# Fast MCMC Algorithms on Polytopes

Raaz Dwivedi, Department of EECS



Joint work with Yuansi Chen, Martin Wainwright, Bin Yu

# Random Sampling

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

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

# Applications

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

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

- Probabilities of Events
- Rare Event Simulations
- Bayesian Posterior Mean
- Volume Computation (polynomial time)

# Applications

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

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

- Probabilities of Events
- Rare Event Simulations
- Bayesian Posterior Mean
- Volume Computation (polynomial time)

# Applications

$$\min_{x \in \mathcal{K}} g(x)$$

- **Zeroth order optimization**: Polynomial time algorithms based on Random Walk
  - **Convex optimization**: Bertsimas and Vempala 2004, Kalai and Vempala 2006, Kannan and Narayanan 2012, Hazan et al. 2015
  - **Non-convex optimization, Simulated Annealing**: Aarts and Korst 1989, Rakhlin et al. 2015

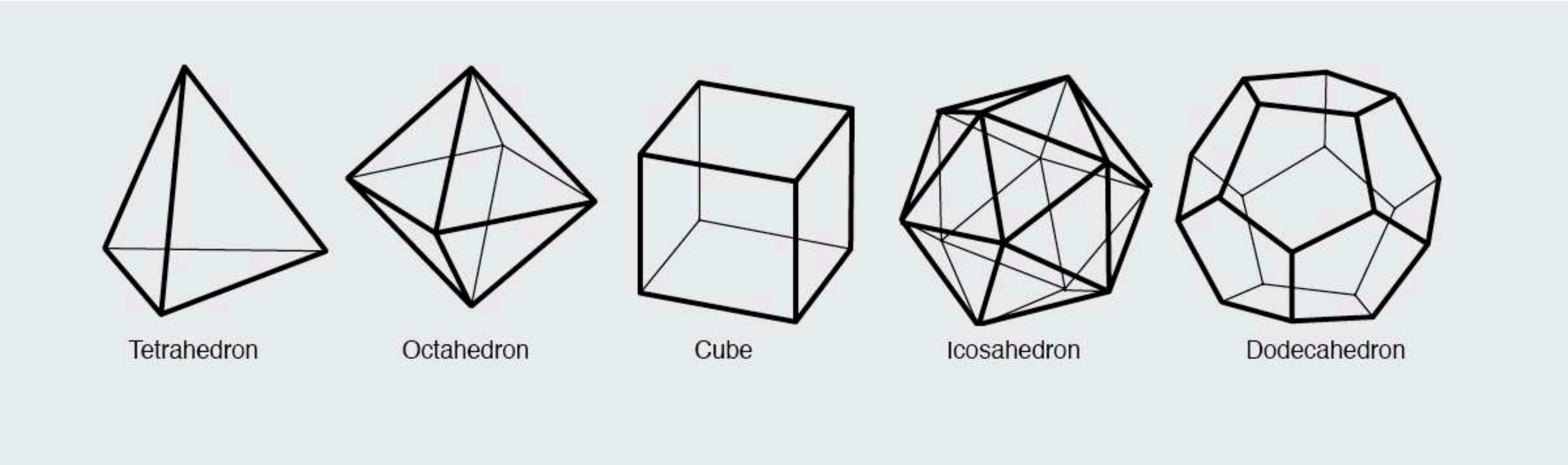
# Uniform Sampling on Polytopes

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

$n$ linear constraints

$d$ dimensions

$n > d$



# Uniform Sampling on Polytopes

- Integration of arbitrary functions under linear constraints
- Mixed Integer Programming
- Sampling non negative integer matrices with specified row and column sums (contingency tables)
- **Connections between optimization and sampling algorithms**

# Goal

Given A and b, and a starting distribution $\mu_0$,

design an MCMC algorithm

that generates a random sample from uniform distribution on

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

*in as few steps as possible*!

Convergence Rate: **Mixing time for total variation**

$$\|\mu_0 P^k - \pi^*\|_{\mathrm{TV}} \leq \epsilon$$

# Markov Chain Monte Carlo

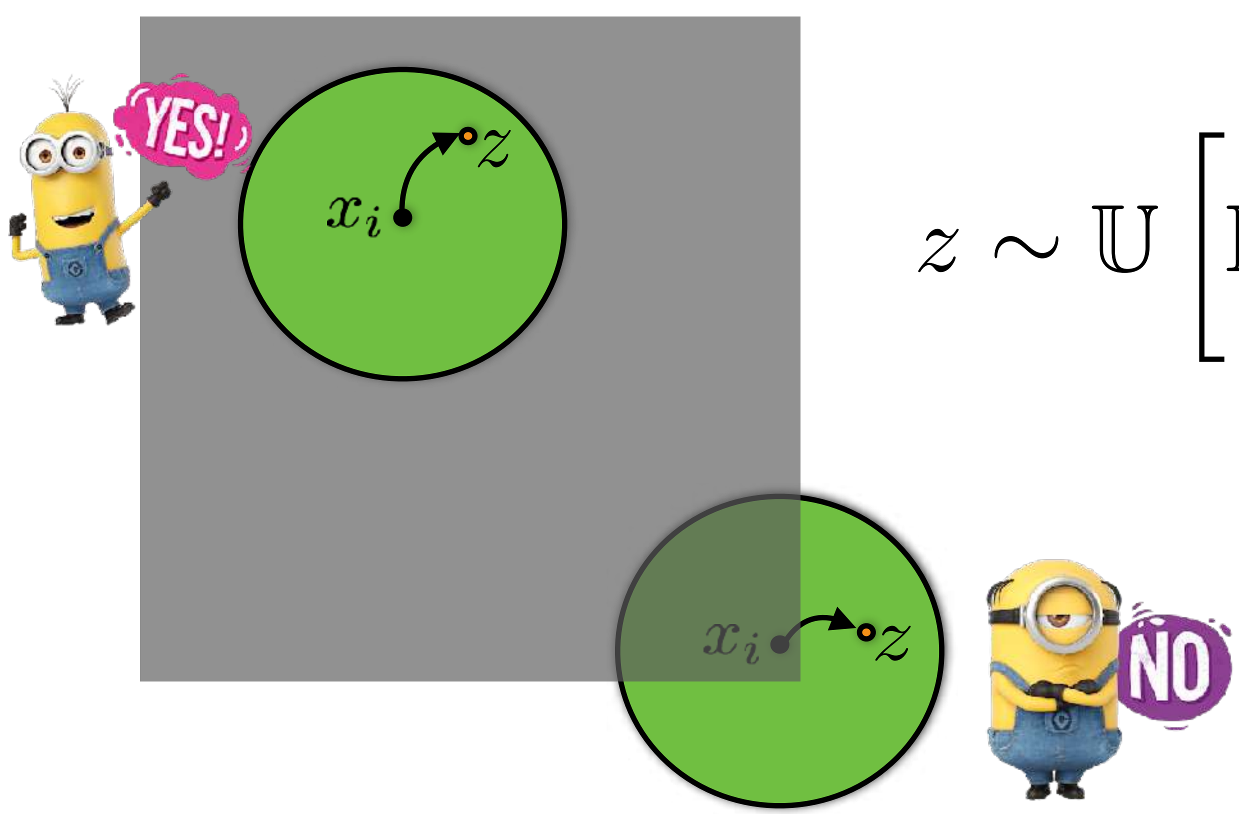
- **Design** a Markov Chain which can converge to the desired distribution
  - Metropolis Hastings Algorithms (1950s), Gibbs Sampling (1980s)
- **Simulate** the Markov chain for **several steps to get a sample**

# Markov Chain Monte Carlo

- Sampling on convex sets: **Ball Walk** (Lovász et al. 1990), **Hit-and-run** (Smith et al. 1993, Lovász 1999),
- Sampling on polytopes: **Dikin Walk** (Kannan and Hariharan 2012, Hariharan 2015, Sachdeva and Vishnoi 2016), **Geodesic Walk** (Lee and Vempala 2016)

# Ball Walk [Lovász and Simonovits 1990]

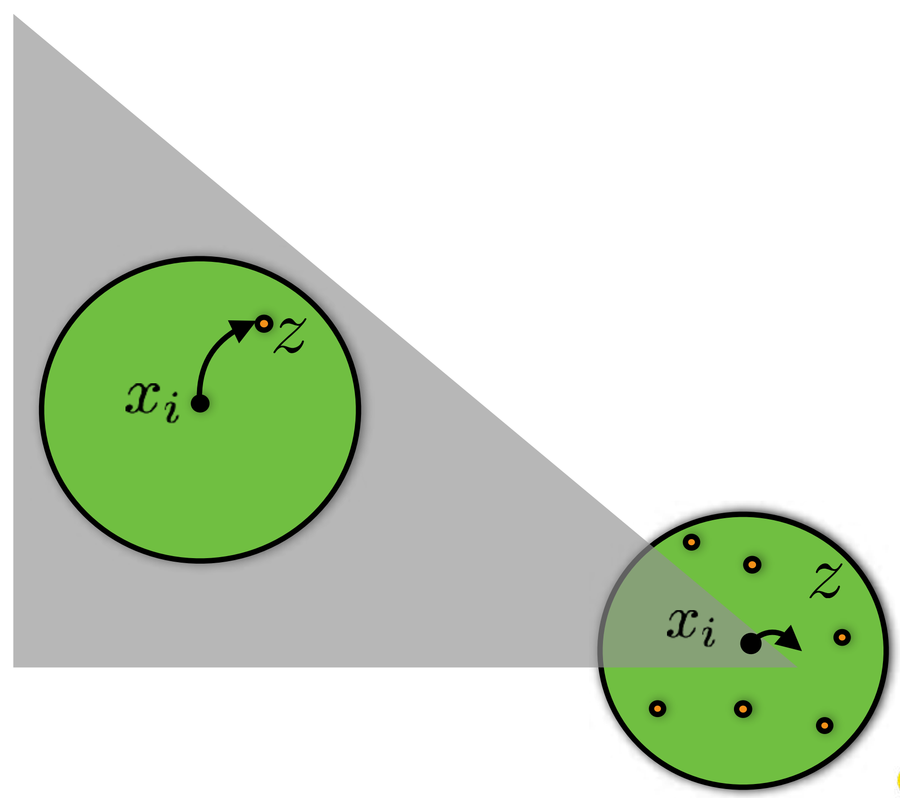
- Propose a uniform point in a ball around x
  - reject if outside the polytope, else move to it




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

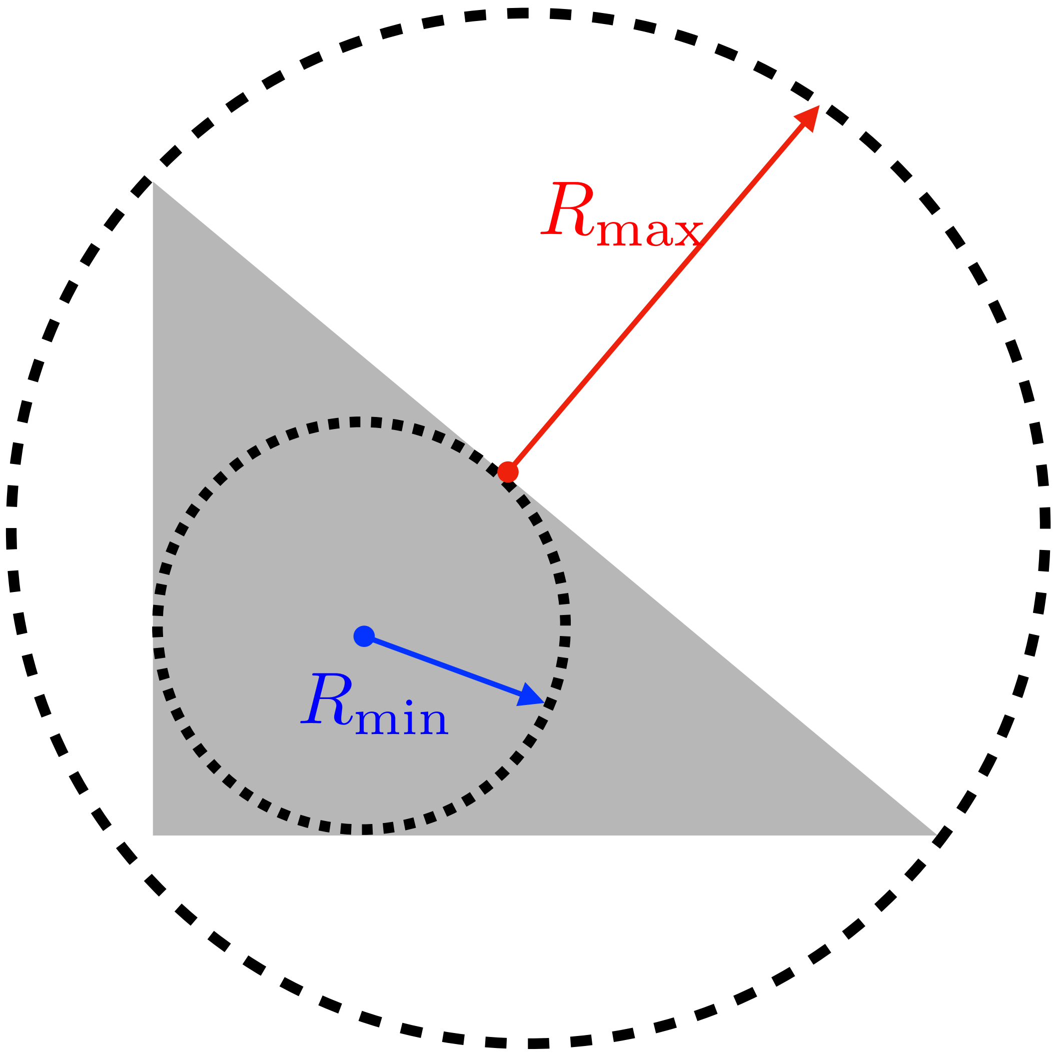
# Ball Walk [Lovász and Simonovits 1990]

- Many rejections near sharp corners




# Ball Walk [Lovász and Simonovits 1990]

- Mixing time depends on *conditioning* of the set

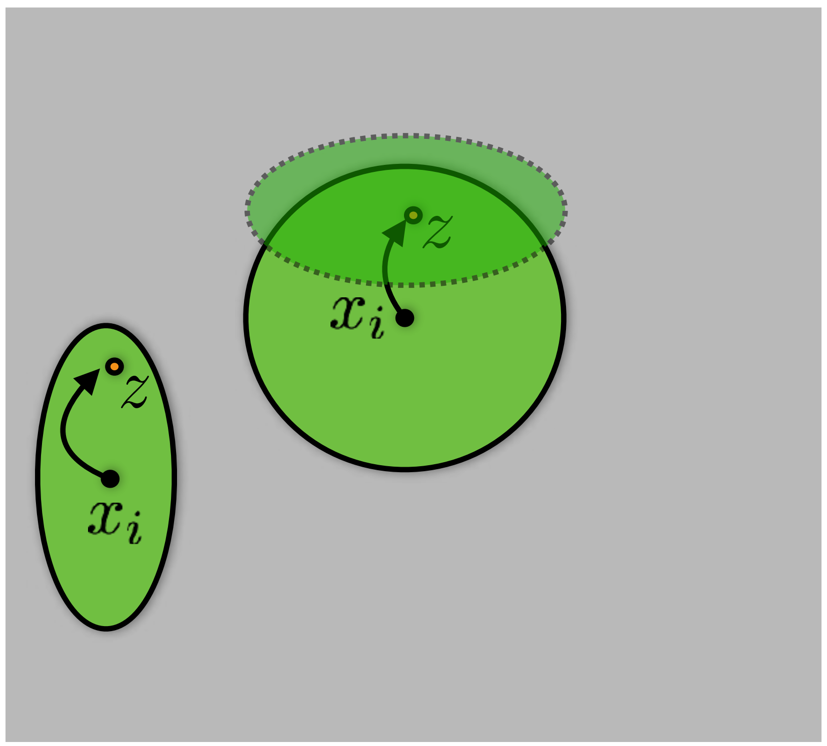



Can be exponential in d

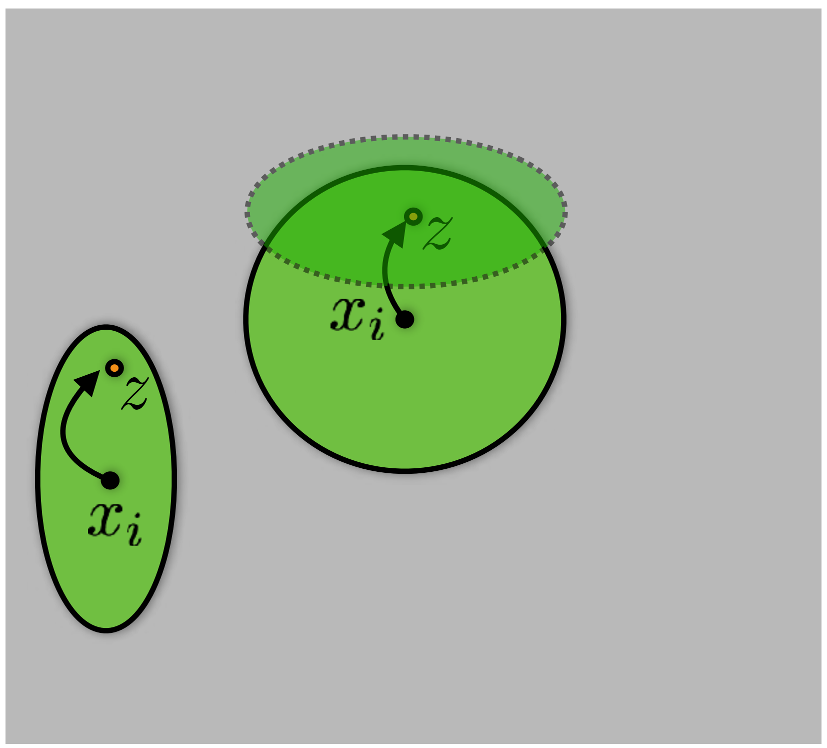
$$\#\text{steps} = \mathcal{O}\left(d^2 \; \frac{R_{\max}^2}{R_{\min}^2}\right)$$

$$\text{per step cost} = nd$$

# May be a variable shape ellipsoid?



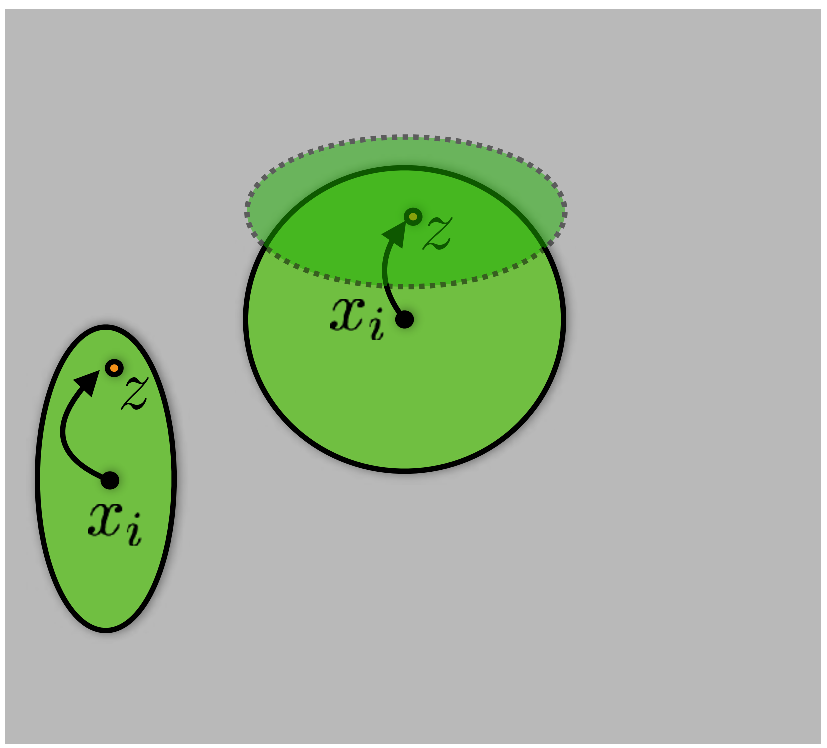
# Dikin Walk [Kannan and Narayanan 2012]




- Proposal $z \sim \mathcal{N}\left(x, \frac{r^2}{d} D_x^{-1}\right)$
- Another variant $z \sim \mathbb{U}\left[D_x(r)\right]$
- Accept Reject:

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

# Dikin Walk [Kannan and Narayanan 2012]




- Proposal $z \sim \mathcal{N}\left(x, \frac{r^2}{d} D_x^{-1}\right)$

$$D_x = \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

$$A = \begin{bmatrix} \text{—} a_1^\top \text{—} \\ \text{—} a_2^\top \text{—} \\ \vdots \\ \text{—} a_n^\top \text{—} \end{bmatrix} \quad \mathcal{K} = \{x \in \mathbb{R}^d | Ax \leq b\}$$

Log Barrier Method (Optimization) [Dikin 1967, Nemirovski 1990]

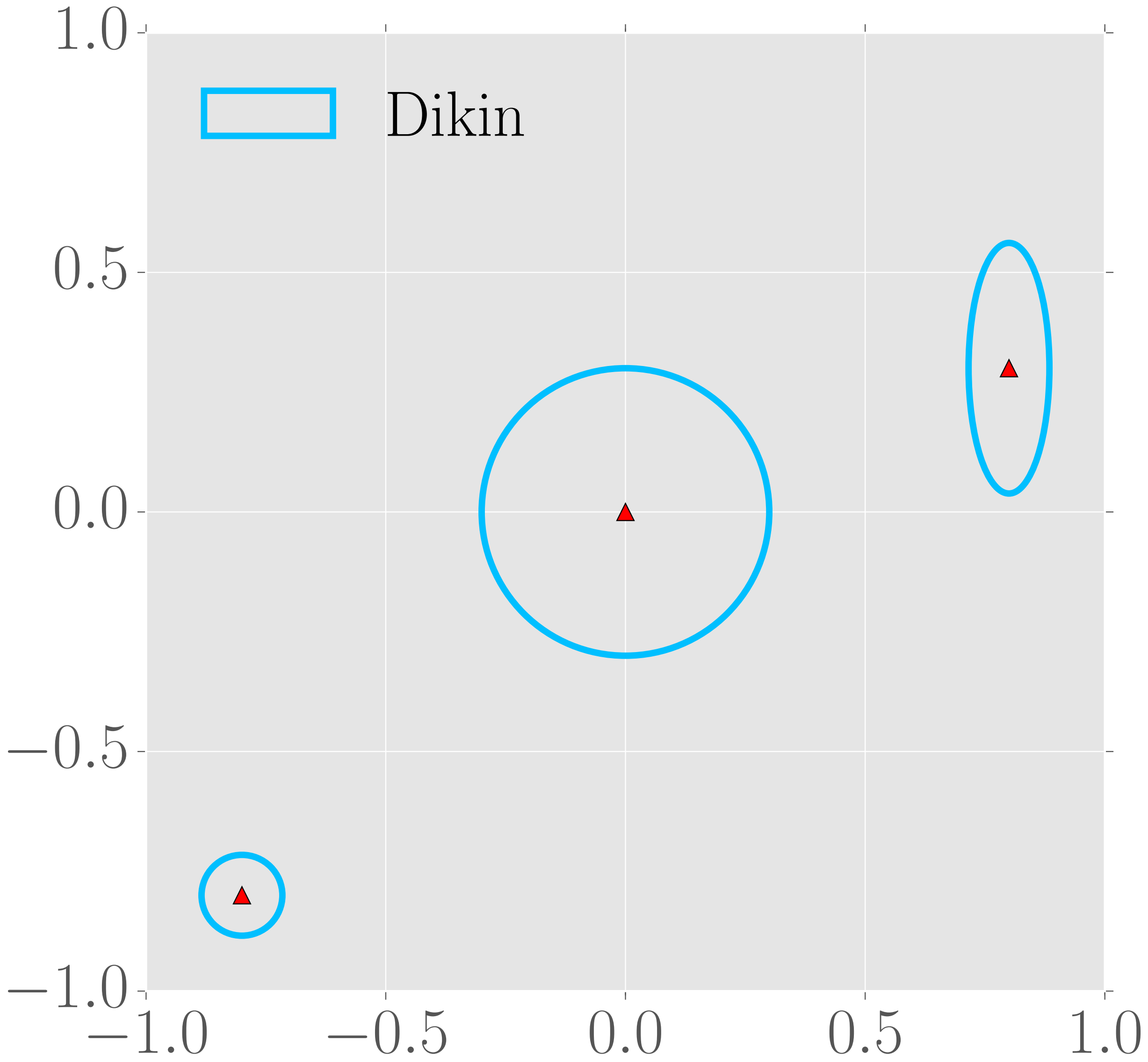
# Upper bounds

| | Ball Walk | Dikin Walk | ? | ? |
|---|---|---|---|---|
| #Steps | $d^2 \ \dfrac{R^2_{\max}}{R^2_{\min}}$ | $nd$ | | |
| Per Step Cost | $nd$ | $nd^2$ | | |

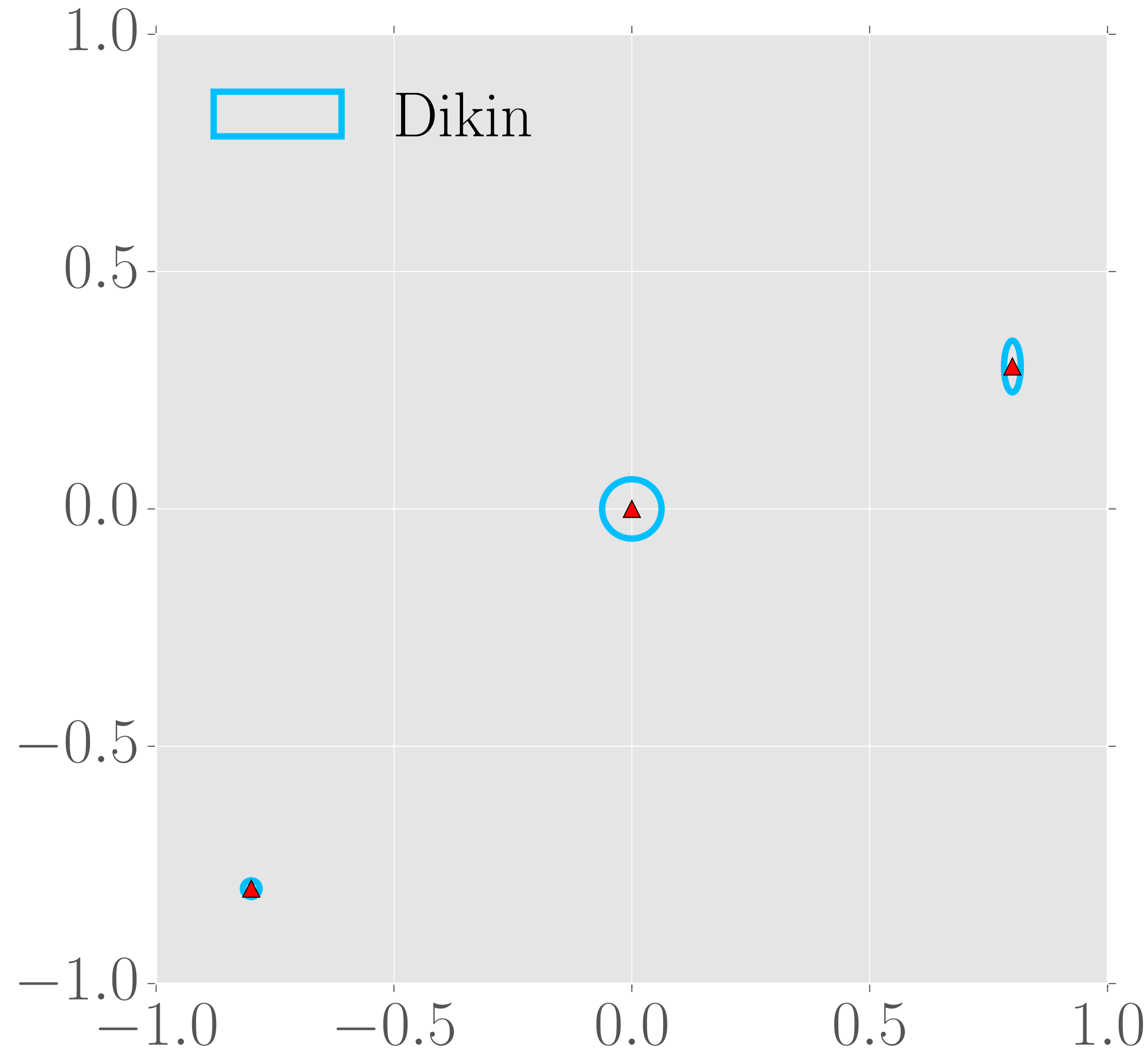
n = #constraints
d = #dimensions
n > d

# Slow mixing of Dikin Walk

#constraints = **4**




#constraints = **128**



“If any two points that are $\Delta$ apart have $\rho$ overlap in their transition regions, then the chain mixes in $\mathcal{O}\left(\frac{1}{\Delta^2\rho^2}\right)$ steps.”

*–Lovász’s Lemma*

(Distance and overlap measured in appropriately)

"If any two points that are $\Delta$ apart have $\rho$ overlap in their transition regions, then the chain mixes in $\mathcal{O}\left(\frac{1}{\Delta^2\rho^2}\right)$ steps."

*–Lovász's Lemma*

For any fixed overlap $\rho$ , we want far away points to have $\rho$ overlapping regions, and hence large ellipsoids (contained within the polytope) are useful.

# Improving Dikin Walk

$$D_x = \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

**Importance weighting of constraints**

$$\sum_{i=1}^{n} \color{red}{w_i(x)} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

# Improving Dikin Walk

[Kannan and Narayanan 2012]

Dikin Proposal

$$z \sim \mathcal{N}\left(x, \frac{\mathbf{r^2}}{\mathbf{d}}\mathbf{D_x^{-1}}\right)$$

$$D_x = \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

Log Barrier Method
[Dikin 1967, Nemirovski 1990]

# Sampling meets optimization (again!!)

[Kannan and Narayanan 2012]

Dikin Proposal

$$z \sim \mathcal{N}\left(x, \frac{\mathbf{r^2}}{\mathbf{d}} \mathbf{D_x^{-1}}\right)$$

$$D_x = \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

Log Barrier Method
[Dikin 1967, Nemirovski 1990]

[Chen, D., Wainwright and Yu 2017]

Vaidya Proposal

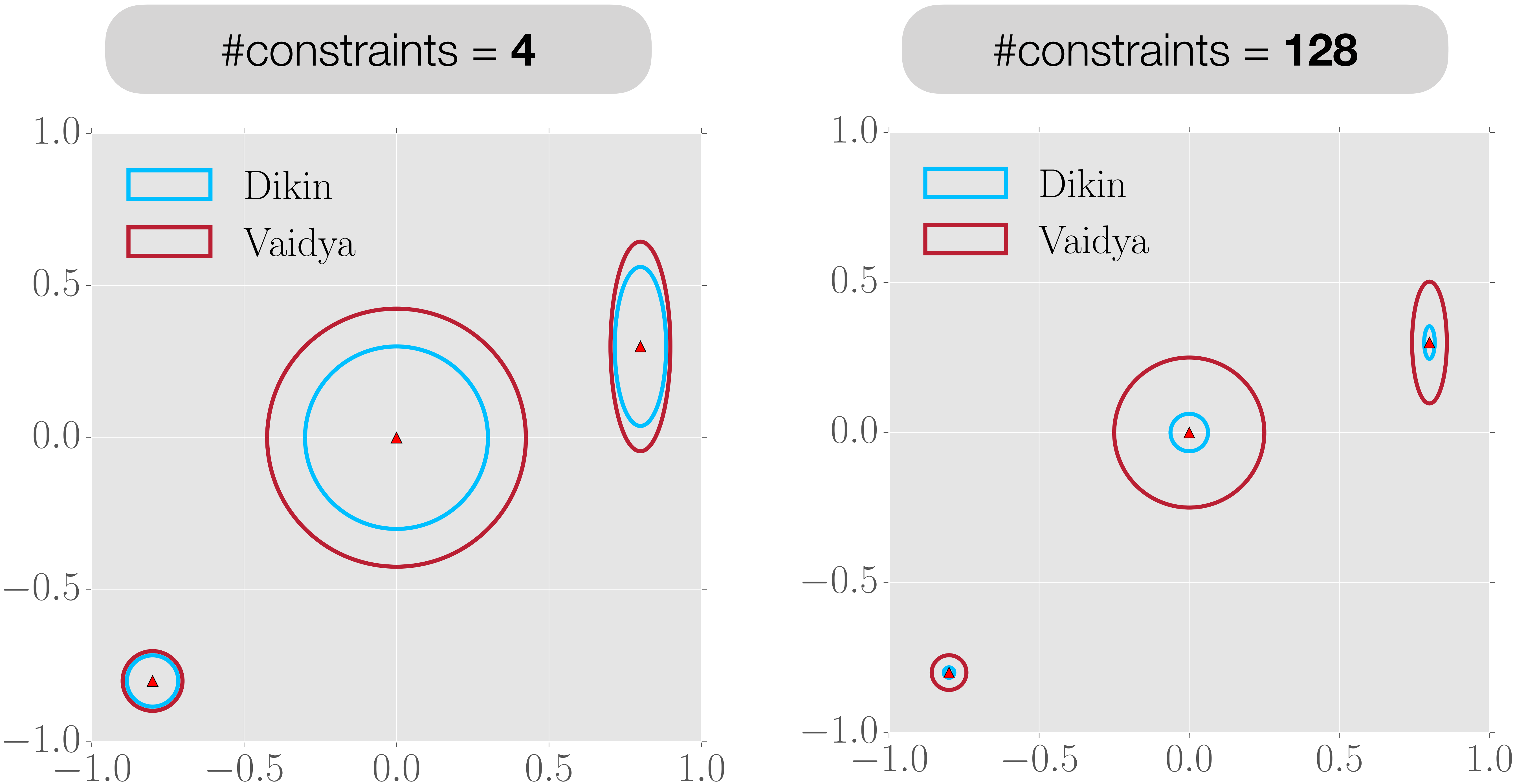
$$z \sim \mathcal{N}\left(x, \frac{r^2}{\sqrt{nd}} V_x^{-1}\right)$$

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

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

Volumetric Barrier Method
[Vaidya 1993]

# Vaidya Walk [Chen, D., Wainwright, Yu 2017]



# Convergence Rates

| | Ball Walk | Dikin Walk | Vaidya Walk |
|---|---|---|---|
| #Steps | $d^2 \; \frac{R_{\max}^2}{R_{\min}^2}$ | $nd$ | $n^{0.5}d^{1.5}$ |
| Per Step Cost | | | |

n constraints
d dimensions
n > d

# Convergence Rates

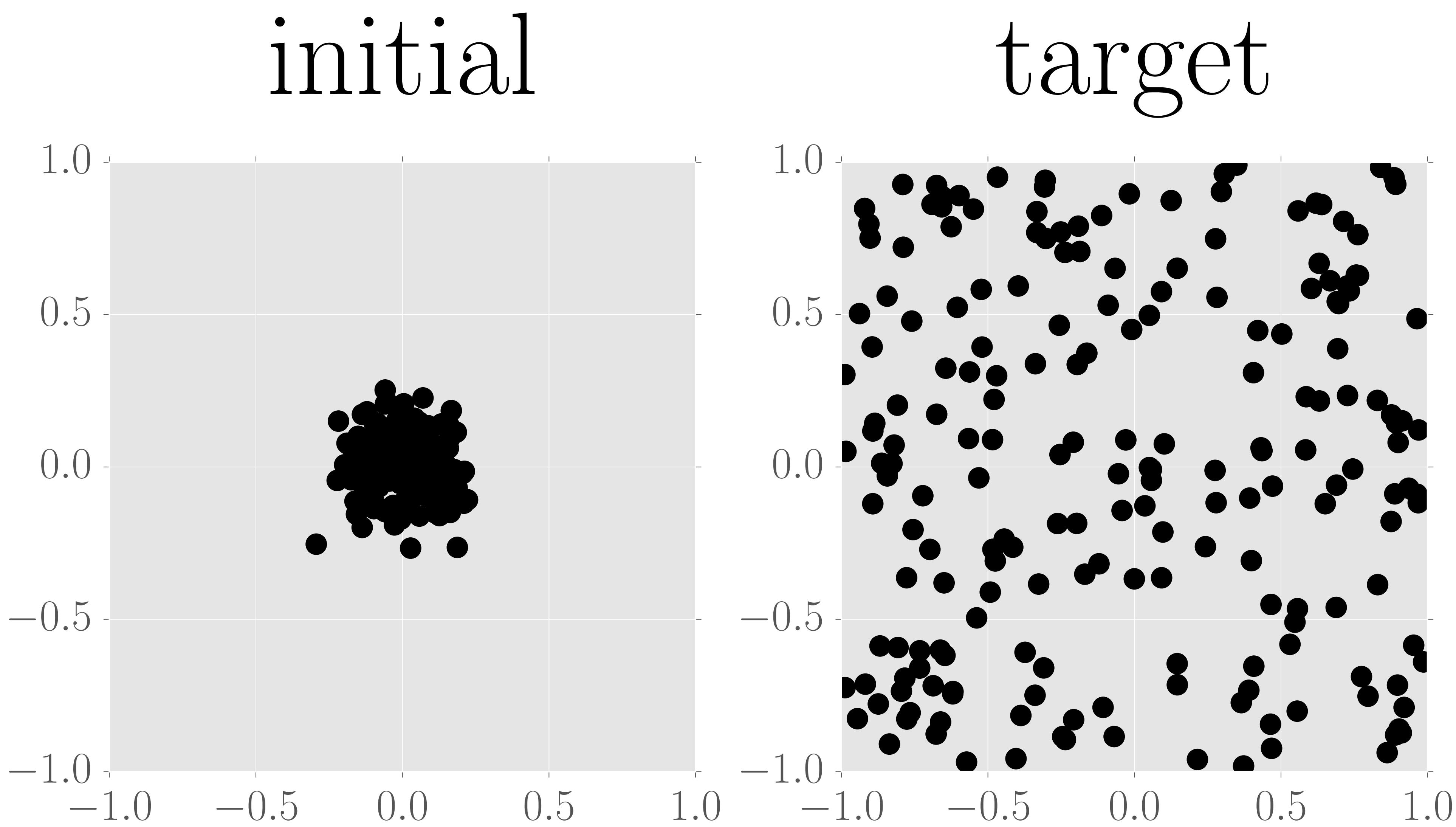
| | **Ball Walk** | **Dikin Walk** | **Vaidya Walk** | |
|---|---|---|---|---|
| **#Steps** | $d^2 \frac{R_{\max}^2}{R_{\min}^2}$ | $nd$ | $n^{0.5} d^{1.5}$ |  |
| **Per Step Cost** | $nd$ | $nd^2$ | $nd^2$ | **n constraints<br>d dimensions<br>n > d** |

# Dikin Walk vs Vaidya Walk

#dimensions = **2**

#experiments = **200**

k = #iterations




$k = 0$

$k = \infty$

# Dikin Walk vs Vaidya Walk

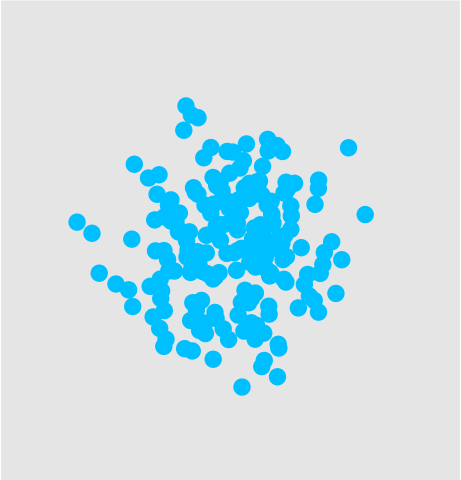
#constraints = **64**

#experiments = **200**

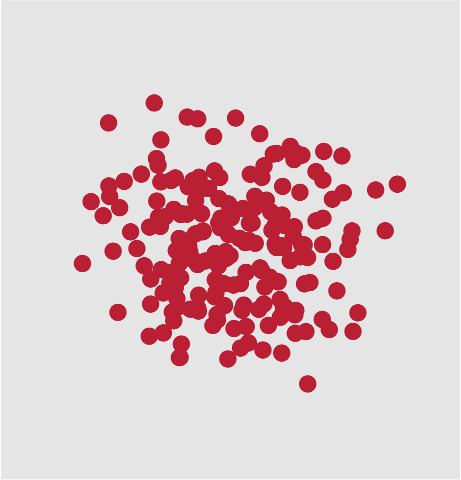
k = #iterations

$k = 10$

Dikin Walk



Vaidya Walk



# Dikin Walk vs Vaidya Walk

#constraints = **64**

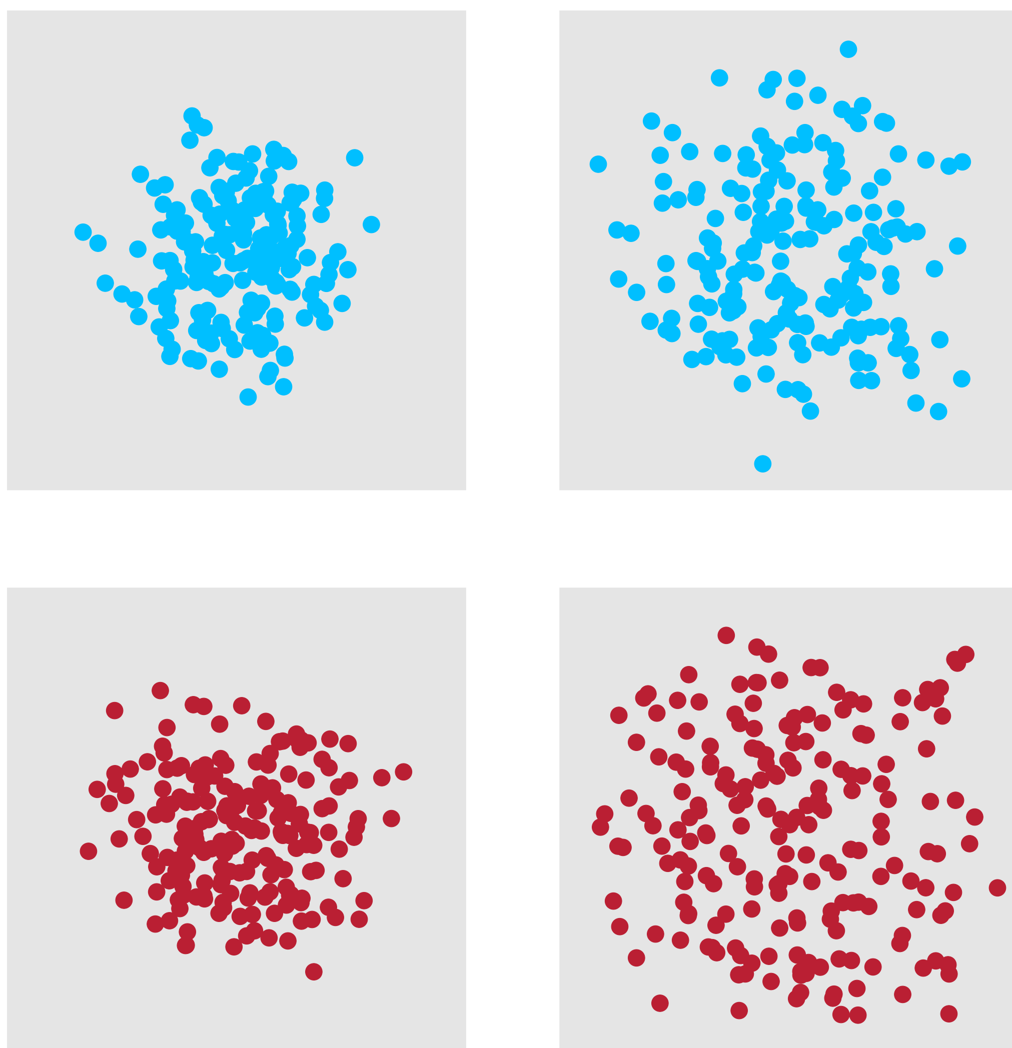
#experiments = **200**

k = #iterations

$k = 10$ $k = 100$

Dikin Walk

Vaidya Walk

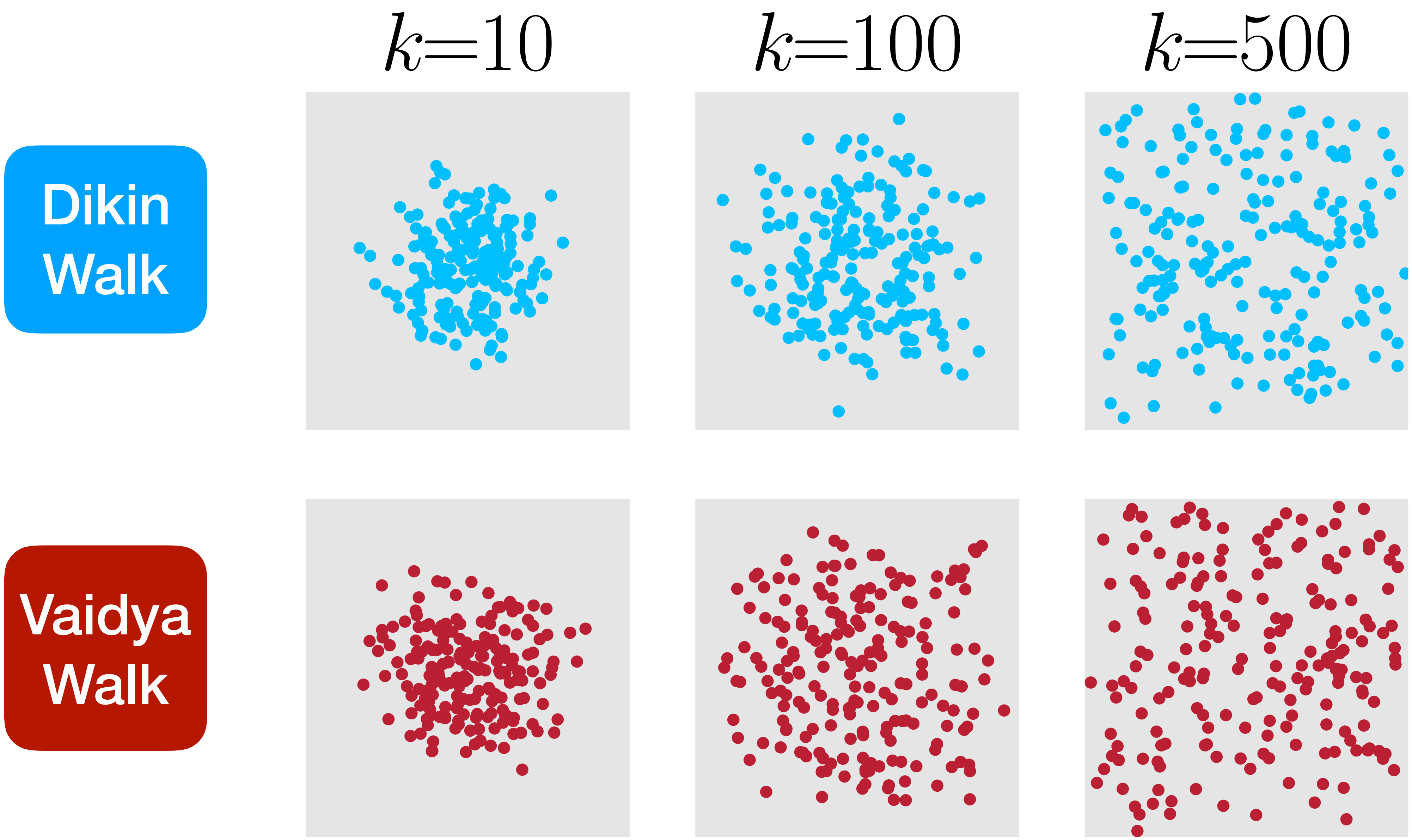


# Dikin Walk vs Vaidya Walk

#constraints = **64**

#experiments = **200**

k = #iterations

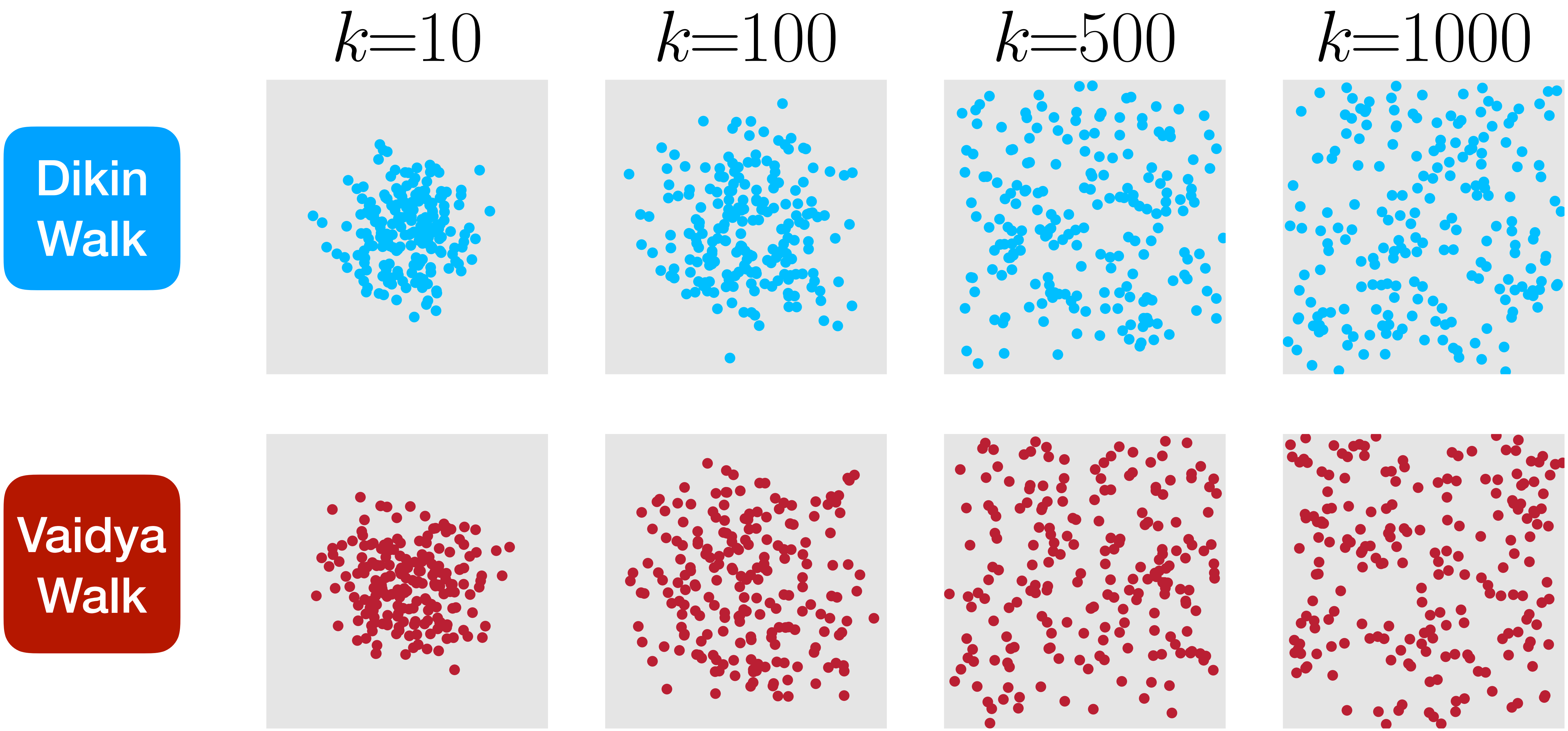


# Small number of constraints: No Winner!

#constraints = **64**

#experiments = **200**

k = #iterations



# Dikin Walk vs Vaidya Walk

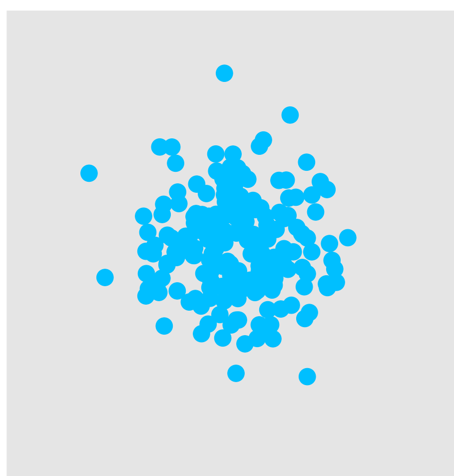
#constraints = **2048**

#experiments = **200**

k = #iterations

$k = 10$

**Dikin Walk**



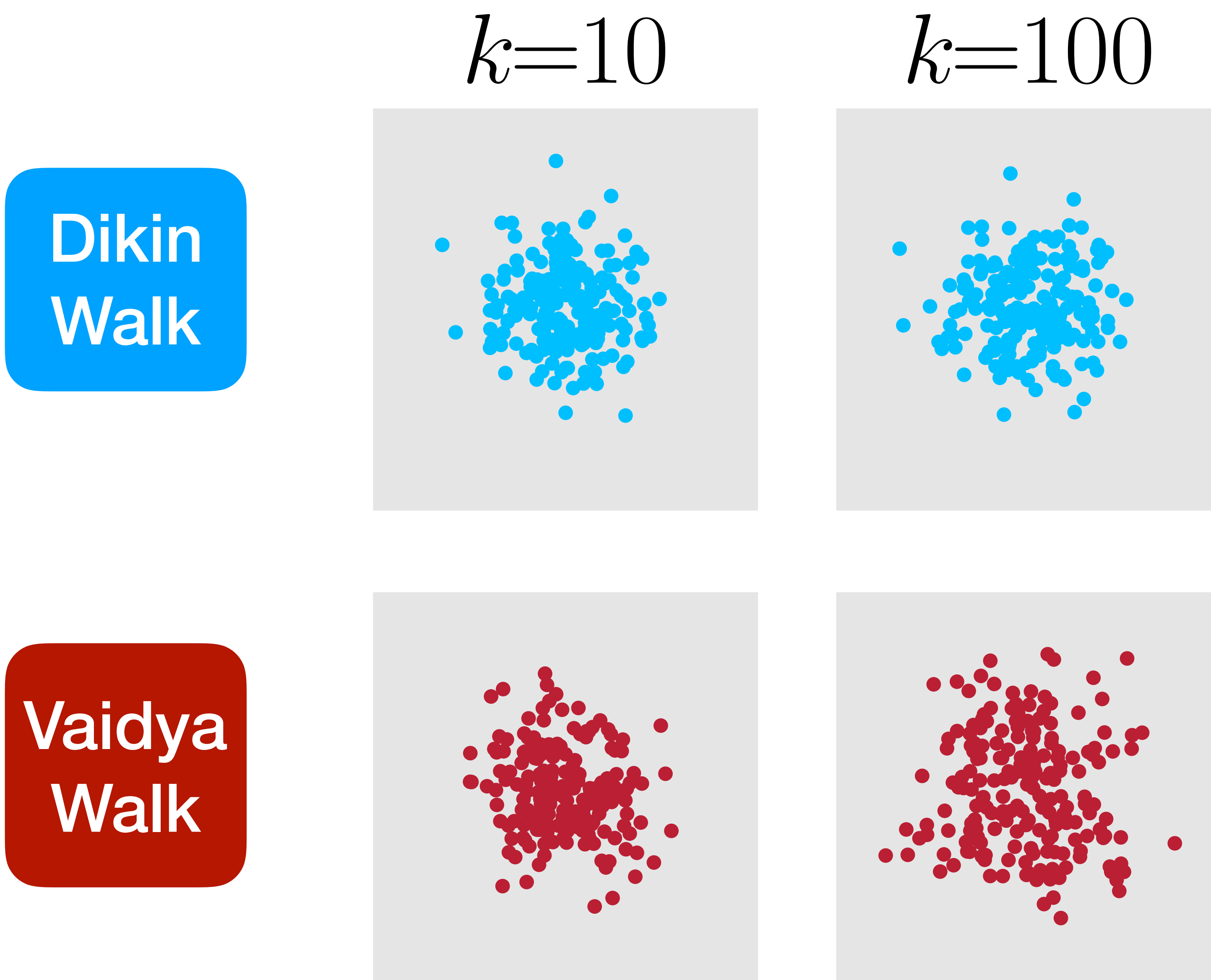
**Vaidya Walk**

# Dikin Walk vs Vaidya Walk

#constraints = **2048**

#experiments = **200**

k = #iterations



# Dikin Walk vs Vaidya Walk

#constraints = **2048**

#experiments = **200**

k = #iterations

$k$=10 $k$=100 $k$=500

Dikin Walk

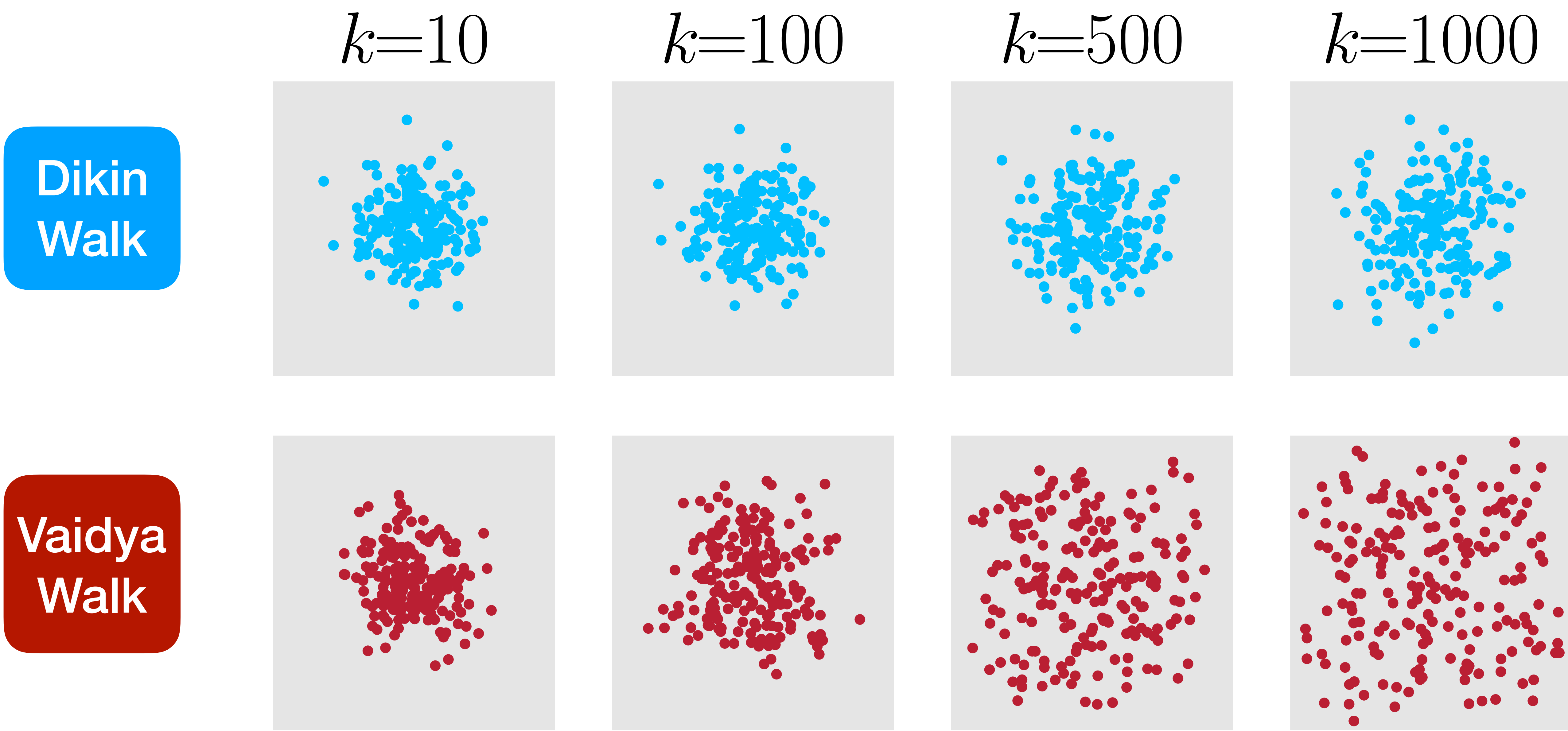
Vaidya Walk

# Vaidya walk wins!

#constraints = **2048**

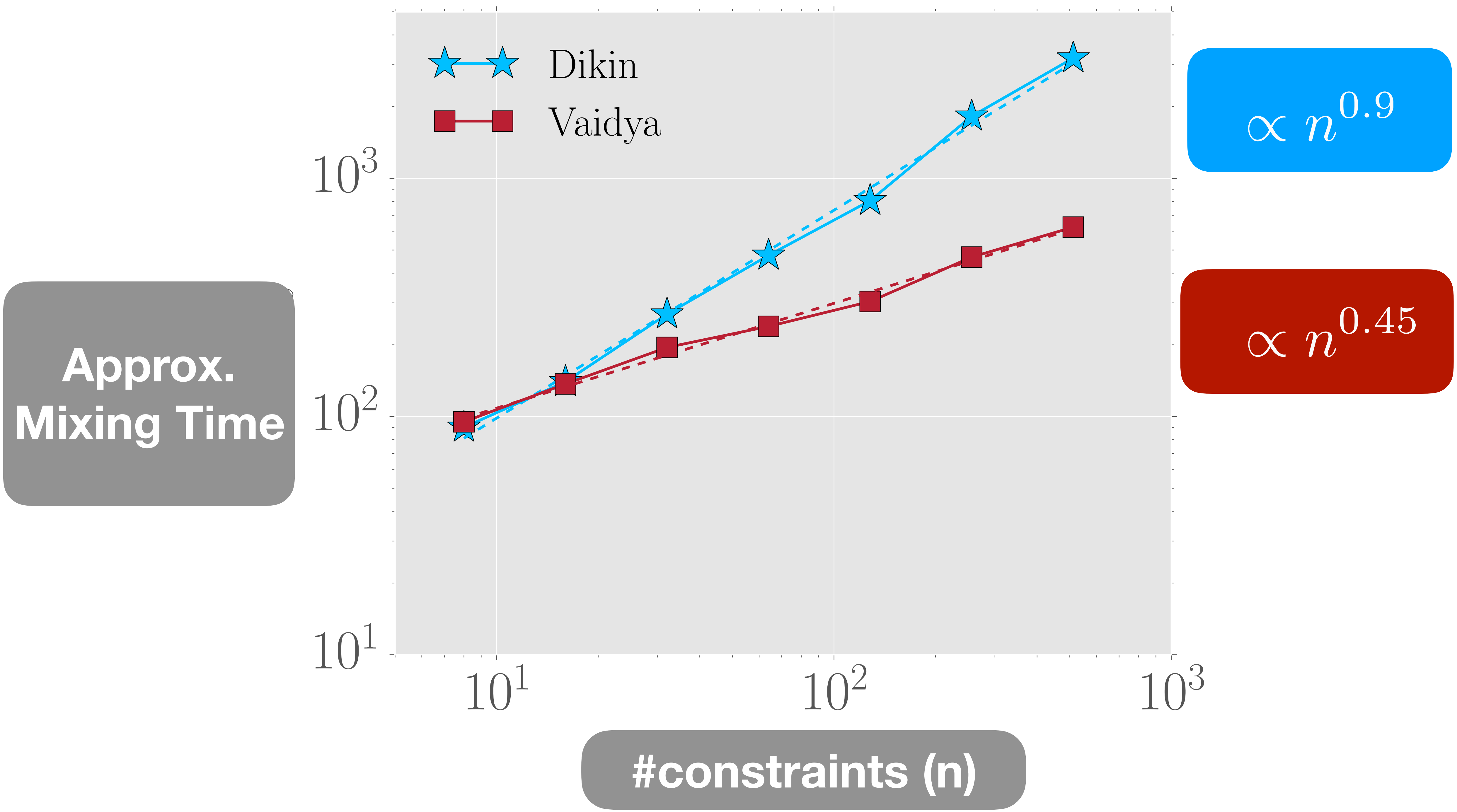
#experiments = **200**

k = #iterations

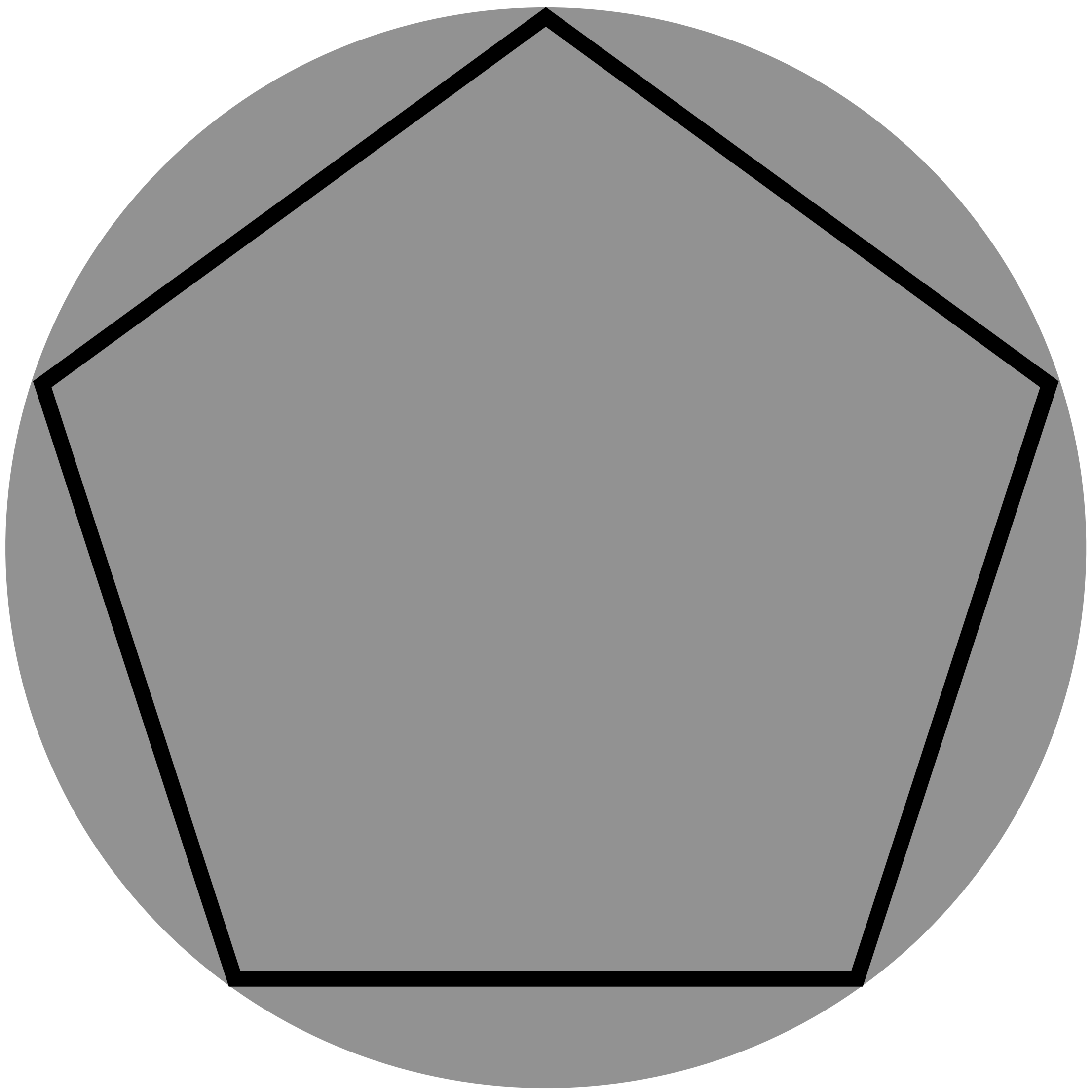


# Dikin Walk vs Vaidya Walk

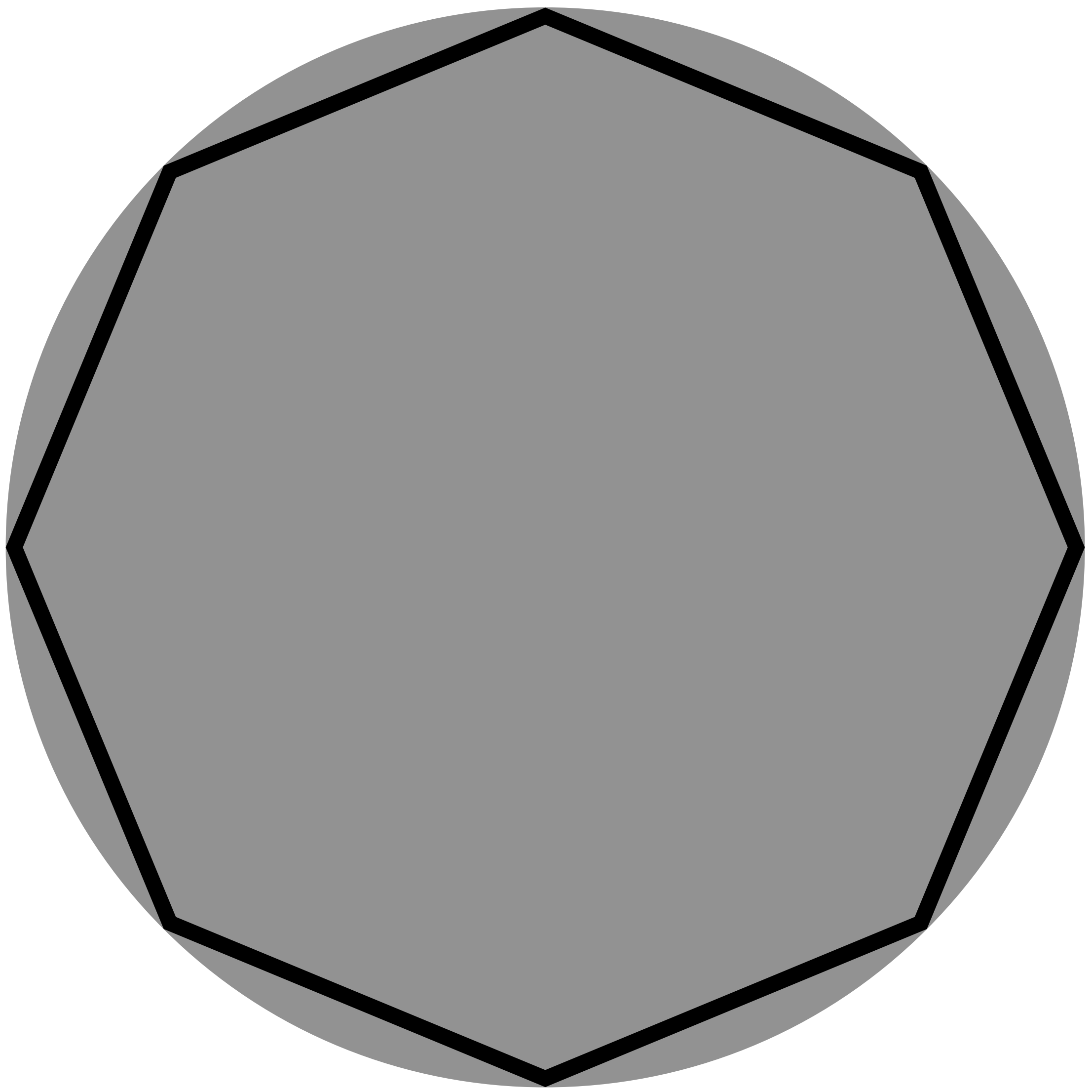
$\mathcal{O}(nd)$ **vs** $\mathcal{O}(n^{0.5}d^{1.5})$



# Polytope approximation to Circle

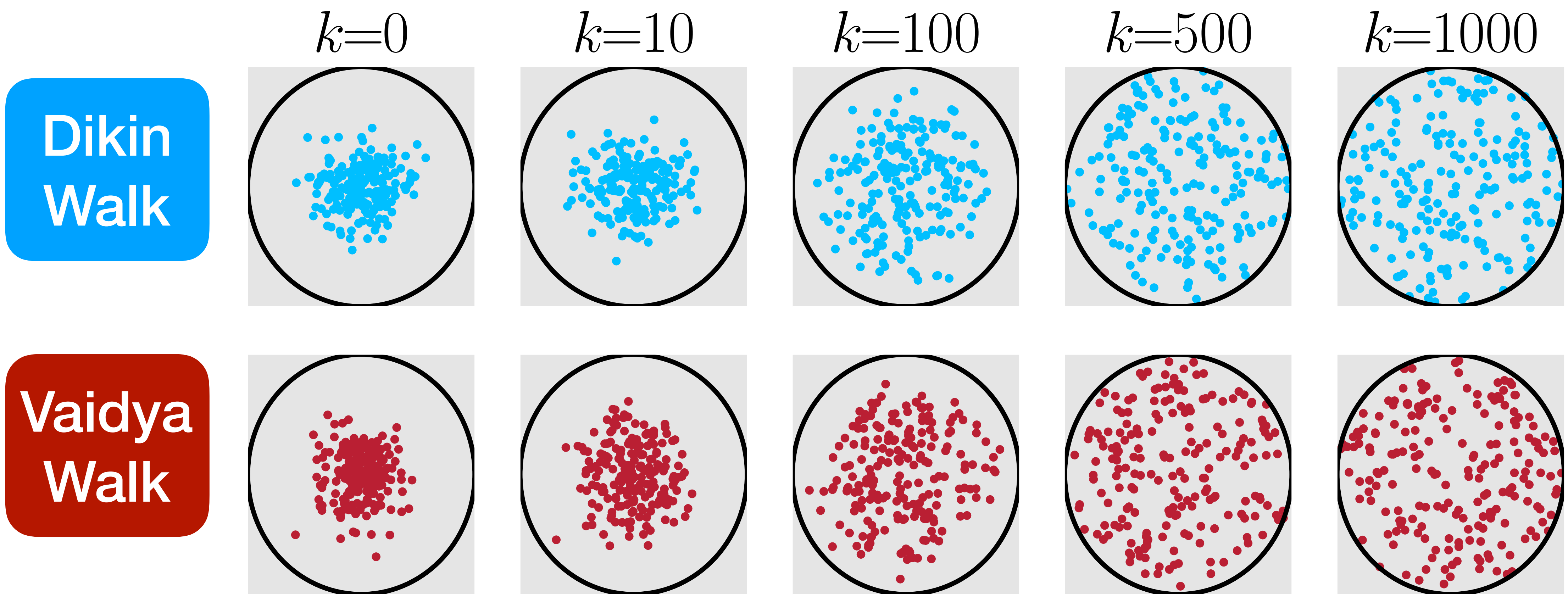


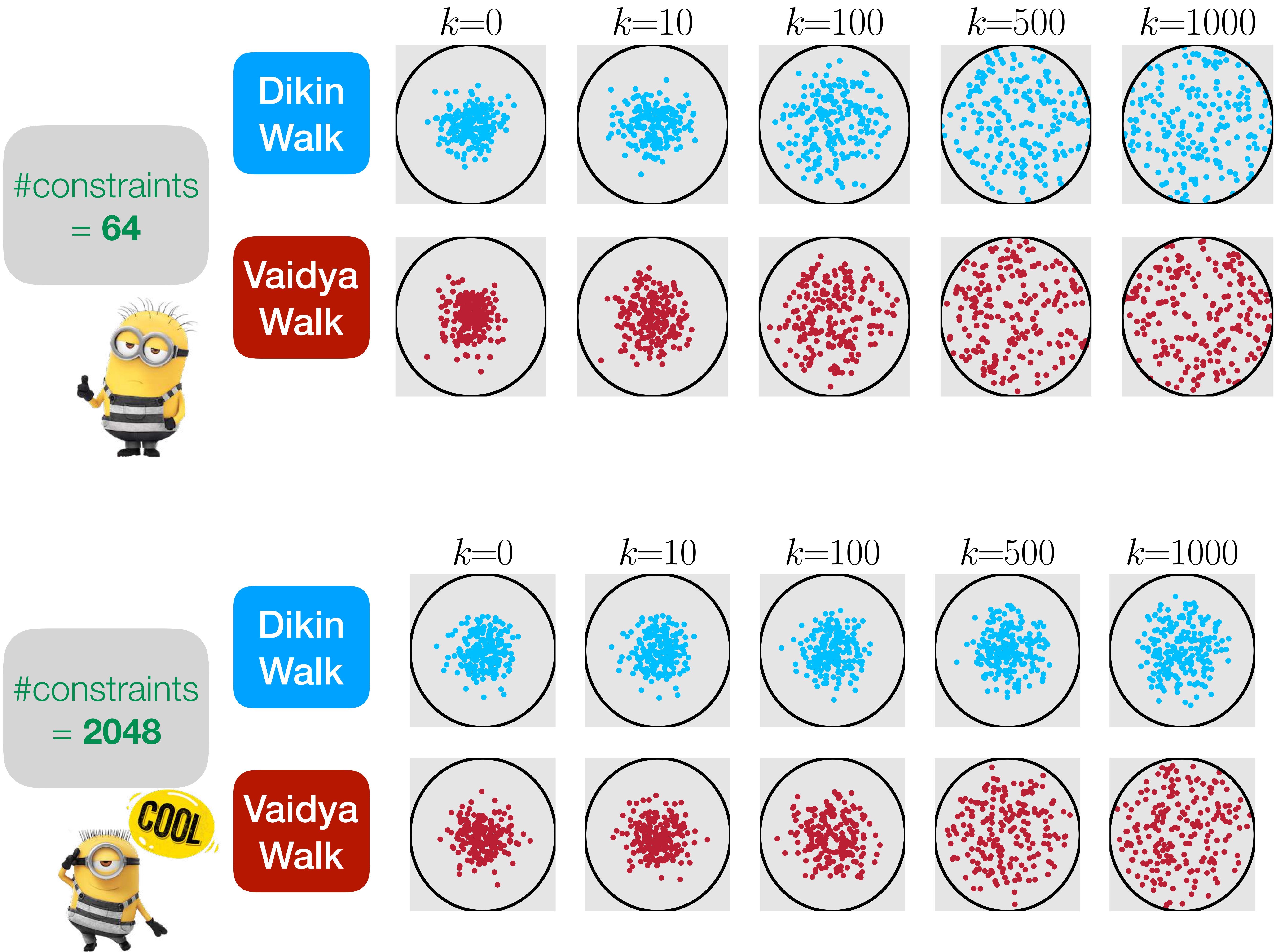
#constraints = **5**



#constraints = **8**

#constraints = 64





k=0
k=10
k=100
k=500
k=1000
#constraints = 64
Dikin Walk
Vaidya Walk
k=0
k=10
k=100
k=500
k=1000
#constraints = 2048
Dikin Walk
Vaidya Walk
COOL

# Can we improve further?

[Kannan and Narayanan, 2012]

Dikin Proposal

$$z \sim \mathcal{N}\left(x, \frac{\mathbf{r^2}}{\mathbf{d}} \mathbf{D_x^{-1}}\right)$$

$$D_x = \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

Vaidya Proposal

$$z \sim \mathcal{N}\left(x, \frac{r^2}{\sqrt{nd}} V_x^{-1}\right)$$

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

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

Log Barrier Method
[Dikin 1967, Nemeirovski 1990]

Vaidya's Volumetric Barrier Method
[Vaidya 1993]

# John Walk

[Kannan and Narayanan, 2012]

[Chen, D., Wainwright, Yu 2017]

Dikin Proposal

$$z \sim \mathcal{N}\left(x, \frac{\mathbf{r^2}}{\mathbf{d}} \mathbf{D_x^{-1}}\right)$$

$$D_x = \sum_{i=1}^{n} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}$$

Vaidya Proposal

$$z \sim \mathcal{N}\left(x, \frac{r^2}{\sqrt{nd}} V_x^{-1}\right)$$

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

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

John Proposal

$$z \sim \mathcal{N}\left(x, \frac{r^2}{d^{1.5}} J_x^{-1}\right)$$

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

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

Log Barrier Method
[Dikin 1967, Nemirovski 1990]

Vaidya's Volumetric Barrier Method
[Vaidya 1993]

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

# Mixing Times

n = #constraints
d = #dimensions
n > d

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

# Mixing Times

n = #constraints
d = #dimensions
n > d

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

# Conjecture

n = #constraints
d = #dimensions
n > d

| | Dikin Walk | Vaidya Walk | John Walk |
|---|---|---|---|
| #Steps | $nd$ | $n^{0.5}d^{1.5}$ | $d^2 \log^c \left(\frac{n}{d}\right)$ |
| Per Step Cost | $nd^2$ | $nd^2$ | $nd^2 \log^2 n$ |

“

For the John walk, the log factors are bottleneck in practice.

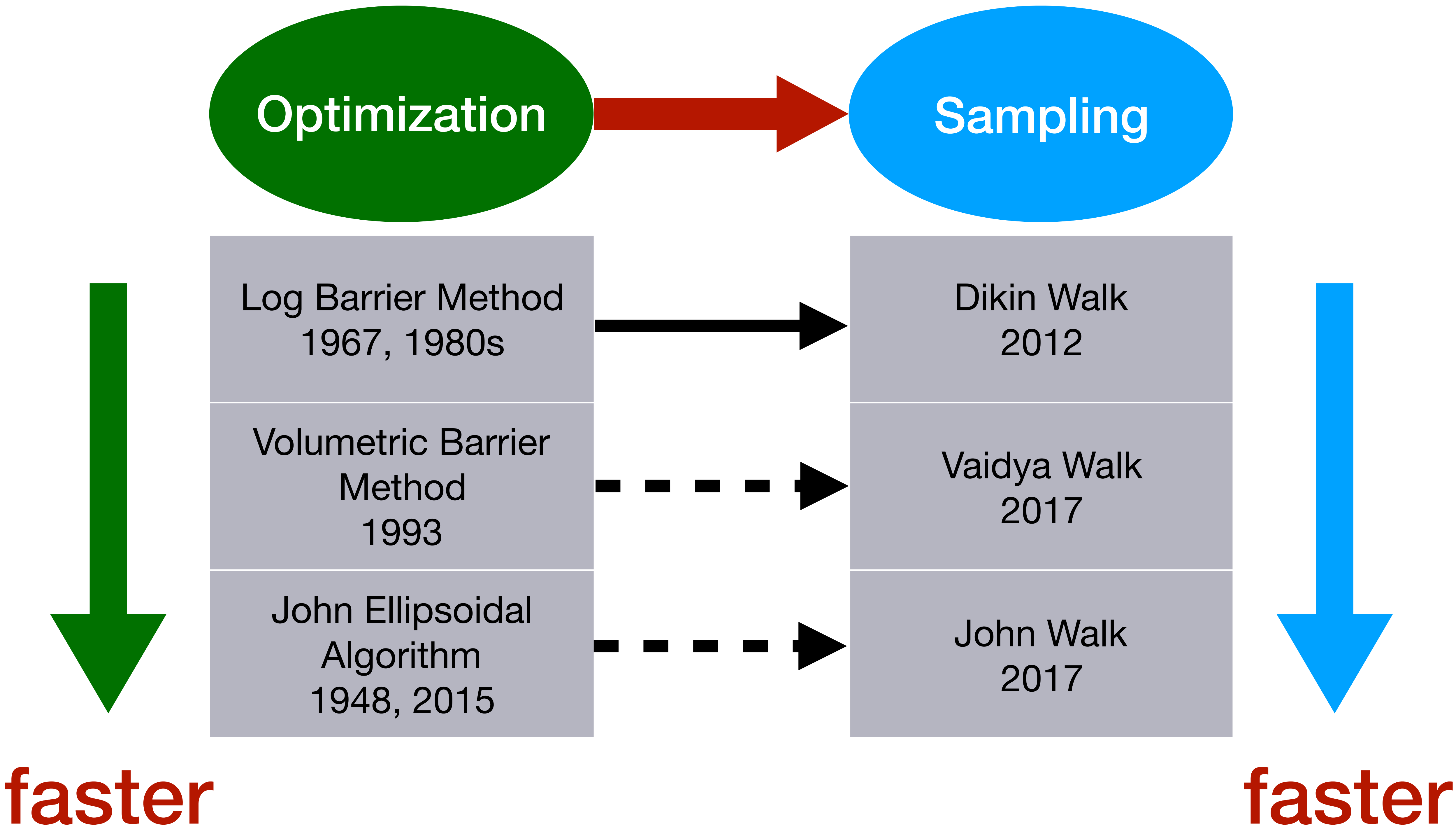
”

– *Numerical Experiments*

# Proof Idea

- Proof relies on Lovasz’s Lemma
- Need to establish that near by points have similar transition distributions
- Have to show that the weighted matrices are sufficiently smooth — *use of weights makes it involved*

# Summary


Optimization
Sampling
Log Barrier Method
1967, 1980s
Dikin Walk
2012
Volumetric Barrier
Method
1993
Vaidya Walk
2017
John Ellipsoidal
Algorithm
1948, 2015
John Walk
2017
faster
faster