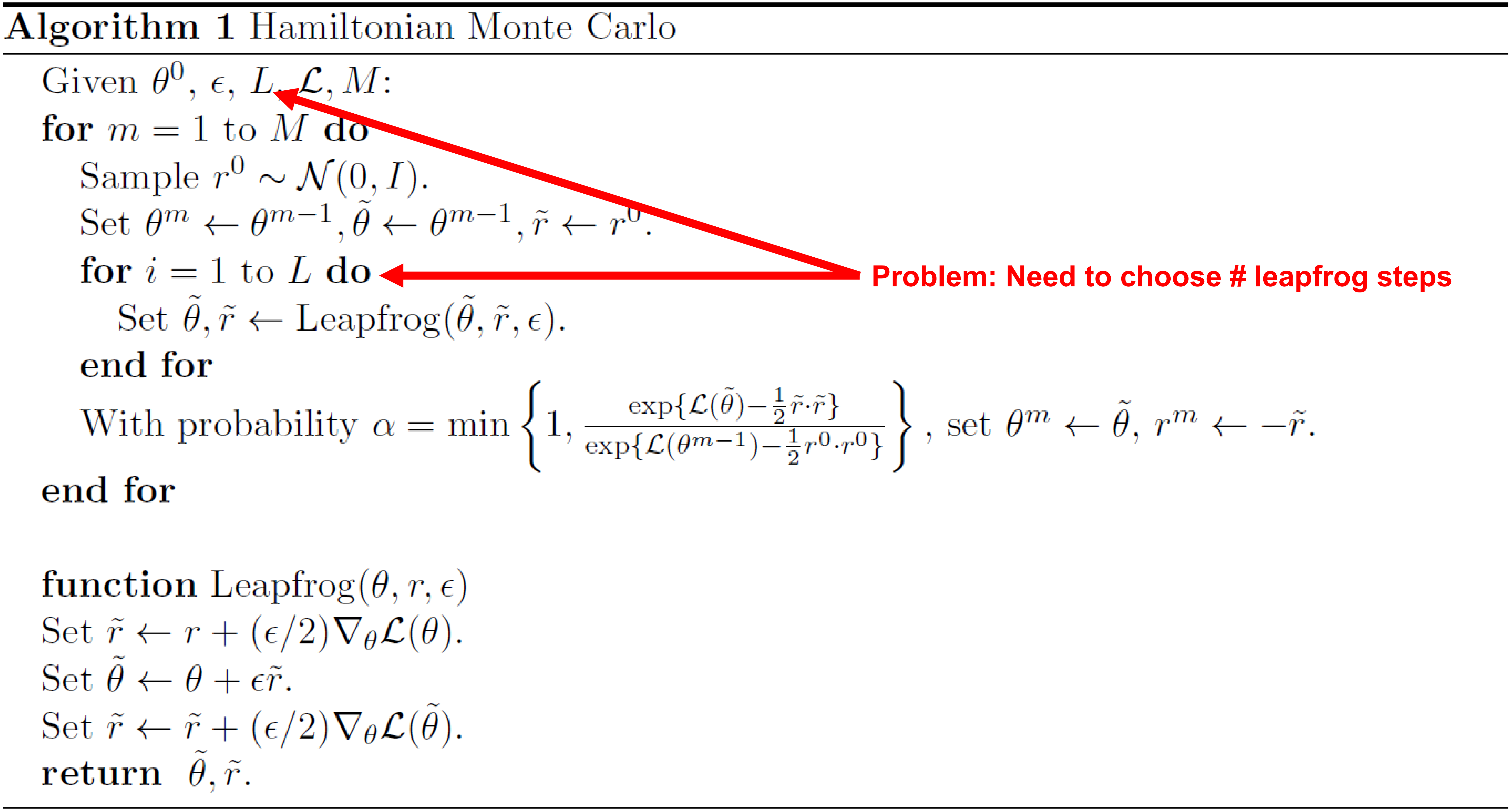
Can simulate Hamiltonian dynamics of our fictitious physical system:

$$\frac{dr}{dt} = \frac{\partial \mathcal{L}(\theta)}{\partial \theta} \qquad\qquad \frac{d\theta}{dt} = \frac{\partial}{\partial r}\frac{1}{2}r^T r = r$$

Need to do this numerically, so we use a “leapfrog” integrator:

$$r^{t+\epsilon/2} = r^t + (\epsilon/2)\nabla_\theta \mathcal{L}(\theta^t); \quad \theta^{t+\epsilon} = \theta^t + \epsilon r^{t+\epsilon/2}; \quad r^{t+\epsilon} = r^{t+\epsilon/2} + (\epsilon/2)\nabla_\theta \mathcal{L}(\theta^{t+\epsilon}),$$

- Simulated θ is a Metropolis-Hastings proposal
- Volume preserving and time-reversible
- Time-reversible
- Satisfies detailed balance ➔ valid MCMC sampler with target $p(\theta)$



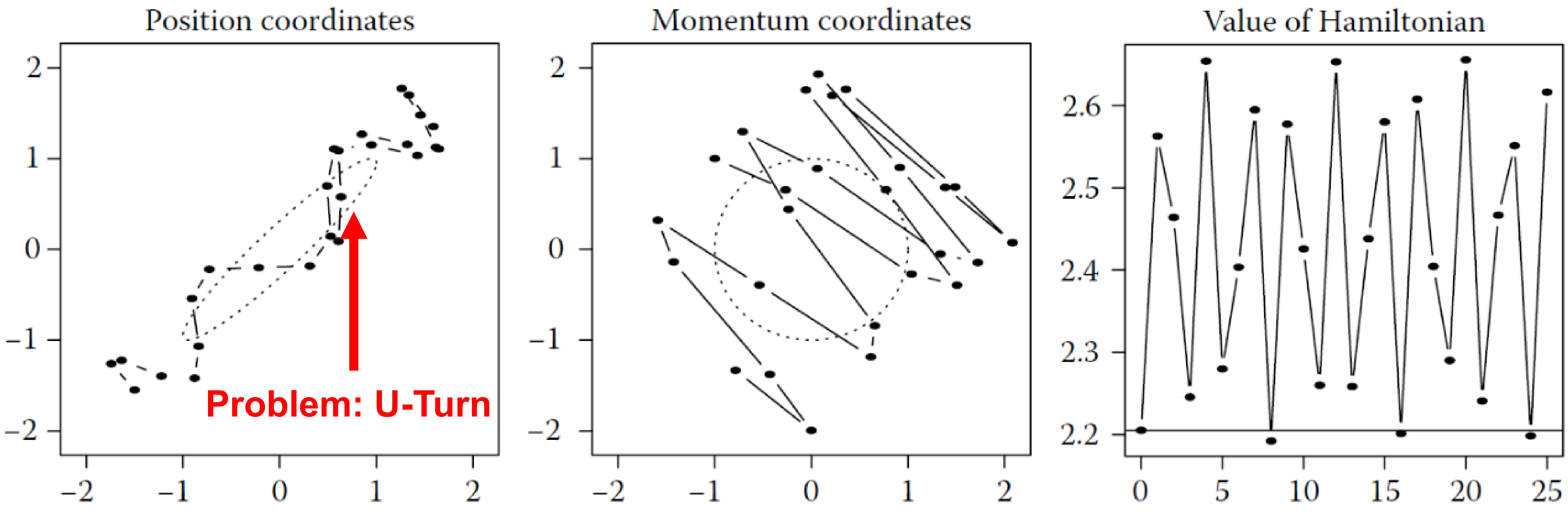
Algorithm 1 Hamiltonian Monte Carlo

```
Given $\theta^0$, $\epsilon$, $L$, $\mathcal{L}$, $M$:
for $m = 1$ to $M$ do
    Sample $r^0 \sim \mathcal{N}(0, I)$.
    Set $\theta^m \leftarrow \theta^{m-1}, \tilde{\theta} \leftarrow \theta^{m-1}, \tilde{r} \leftarrow r^0$.
    for $i = 1$ to $L$ do
        Set $\tilde{\theta}, \tilde{r} \leftarrow \text{Leapfrog}(\tilde{\theta}, \tilde{r}, \epsilon)$.
    end for
    With probability $\alpha = \min\left\{1, \frac{\exp\{\mathcal{L}(\tilde{\theta}) - \frac{1}{2}\tilde{r}\cdot\tilde{r}\}}{\exp\{\mathcal{L}(\theta^{m-1}) - \frac{1}{2}r^0\cdot r^0\}}\right\}$, set $\theta^m \leftarrow \tilde{\theta}$, $r^m \leftarrow -\tilde{r}$.
end for

function Leapfrog($\theta, r, \epsilon$)
Set $\tilde{r} \leftarrow r + (\epsilon/2)\nabla_\theta \mathcal{L}(\theta)$.
Set $\tilde{\theta} \leftarrow \theta + \epsilon\tilde{r}$.
Set $\tilde{r} \leftarrow \tilde{r} + (\epsilon/2)\nabla_\theta \mathcal{L}(\tilde{\theta})$.
return $\tilde{\theta}, \tilde{r}$.
```

Problem: Need to choose # leapfrog steps

A trajectory for a two-dimensional Gaussian distribution, simulated using 25 leapfrog steps with a step-size of 0.25. The ellipses plotted are one standard deviation from the means. The initial state had $q = [-1.50, -1.55]^T$ and $p = [-1, 1]^T$.



Notice that this trajectory does not resemble a random walk. Instead, starting from the lower left-hand corner, the position variables systematically move upward and to the right, until they reach the upper right-hand corner, at which point the direction of motion is reversed. The consistency of this motion results from the role of the momentum variables.

Components of No U-Turn Sampler

Combines many MCMC components that we have explicitly covered (or covered in readings)

- Gibbs sampler
- Slice Sampler (also involves Gibbs updates)
- Metropolis
- HMC simulation via leapfrog integrator

No U-Turn Sampler : In a NUTShell

Solves 2 problems with HMC

1. Automatically select number of leapfrog steps L
2. Avoid U-turn phenomenon (by selecting L)

Approach

- Simulate backwards-and-forward random number of steps
- Step simulation if a U-turn is happening
- Do extra technical stuff to ensure detailed balance satisfied

On to the technical bits!

Figuring out a good L is hard…

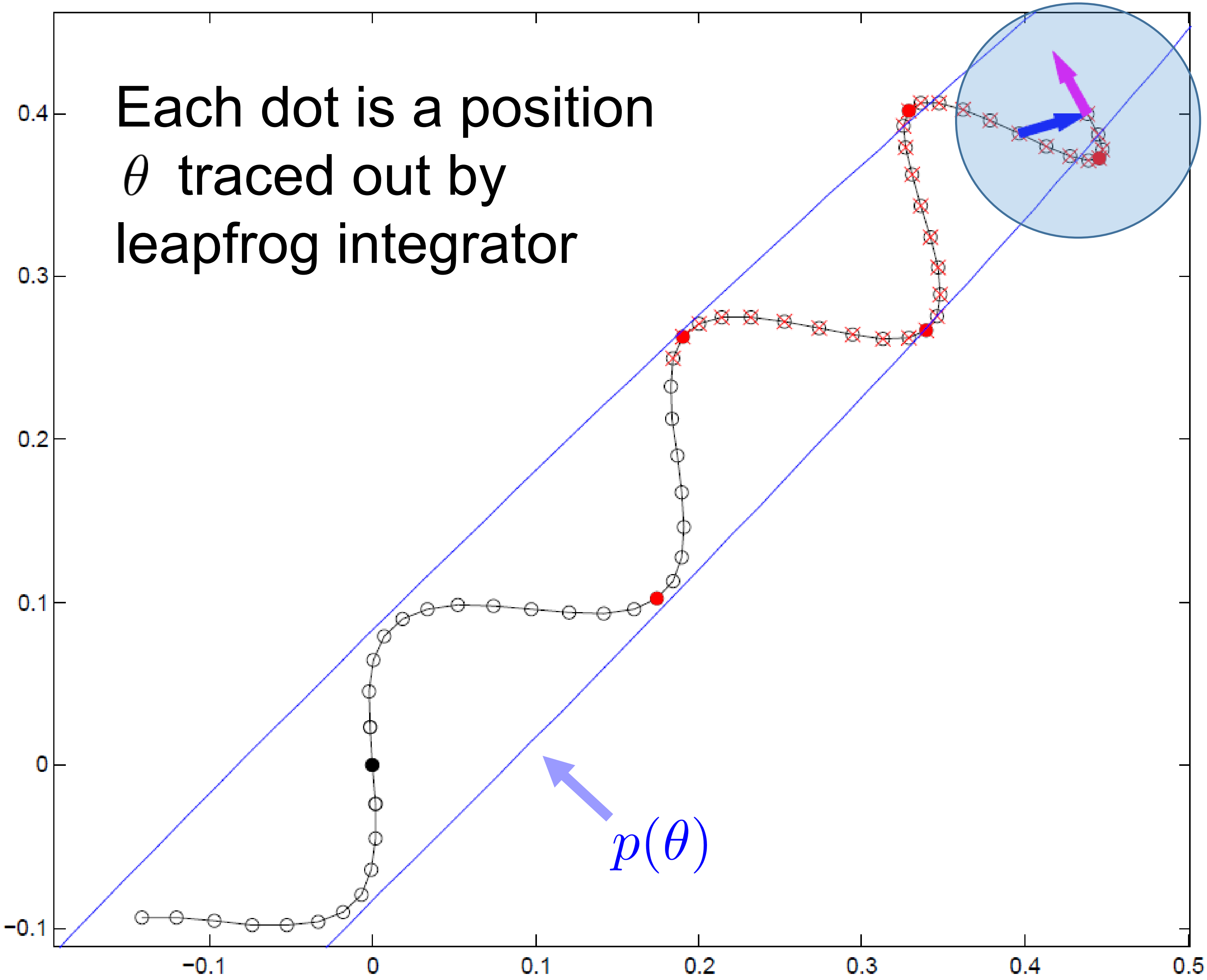
- Need to figure out if simulation is too long, too short, or "just right"
- Typically need to rely on heuristics
- Need a useful criterion to tell if simulation is "long enough"

Let θ be initial value of simulator and $\widetilde{\theta}$ eventual proposal with momentum $\widetilde{r}$ then:

$$\frac{d}{dt}\frac{(\tilde{\theta}-\theta)\cdot(\tilde{\theta}-\theta)}{2} = (\tilde{\theta}-\theta)\cdot\frac{d}{dt}(\tilde{\theta}-\theta) = (\tilde{\theta}-\theta)\cdot\tilde{r}.$$

is proportional to progress we *would make* if we continue to run simulator.

- Less than 0 means we have a U-turn


Each dot is a position θ traced out by leapfrog integrator

$p(\theta)$

$$(\theta - \widetilde{\theta}) \cdot \widetilde{r} < 0$$

Idea Simulate HMC until we hit a U-turn then stop

Problem This naïve approach violates time reversibility and detailed balance!

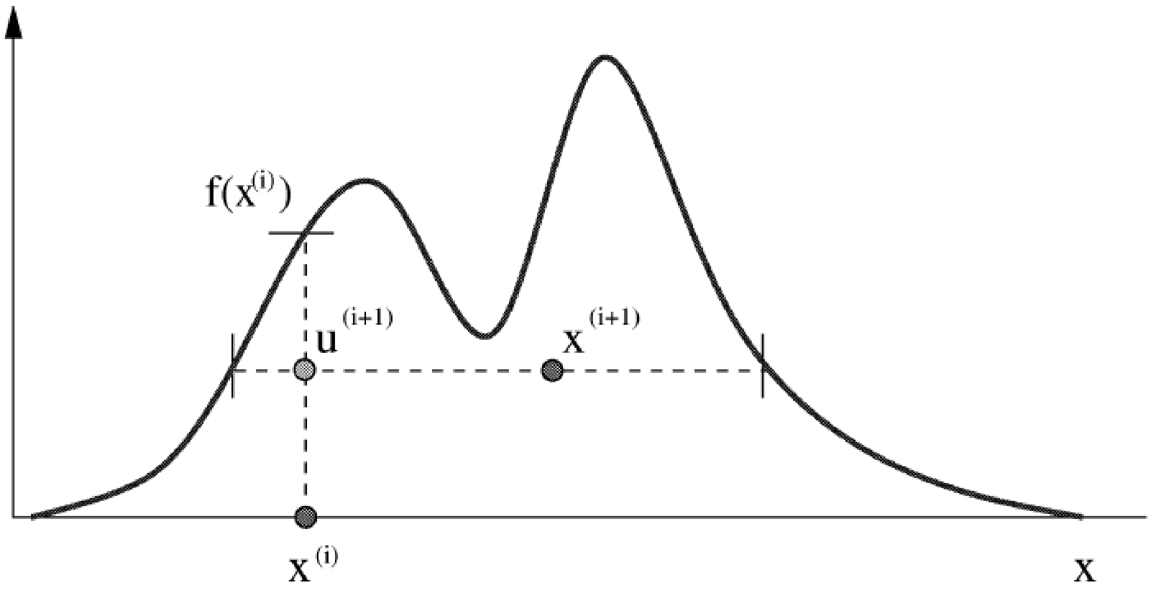
Approach Simulate HMC forward-and-backward and ensure detailed balance holds

Slice Sampler

Target to sample: $p(x) \propto f(x)$

Augment with $u \in \mathbb{R}$ as:

$$p^\star(x,u) = \begin{cases} 1 & \text{if } 0 \le u \le p(x) \\ 0 & \text{otherwise.} \end{cases}$$



Note that marginal is unchanged:

$$\int p^\star(x,u)du = \int_0^{p(x)} du = p(x)$$

Can do this with unnormalized f(x)

So sample from new target $p^*(x,u)$ then ignore u for samples x:

$$u^{(i+1)} \mid x^i \sim \text{Uniform}([0, p(x^i)]) \qquad x^{(i+1)} \mid u^{(i+1)} \sim \text{Uniform}(\{x : p(x) \ge u^{(i+1)}\})$$

Samples from conditionals as in a Gibbs sampler

NUTS : Slice Sampler View

Hamiltonian target to sample :

$$p(\theta, r) \propto \exp\left(\mathcal{L}(\theta) - \frac{1}{2} r \cdot r\right)$$

Augment with slice variable $u \in \mathbb{R}$ to yield new target:

$$p(\theta, r, u) \propto \mathbb{I}[u \in [0, \exp\{\mathcal{L}(\theta) - \tfrac{1}{2} r \cdot r\}]]$$

Slice sampling from each of the conditionals (both Uniform):

$$u \mid \theta, r \sim \text{Uniform}([0, \exp\{\mathcal{L}(\theta) - \frac{1}{2} r \cdot r\}])$$

$$\theta, r \mid u \sim \text{Uniform}(\{\theta, r : u \leq \exp(\mathcal{L}(\theta) - \frac{1}{2} r \cdot r)\})$$

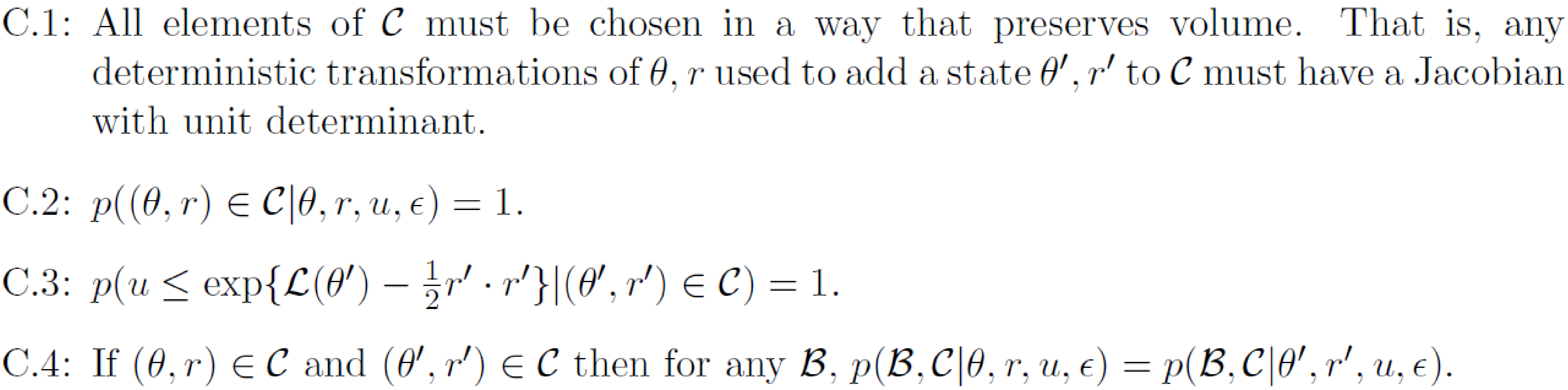
How do we sample this?

Simulate HMC via leapfrog

Some Complications

The previous approach is not guaranteed to satisfy detailed balance…

- Let $\mathcal{B}$ be all position-momentum states generated by leapfrog
- Let $\mathcal{C} \subseteq \mathcal{B}$ be subset of states that ensure detailed balance satisfied
- Sample from new target $p(\theta, r, u, \mathcal{B}, \mathcal{C} \mid \epsilon)$ and ensure:



C.1: All elements of $\mathcal{C}$ must be chosen in a way that preserves volume. That is, any deterministic transformations of θ, r used to add a state θ', r' to $\mathcal{C}$ must have a Jacobian with unit determinant.

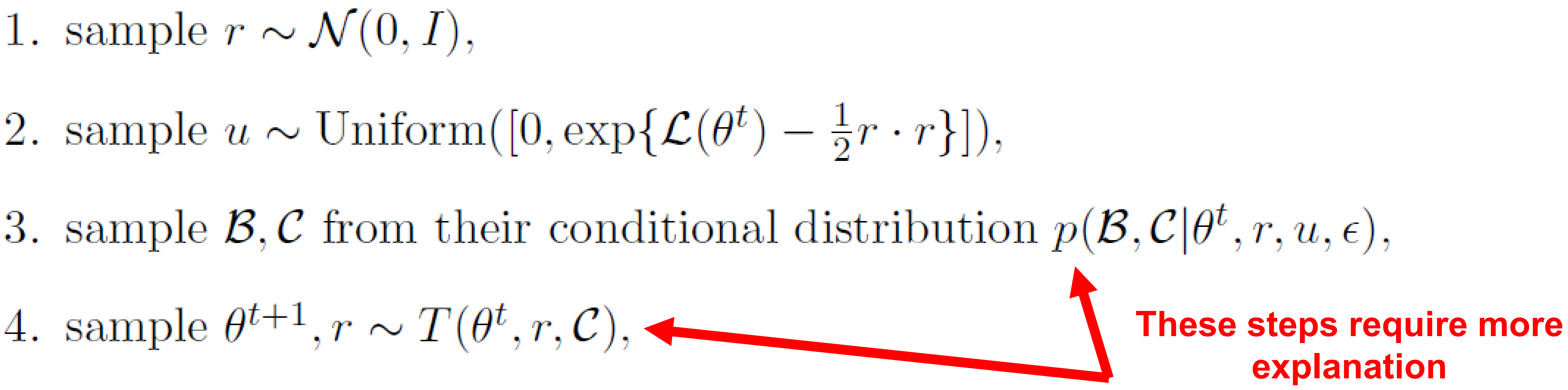
C.2: $p((\theta, r) \in \mathcal{C} | \theta, r, u, \epsilon) = 1$.

C.3: $p(u \leq \exp\{\mathcal{L}(\theta') - \frac{1}{2} r' \cdot r'\} | (\theta', r') \in \mathcal{C}) = 1$.

C.4: If $(\theta, r) \in \mathcal{C}$ and $(\theta', r') \in \mathcal{C}$ then for any $\mathcal{B}$, $p(\mathcal{B}, \mathcal{C} | \theta, r, u, \epsilon) = p(\mathcal{B}, \mathcal{C} | \theta', r', u, \epsilon)$.

The Basic NUTS Algorithm : Skipping Details

Samples from augmented target: $p(\theta, r, u, \mathcal{B}, \mathcal{C} \mid \epsilon)$



1. sample $r \sim \mathcal{N}(0, I)$,
2. sample $u \sim \mathrm{Uniform}([0, \exp\{\mathcal{L}(\theta^t) - \frac{1}{2} r \cdot r\}])$,
3. sample $\mathcal{B}, \mathcal{C}$ from their conditional distribution $p(\mathcal{B}, \mathcal{C} | \theta^t, r, u, \epsilon)$,
4. sample $\theta^{t+1}, r \sim T(\theta^t, r, \mathcal{C})$,

These steps require more explanation

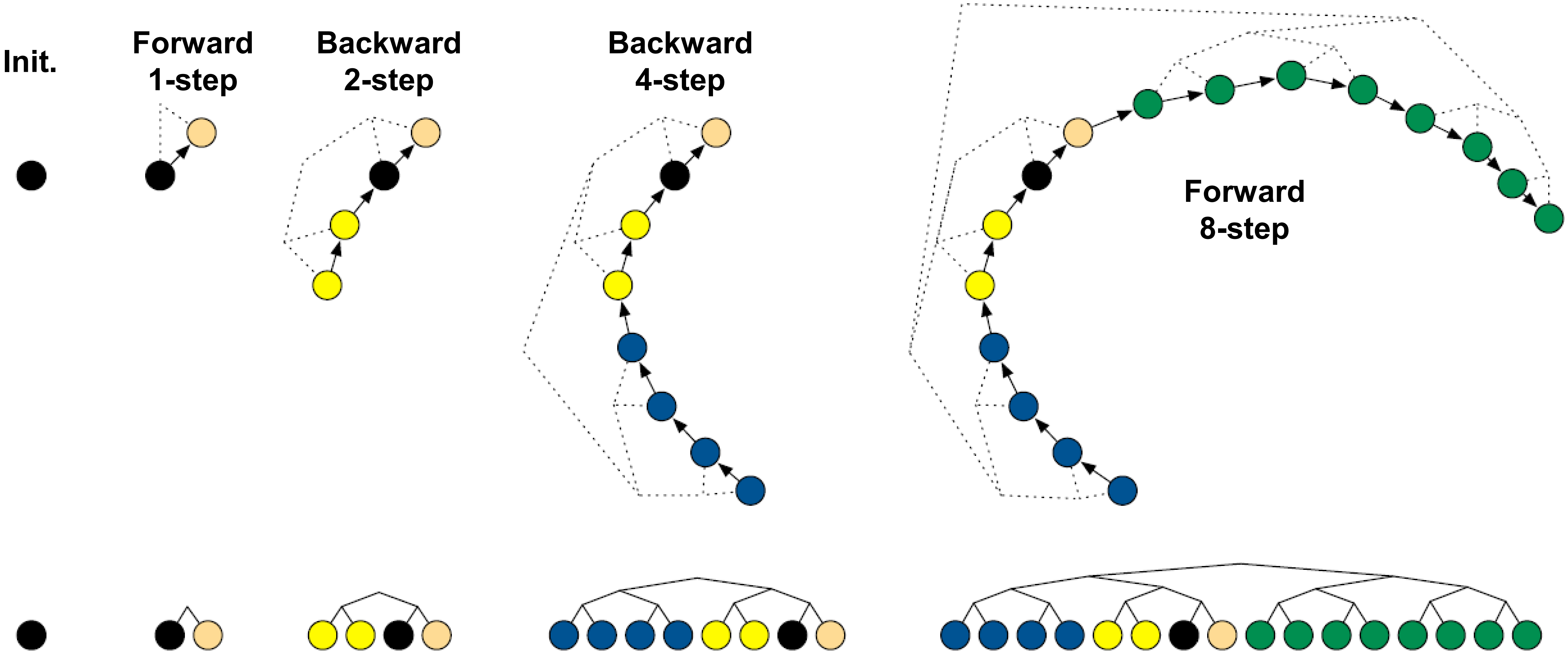
- Steps 1-3 sample $r, u, \mathcal{B}, \mathcal{C}$ conditional on θ^t
- Step 4 samples new $\theta^{t+1} \sim p(\theta \mid \mathcal{B}, \mathcal{C}, u, r, \epsilon)$

NUTS : Step 3

3. sample $\mathcal{B}, \mathcal{C}$ from their conditional distribution $p(\mathcal{B}, \mathcal{C} | \theta^t, r, u, \epsilon)$

- Simulate all points via leapfrog
- Build $\mathcal{B}$ by simulating in, both, forward- and reverse-time
- Use *repeated doubling* method
  - At stage j choose forward (+1) or backward (-1) as : $v_j \sim \mathrm{Uniform}(\{-1, +1\})$
  - Simulate 2^j steps of size $v_j \epsilon$
- Keep doing this until we detect a U-turn (or hit maximum steps)

This builds a balanced binary "tree" of simulations forward- and backward- from an initial point. Better shown by picture…



Binary simulation tree built by *repeated doubling*. At stage j randomly simulate forwards or backwards 2^j leapfrog steps. Note that binary tree is never explicitly represented, only the simulation chain.

NUTS : Step 4

$$4.\ \text{sample } \theta^{t+1}, r \sim T(\theta^t, r, \mathcal{C})$$

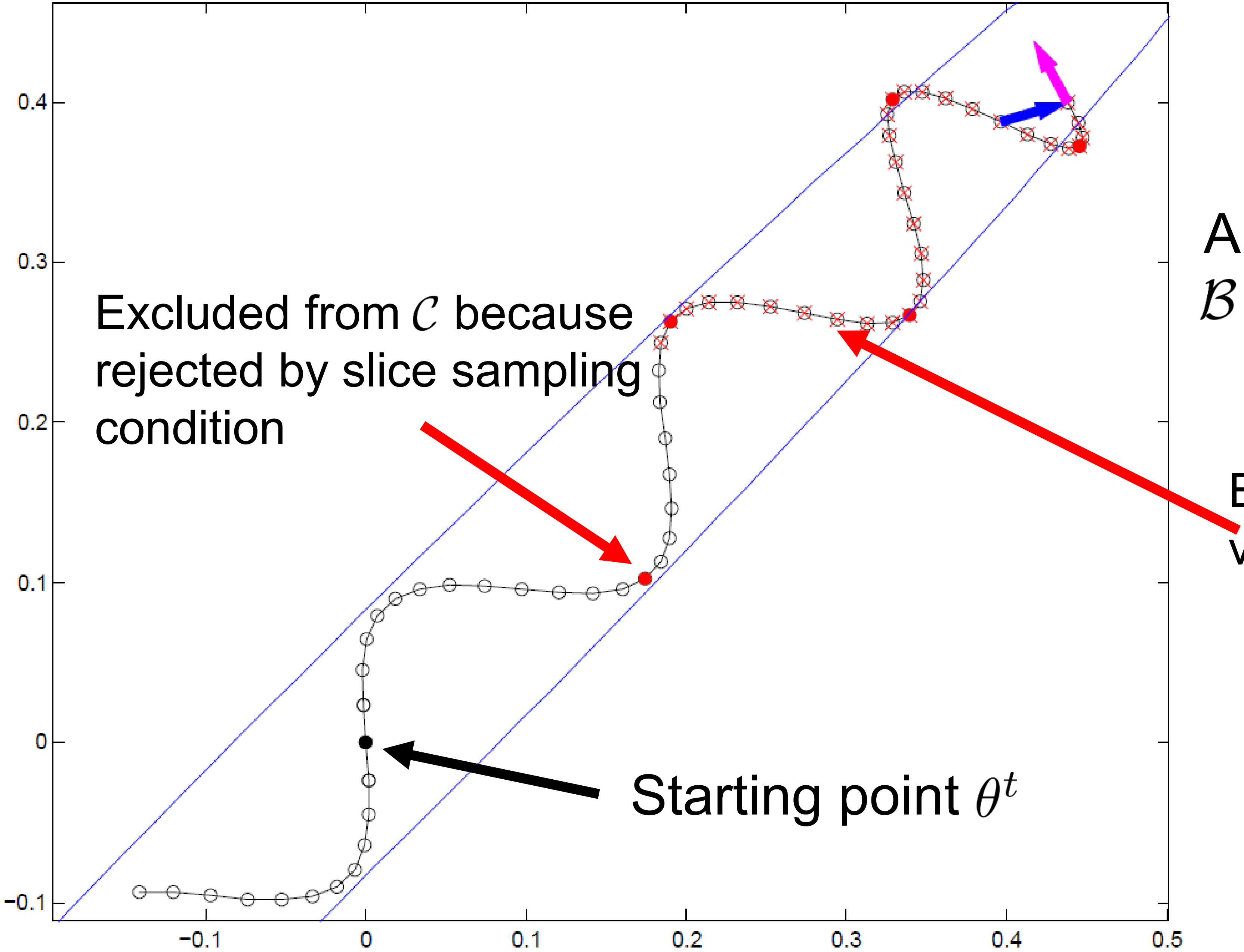
Where T(.) is transition that leaves uniform distribution over $\mathcal{C}$ invariant,

$$\frac{1}{|\mathcal{C}|} \sum_{(\theta,r)\in\mathcal{C}} T(\theta', r' | \theta, r, \mathcal{C}) = \frac{\mathbb{I}[(\theta', r') \in \mathcal{C}]}{|\mathcal{C}|}$$

So, once we figure out position-momentum points in $\mathcal{C}$ then we can choose uniformly among them for position-momentum sample

Step 4 is valid because:

$$\begin{aligned} p(\theta, r | u, \mathcal{B}, \mathcal{C}, \epsilon) &\propto p(\mathcal{B}, \mathcal{C} | \theta, r, u, \epsilon) p(\theta, r | u) && \text{(Bayes' rule + chain rule)} \\ &\propto p(\mathcal{B}, \mathcal{C} | \theta, r, u, \epsilon) \mathbb{I}[u \leq \exp\{\mathcal{L}(\theta) - \tfrac{1}{2} r \cdot r\}] && \text{(Condition C.1)} \\ &\propto \mathbb{I}[(\theta, r) \in \mathcal{C}]. && \text{(Condition C.2 and C.4)} \end{aligned}$$

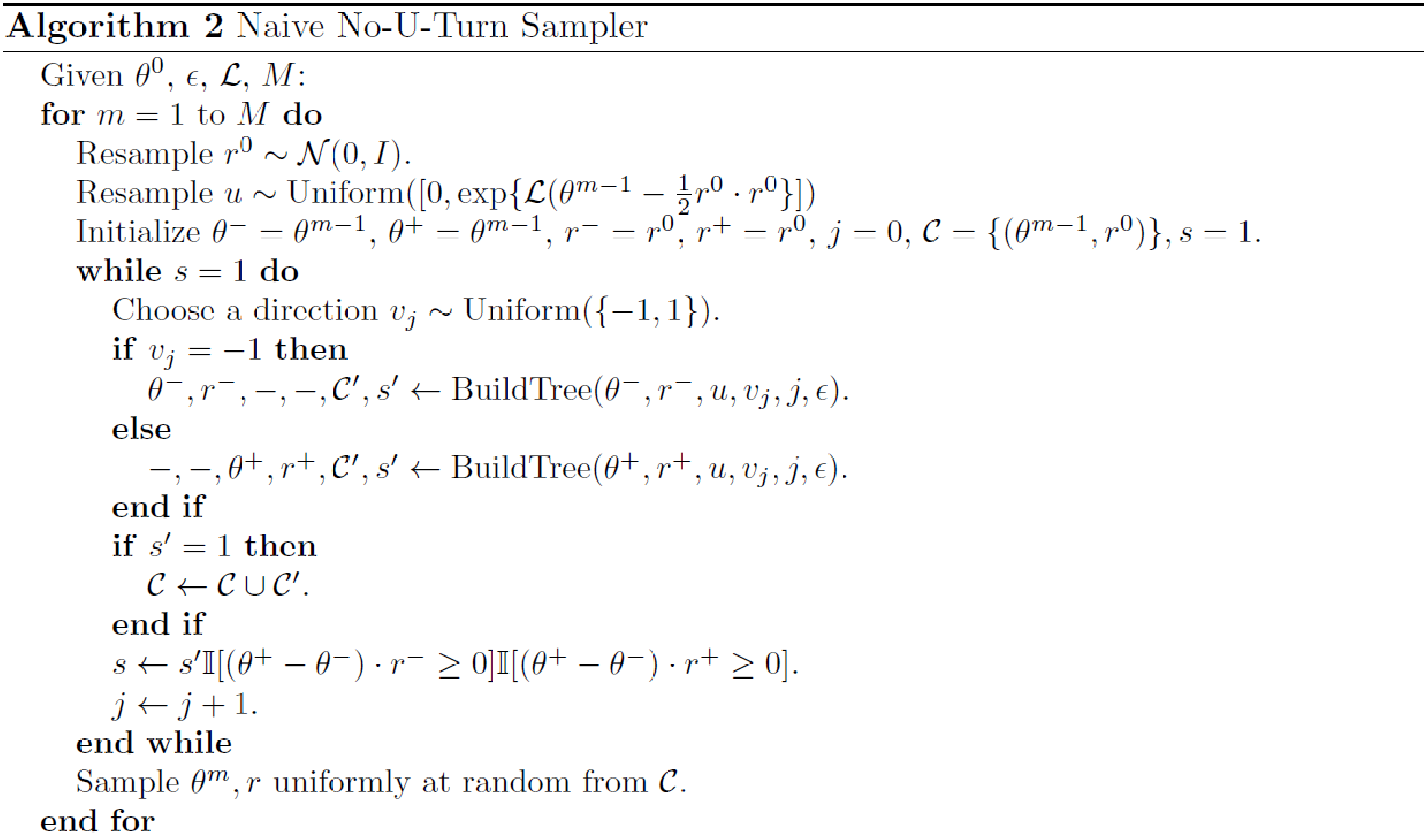


All points belong to set $\mathcal{B}$ of HMC simulations

Excluded from $\mathcal{C}$ because rejected by slice sampling condition

Excluded from $\mathcal{C}$ because violate detailed balance

Starting point θ^t



Algorithm 2 Naive No-U-Turn Sampler

```
Given $\theta^0$, $\epsilon$, $\mathcal{L}$, $M$:
for $m = 1$ to $M$ do
    Resample $r^0 \sim \mathcal{N}(0, I)$.
    Resample $u \sim \mathrm{Uniform}([0, \exp\{\mathcal{L}(\theta^{m-1} - \frac{1}{2} r^0 \cdot r^0\}])$
    Initialize $\theta^- = \theta^{m-1}$, $\theta^+ = \theta^{m-1}$, $r^- = r^0$, $r^+ = r^0$, $j = 0$, $\mathcal{C} = \{(\theta^{m-1}, r^0)\}$, $s = 1$.
    while $s = 1$ do
        Choose a direction $v_j \sim \mathrm{Uniform}(\{-1, 1\})$.
        if $v_j = -1$ then
            $\theta^-, r^-, -, -, \mathcal{C}', s' \leftarrow \mathrm{BuildTree}(\theta^-, r^-, u, v_j, j, \epsilon)$.
        else
            $-, -, \theta^+, r^+, \mathcal{C}', s' \leftarrow \mathrm{BuildTree}(\theta^+, r^+, u, v_j, j, \epsilon)$.
        end if
        if $s' = 1$ then
            $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}'$.
        end if
        $s \leftarrow s' \mathbb{I}[(\theta^+ - \theta^-) \cdot r^- \geq 0] \mathbb{I}[(\theta^+ - \theta^-) \cdot r^+ \geq 0]$.
        $j \leftarrow j + 1$.
    end while
    Sample $\theta^m, r$ uniformly at random from $\mathcal{C}$.
end for
```

Example : Bayesian Logistic Regression

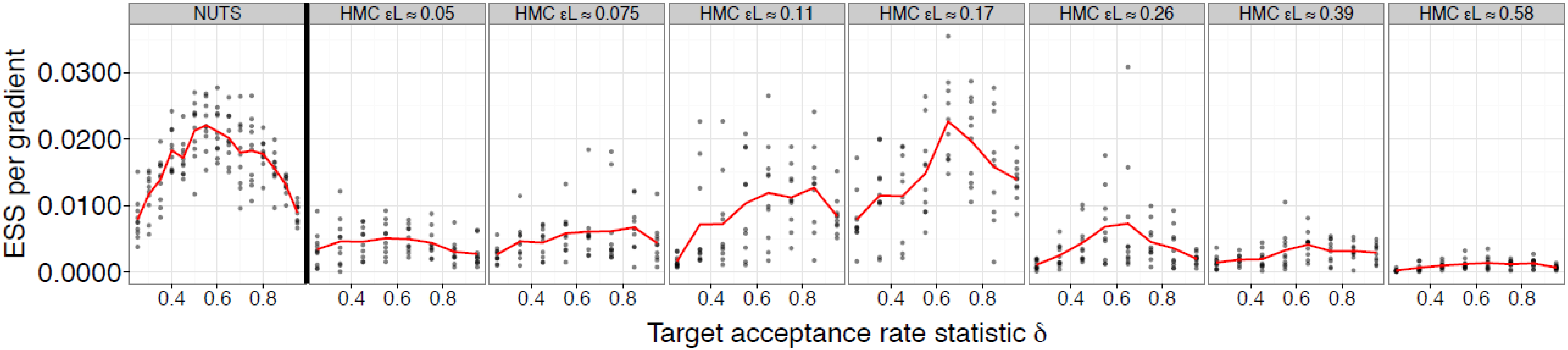
Logistic regression model:

$$\begin{aligned} p(\alpha, \beta | x, y) &\propto p(y | x, \alpha, \beta) p(\alpha) p(\beta) \\ &\propto \exp\{-\sum_i \log(1 + \exp\{-y_i(\alpha + x_i \cdot \beta\}) - \tfrac{1}{2\sigma^2}\alpha^2 - \tfrac{1}{2\sigma^2}\beta \cdot \beta\} \end{aligned}$$

Fit to German credit data from UCI benchmark datasets:

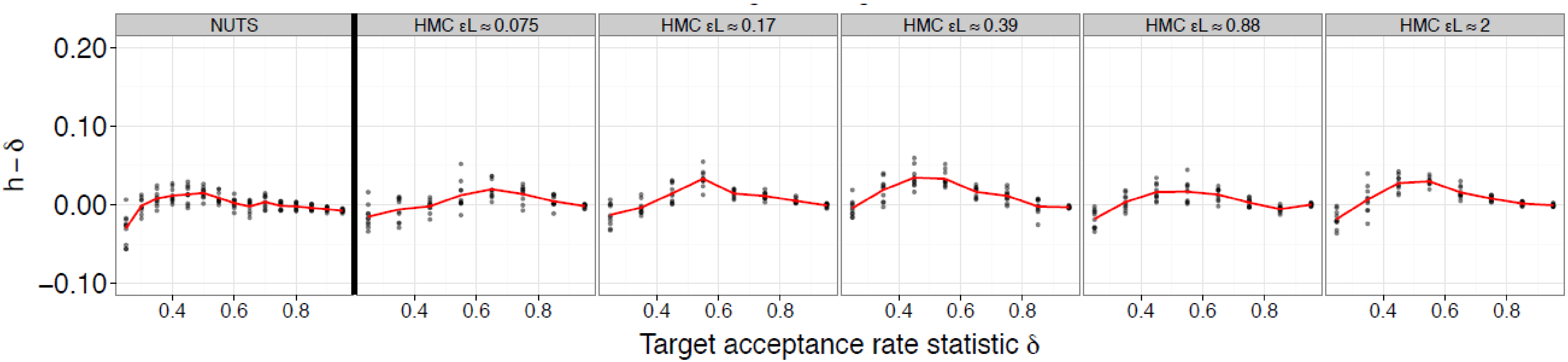
- x_i is 24-dim feature vector of predictors (zero-mean, unit variance)
- Output y_i: denied credit (-1) extended credit (+1)
- 24-dim feature weights β
- Scalar intercept α
- Priors of α and β zero-mean normal w/ independent $\sigma^2 = 100$ variance

Example : Bayesian Logistic Regression



Effective sample size (ESS) as a function of δ and (for HMC) simulation length ϵL for the ~~multivariate normal~~, logistic regression, ~~hierarchical logistic regression~~, ~~and stochastic volatility models~~. Each point shows the ESS divided by the number of gradient evaluations for a separate experiment; lines denote the average of the points' y-values for a particular δ. Leftmost plots are NUTS's performance, other plots shows HMC's performance for various settings of ϵL.

Example : Bayesian Logistic Regression



Discrepancies between the realized average acceptance probability statistic h and its target δ for the ~~multivariate normal~~, logistic regression, ~~hierarchical logistic regression, and stochastic volatility~~ models. Each point's distance from the x-axis shows how effectively the dual averaging algorithm tuned the step size ϵ for a single experiment. Leftmost plots show experiments run with NUTS, other plots show experiments run with HMC with various settings of ϵL.