Theoretical Guarantees for Markov Chain Monte Carlo (MCMC) Algorithms

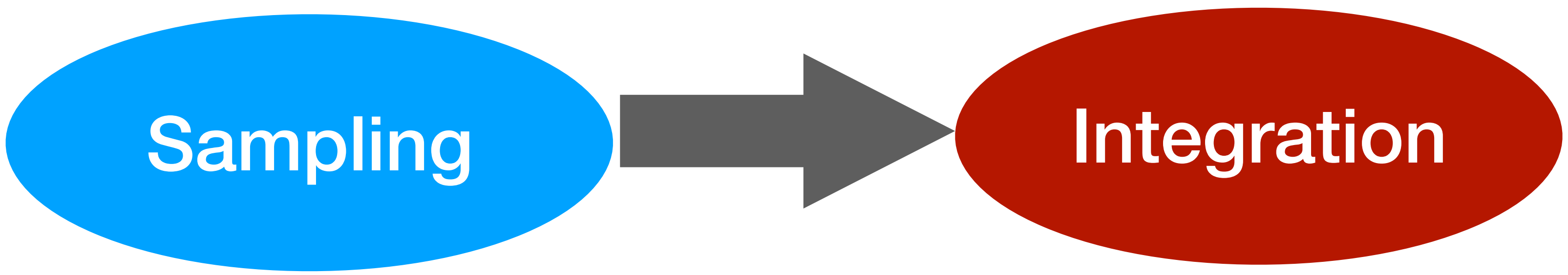
Raaz Dwivedi
Advised by Prof Martin Wainwright and Prof Bin Yu
Department of EECS

Random Sampling

- We consider the problem of drawing random samples from a given density (known up-to proportionality)

$$X_1, X_2, \ldots, X_m \sim \pi$$

Sampling: A fundamental task



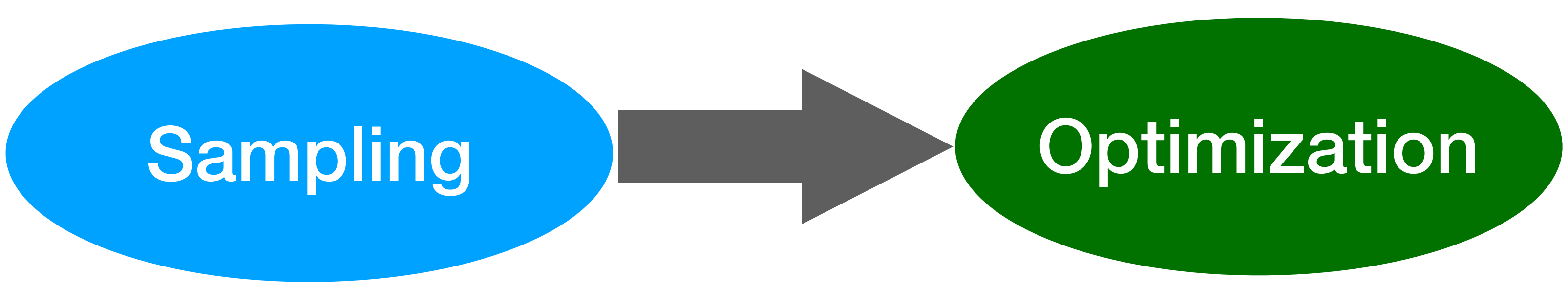
$$\mathbb{E}\left[g(X)\right] = \int g(x)\pi(x)dx \approx \frac{1}{m}\sum_{i=1}^{m} g(X_i)$$

$$X_1, X_2, \ldots, X_m \sim \pi$$

Monte Carlo Approximations

Rare event simulations

Bayesian inference



$$\min_x g(x) \quad \Longleftrightarrow \quad \text{sample from } e^{-g(x)/T}$$

Zeroth order optimization

Escaping saddle points

Simulated annealing

Starting point: The reverse direction! From optimization to sampling

Optimization

- Find the global minimum (or a stationary point)

$$\min_{x \in \mathbb{R}^d} f(x)$$

- Gradient descent:

$$x_{k+1} = x_k - h\nabla f(x_k)$$

- Stochastic Gradient Algorithm:

$$X_{k+1} = X_k - h\nabla f(X_k) + {\color{blue} h\xi_{k+1}}$$

Sampling

- Sampling: draw samples from the density

$$\pi(x) \propto e^{-f(x)}$$

- Unadjusted Langevin algorithm (ULA):

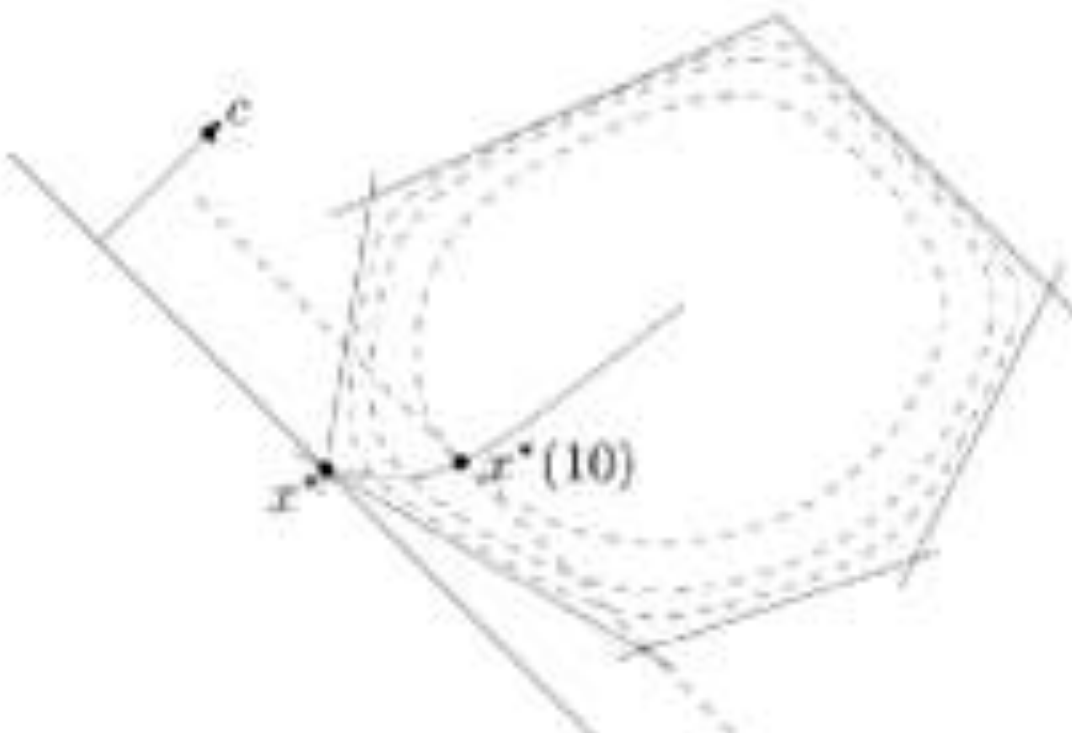
$$X_{k+1} = X_k - h\nabla f(X_k) + {\color{red} \sqrt{2h}\xi_{k+1}}$$

$$\xi_k \overset{i.i.d.}{\sim} \mathcal{N}(0, I_{d\times d})$$

[Parisi 1981, Grenander & Miller 1994, Roberts & Tweedie 1996]

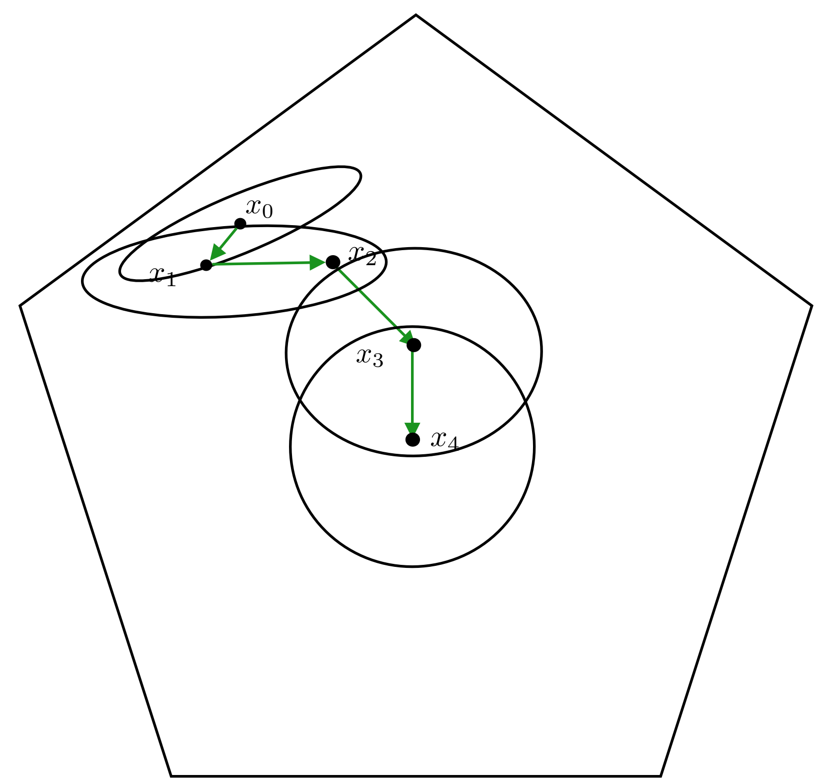
Starting point: The reverse direction! From optimization to sampling

Interior point method for linear programming



[Dikin 1967, Nemivroski 1990]

Sampling from polytopes

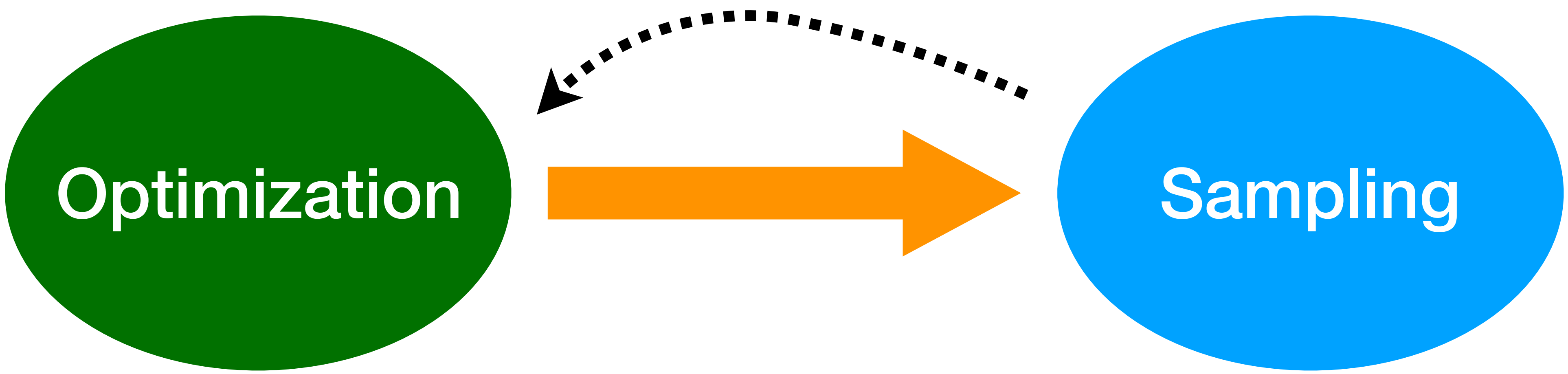


[Kannan and Narayanan 2012]

Motivation for current work: Better understanding of sampling for continuous spaces

- Metropolis Hastings Algorithms [1953, 1970] literature rich with numerous algorithms
- Good understanding for sampling on discrete state space in literature
- Theoretical understanding for sampling from continuous spaces: an active area of research
- Explicit theoretical guarantees gain us
    - Provably correct benchmark for comparison, sometime further insight into the pros and cons of the algorithm,
    - Breadcrumbs for designing better algorithms

Today’s talk:



Optimization subject to linear constraints	Sampling from Polytopes
Convex Optimization	Log-Concave Sampling

Part I: Uniform Sampling on Polytopes

Joint work with Yuansi Chen, Martin Wainwright and Bin Yu

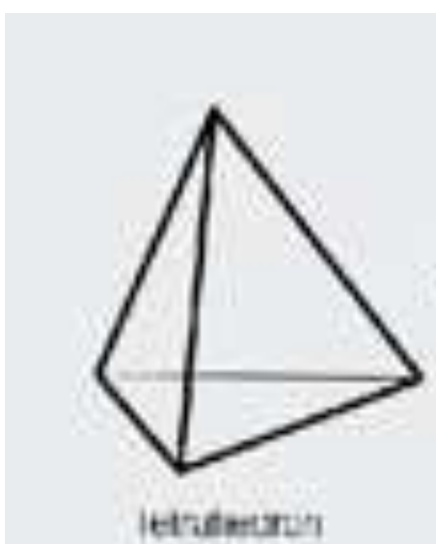
$$\mathcal{X} = \left\{ x \in \mathbb{R}^{\color{red}{d}} \;\middle|\; Ax \leq b \right\}$$

n linear constraints

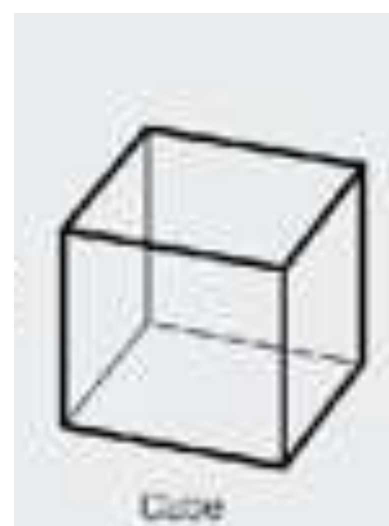
d dimensions

$n > d$

A and b are known



n=4



n=6

n=12

Part I: Uniform Sampling on Polytopes

Joint work with Yuansi Chen, Martin Wainwright and Bin Yu

$$\mathcal{X} = \left\{ x \in \mathbb{R}^{d} \;\middle|\; Ax \leq b \right\}$$

n linear constraints

d dimensions

$n > d$

A and b are known

- Applications in
  - Statistical physics: Hard disk simulations
  - Sampling contingency tables
  - Mixed integer convex programming

Uniform sampling: Existing methods

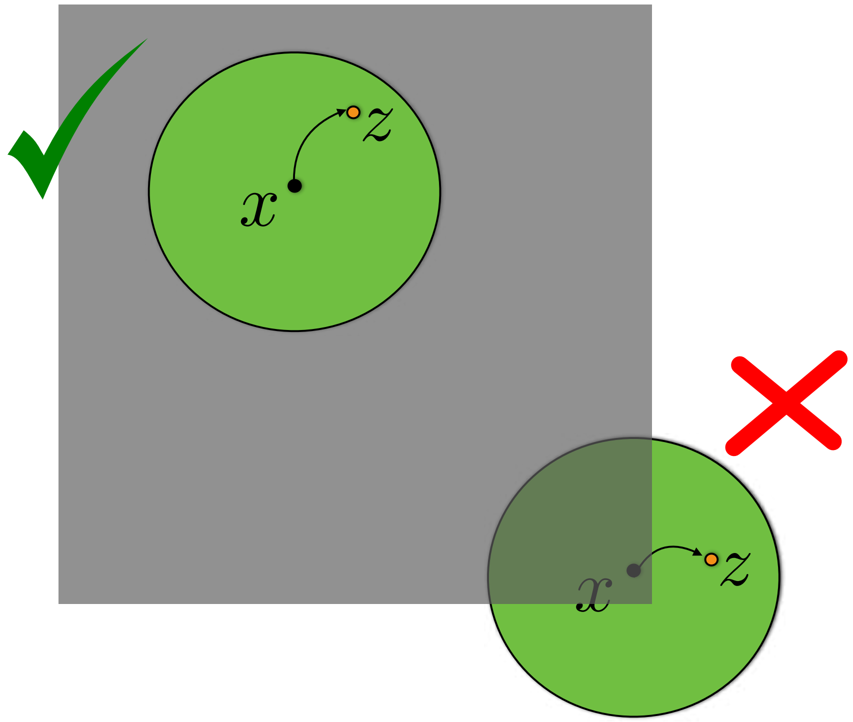
- Sampling on convex sets:
  - **Ball Walk** [Lovász and Simonovits 1990, 1992, 1993]
  - **Hit-and-run** [Berbee et al. 1987, Bélisle et al. 1993, Lovász 1999, Lovász and Vempala 2003, 2004]
- Sampling on polytopes:
  - **Dikin Walk** [Kannan and Narayanan 2012, Narayanan 2015, Sachdeva and Vishnoi 2016]
  - **Geodesic Walk** [Lee and Vempala 2016], **Riemannian Hamiltonian Monte Carlo** [Lee and Vempala 2017]

Ball Walk [Lovász and Simonovits 1990]

- Propose a uniform point in a ball around x

- Reject if outside the polytope, else move to it

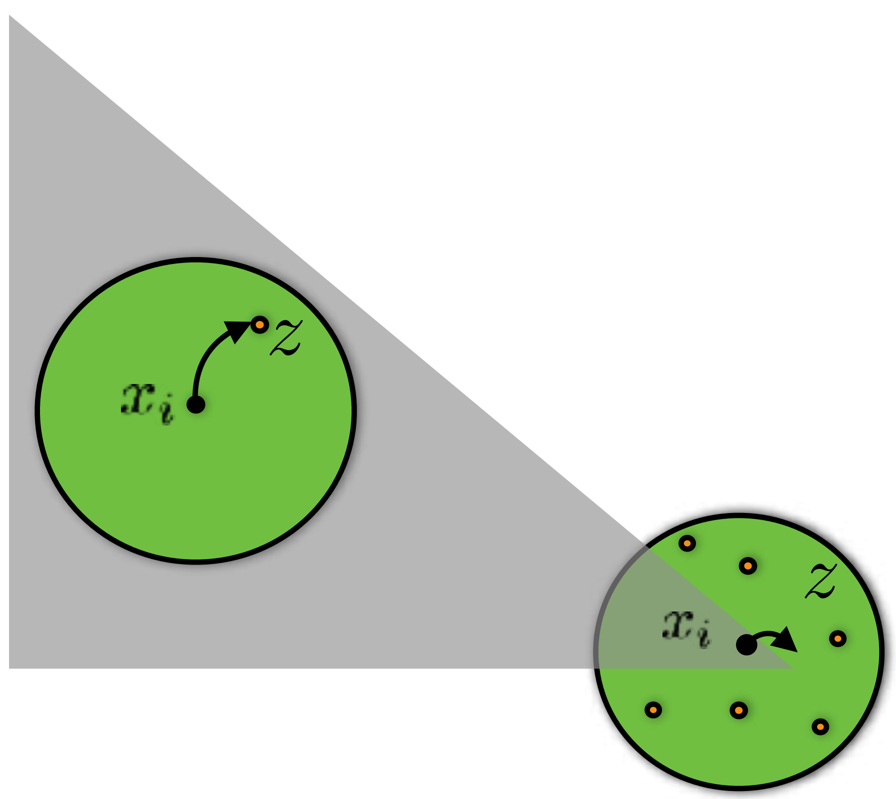
- In case of rejection, define next state as x



$$z \sim \mathrm{Unif}\left[\mathbb{B}\left(x, \frac{c}{\sqrt{d}}\right)\right]$$

Ball walk mixes slowly for sharp sets

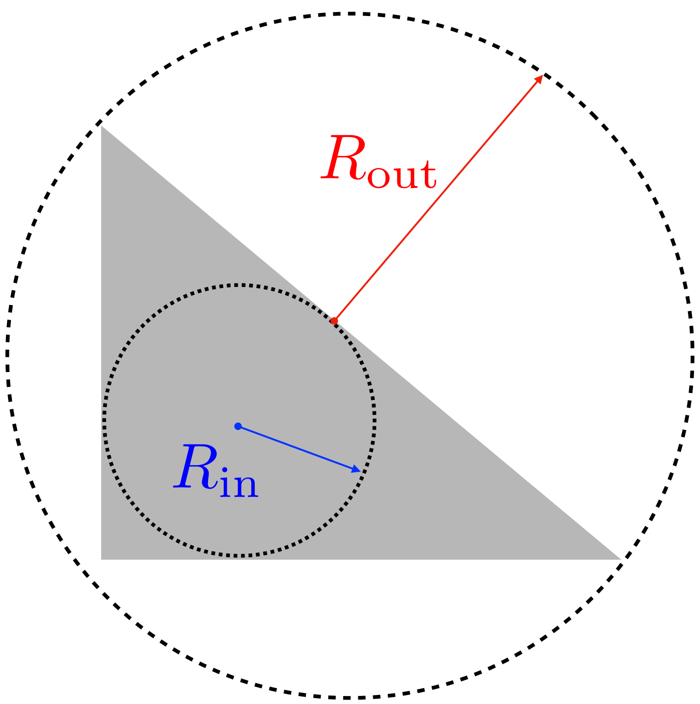
- Many rejections near sharp corners



Ball walk mixes slowly for sharp sets

- Mixing time depends on conditioning of the set

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \delta$$

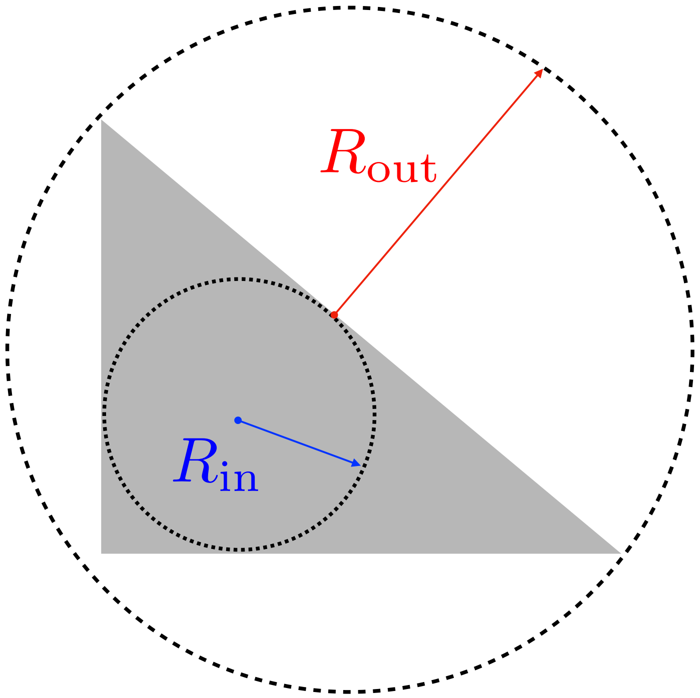


- Number of steps $k \geq \mathcal{O}\left(\frac{d^2}{\delta^2}\ \frac{R_{\mathrm{out}}^2}{R_{\mathrm{in}}^2}\right)$
- Per step cost = $\mathcal{O}\left(nd\right)$

Ball walk mixes slowly for sharp sets

- Mixing time depends on conditioning of the set

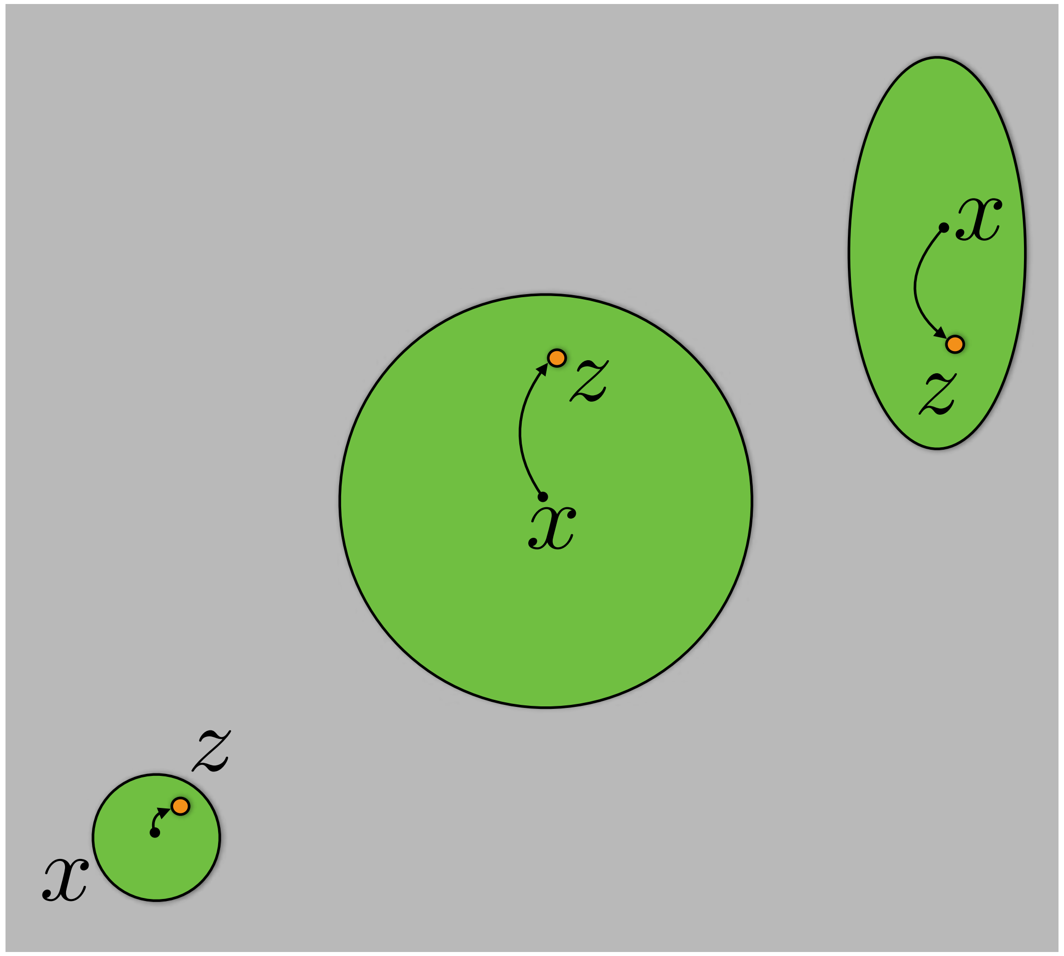
$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \delta$$



- Number of steps $k \geq \mathcal{O}\left(\frac{d^2}{\delta^2} \; \frac{R_{\mathrm{out}}^2}{R_{\mathrm{in}}^2}\right)$
- Per step cost = $\mathcal{O}\left(nd\right)$

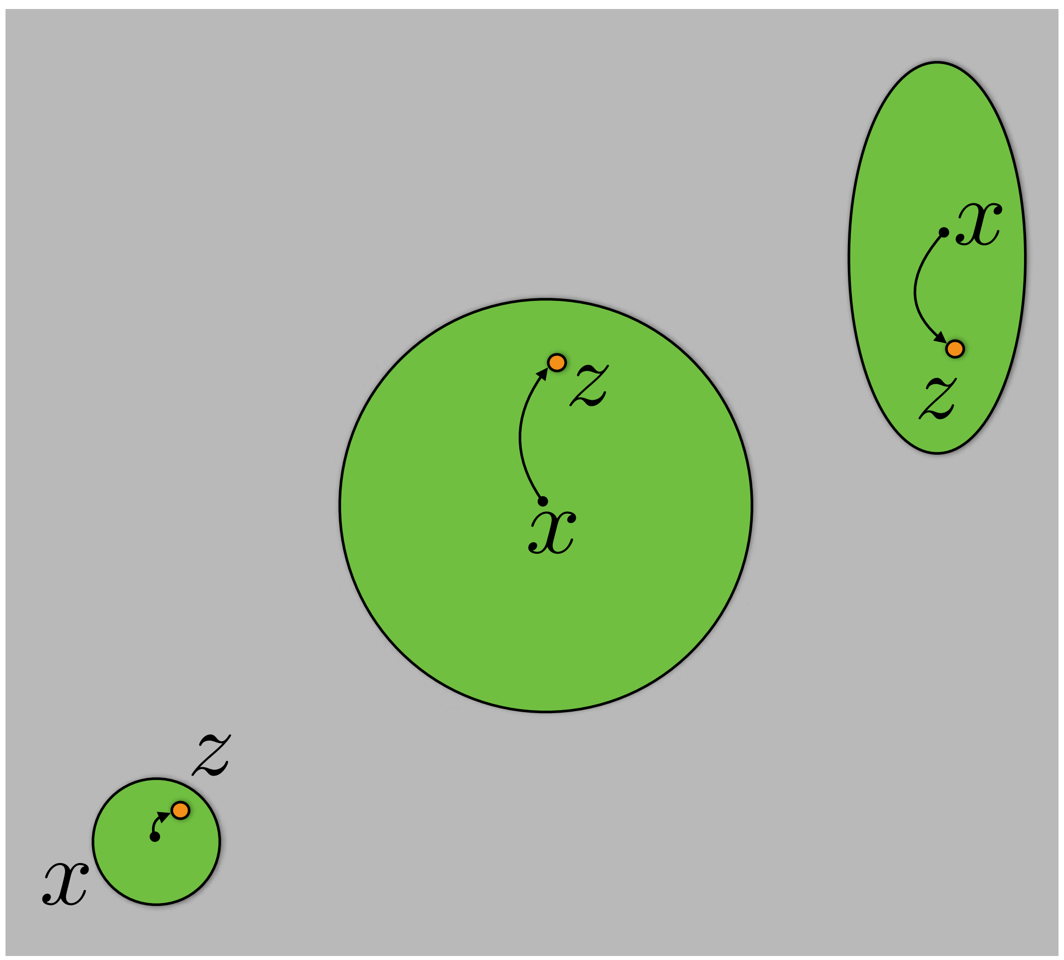
Conditioning ratio:
Unknown
Can be exponential in d

Improving Ball Walk: Adaptive ellipsoids?

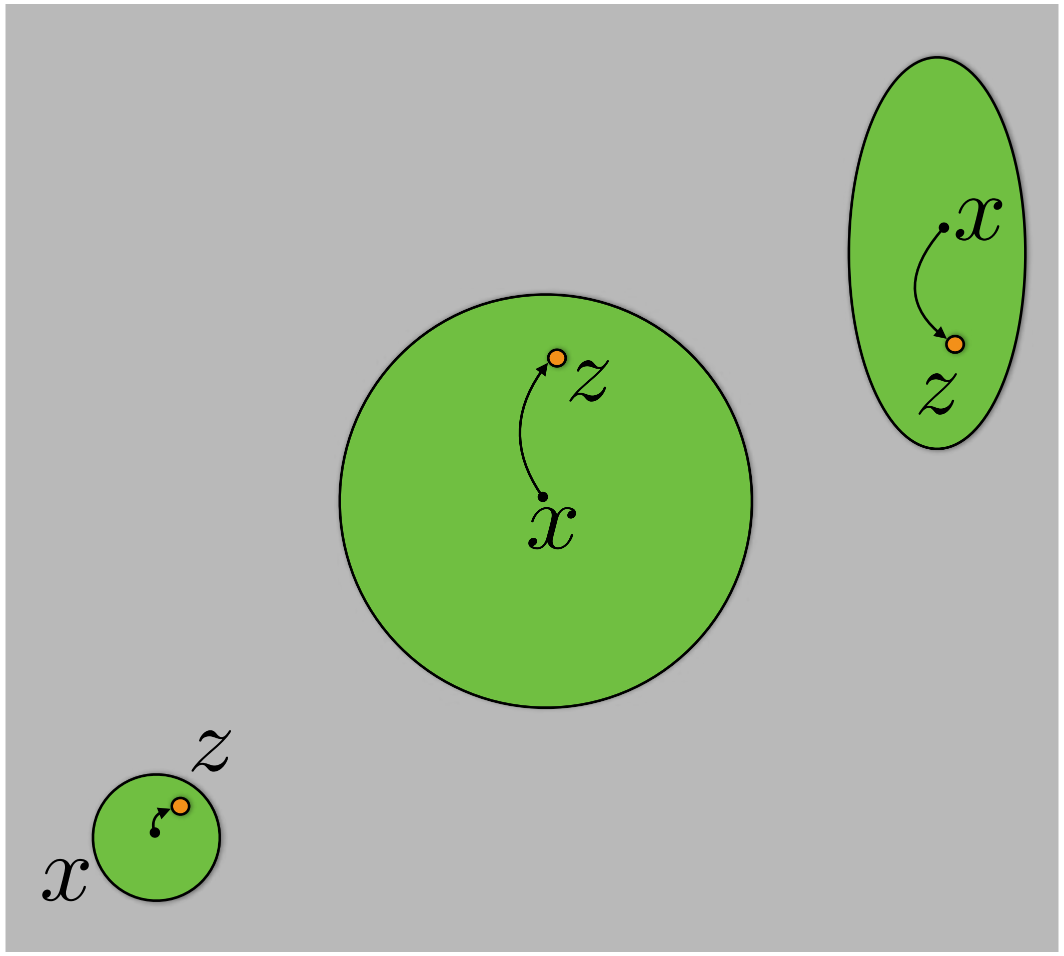


Dikin Walk [Kannan and Narayanan 2012]

- Based on log barrier for polytope used in interior point methods [Dikin 1967, Nemirovski 1990]



Dikin Walk [Kannan and Narayanan 2012]

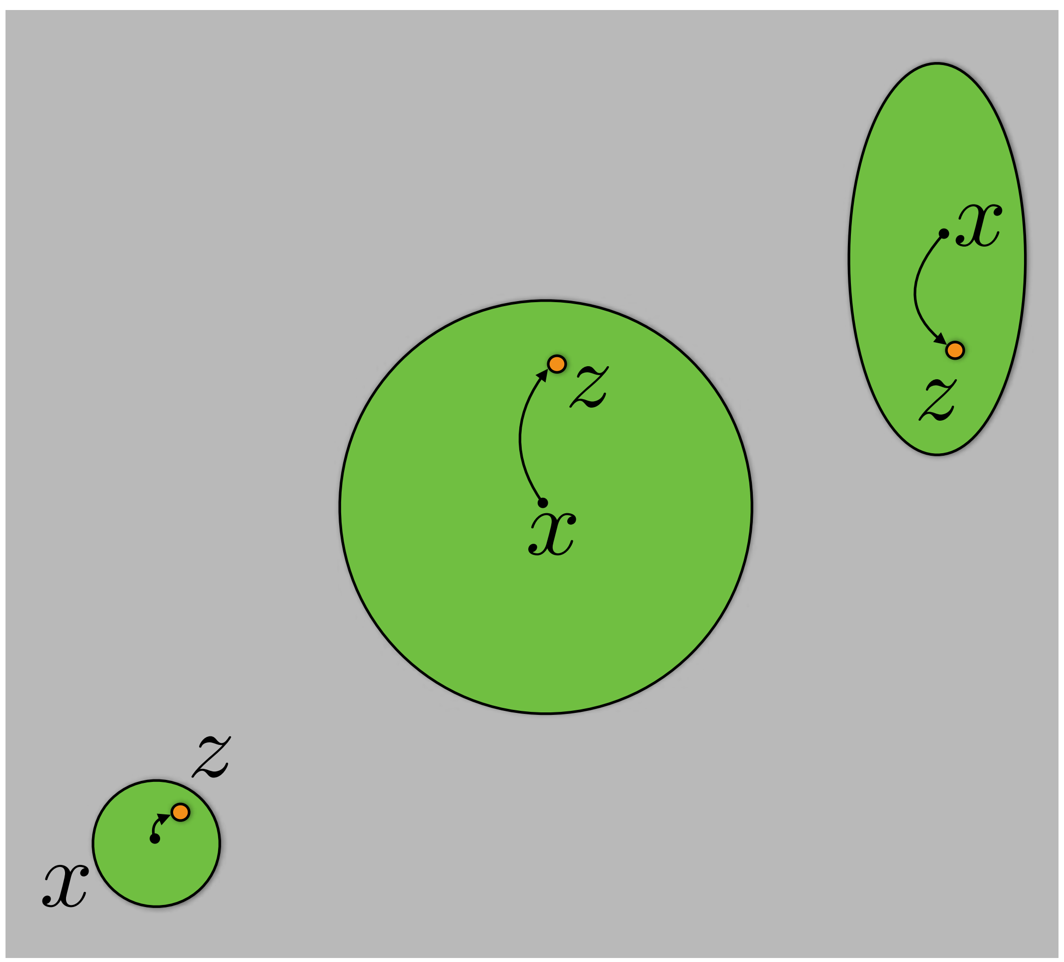


- Propose $z \sim \mathcal{N}\left(x, \frac{1}{d}\mathcal{D}_x^{-1}\right)$
- The inverse covariance defined by the **Hessian of the log barrier**

$$\mathcal{D}_x \propto \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

$$A = \begin{bmatrix} \text{—} a_1^\top \text{—} \\ \text{—} a_2^\top \text{—} \\ \vdots \\ \text{—} a_n^\top \text{—} \end{bmatrix}$$

Dikin Walk [Kannan and Narayanan 2012]



- Propose $z \sim \mathcal{N}\left(x, \frac{1}{d}\mathcal{D}_x^{-1}\right)$
- Reject z if it is outside the set
- Otherwise, accept z with probability

$$P(\text{accept } z) = \min\left\{1, \frac{P(z \to x)}{P(x \to z)}\right\}$$

- In case of rejection, define next state as x

Upper bounds on mixing times

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \delta$$

	Ball Walk	Dikin Walk		
#steps (k)	$\frac{d^2}{\delta^2} \frac{R_{\mathrm{out}}^2}{R_{\mathrm{in}}^2}$	$nd \log \frac{1}{\delta}$		
cost/step	nd	nd^2		

$n = \#\text{linear constraints}$

$d = \#\text{dimensions}$

$n > d$

$\delta = \text{accuracy}$

$\frac{R_{\mathrm{out}}}{R_{\mathrm{in}}} = \text{conditioning}$

Upper bounds on mixing times

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \delta$$

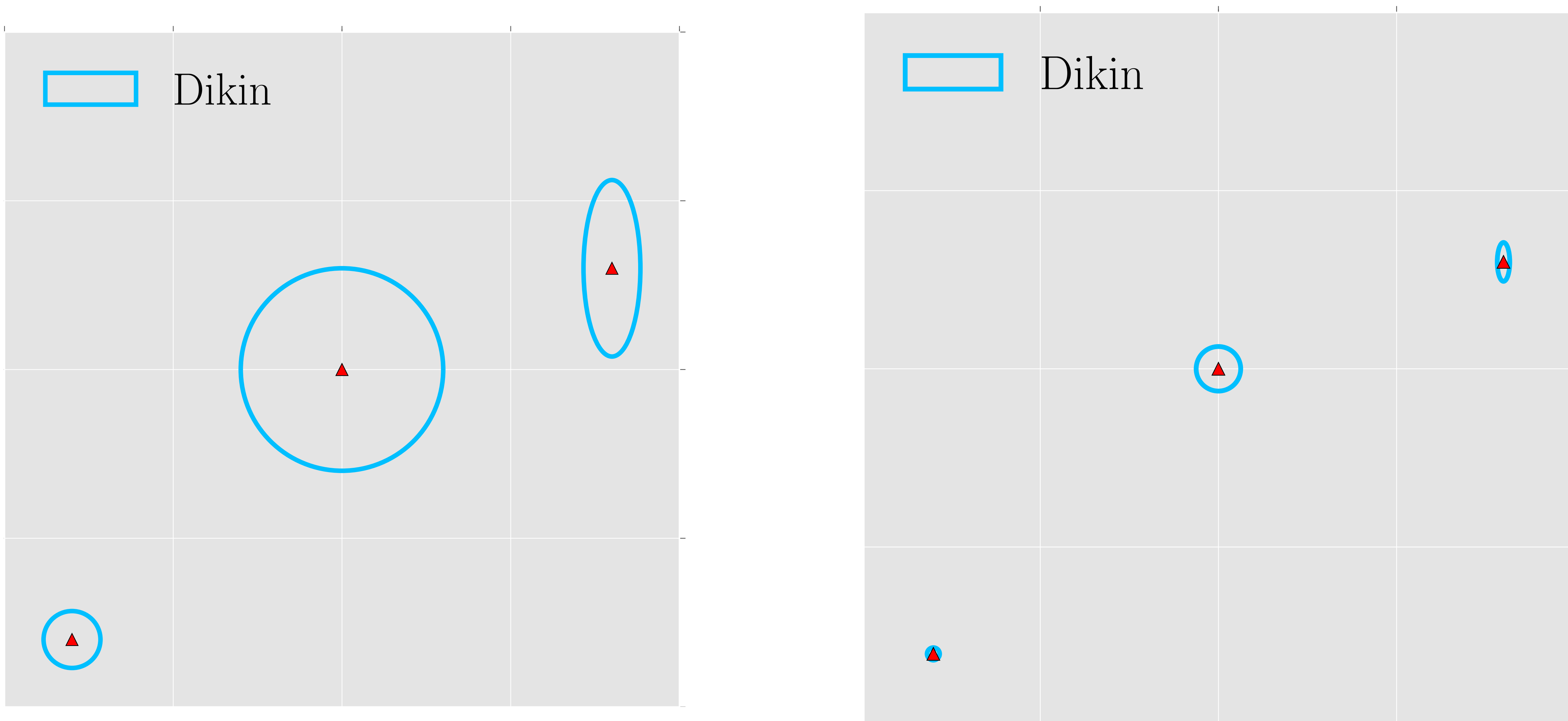
	Ball Walk	Dikin Walk	?	?
#steps (k)	$\frac{d^2}{\delta^2}\frac{R_{\mathrm{out}}^2}{R_{\mathrm{in}}^2}$	$nd\log\frac{1}{\delta}$		
cost/step	nd	nd^2		

What if $n \gg d$?

A closer look at Dikin walk: Proposals shrink with # constraints

Square, **4 constraints**

Square, **overparameterized**



[Similar argument holds even when the set is not overparameterized.]

How to improve the Dikin walk?: Even better ellipsoids?

Put weights on constraints

$$D_x = \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2} \quad \Rightarrow \quad \mathcal{V}_x = \sum_{i=1}^{n} {\color{red} w_i(x)} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

Hessians of weighted barriers in optimization

Our work:
Exploiting improved barriers for sampling

[Kannan and Narayanan 2012]

Dikin Proposal

$$z \sim \mathcal{N}\left(x, \frac{\mathbf{1}}{\mathbf{d}} \mathcal{D}_{\mathbf{x}}^{-\mathbf{1}}\right)$$

$$\mathcal{D}_x \propto \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

[Chen, D., Wainwright and Yu 2017]

Vaidya Proposal

$$z \sim \mathcal{N}\left(x, \frac{1}{\sqrt{nd}} \mathcal{V}_x^{-1}\right)$$

$$\mathcal{V}_x \propto \sum_{i=1}^{n} \left(\sigma_{x,i} + \frac{d}{n}\right) \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

$$\sigma_{x,i} = \frac{a_i^\top \mathcal{D}_x^{-1} a_i}{(b_i - a_i^\top x)^2}$$

Our work:
Exploiting improved barriers for sampling

[Kannan and Narayanan 2012]

Dikin Proposal

$$z \sim \mathcal{N}\left(x, \frac{\mathbf{1}}{\mathbf{d}}\mathcal{D}_{\mathbf{x}}^{-1}\right)$$

$$\mathcal{D}_x \propto \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

[Chen, D., Wainwright and Yu 2017]

Vaidya Proposal

$$z \sim \mathcal{N}\left(x, \frac{1}{\sqrt{nd}}\mathcal{V}_x^{-1}\right)$$

$$\mathcal{V}_x \propto \sum_{i=1}^{n} \left(\sigma_{x,i} + \frac{d}{n}\right) \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

$$\sigma_{x,i} = \frac{a_i^\top \mathcal{D}_x^{-1} a_i}{(b_i - a_i^\top x)^2}$$

Inspiration from Optimization:

Log Barrier Method
[Dikin 1967, Nemirovski 1990]

Volumetric Barrier Method
[Vaidya 1993]

Our work:
Exploiting improved barriers for sampling

[Kannan and Narayanan 2012]

Dikin Proposal

$$z \sim \mathcal{N}\left(x, \frac{\mathbf{1}}{\mathbf{d}} \mathcal{D}_{\mathbf{x}}^{-1}\right)$$

$$\mathcal{D}_x \propto \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

Unit weight, sums to n

[Chen, D., Wainwright and Yu 2017]

Vaidya Proposal

$$z \sim \mathcal{N}\left(x, \frac{1}{\sqrt{nd}} \mathcal{V}_x^{-1}\right)$$

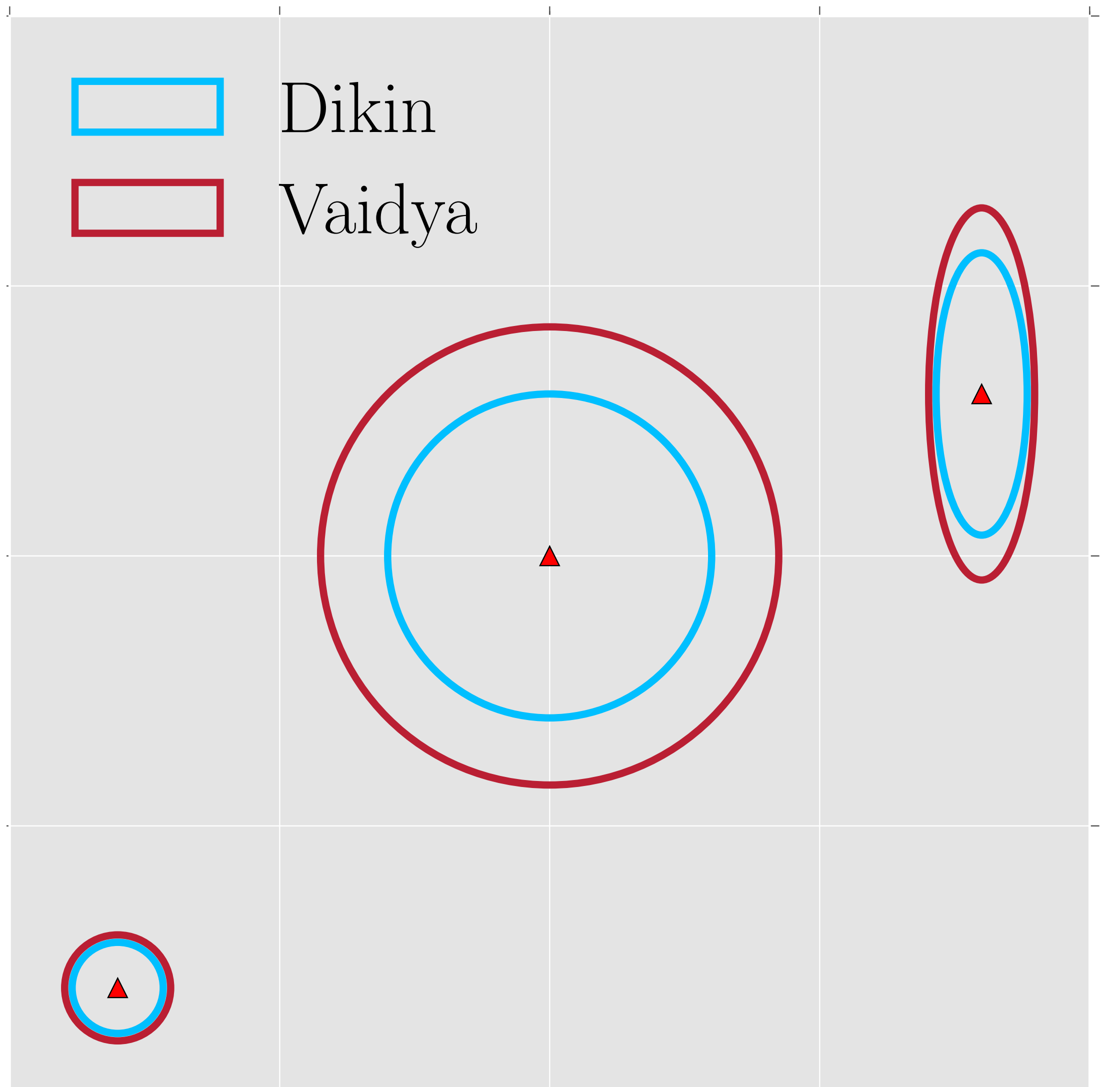
$$\mathcal{V}_x \propto \sum_{i=1}^{n} \left(\sigma_{x,i} + \frac{d}{n}\right) \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

$$\sigma_{x,i} = \frac{a_i^\top \mathcal{D}_x^{-1} a_i}{(b_i - a_i^\top x)^2}$$

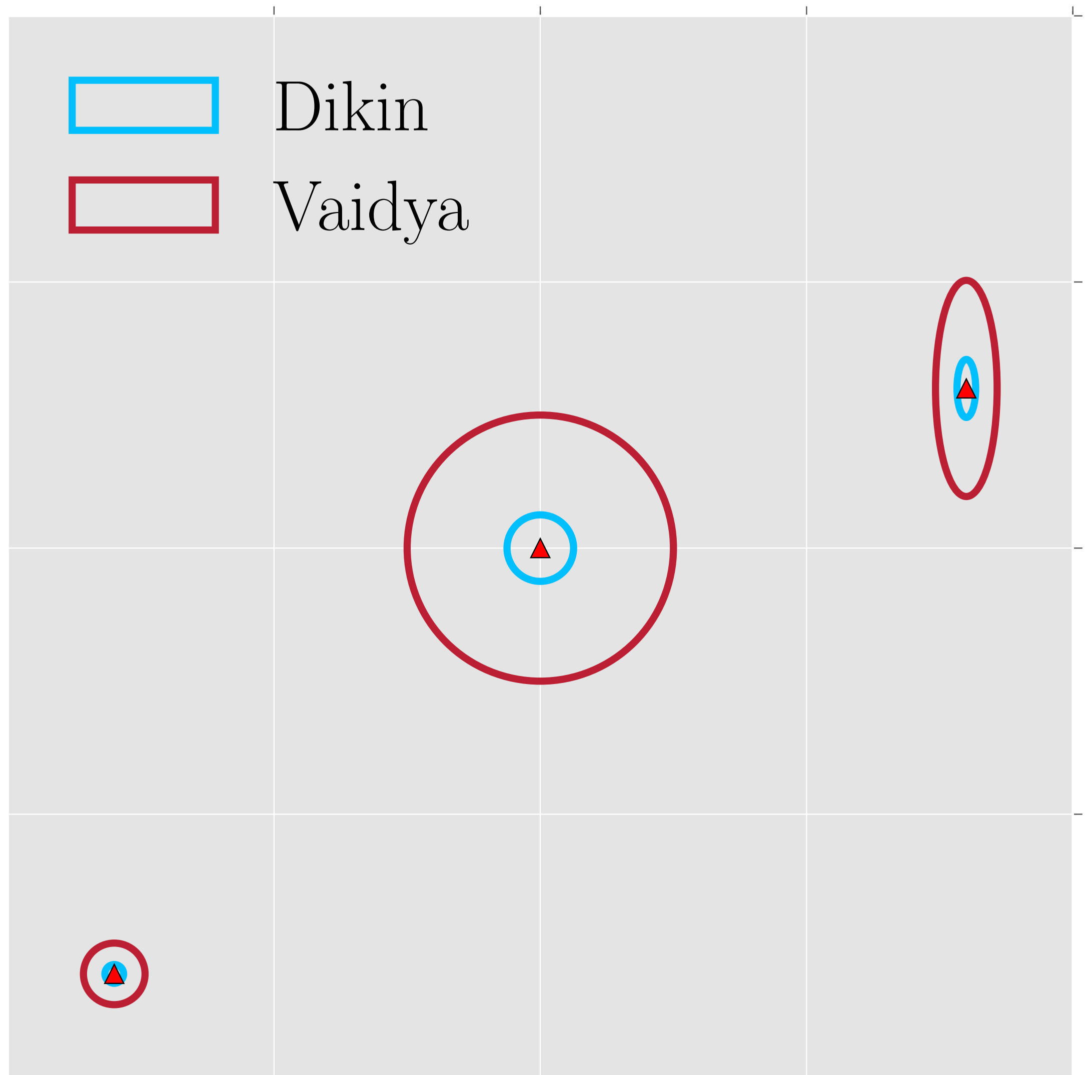
[0, 1] valued, sums to d

Vaidya vs Dikin proposals

Square, **4 constraints**



Square, **overparameterized**



Upper bounds:
Vaidya walk mixes in fewer steps!

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \delta$$

	Ball Walk	Dikin Walk	Vaidya Walk	
#Steps	$\frac{d^2}{\delta^2} \frac{R_{\max}^2}{R_{\min}^2}$	$nd \log \frac{1}{\delta}$	$n^{0.5} d^{1.5} \log \frac{1}{\delta}$	**n constraints d dimensions n > d**
Per Step Cost	nd	nd^2	nd^2	**similar cost/step as Dikin walk**

Upper bounds:
Vaidya walk mixes in fewer steps!

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \delta$$

	Ball Walk	Dikin Walk	Vaidya Walk	
#Steps	$\frac{d^2}{\delta^2} \frac{R_{\max}^2}{R_{\min}^2}$	$nd \log \frac{1}{\delta}$	$n^{0.5} d^{1.5} \log \frac{1}{\delta}$	**n constraints d dimensions n > d**
Per Step Cost	nd	nd^2	nd^2	**similar cost/step as Dikin walk**

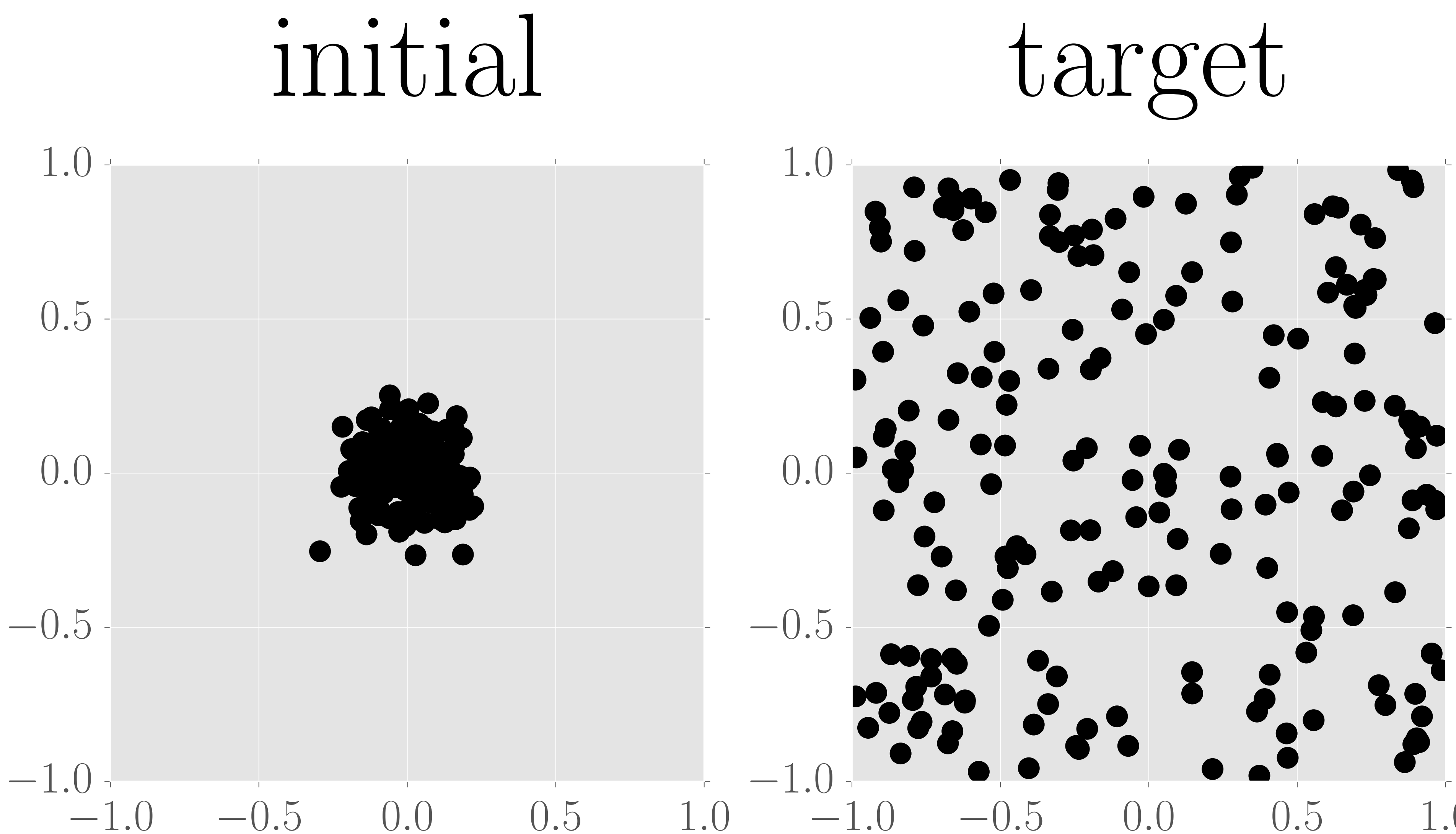
What if $n \gg d$?

Simulation: Dikin Walk vs Vaidya Walk

#dimensions = **2**

k = #iterations

#experiments = **200**



$k = 0$

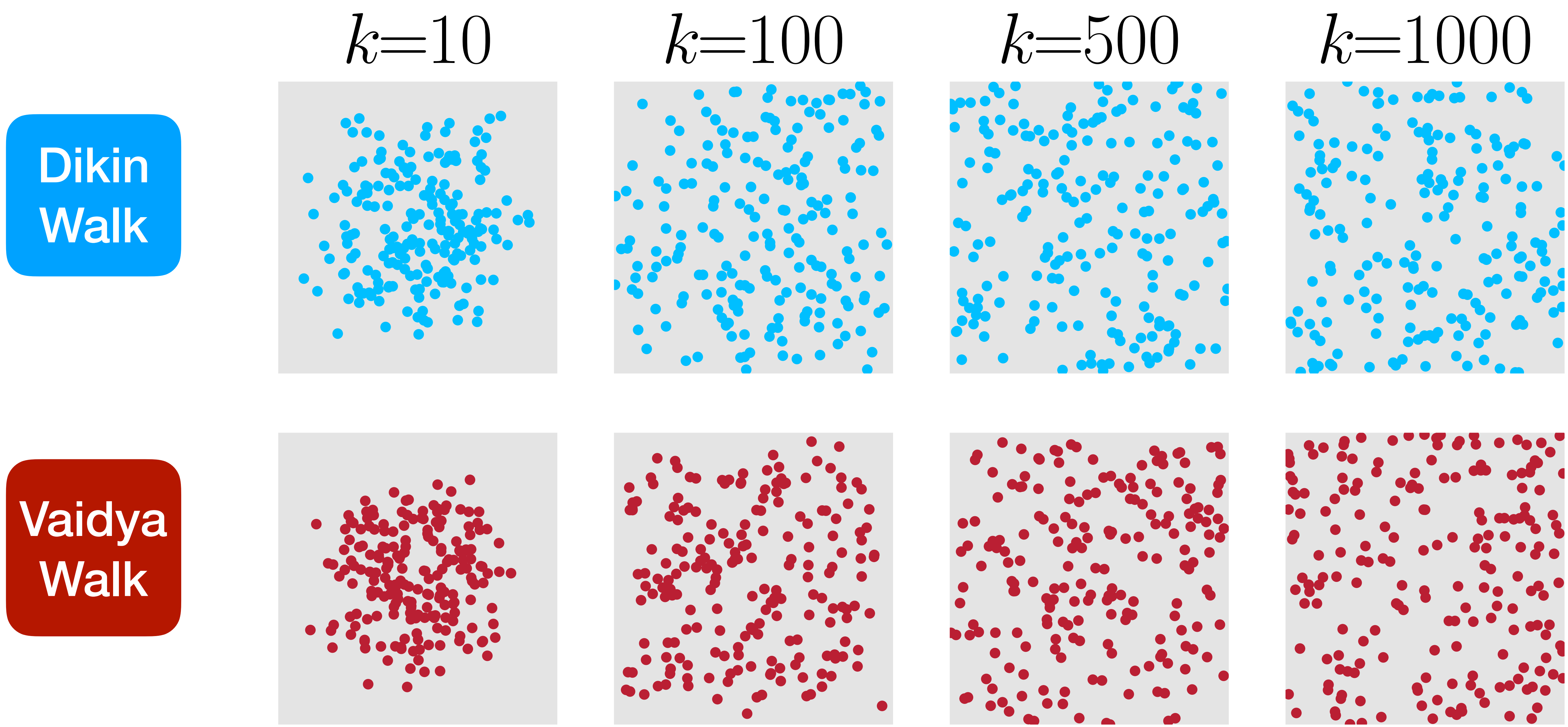
$k = \infty$

Small #constraints: No Winner!

#constraints = **4**

k = #iterations

#experiments = **200**

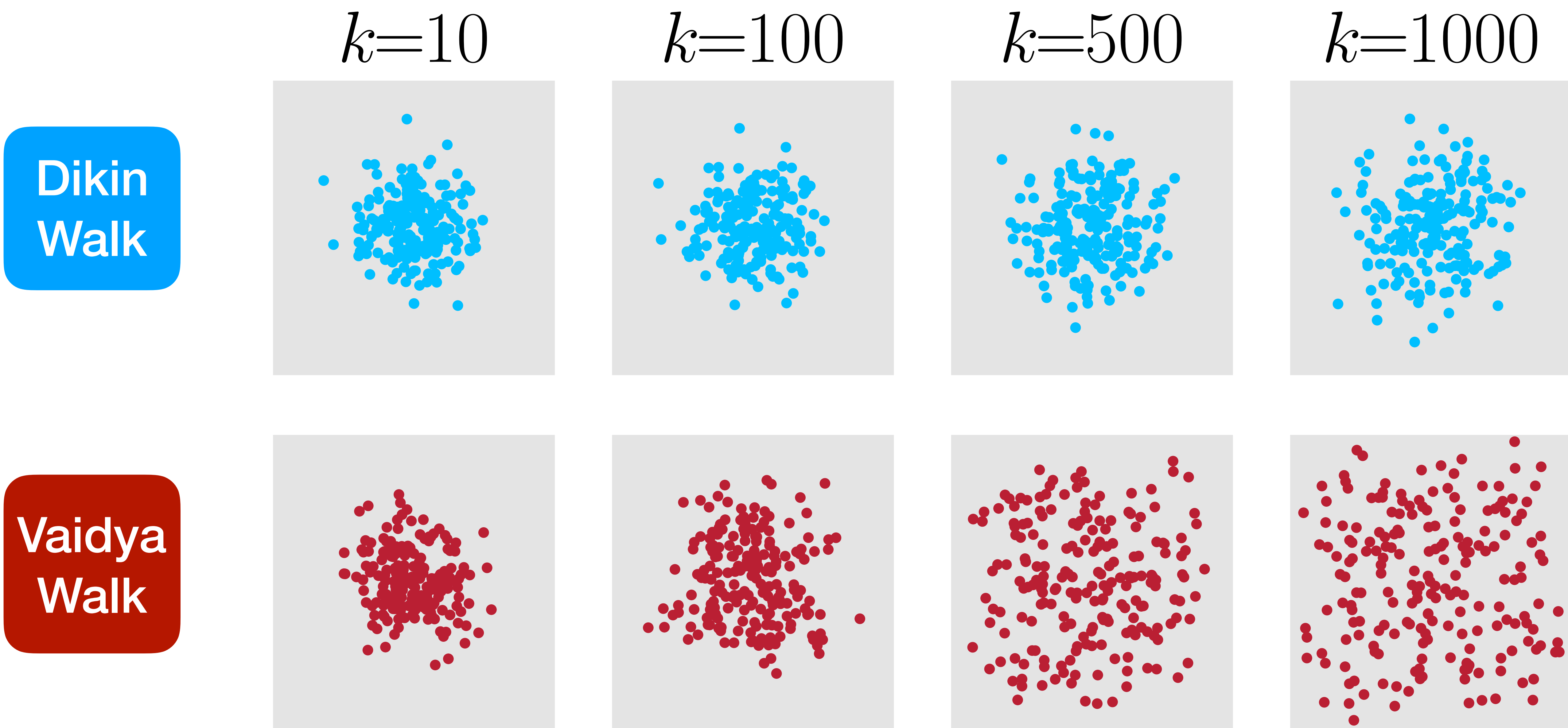


What if $n \gg d$? Vaidya walk wins!

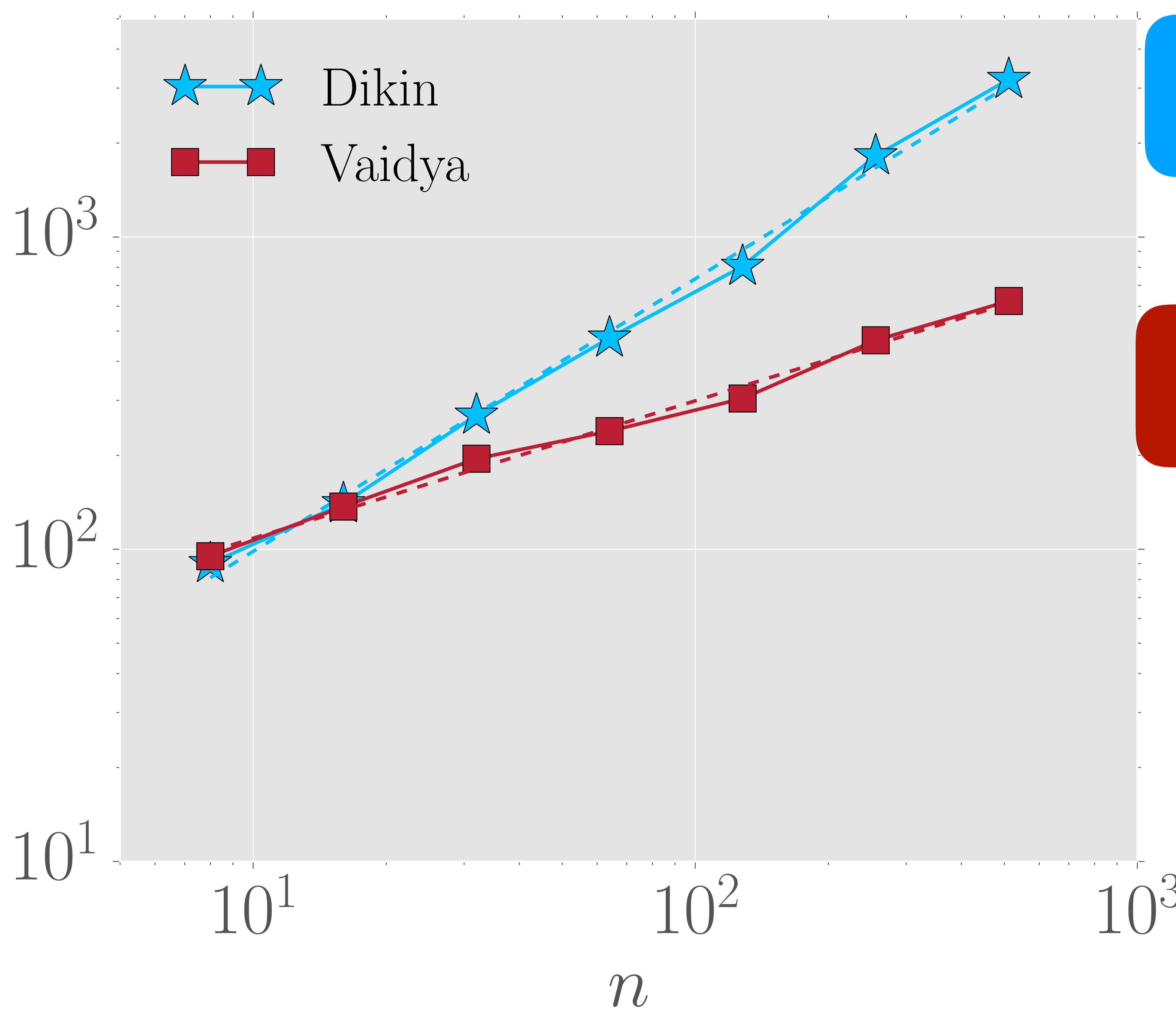
#constraints = **2048**

k = #iterations

#experiments = **200**



Scaling with #constraints



$\propto n^{0.9}$

$\propto n^{0.45}$

Approx. Mixing Time

#constraints (n)

Can we improve further?

[Kannan and Narayanan 2012]

Dikin Proposal

$$z \sim \mathcal{N}\left(x, \frac{\mathbf{1}}{\mathbf{d}}\mathcal{D}_{\mathbf{x}}^{-1}\right)$$

$$\mathcal{D}_x \propto \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

[Chen, D., Wainwright, Yu 2017]

Vaidya Proposal

$$z \sim \mathcal{N}\left(x, \frac{1}{\sqrt{nd}}\mathcal{V}_x^{-1}\right)$$

$$\mathcal{V}_x \propto \sum_{i=1}^{n} \left(\sigma_{x,i} + \frac{d}{n}\right) \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

$$\sigma_{x,i} = \frac{a_i^\top \mathcal{D}_x^{-1} a_i}{(b_i - a_i^\top x)^2}$$

Inspiration from Optimization:

Log Barrier Method [Dikin 1967, Nemirovski 1990]

Volumetric Barrier Method [Vaidya 1993]

Yes..via the John Walk!

[Kannan and Narayanan 2012]

Dikin Proposal

$$z \sim \mathcal{N}\left(x, \frac{\mathbf{1}}{\mathbf{d}} \mathcal{D}_{\mathbf{x}}^{-1}\right)$$

$$\mathcal{D}_x \propto \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

[Chen, D., Wainwright, Yu 2017]

Vaidya Proposal

$$z \sim \mathcal{N}\left(x, \frac{1}{\sqrt{nd}} \mathcal{V}_x^{-1}\right)$$

$$\mathcal{V}_x \propto \sum_{i=1}^{n} \left(\sigma_{x,i} + \frac{d}{n}\right) \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

$$\sigma_{x,i} = \frac{a_i^\top \mathcal{D}_x^{-1} a_i}{(b_i - a_i^\top x)^2}$$

[Chen, D., Wainwright, Yu 2017]

John Proposal

$$z \sim \mathcal{N}\left(x, \frac{1}{d^{1.5}} \mathcal{J}_x^{-1}\right)$$

$$\mathcal{J}_x \propto \sum_{i=1}^{n} j_{x,i} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

$$j_{x,i} = \text{convex program}$$

Inspiration from Optimization:

Log Barrier Method [Dikin 1967, Nemirovski 1990]

Volumetric Barrier Method [Vaidya 1993]

John's Ellipsoidal Algorithm [John 1948, Lee and Sidford 2015]

John walk is “faster” for large #constraints (n)

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \delta$$

	Dikin Walk	Vaidya Walk	John Walk
#Steps	$nd \log \frac{1}{\delta}$	$n^{0.5} d^{1.5} \log \frac{1}{\delta}$	$d^{2.5} \log^4 \frac{n}{d} \log \frac{1}{\delta}$

Per Step Cost			n = #constraints
			d = #dimensions
			n > d

John walk is "faster" for large #constraints (n)

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \delta$$

	Dikin Walk	Vaidya Walk	John Walk
#Steps	$nd \log \frac{1}{\delta}$	$n^{0.5} d^{1.5} \log \frac{1}{\delta}$	$d^{2.5} \log^4 \frac{n}{d} \log \frac{1}{\delta}$
Per Step Cost	nd^2	nd^2	$nd^2 \log^2 n$

John walk is “faster” for large #constraints (n)

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \delta$$

	Dikin Walk	Vaidya Walk	John Walk
#Steps	$nd \log \frac{1}{\delta}$	$n^{0.5} d^{1.5} \log \frac{1}{\delta}$	$d^{2.5} \log^4 \frac{n}{d} \log \frac{1}{\delta}$
Per Step Cost	nd^2	nd^2	$nd^2 \log^2 n$

What if $n \gg d$?

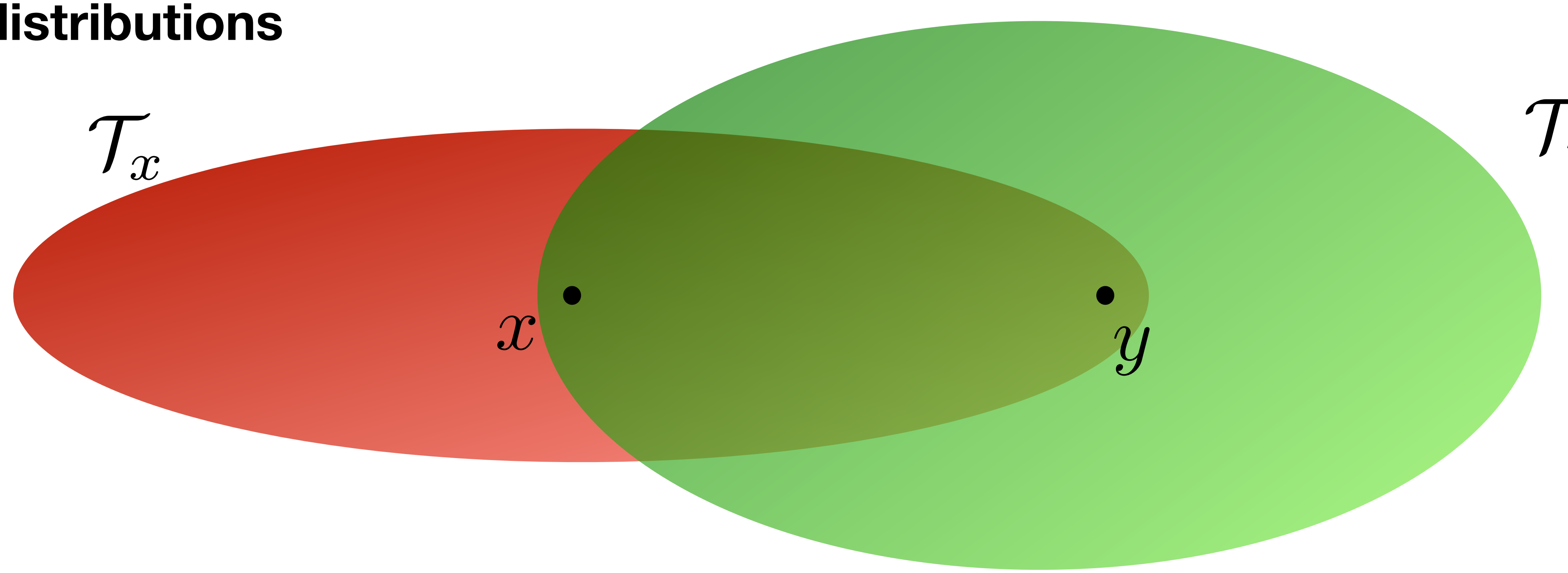
Conjecture: Faster mixing for John walk

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \delta$$

	Dikin Walk	Vaidya Walk	John Walk
#Steps	$nd \log \frac{1}{\delta}$	$n^{0.5} d^{1.5} \log \frac{1}{\delta}$	$d^2 \log^c \frac{n}{d} \log \frac{1}{\delta}$
Per Step Cost	nd^2	nd^2	$nd^2 \log^2 n$

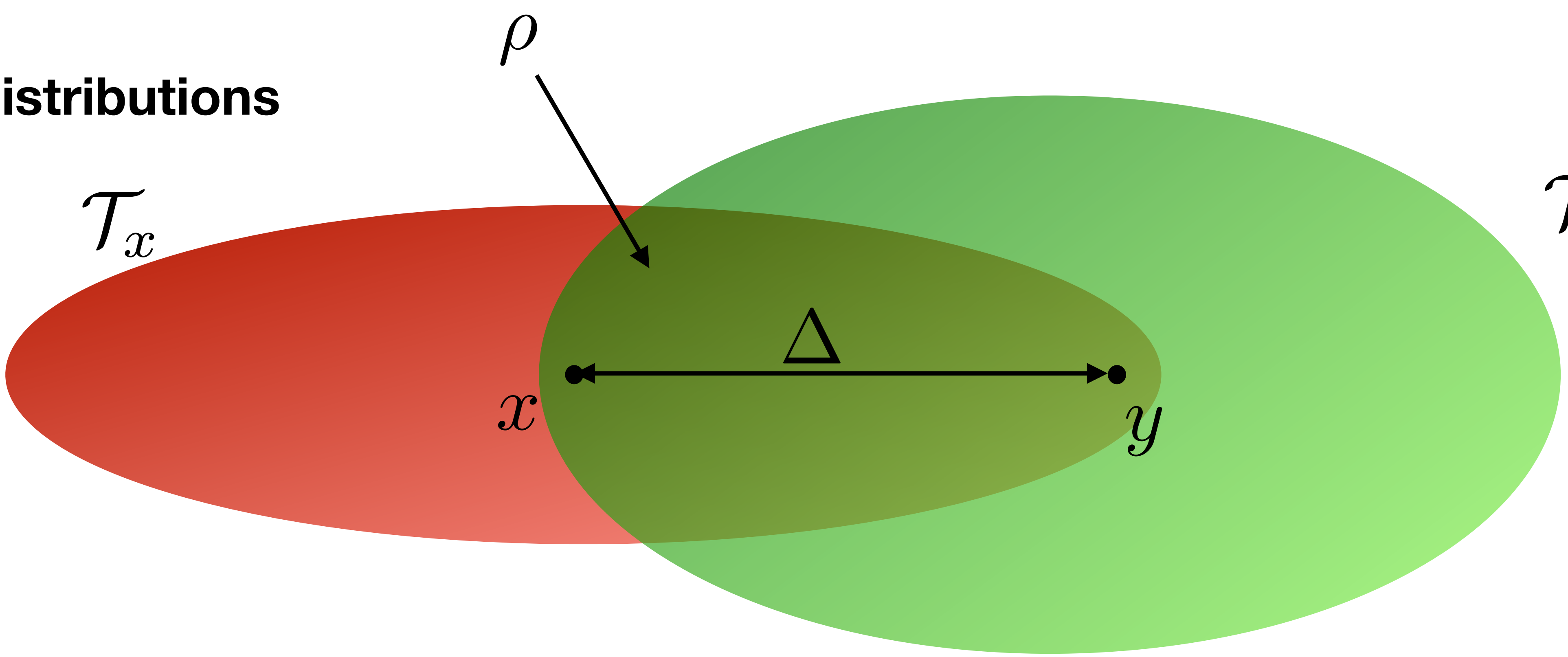
Proof Outline

Transition distributions



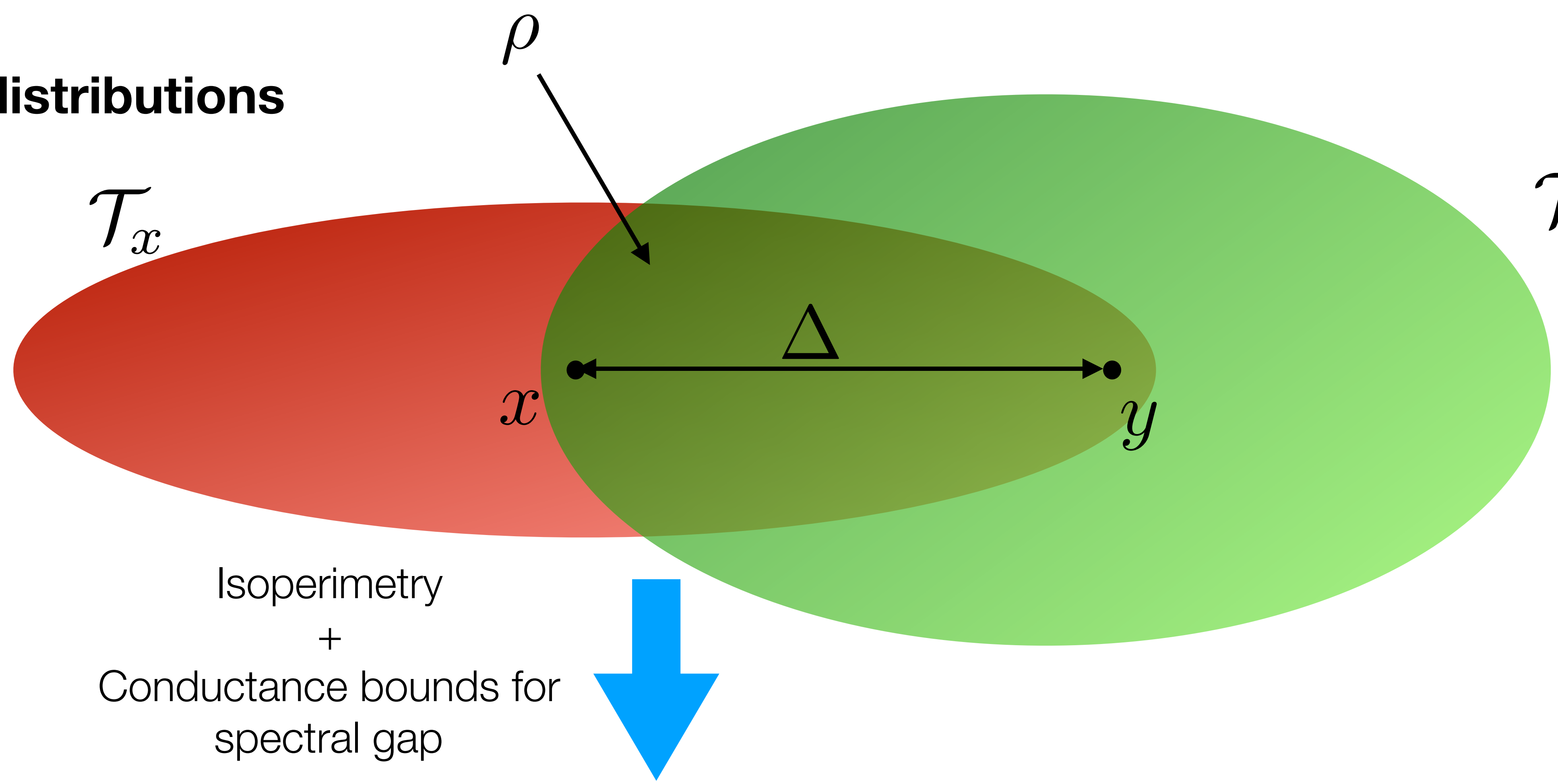
Proof Outline

Transition distributions



Proof Outline

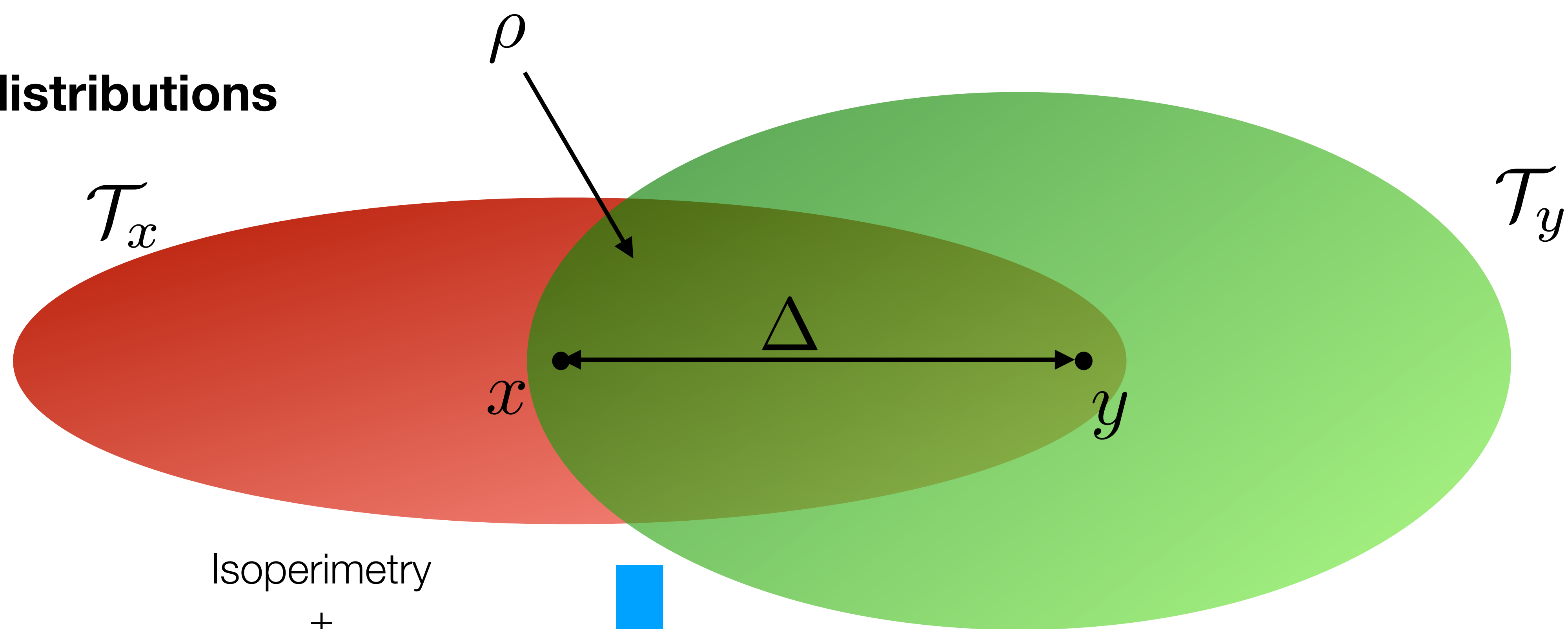
Transition distributions



$$\text{spectral gap} \geq 1 - \frac{\rho^2 \Delta^2}{2}$$

Proof Outline

Transition distributions



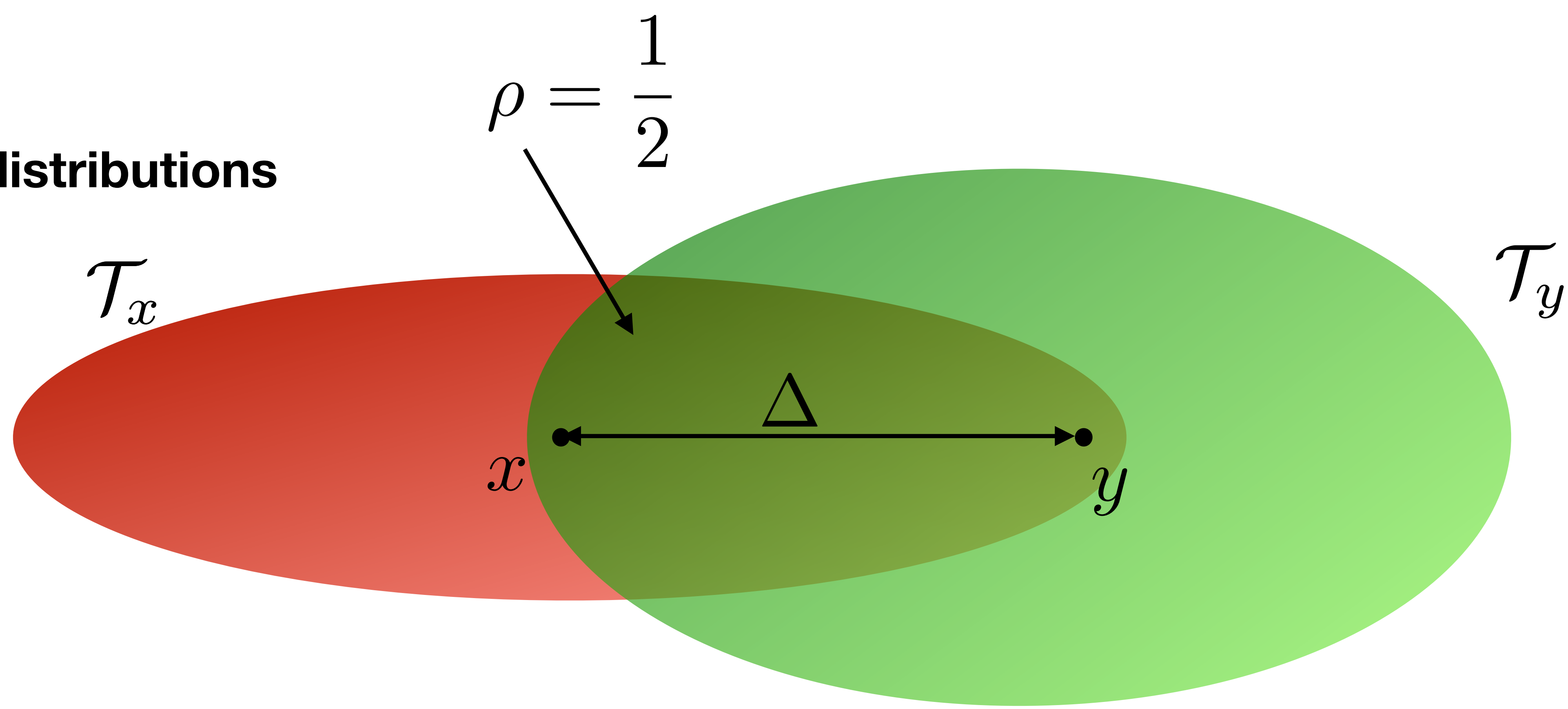
Isoperimetry
+
Conductance bounds for spectral gap

$$\text{spectral gap} \geq 1 - \frac{\rho^2 \Delta^2}{2}$$

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \delta \text{ for } k \geq \mathcal{O}\left(\frac{\log(1/\delta)}{\Delta^2 \rho^2}\right)$$

Proof Outline

Transition distributions

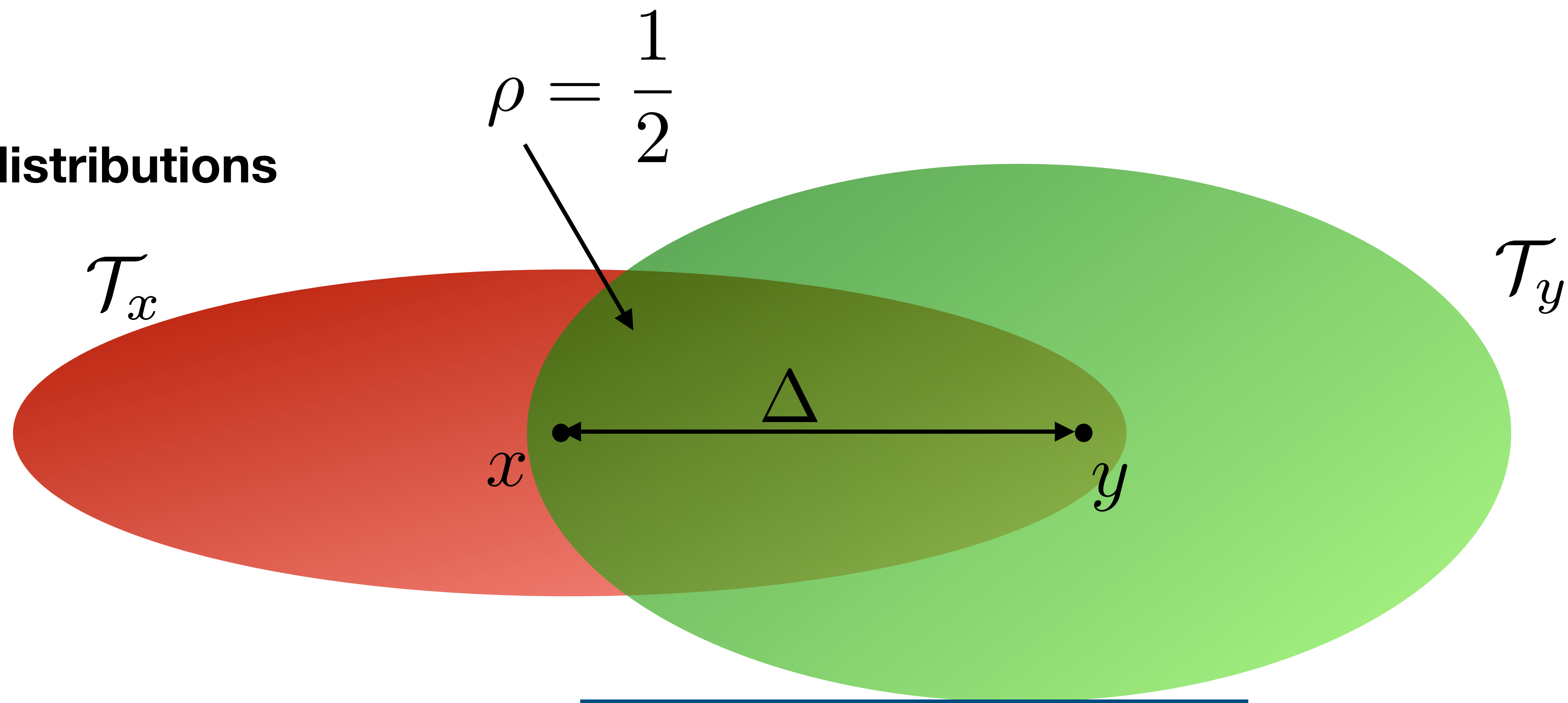


$$\|\mathcal{T}_x - \mathcal{T}_y\|_{\mathrm{TV}} \leq \tfrac{1}{2} \text{ whenever } d(x,y) \leq \Delta$$

$$\begin{aligned}\|\mathcal{T}_x - \mathcal{T}_y\|_{\mathrm{TV}} \leq \|\mathcal{T}_x - \mathcal{P}_x\|_{\mathrm{TV}} + \|\mathcal{T}_y - \mathcal{P}_y\|_{\mathrm{TV}} \\ + \|\mathcal{P}_x - \mathcal{P}_y\|_{\mathrm{TV}}\end{aligned}$$

Proof Outline

Transition distributions



Difference in proposal and transition distribution due to accept-reject step

$$\|\mathcal{T}_x - \mathcal{T}_y\|_{\mathrm{TV}} \leq \|\mathcal{T}_x - \mathcal{P}_x\|_{\mathrm{TV}} + \|\mathcal{T}_y - \mathcal{P}_y\|_{\mathrm{TV}} + \|\mathcal{P}_x - \mathcal{P}_y\|_{\mathrm{TV}}$$

Difference in proposal distributions at two points

Easy part: Analyzing difference in the proposal distributions

$$\mathcal{P}_x = \mathcal{N}\left(x, \frac{c}{\sqrt{nd}} \mathcal{V}_x^{-1}\right)$$

$$\mathcal{V}_x = \sum_{i=1}^{n} \left(\sigma_{x,i} + \frac{d}{n}\right) \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

$\|\mathcal{P}_x - \mathcal{P}_y\|_{\mathrm{TV}}$ is small, if

$$x \approx y$$
$$\mathcal{V}_x \approx \mathcal{V}_y \longrightarrow \text{Smoothness of weights}$$

Hard part: Analyzing the accept-reject step

Difference caused by **accept-reject step** at each point

$$\|\mathcal{T}_x - \mathcal{P}_x\|_{\mathrm{TV}} \leq 2\mathbb{P}(z \notin \mathcal{X}) + \mathbb{E}\left[\min\left\{1, \frac{P(z \to x)}{P(x \to z)}\right\}\right]$$

Hard part: Analyzing the accept-reject step

Difference caused by **accept-reject step** at each point

Easy!

Not easy!

$$\|\mathcal{T}_x - \mathcal{P}_x\|_{\mathrm{TV}} \leq 2\mathbb{P}(z \notin \mathcal{X}) + \mathbb{E}\left[\min\left\{1, \frac{P(z \to x)}{P(x \to z)}\right\}\right]$$

Hard part: Analyzing the accept-reject step

Difference caused by **accept-reject step** at each point

Easy!

Not easy!

$$\|\mathcal{T}_x - \mathcal{P}_x\|_{\mathrm{TV}} \leq 2\mathbb{P}(z \notin \mathcal{X}) + \mathbb{E}\left[\min\left\{1, \frac{P(z \to x)}{P(x \to z)}\right\}\right]$$

Randomness in z
+
Smoothness of weights

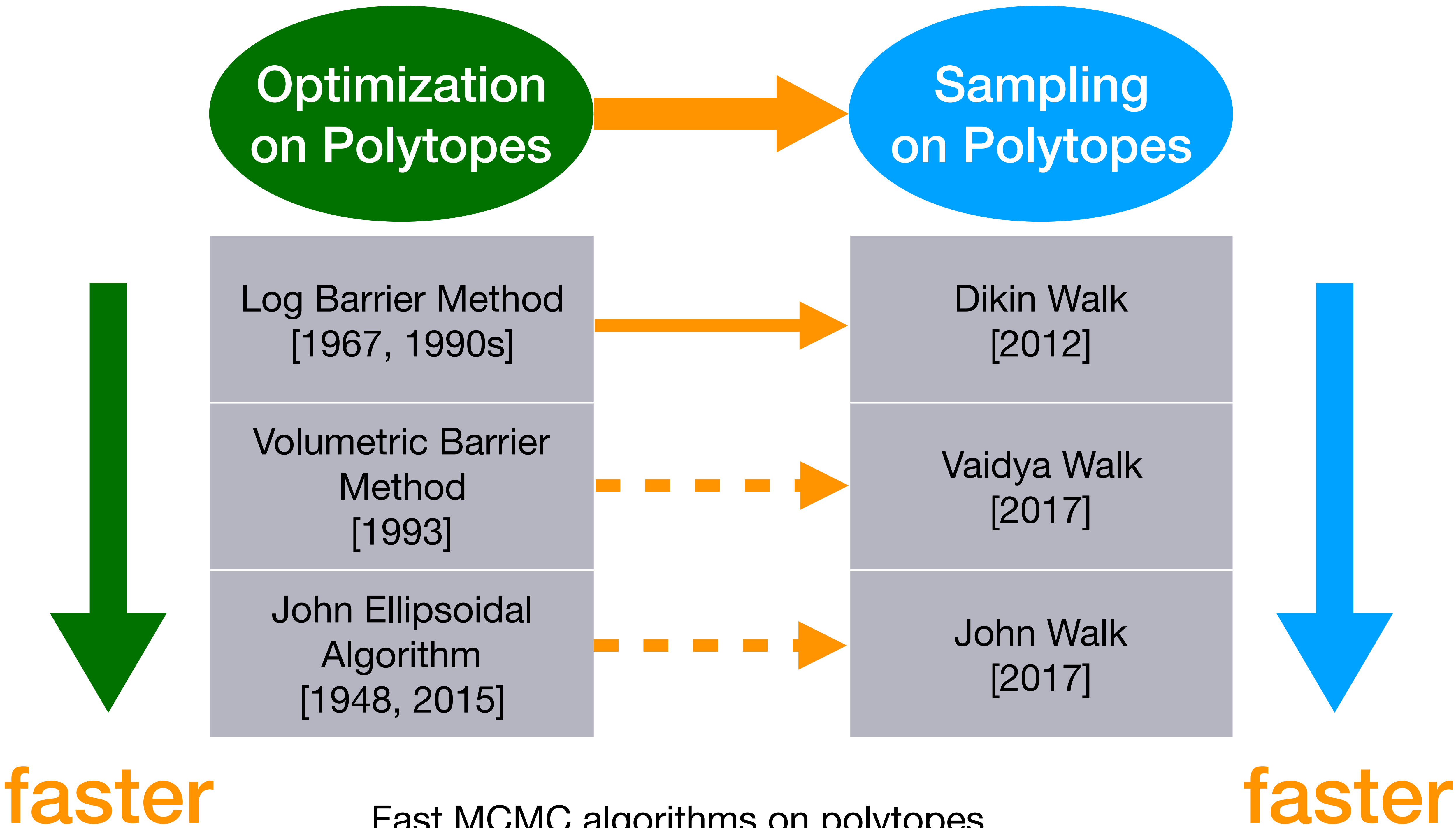
Taylor Series
+
Gaussian polynomial tail bounds

$$(\mathbf{z} - x)^\top \mathcal{V}_{\mathbf{z}}(\mathbf{z} - x) \approx (\mathbf{z} - x)^\top \mathcal{V}_x(\mathbf{z} - x)$$

$$\log\det \mathcal{V}_{\mathbf{z}} \approx \log\det \mathcal{V}_x$$

$$\text{for random } \mathbf{z} \sim \mathcal{P}_x$$

Part I Summary:
Sampling meets optimization



Fast MCMC algorithms on polytopes
https://arxiv.org/abs/1710.08165

Future Directions

Improving dependency on d
[Lee and Vempala 2016, 2017]

Sampling on sketched polytopes

Non-uniform sampling
[Rakhlin et al. 2015, Bubeck et al. 2015]

Part II: Log-Concave Sampling

Joint work with Yuansi Chen, Martin Wainwright and Bin Yu

$$\pi(x) \propto e^{-f(x)} \quad \text{where } f : \mathbb{R}^d \to \mathbb{R} \text{ is convex}$$

Part II: Log-Concave Sampling

Joint work with Yuansi Chen, Martin Wainwright and Bin Yu

$$\pi(x) \propto e^{-f(x)} \quad \text{where } f : \mathbb{R}^d \to \mathbb{R} \text{ is convex}$$

- Examples include Gaussian distributions, Laplace distributions, exponential and logistic distributions
- Frequentist set ups: form confidence intervals around the MLE
- Bayesian inference and inverse problems: MAP and credible interval estimation
- Large scale stochastic/Bayesian optimization

From optimization to sampling

- Optimization: find the global minimum (or a stationary point)

$$\min_{x \in \mathbb{R}^d} f(x)$$

- Gradient descent:

$$x_{k+1} = x_k - h \nabla f(x_k)$$

- Stochastic Gradient Algorithm:

$$X_{k+1} = X_k - h \nabla f(X_k) + {\color{blue} h \xi_{k+1}}$$

- Sampling: draw samples from the density

$$\pi(x) \propto e^{-f(x)}$$

- Unadjusted Langevin algorithm (ULA):

$$X_{k+1} = X_k - h \nabla f(X_k) + {\color{red} \sqrt{2h} \xi_{k+1}}$$

$$\xi_k \overset{i.i.d.}{\sim} \mathcal{N}(0, I_{d \times d})$$

[Parisi 1981, Grenander & Miller 1994, Roberts & Tweedie 1996]

Langevin algorithms: Origins?

- Classical Langevin stochastic differential equation

$$dX_t = -\nabla f(X_t)dt + \sqrt{2}dB_t \quad \text{where } B_t \text{ is standard Brownian motion}$$

- Under mind regularity conditions: as $t \to \infty$, distribution of X_t converges to $\pi(x) \propto e^{-f(x)}$

$$\|P(X_t) - \pi\|_{\mathrm{TV}} \xrightarrow{t \uparrow \infty} 0$$

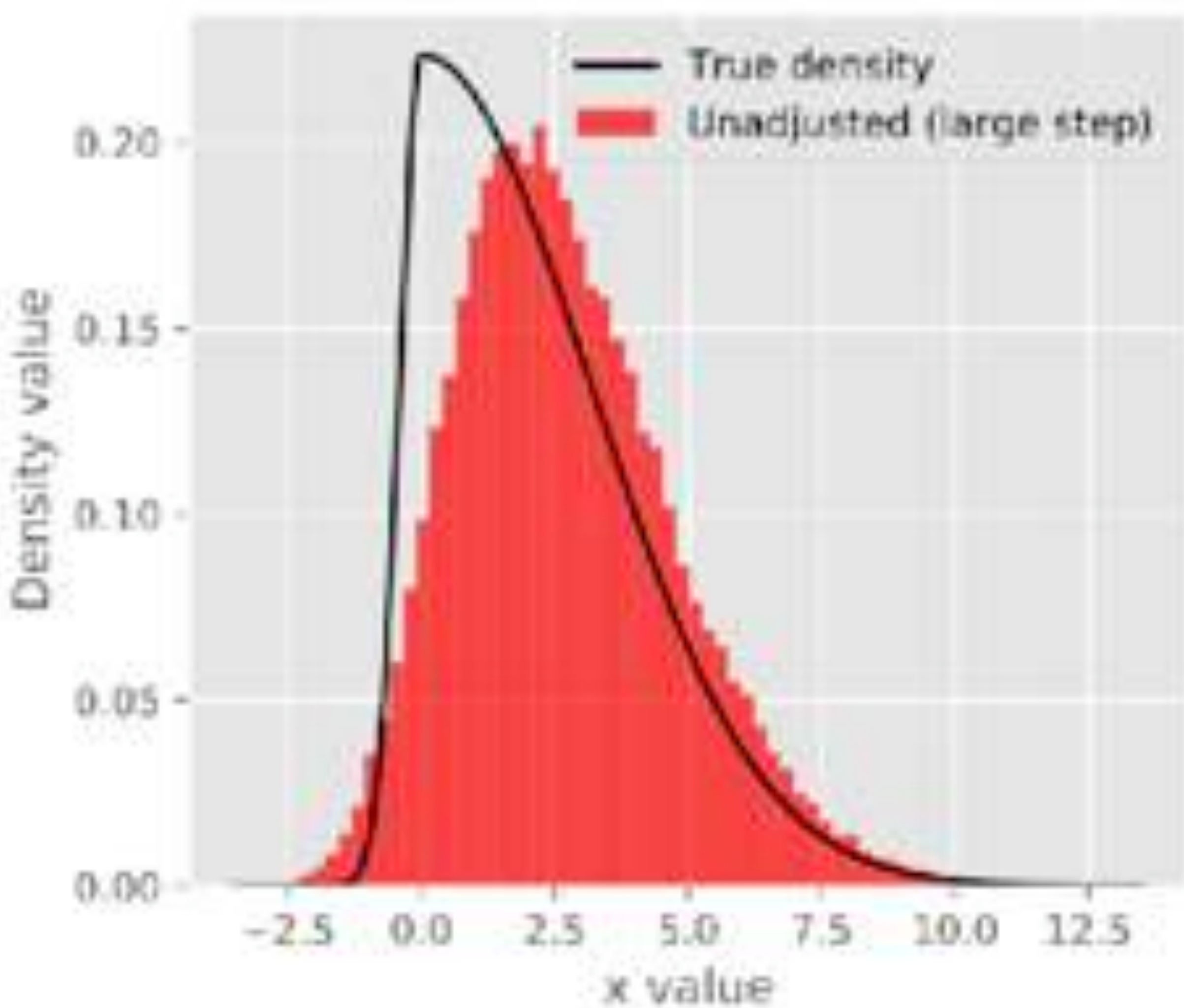
- ULA updates: forward discretization of the Langevin SDE

$$X_{k+1} - X_k = -h\nabla f(X_k) + \sqrt{2h}\xi_{k+1}$$

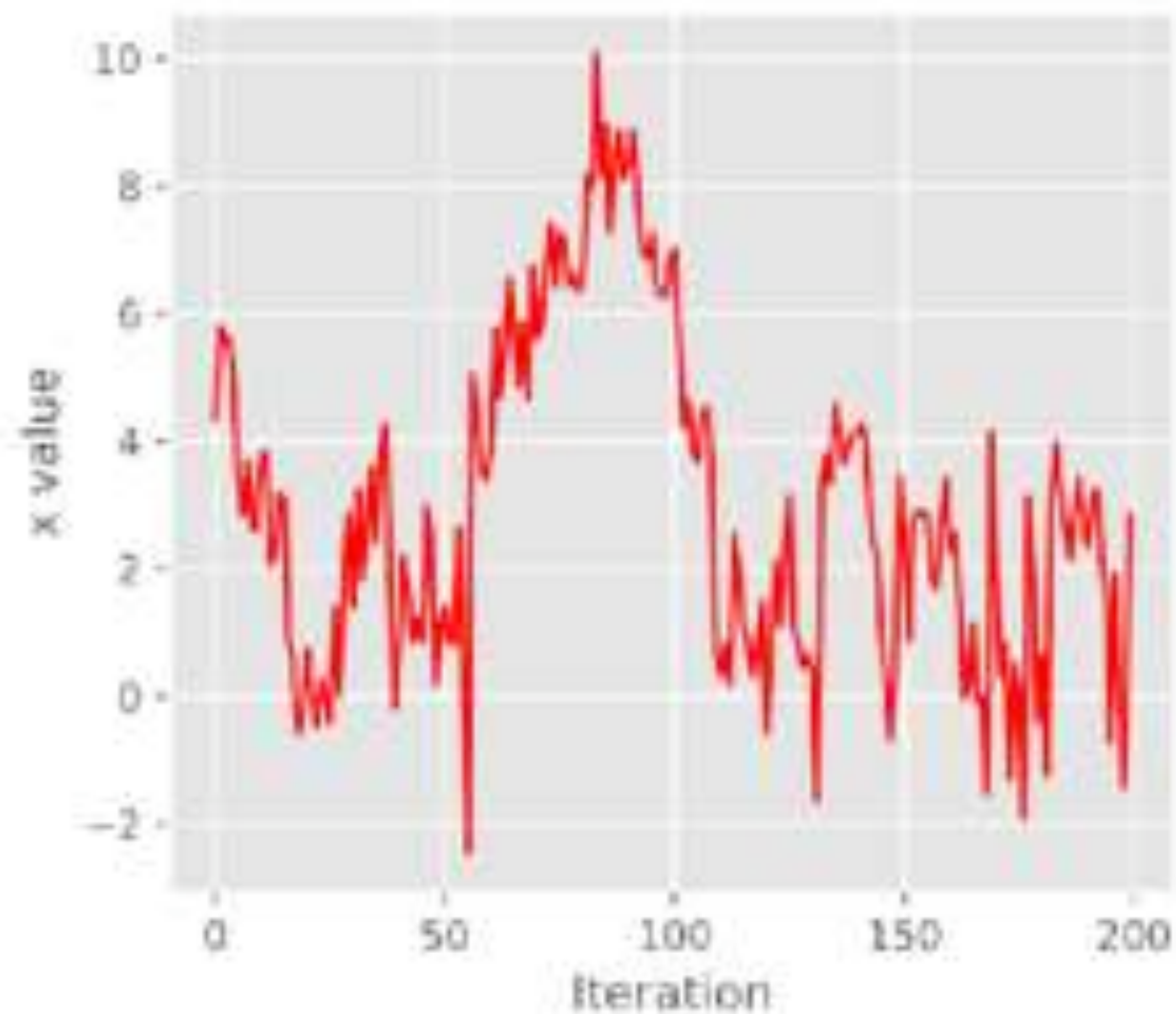
(no accept-reject step)

ULA performance:
Large step size leads to large bias!

Histogram (multiple runs)
upon convergence

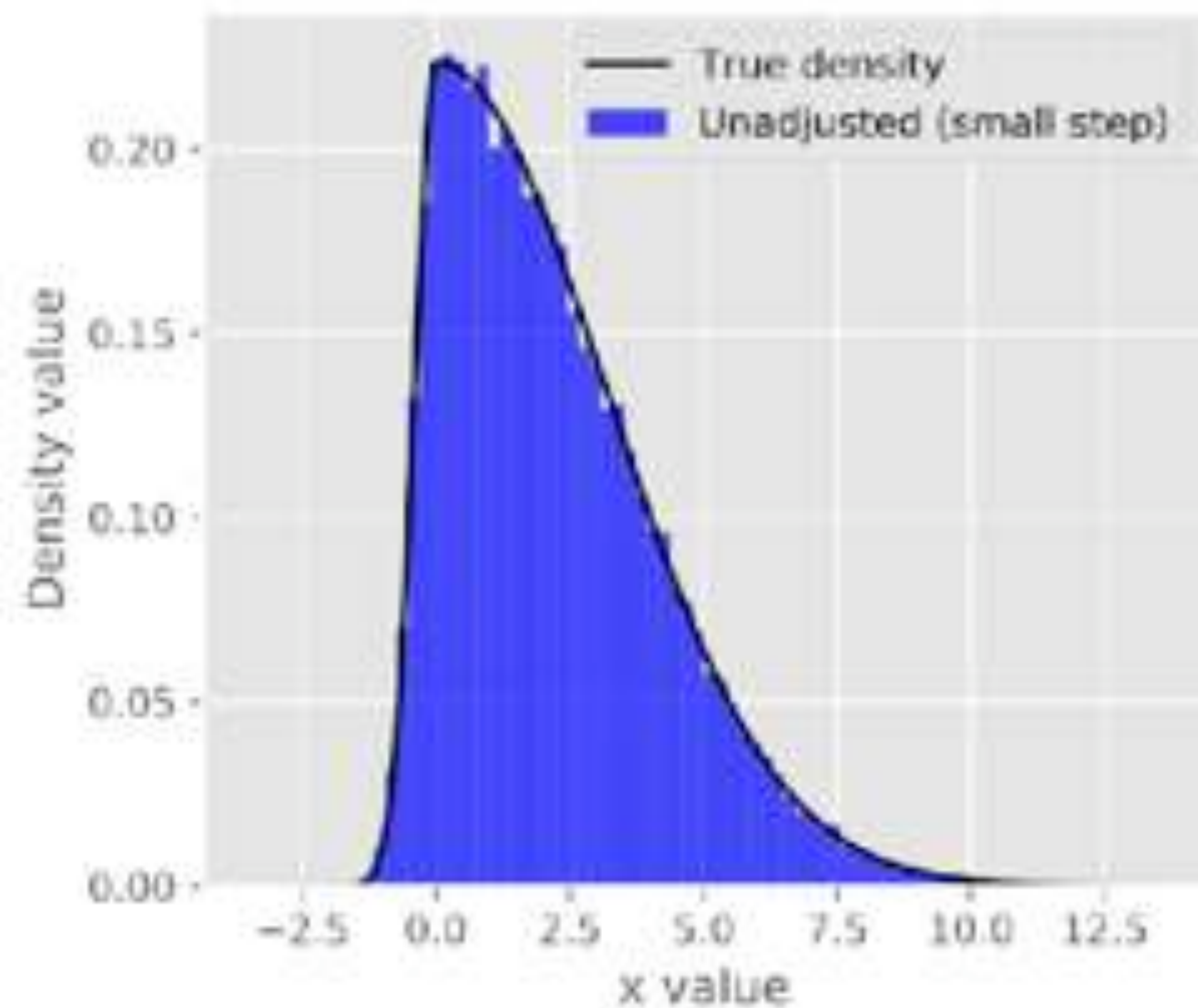


Trace-plot for one run

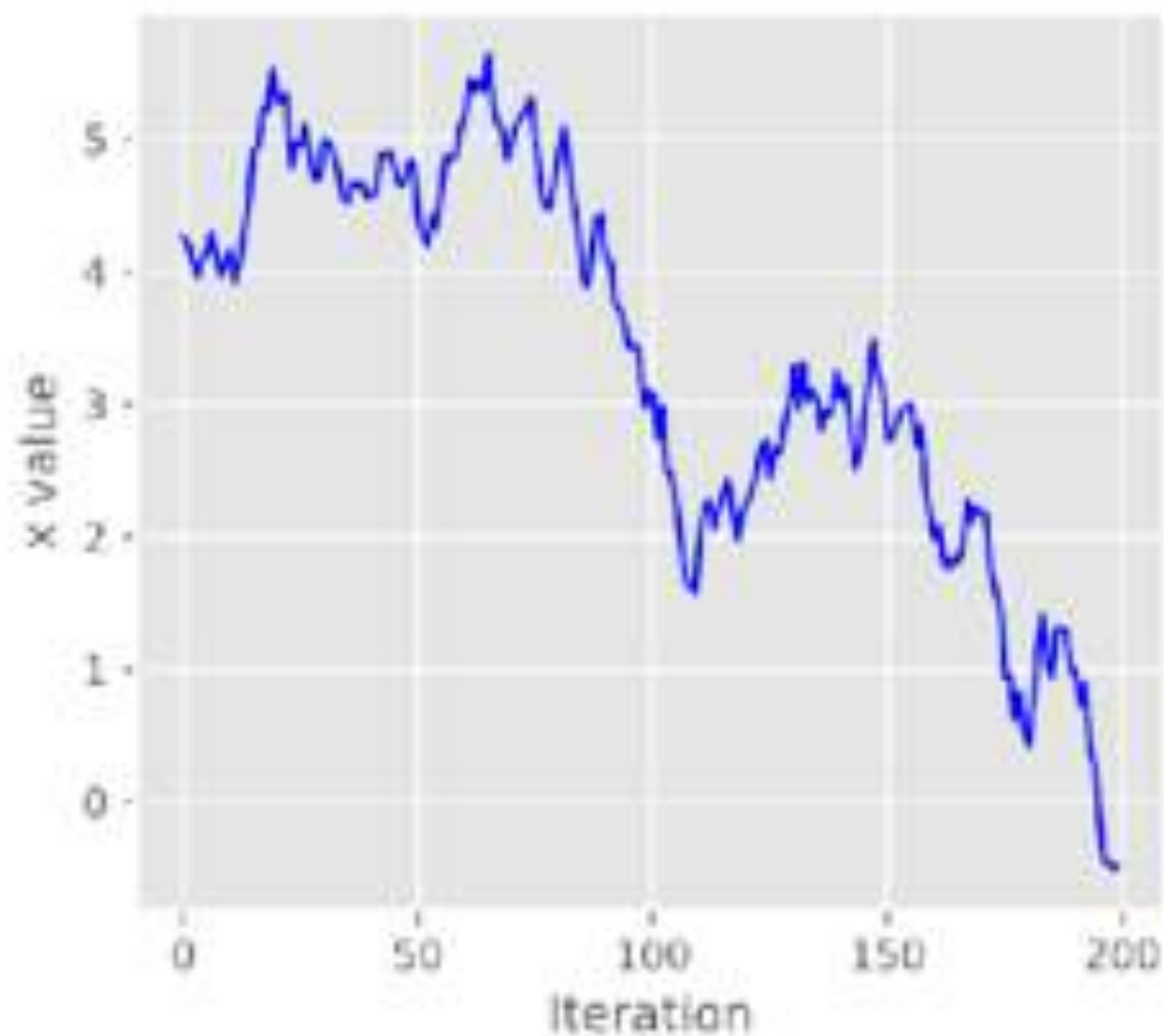


ULA performance:
Small step mixes slowly!

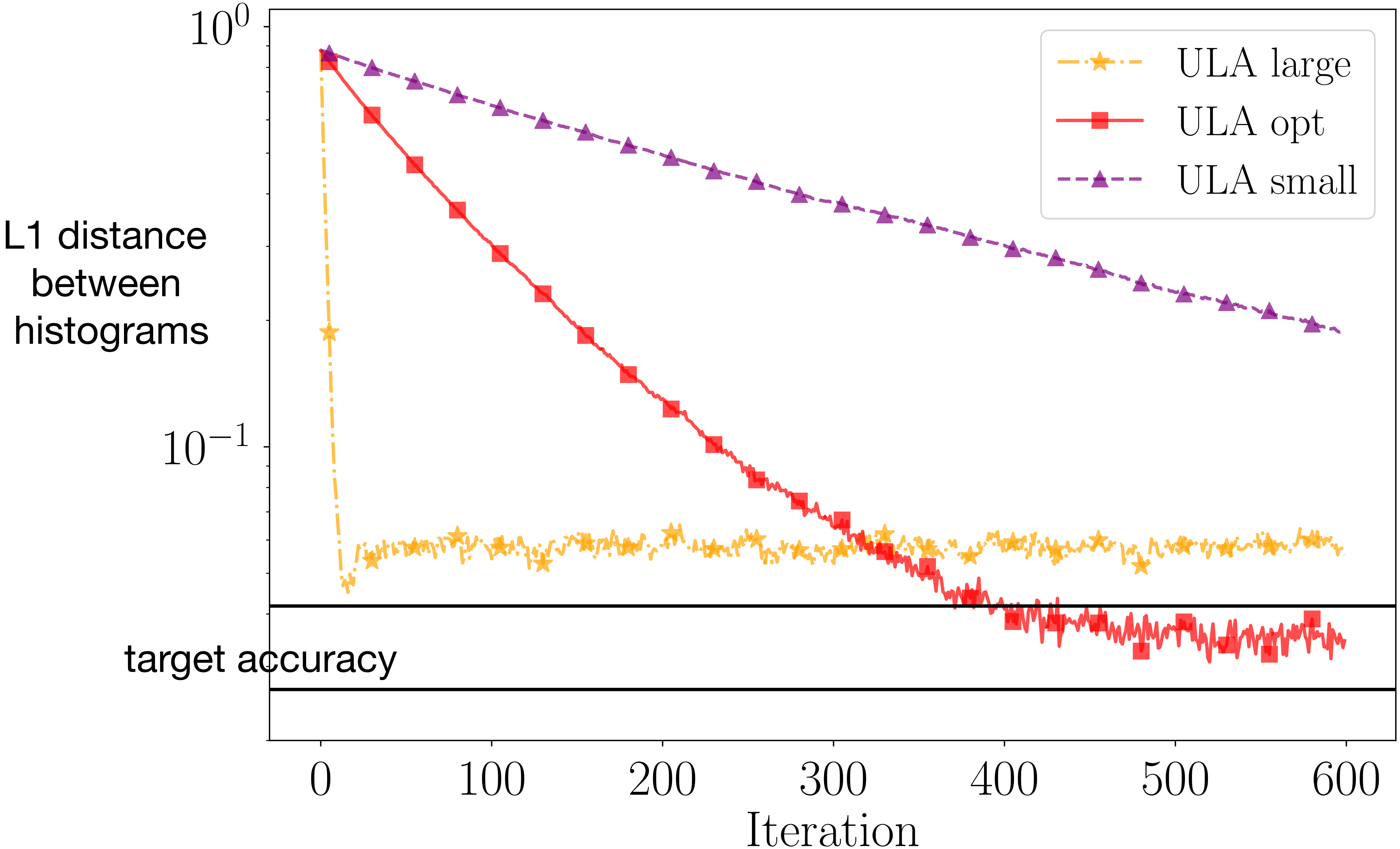
Histogram (multiple runs)
upon convergence



Trace-plot for one run



ULA: Step-size and speed/bias tradeoff



How does one remove the asymptotic bias?

- Via the **classical** Metropolis-Hastings correction step
- Metropolis adjusted Langevin algorithm (MALA):

  1. Use ULA updates as proposals

  $$z = x - h\nabla f(x) + \sqrt{2h}\xi$$

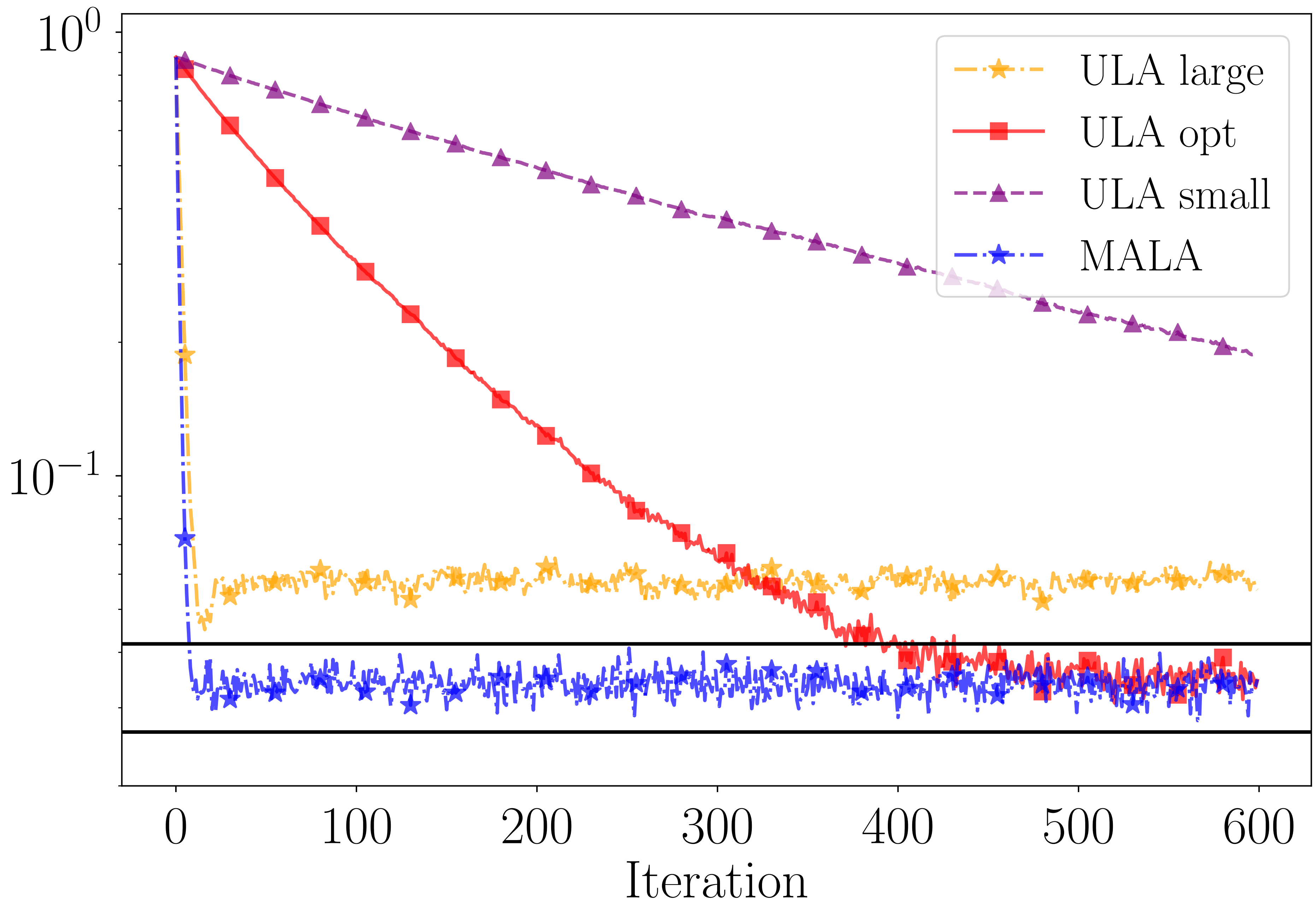
  2. Accept z with probability

  $$\min\left\{1, \frac{e^{-f(z)}}{e^{-f(x)}} \frac{P(z \to x)}{P(x \to z)}\right\}$$

  3. In case of rejection, stay at x

MALA: Fast convergence with no bias

L1 distance between histograms



Langevin algorithms: Traditional wisdom

- Rich body of work for Langevin algorithms
- ULA and MALA first suggested by Parisi in 1981 and formally introduced by Grenander & Miller in 1994
- Sufficient conditions for convergence first established by Roberts and Tweedie in 1996

Langevin algorithms: Prior work

Type of results	Existing Literature
Discretization & integration errors, Ergodicity, Asymptotic convergence	[Talay & Tubaro '90], [Meyn & Tweedie '95], [Roberts & Rosenthal '96, '01, '02]

- Find conditions on π and a Lyapunov function V that contracts outside a ball

$$E[V(X_1)|X_0 = x] \leq \lambda V(x) + C\mathbb{I}_{\mathbb{B}(0,R)}(x), \quad \lambda < 1$$

sufficient to establish geometric ergodicity

$$\|P(x_k|x_0 = x) - \pi\|_{\mathrm{TV}} \leq V(x)R\rho^k \quad \text{for some } \rho < 1 \text{ and } R > 0$$

For limited class of distributions,
non-explicit rates,
hard to track dependency on problem parameters

Langevin algorithms: Related work

Type of results	Existing Literature
Discretization & integration errors, Ergodicity, Asymptotic convergence	[Talay & Tubaro ‘90], [Meyn & Tweedie ‘95], [Roberts & Rosenthal ‘96, ‘01, ‘02]
Revived interest for non-asymptotic results	[Bou-Rabee & Hairer ‘09], [Roberts & Rosenthal ‘14]
Explicit non-asymptotic bounds	[Dalalyan ‘15, ‘17], [Durmus & Moulines ‘15, ‘16], [Cheng & Bartlett ‘17]

Recent work uses coupling arguments for diffusions

Mixing time bounds: Strongly log-concave

$$\pi(x) \propto e^{-f(x)}$$

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \delta$$

Algorithm	ULA [Dalalyan 2016]	
f is L-smooth and m-strongly-convex	$d\left(\frac{L}{m}\right)^2 \frac{1}{\delta^2}$	

Mixing time bounds: Strongly log-concave

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \delta$$

$$\pi(x) \propto e^{-f(x)}$$

Algorithm	ULA [Dalalyan 2016]	MALA [D., Chen, Wainwright, Yu 2018]
f is L-smooth and m-strongly-convex	$d\left(\frac{L}{m}\right)^2 \frac{1}{\delta^2}$	$d\left(\frac{L}{m}\right) \log \frac{1}{\delta}$

Mixing time bounds: Strongly log-concave

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \delta$$

$\pi(x) \propto e^{-f(x)}$

Algorithm	ULA [Dalalyan 2016]	MALA [D., Chen, Wainwright, Yu 2018]
f is L-smooth and m-strongly-convex	$d\left(\frac{L}{m}\right)^2 \frac{1}{\delta^2}$	$d\left(\frac{L}{m}\right) \log \frac{1}{\delta}$
	Mixing time of MALA has • exponentially better dependence on accuracy δ • better dependence on conditioning L/m	

Mixing time bounds: Strongly and weakly log-concave

$$\pi(x) \propto e^{-f(x)}$$

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \delta$$

Algorithm	ULA [Dalalyan 2016]	MALA [D., Chen, Wainwright, Yu 2018]
f is L-smooth and m-strongly-convex	$d\left(\frac{L}{m}\right)^2 \frac{1}{\delta^2}$	$d\left(\frac{L}{m}\right) \log \frac{1}{\delta}$
f is convex and L-smooth	$d^3 L^2 \frac{1}{\delta^4}$	$d^2 L^{1.5} \frac{1}{\delta^{1.5}}$

Mixing time bounds:
Strongly and weakly log-concave

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \delta$$

$\pi(x) \propto e^{-f(x)}$

Algorithm	ULA [Dalalyan 2016]	MALA [D., Chen, Wainwright, Yu 2018]
f is L-smooth and m-strongly-convex	$d\left(\frac{L}{m}\right)^2 \frac{1}{\delta^2}$	$d\left(\frac{L}{m}\right) \log \frac{1}{\delta}$
f is convex and L-smooth	$d^3 L^2 \frac{1}{\delta^4}$	$d^2 L^{1.5} \frac{1}{\delta^{1.5}}$

Faster!

The difference between MALA and ULA: An informal proof

- Both algorithms have a good spectral gap in a high probability region

The difference between MALA and ULA: An informal proof

- Both algorithms have a good spectral gap in a high probability region

- ULA has a biased stationary distribution

Bias

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \|P(x_k) - \pi_{\mathrm{ULA}}\|_{\mathrm{TV}} + \|\pi_{\mathrm{ULA}} - \pi\|_{\mathrm{TV}}$$

The difference between MALA and ULA: An informal proof

- Both algorithms have a good spectral gap in a high probability region

- ULA has a biased stationary distribution

Bias

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \underbrace{\|P(x_k) - \pi_{\mathrm{ULA}}\|_{\mathrm{TV}}}_{\mathcal{O}(e^{-kh})} + \underbrace{\|\pi_{\mathrm{ULA}} - \pi\|_{\mathrm{TV}}}_{\mathcal{O}(\sqrt{h})}$$

The difference between MALA and ULA: An informal proof

- Both algorithms have a good spectral gap in a high probability region

- ULA has a biased stationary distribution

Bias

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \|P(x_k) - \pi_{\mathrm{ULA}}\|_{\mathrm{TV}} + \|\pi_{\mathrm{ULA}} - \pi\|_{\mathrm{TV}}$$

$$\mathcal{O}(e^{-kh}) \leq \delta/2 \qquad \mathcal{O}(\sqrt{h}) \leq \delta/2$$

$$\Rightarrow \quad k \geq \mathcal{O}\left(\frac{1}{h}\log\frac{1}{\delta}\right) = \mathcal{O}\left(\frac{1}{\delta^2}\right)$$

The difference between MALA and ULA: An informal proof

- Both algorithms have a good spectral gap in a high probability region

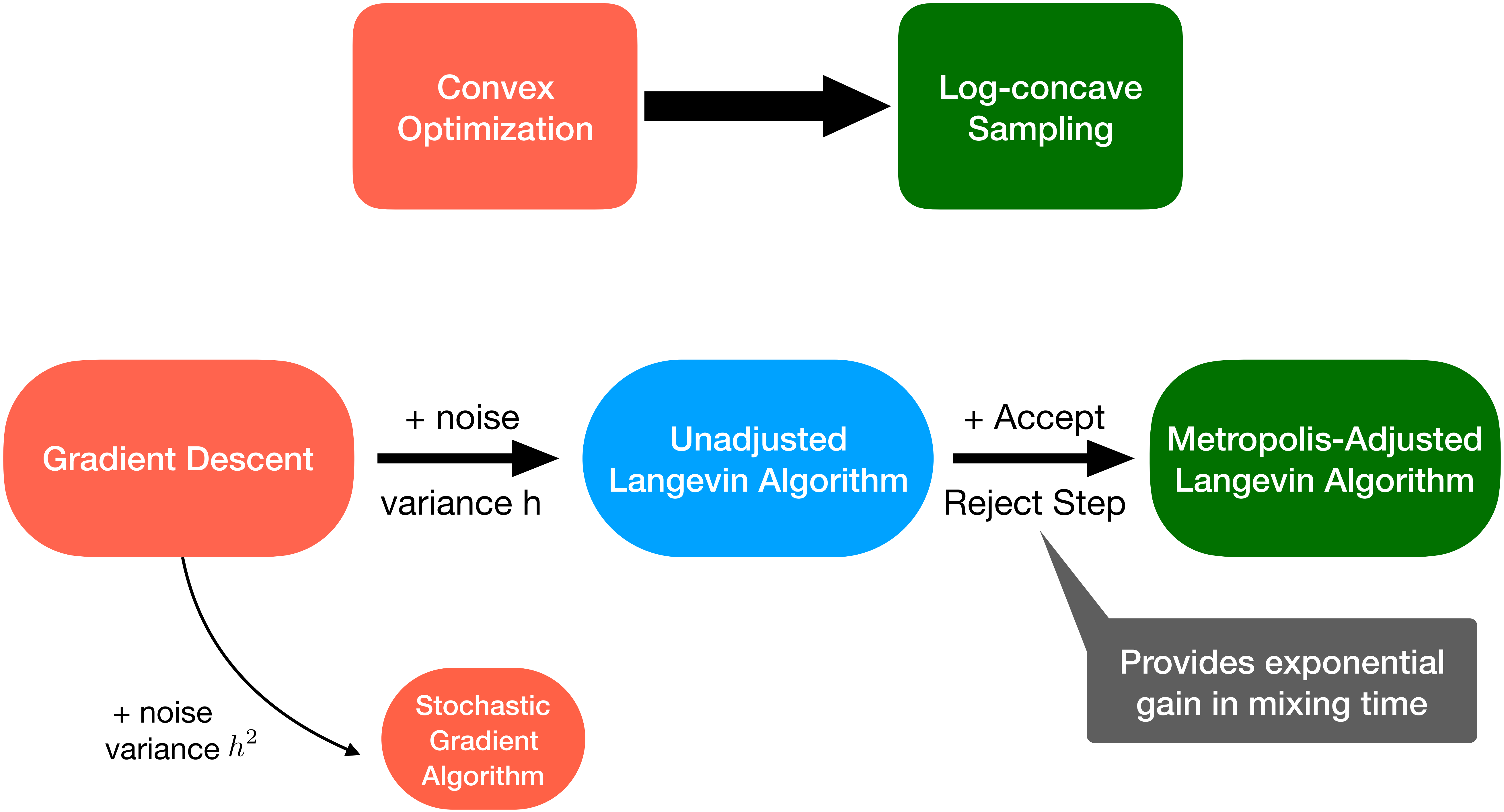
- ULA has a biased stationary distribution

$$\|P(x_k) - \pi\|_{\mathrm{TV}} \leq \underbrace{\|P(x_k) - \pi_{\mathrm{ULA}}\|_{\mathrm{TV}}}_{\mathcal{O}(e^{-kh}) \ \leq \ \delta/2} + \overbrace{\underbrace{\|\pi_{\mathrm{ULA}} - \pi\|_{\mathrm{TV}}}_{\mathcal{O}(\sqrt{h}) \ \leq \ \delta/2}}^{\text{Bias}}$$

$$k \geq \mathcal{O}\left(\frac{1}{h} \log \frac{1}{\delta}\right) = \mathcal{O}\left(\frac{1}{\delta^2}\right)$$

- MALA is unbiased: larger step size implies faster mixing

Part II: Summary



Future Directions

No gradient:
Metropolis random walk
O(d) slower!
[D., Chen, Wainwright, Yu 2018]

With Hessian:
Can we have a faster algorithm?

Higher order methods:
Hamiltonian Monte Carlo
Underdamped Langevin
[Cheng et al. 2017, Smith et al. 2018]

Framework for lower bounds on mixing times?

General/Mixture distributions:
Non-log concave sampling
(Simulated Tempering)

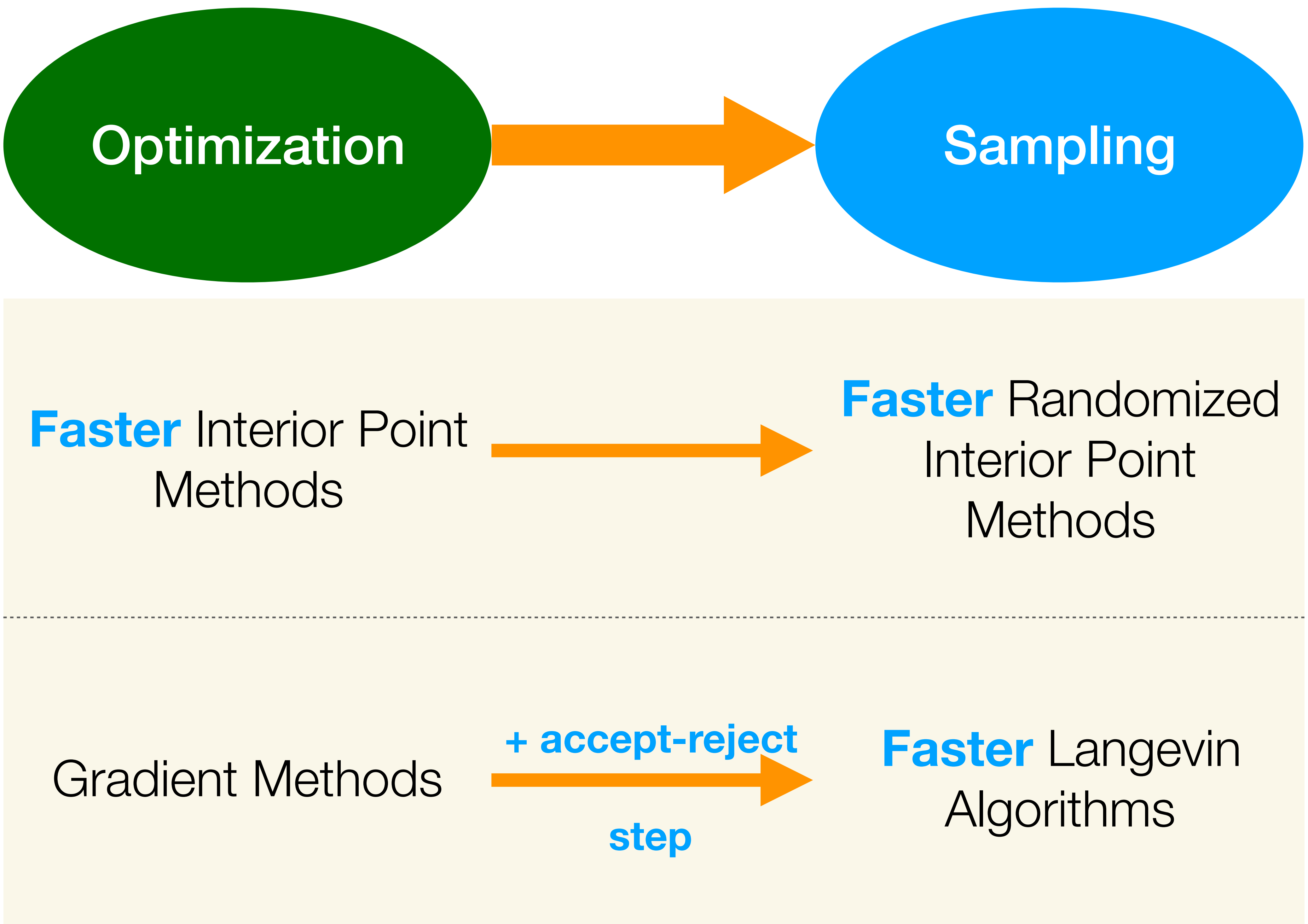
Summary: Connections

Optimization → **Sampling**

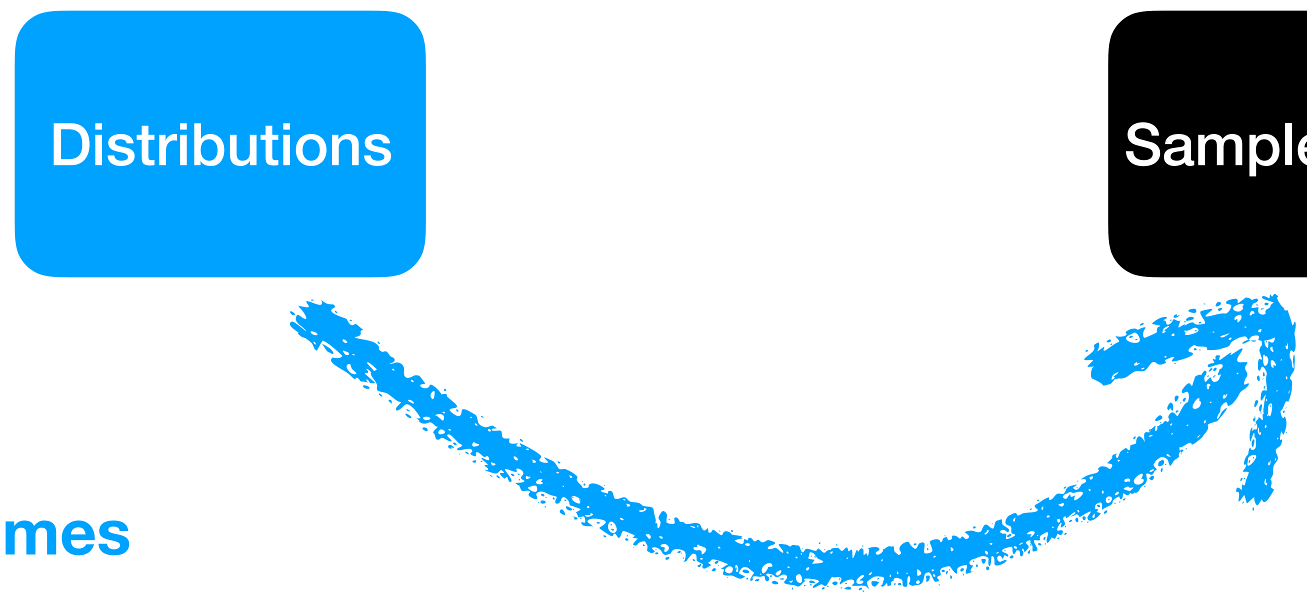
Interior Point Methods → Randomized Interior Point Methods

Gradient Methods → Langevin Algorithms

Summary: Findings



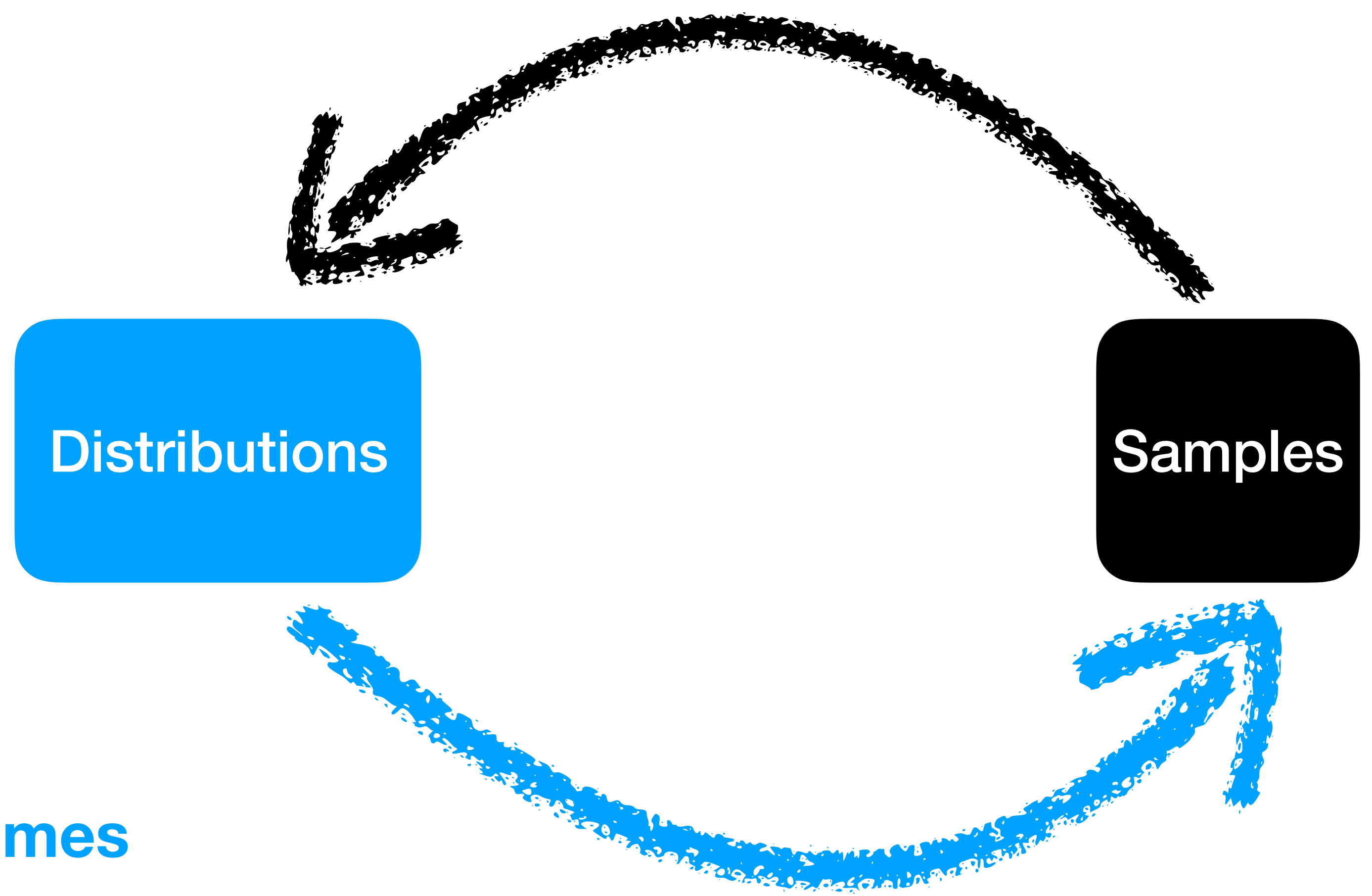
So far…



- **Mixing times**
- Function specific mixing times:
  Estimating mean and covariance

Looking forward..

- Learning from data

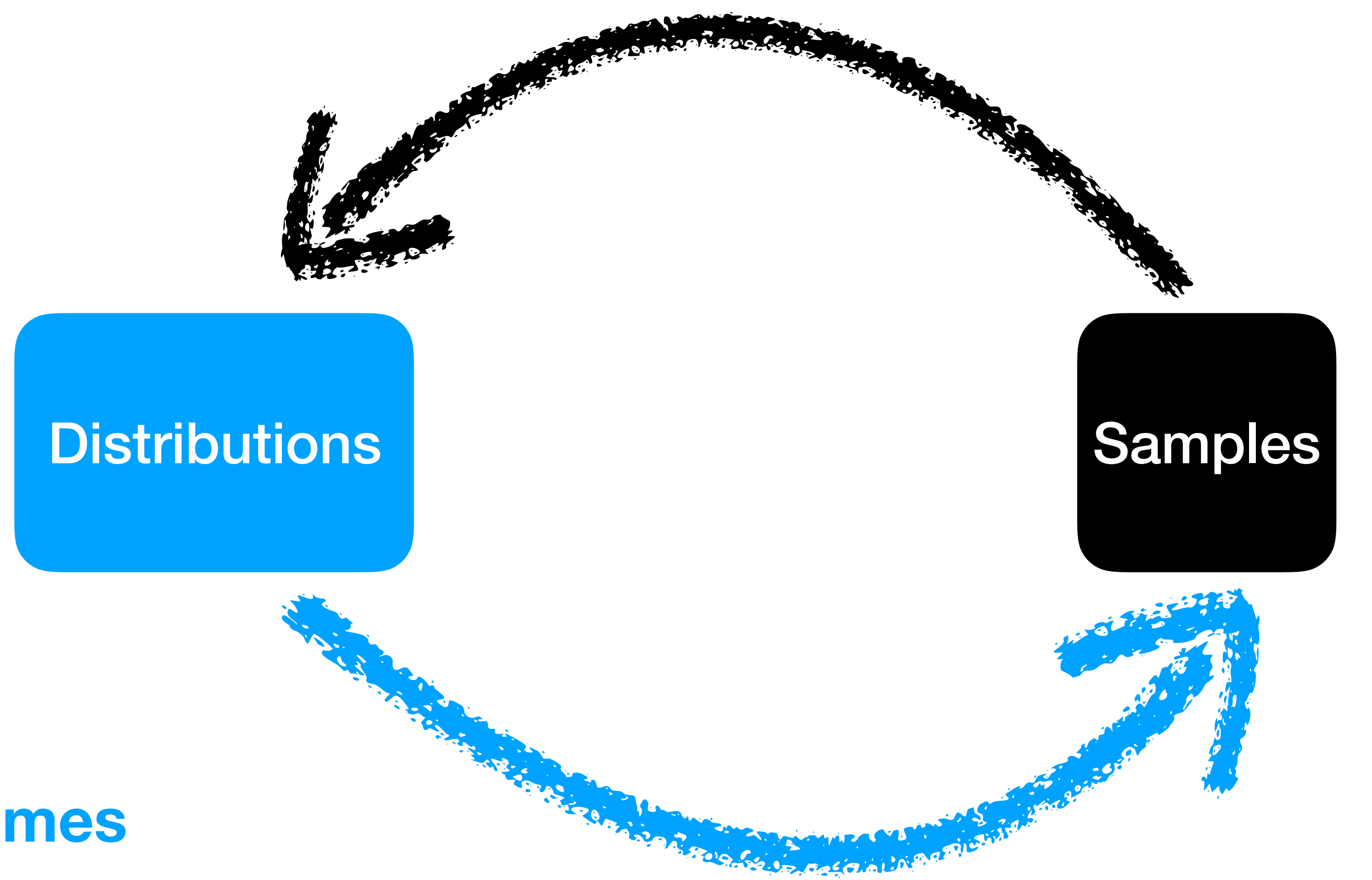


- **Mixing times**
- Function specific mixing times:
  Estimating mean and covariance

Looking forward..

Algorithmic and statistical guarantees for learning mixture models from samples when number of mixtures is not known

- Learning from data



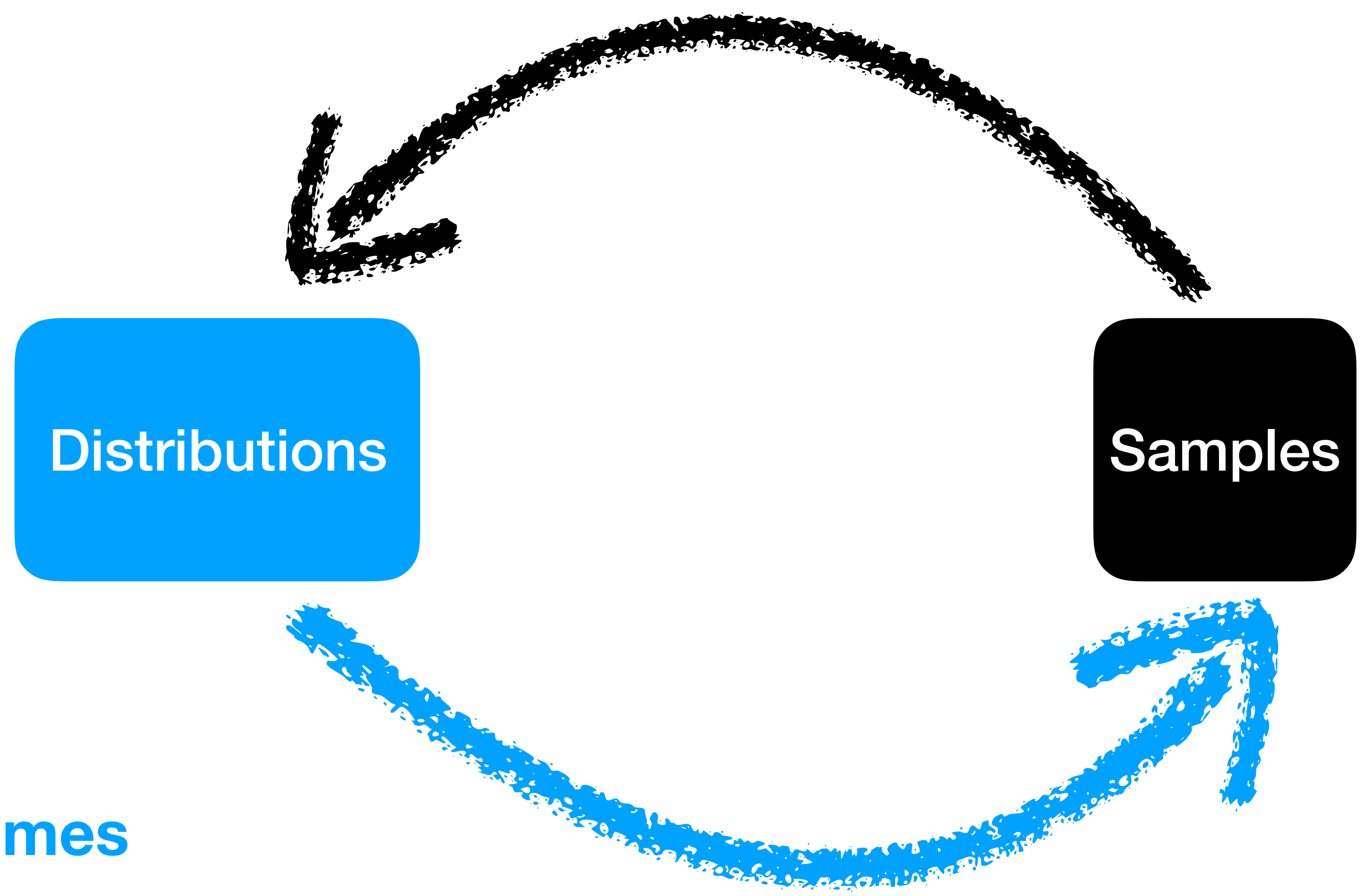
- **Mixing times**
- Function specific mixing times: Estimating mean and covariance

Looking forward..

Algorithmic and statistical guarantees for learning mixture models from samples when number of mixtures is not known

Data driven manifold learning: Low dimensional structure in deep networks

- Learning from data



- **Mixing times**
- Function specific mixing times: Estimating mean and covariance

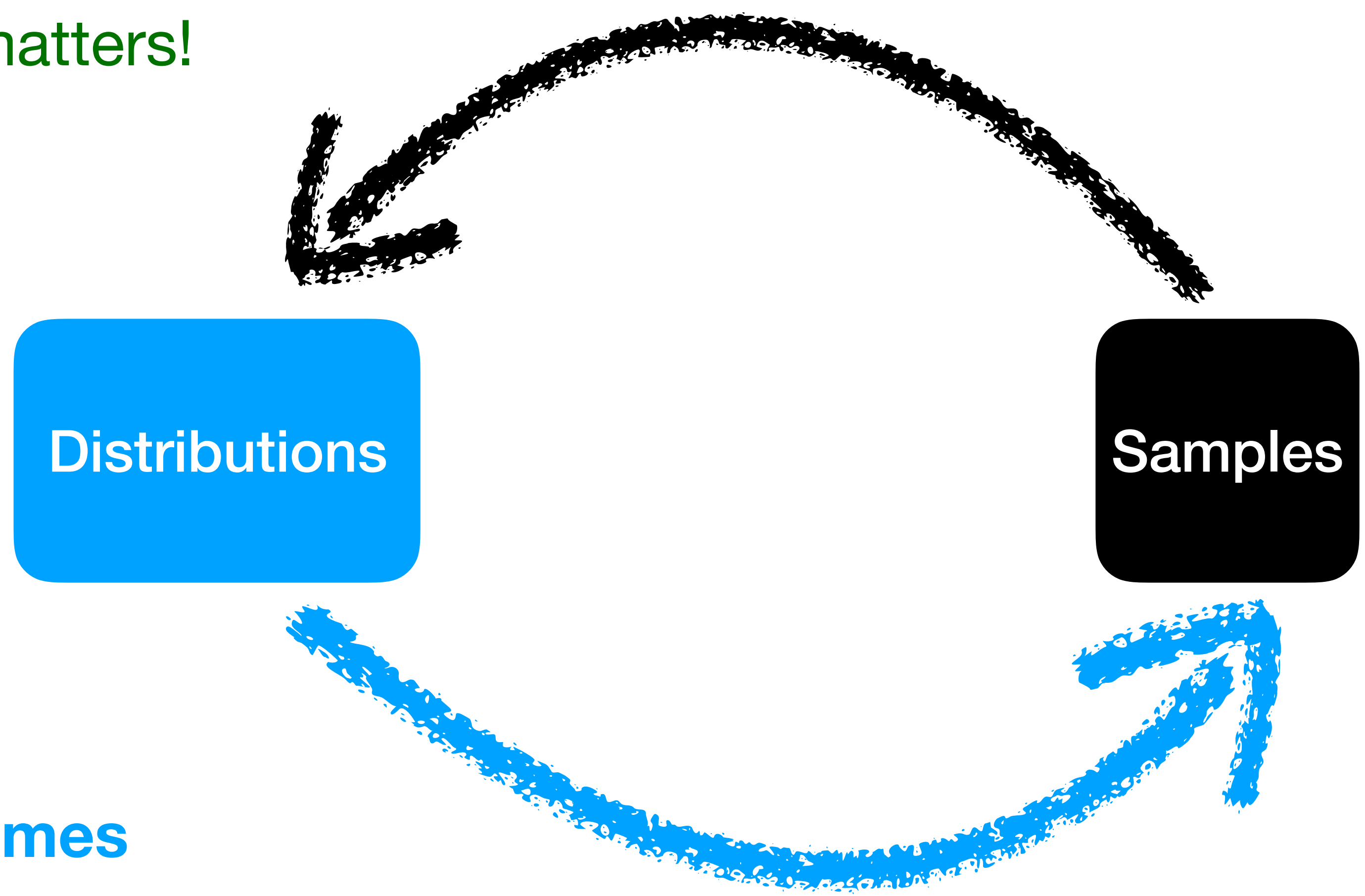
Looking forward..

Algorithmic and statistical guarantees for learning mixture models from samples when number of mixtures is not known

Data driven manifold learning: Low dimensional structure in deep networks

- Learning from data

Will the model generalize or not?
Choice of kernel matters!



- **Mixing times**
- Function specific mixing times: Estimating mean and covariance

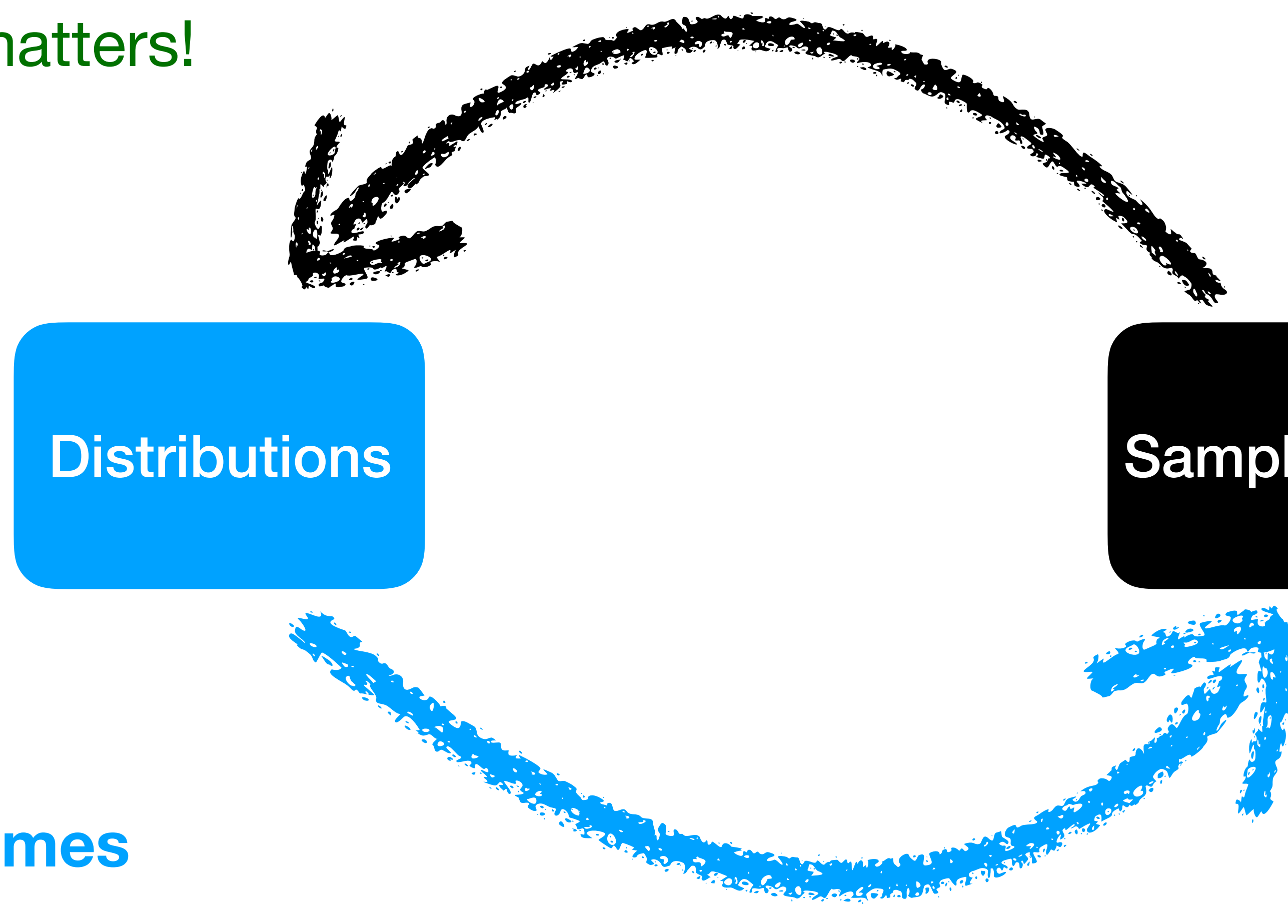
Looking forward..

Algorithmic and statistical guarantees for learning mixture models from samples when number of mixtures is not known

Data driven manifold learning: Low dimensional structure in deep networks

- Learning from data

Will the model generalize or not?
Choice of kernel matters!



Improving sample quality:

From Monte Carlo to Quasi Monte Carlo to reduce discrepancy

- **Mixing times**
- Function specific mixing times: Estimating mean and covariance

References

Fast MCMC algorithms on polytopes
https://arxiv.org/abs/1710.08165

Log-concave sampling: Metropolis Hastings Algorithms are fast!
http://arxiv.org/abs/1801.02309