PASCAL

A Parallel Algorithmic SCALable Framework for N-body Problems

Laleh Aghababaie Beni, Aparna Chandramowlishwaran

Euro-Par 2017

Outline

- Introduction
- PASCAL Framework
  - Space Partitioning Trees
  - Tree Traversal
  - Prune/Approximate Generators
- Optimizations & Parallelization
- Experiments & Results
- Conclusions & Future Work

Outline

- Introduction
- PASCAL Framework
    - Space Partitioning Trees
    - Tree Traversal
    - Prune/Approximate Generators
- Optimizations & Parallelization
- Experiments & Results
- Conclusions & Future Work

N-body calculations

$$\forall q \in Q : \quad F(q) = \sum_{r \in (Q - \{q\})} C \frac{r - q}{||r - q||^3}$$ *Force computation*

$$\forall q \in Q : \quad \text{AllNN}(q) = \text{argmin}_{r \in R} \, \text{d}(q, r)$$ *Nearest neighbors*

$$\forall q \in Q : \quad \text{KDE}(q) = \frac{1}{|R|} \sum_{r \in R} K(q, r)$$ *Kernel density estimation*

$$\forall q \in Q : \quad \text{Range}(q) = \sum_{r \in R} I(\text{dist}(q, r)) \leq h)$$ *Range count*

N-body calculations

What do these have in common?

$$\forall q \in Q : \quad F(q) = \sum_{r \in (Q - \{q\})} C \frac{r - q}{||r - q||^3}$$

Force computation

$$\forall q \in Q : \quad \text{AllNN}(q) = \text{argmin}_{r \in R}\, \text{d}(q, r)$$

Nearest neighbors

$$\forall q \in Q : \quad \text{KDE}(q) = \frac{1}{|R|} \sum_{r \in R} K(q, r)$$

Kernel density estimation

$$\forall q \in Q : \quad \text{Range}(q) = \sum_{r \in R} I(\text{dist}(q, r)) \leq h)$$

Range count

N-body calculations

What do these have in common?

$$\forall q \in Q: \qquad F(q) = \sum_{r \in (Q - \{q\})} C \frac{r - q}{||r - q||^3}$$

Force computation

$$\forall q \in Q: \qquad \text{AllNN}(q) = \text{argmin}_{r \in R}\, \text{d}(q, r)$$

Nearest neighbors

$$\forall q \in Q: \qquad \text{KDE}(q) = \frac{1}{|R|} \sum_{r \in R} K(q, r)$$

Kernel density estimation

$$\forall q \in Q: \qquad \text{Range}(q) = \sum_{r \in R} I(\text{dist}(q, r)) \leq h)$$

Range count

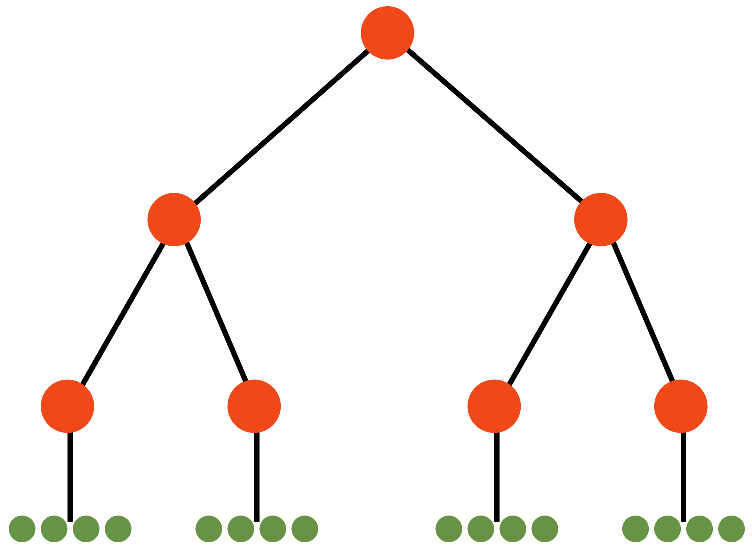
Consider pairs of points – naïvely O(N^2)

Commonality: Optimal approximation algorithms

$$\forall q \in Q : \qquad F(q) = \sum_{r \in (Q - \{q\})} C \frac{r - q}{||r - q||^3}$$

Force computation

- Hierarchical tree-based approximation algorithms for force computations, *e.g.*, Barnes-Hut or FMM



Evaluate interactions
→ Tree traversals

Store aggregate data at nodes, e.g., bounding box, mass

N-body problems in other domains

Problem	Operators	Kernel Function
All Nearest Neighbors	$\forall, \arg\min$	$\|\|x_q - x_r\|\|$
All Range Search	$\forall, \cup \arg$	$I(h_{min} < \|\|x_q - x_r\|\| < h_{max})$
All Range Count	$\forall, \Sigma$	$I(h_{min} < \|\|x_q - x_r\|\| < h_{max})$
Naive Bayes Classifier	$\forall, \arg\max$	$(1/\sqrt{2\pi\|\Sigma_k\|})e^{-\frac{1}{2}(x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)}P(C_k)$
Mixture Model E-step	$\forall, \forall$	$(1/\sqrt{2\pi\|\Sigma_k\|})e^{-\frac{1}{2}(x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)}$
K-means E-step	$\forall, \arg\min$	$\|\|x_q - x_r\|\|$
Mixture Model Log-likelihood	$\sum, \log \sum$	$(1/\sqrt{2\pi\|\Sigma_k\|})e^{-\frac{1}{2}(x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)}$
Kernel Density Estimation	$\forall, \Sigma$	$\phi(\frac{\|\|x_q - x_r\|\|}{h})$
Kernel Density Bayes Classifier	$\forall, \arg\max \Sigma$	$\phi(\frac{\|\|x_q - x_r\|\|}{h})P(C_k)$
2-point (cross-)correlation	Σ, Σ	$I(\|\|x_q - x_r\|\| < h)$
Nadaraya-Watson Regression	$\forall, \Sigma$	$y_r \ \phi(\frac{\|\|x_q - x_r\|\|}{h})$
Thermodynamic Average	Σ, Σ	$\phi(\|\|x_q - x_r\|\|)$
Largest-span set	$\max, ..., \max$	$\Sigma(\|\|x_q - x_r\|\|)$
Closest Pair	$\arg\min, \arg\min$	$\|\|x_q - x_r\|\|$
Minimum Spanning Tree	$\forall, \arg\min$	$\|\|x_q - x_r\|\|$
Coulombic Interaction	$\forall, \Sigma$	$\frac{\alpha_q \alpha_r}{\|\|x_q - x_r\|\|}$
Average Density	Σ, Σ	$I(\|\|x_q - x_r\|\| < h)$
Wave function	$\forall, \Pi$	$\phi(\|\|x_q - x_r\|\|)$
Hausdorff Distance	$\max, \min$	$\|\|x_q - x_r\|\|$
Intrinsic (fractal) Dimension	Σ, Σ	$I(\|\|x_q - x_r\|\| < h)$

Each problem has a set of operators and a kernel function

Why N-body methods?

- One of the original seven dwarfs or motifs
- FMM listed among the top 10 algorithms having the greatest influence in 20th century
- EM is one of the top 10 algorithms having the highest impact in data mining
- Applications
  - Machine learning
  - Computer vision
  - Computational geometry
  - Scientific computing …

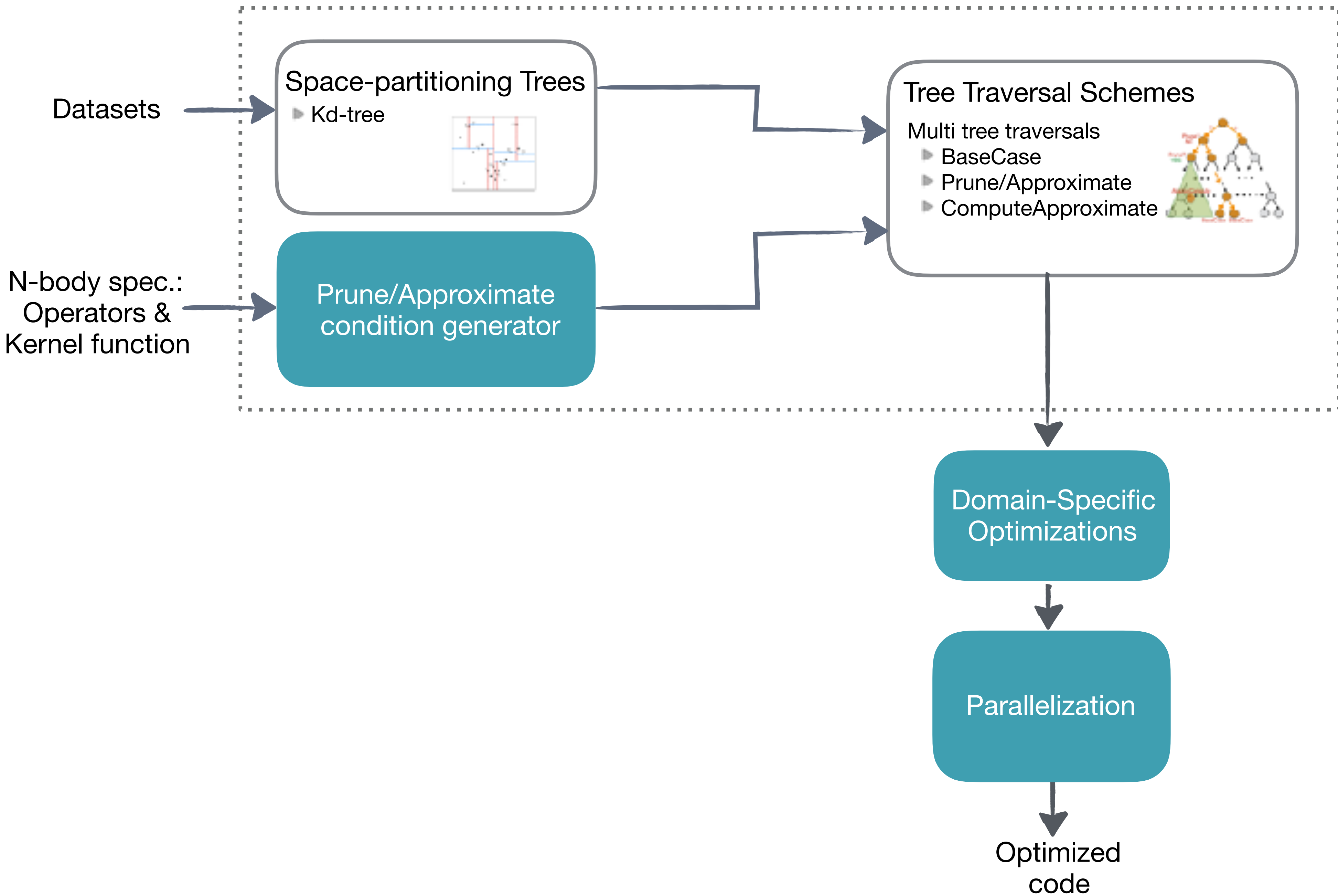
Key Ideas and Findings

- An algorithmic framework for N-body problems
  - Automatically generates prune & approximation conditions
  - Results in O(N log N) and O(N) algorithms
  - Domain-specific optimizations and parallelization
- Show 10-230x speedup on 6 different algorithms compared to state-of-art libraries/softwares
- Out-of-the-box new optimal algorithms
  - O(N log N) EM algorithm for GMM’s
  - O(N) algorithm for Hausdorff distance

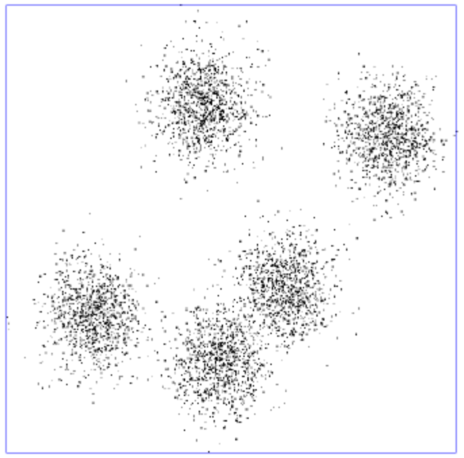
Outline

- Introduction
- PASCAL Framework
  - Space Partitioning Trees
  - Tree Traversal
  - Prune/Approximate Generators
- Optimizations & Parallelization
- Experiments & Results
- Conclusions & Future Work

PASCAL Framework



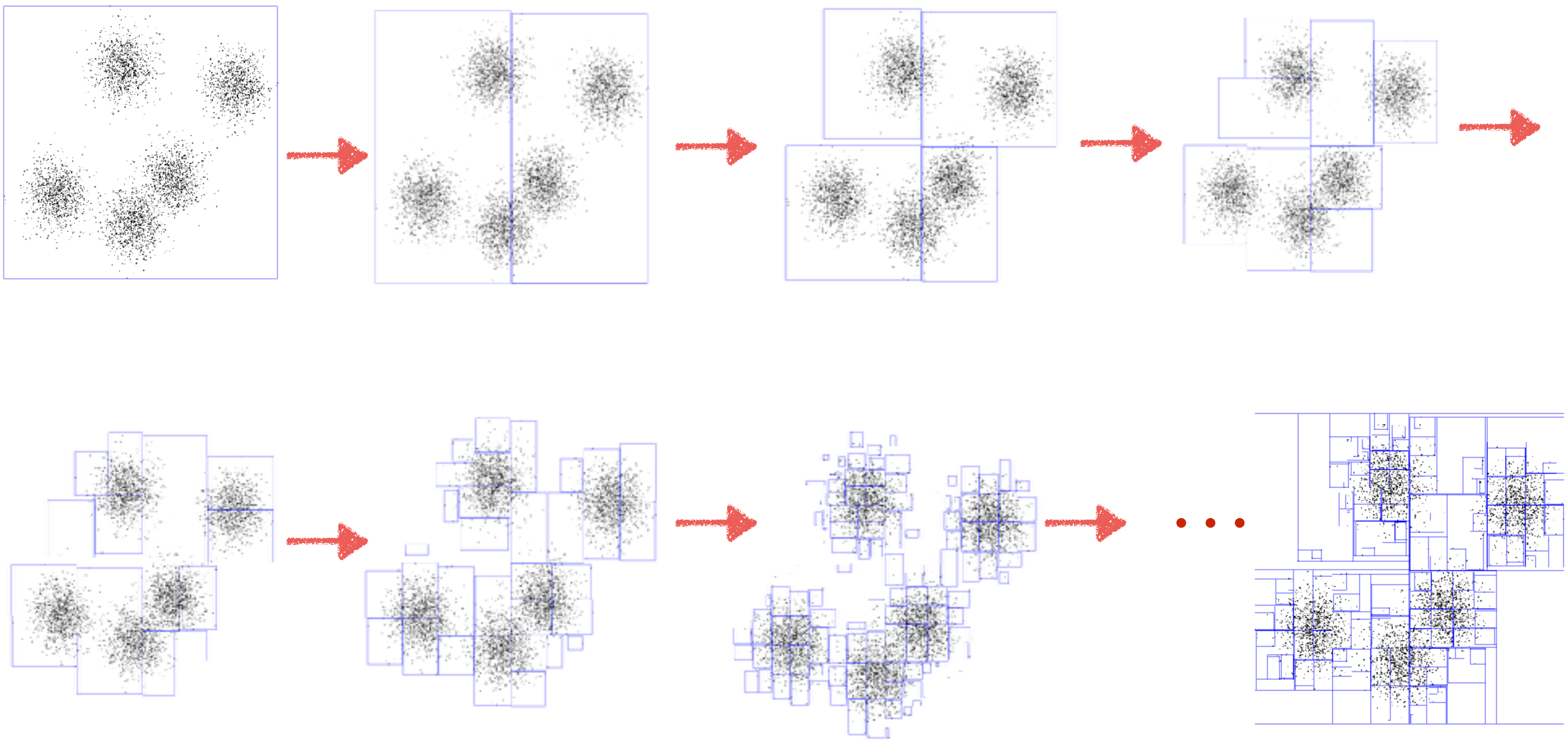
Tree Construction



Recursively divide space until each box has **at most *q* points**.

http://www.cs.cmu.edu/~dpelleg/kmeans.html

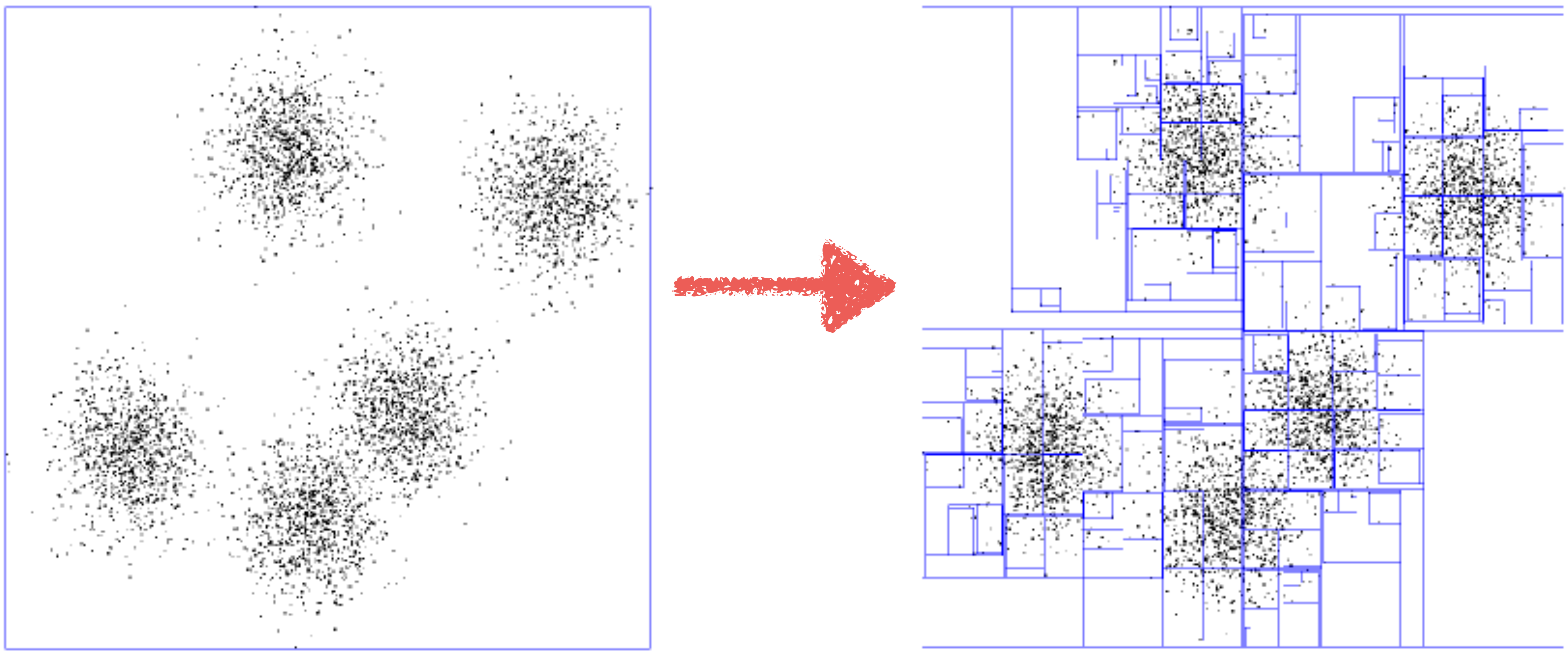
Tree Construction



Recursively divide space until each box has **at most *q* points**.

http://www.cs.cmu.edu/~dpelleg/kmeans.html

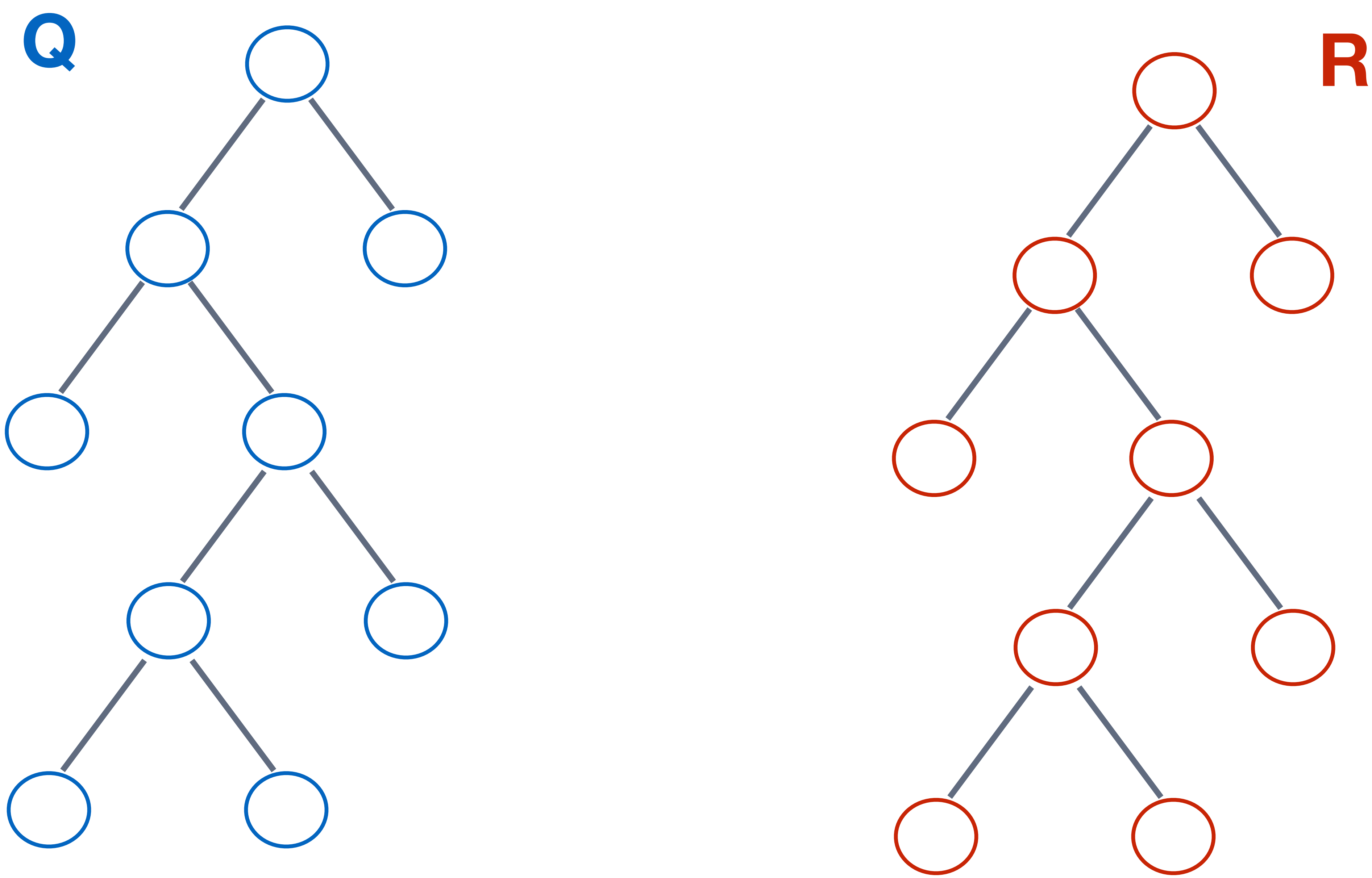
Tree Construction



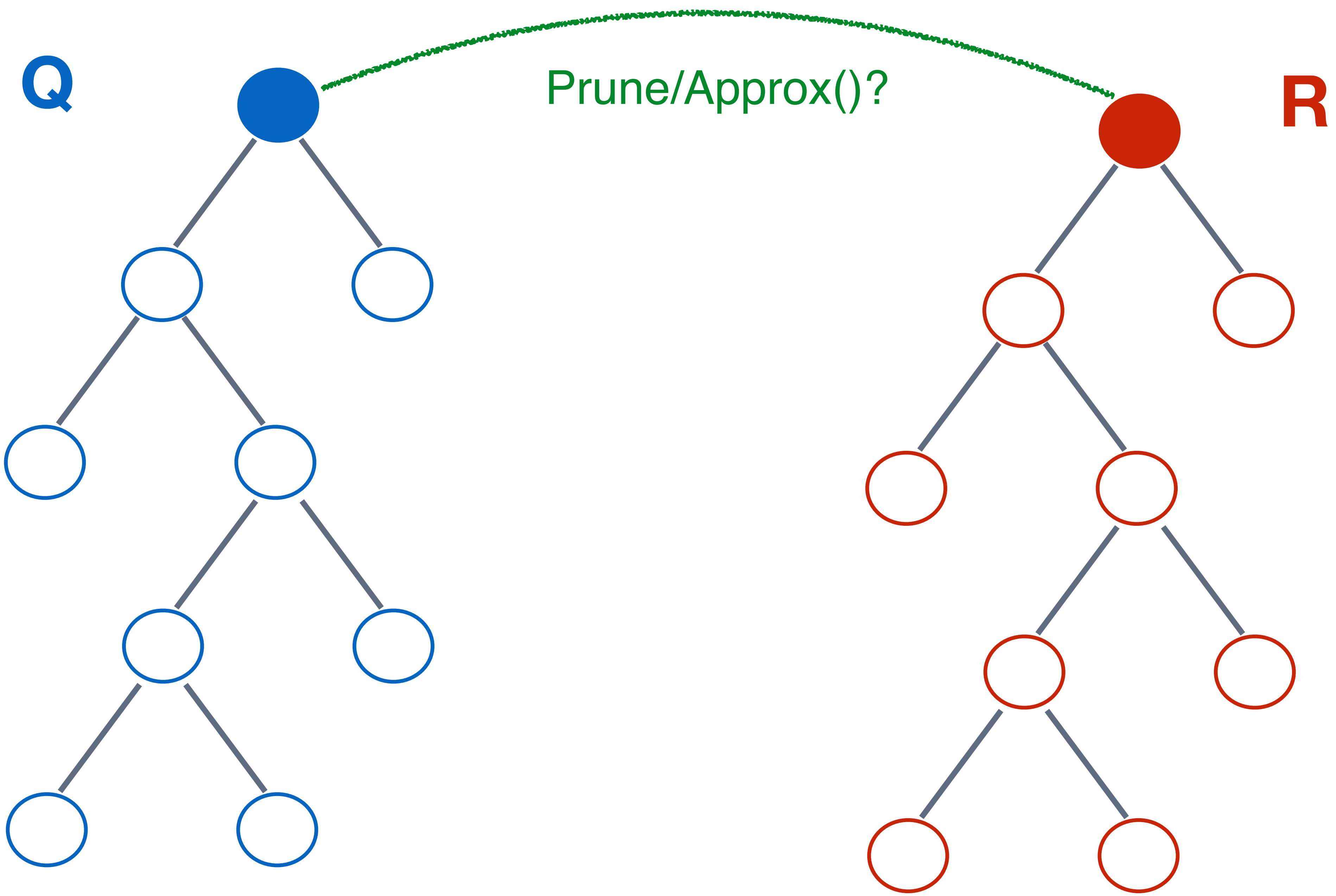
Recursively divide space until each box has **at most *q* points**.

http://www.cs.cmu.edu/~dpelleg/kmeans.html

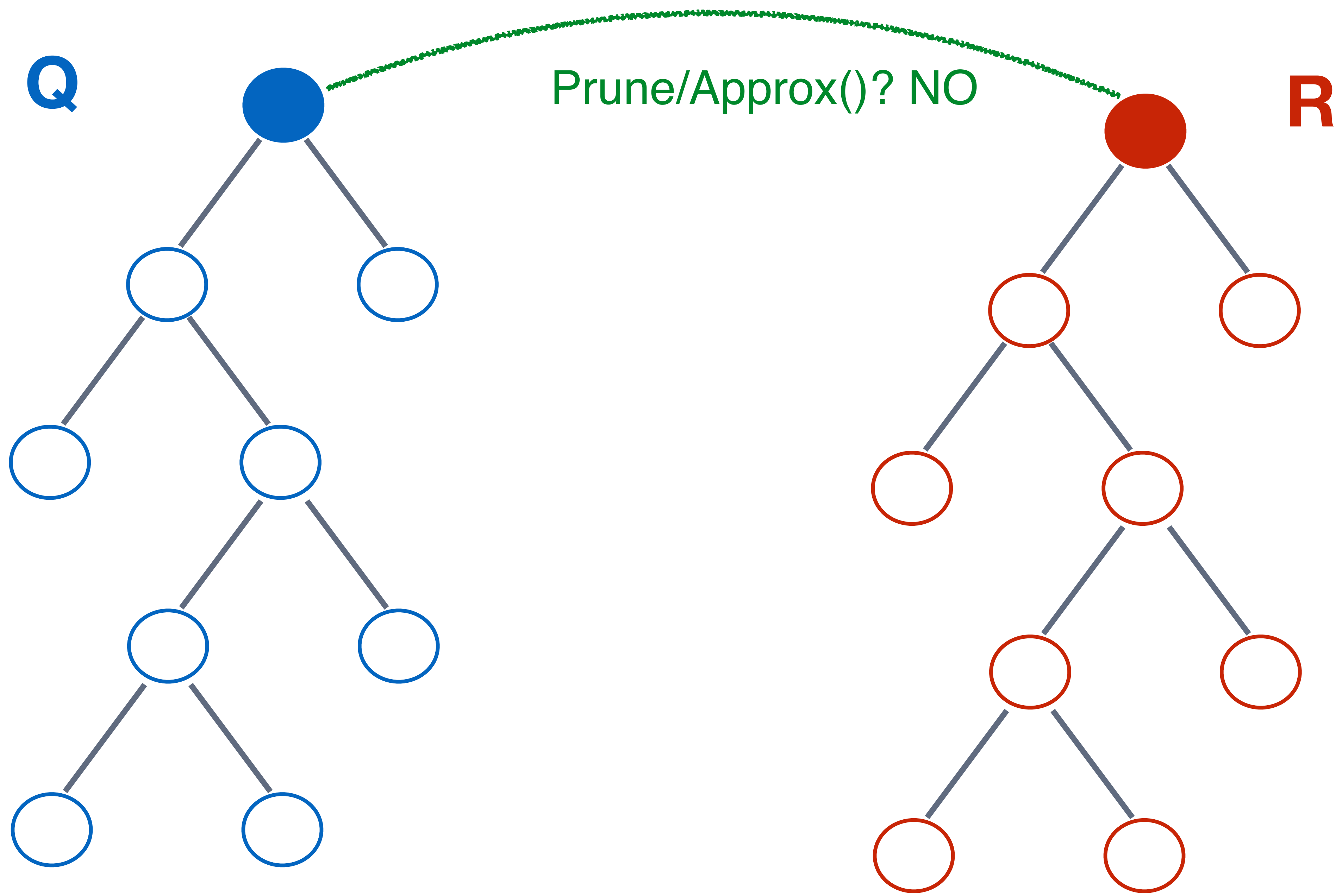
Tree Traversal



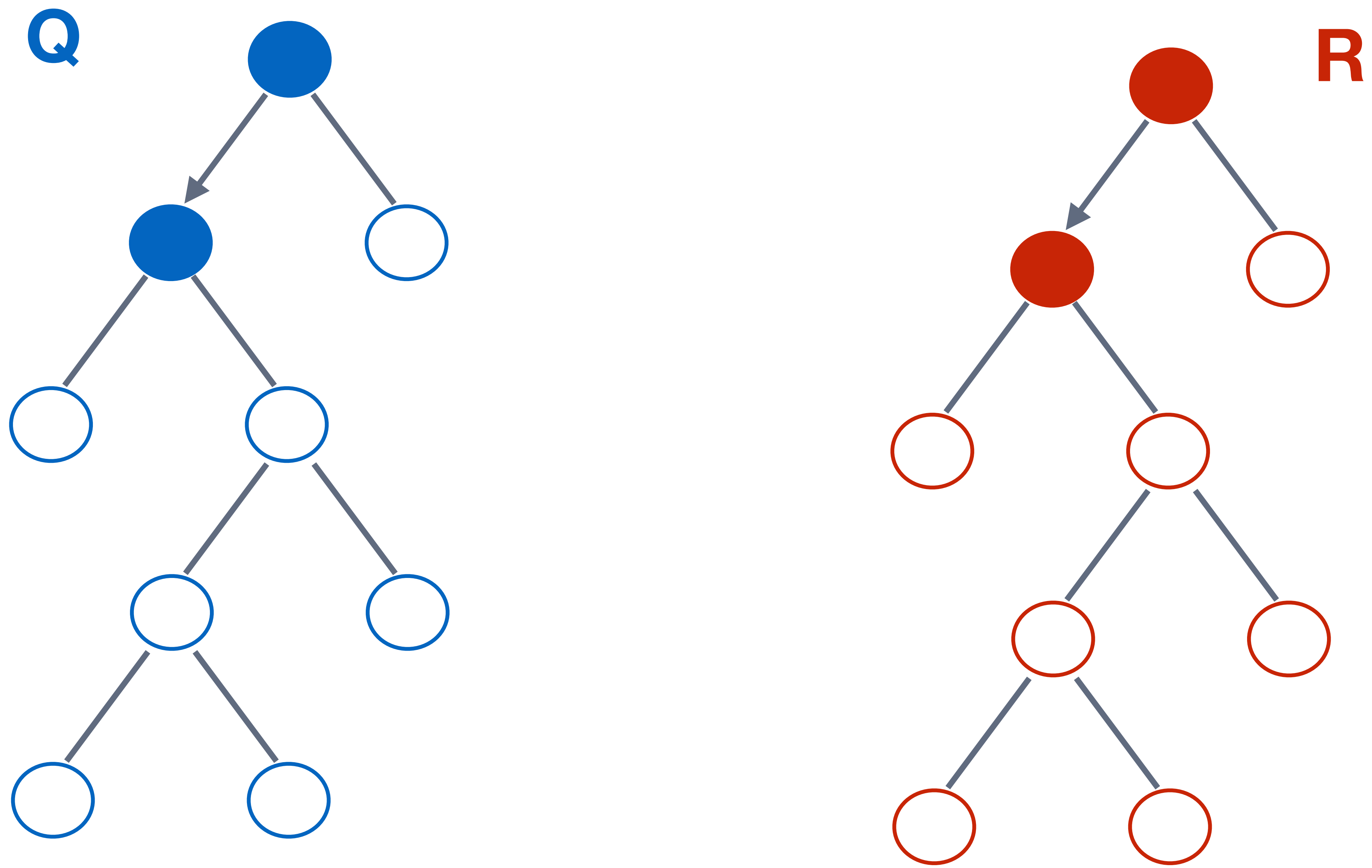
Tree Traversal



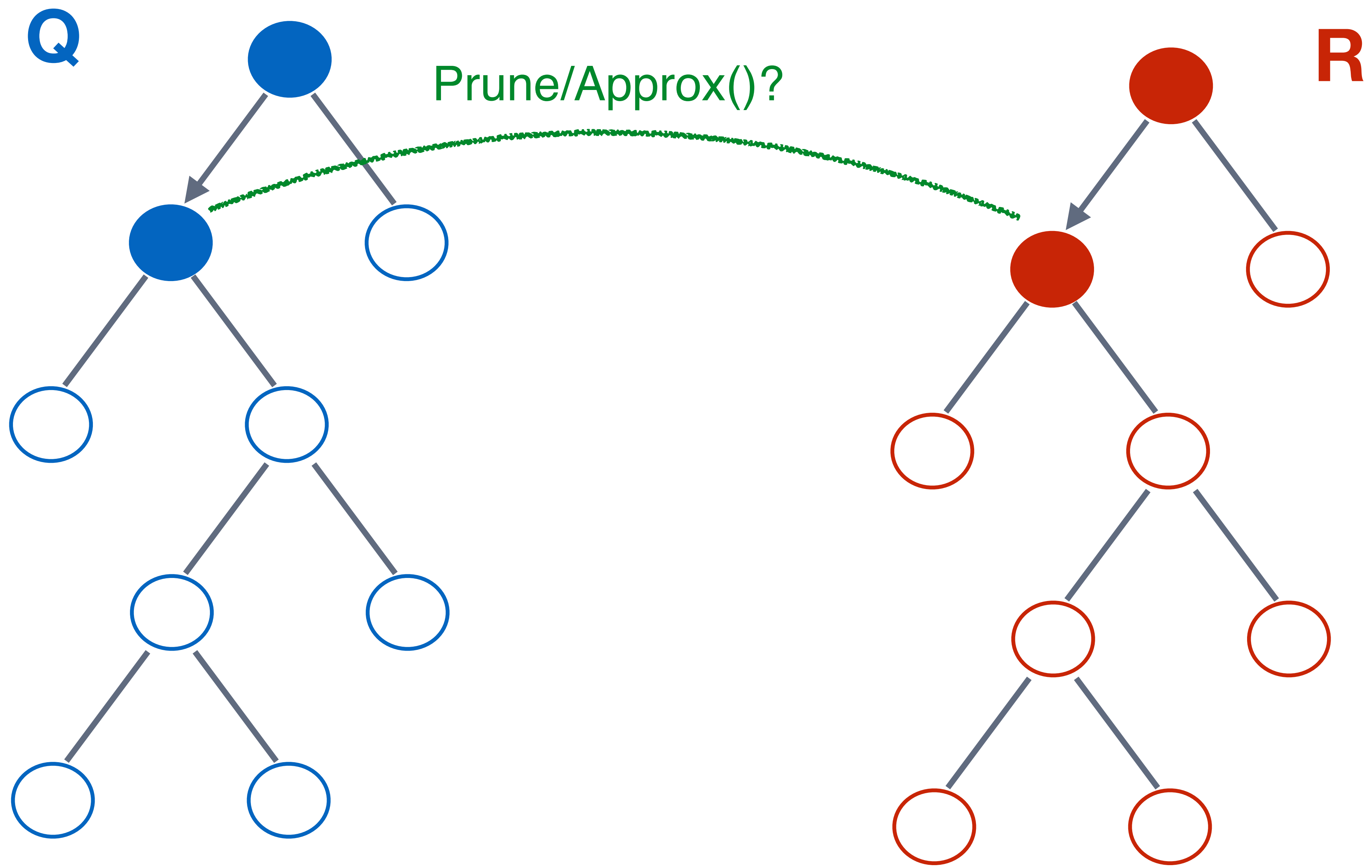
Tree Traversal



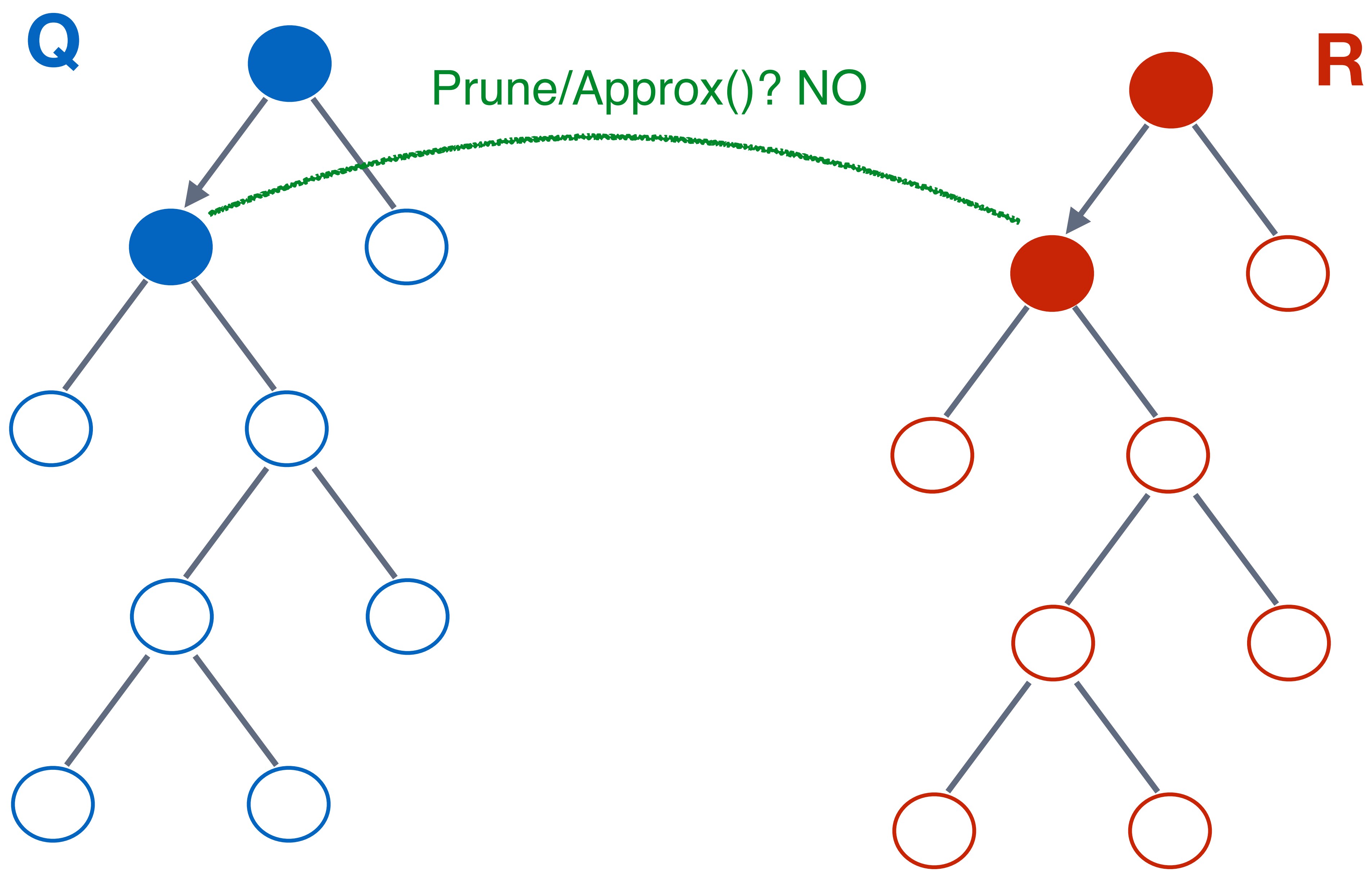
Tree Traversal



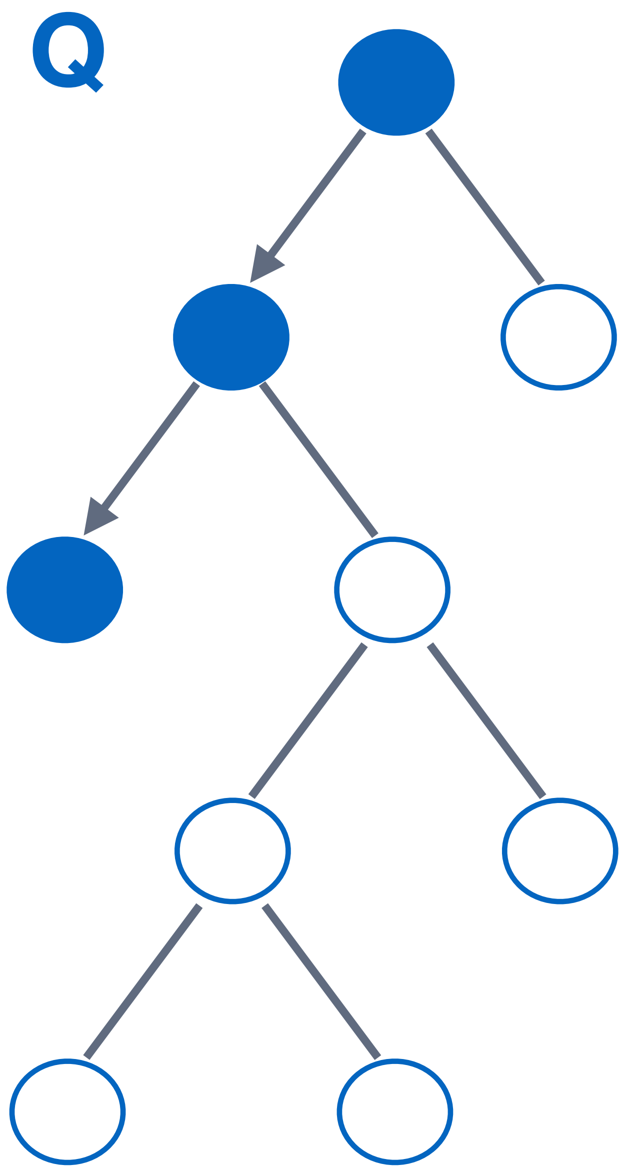
Tree Traversal

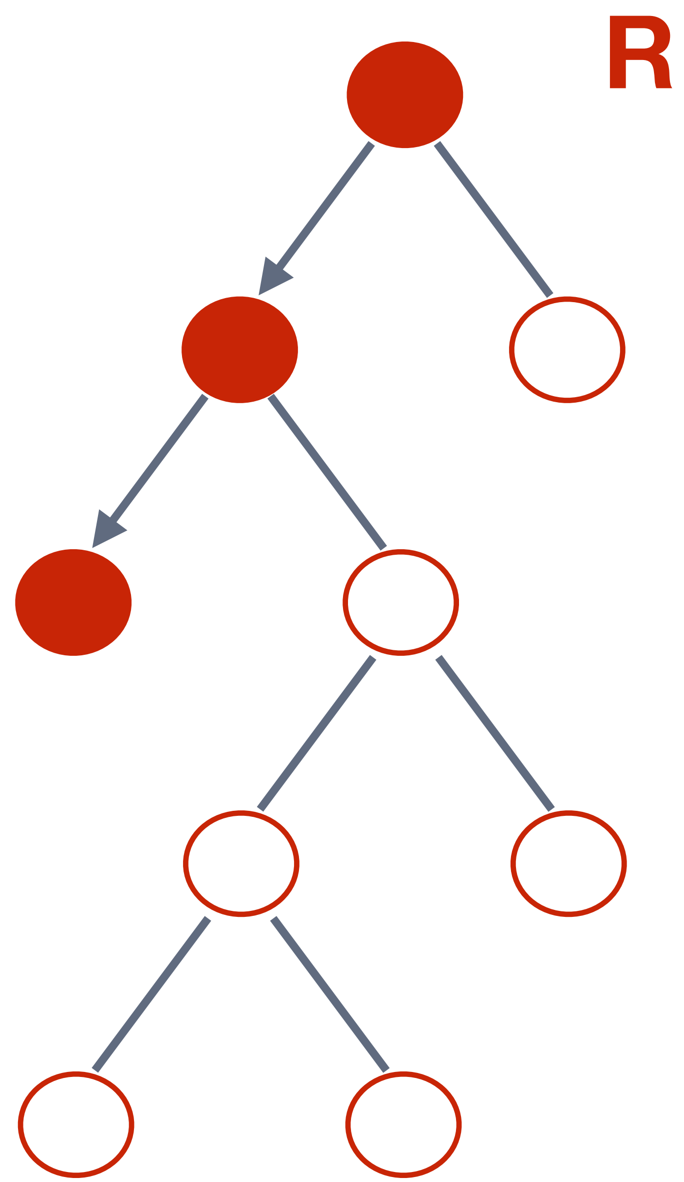


Tree Traversal

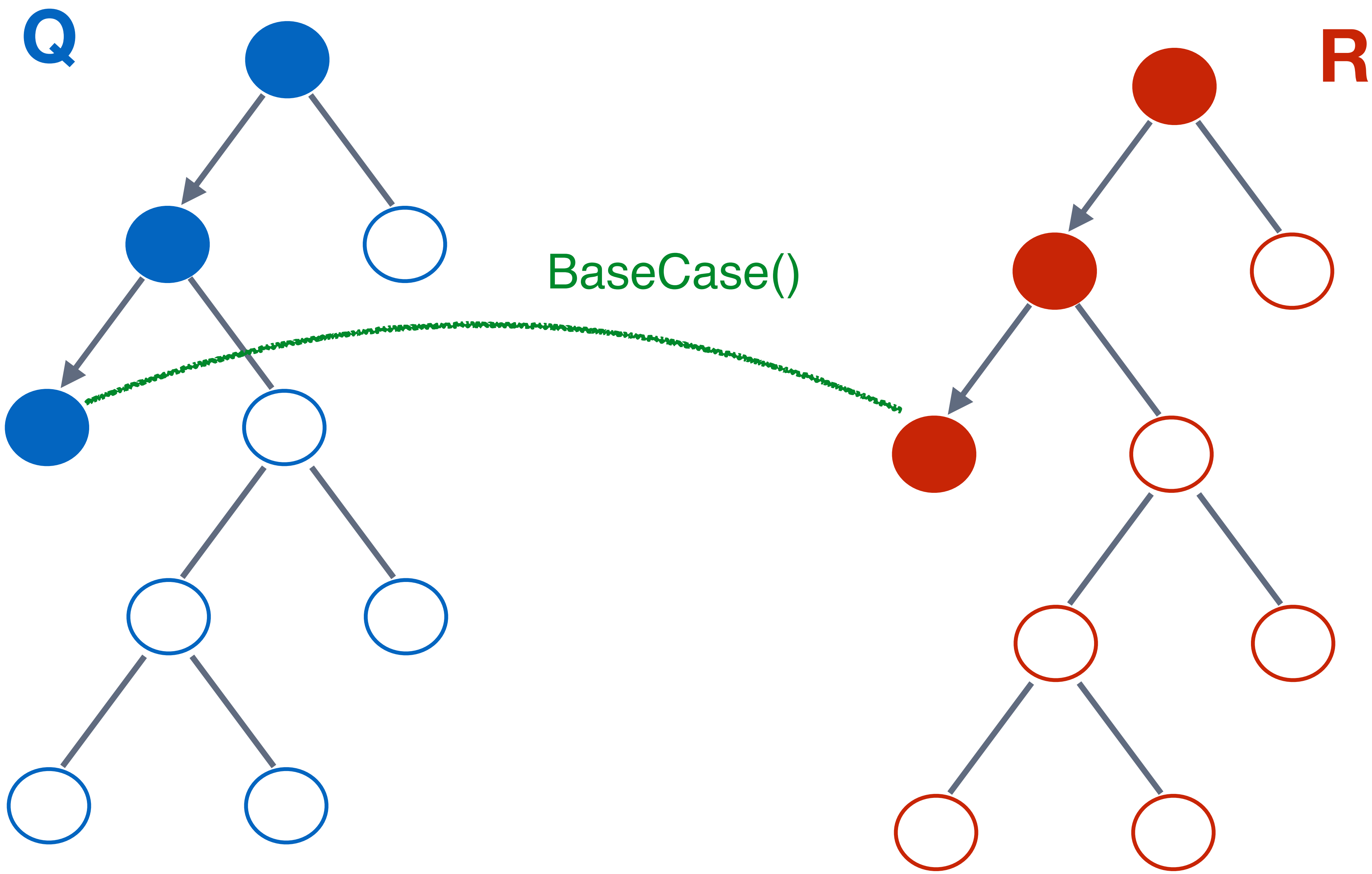


Tree Traversal



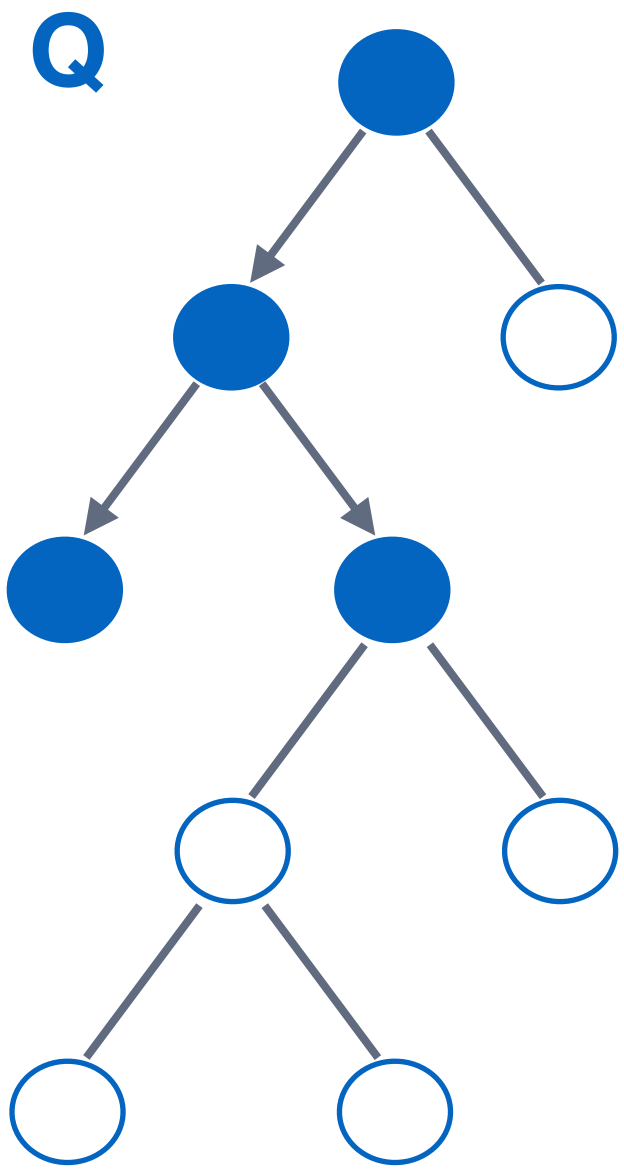


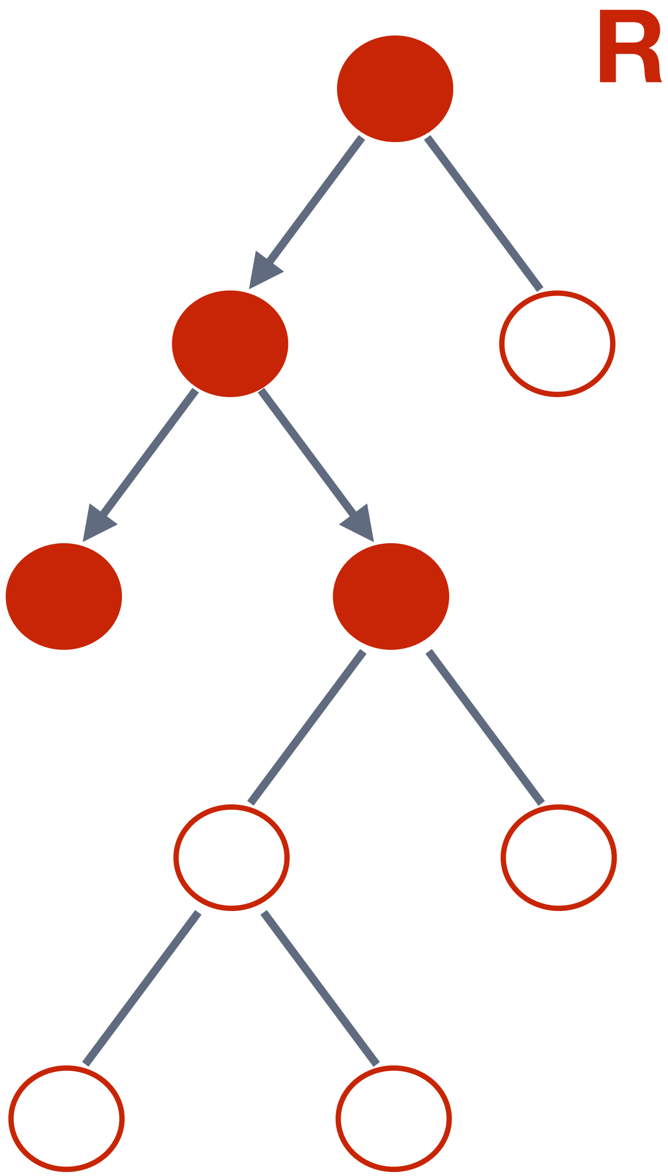
Tree Traversal



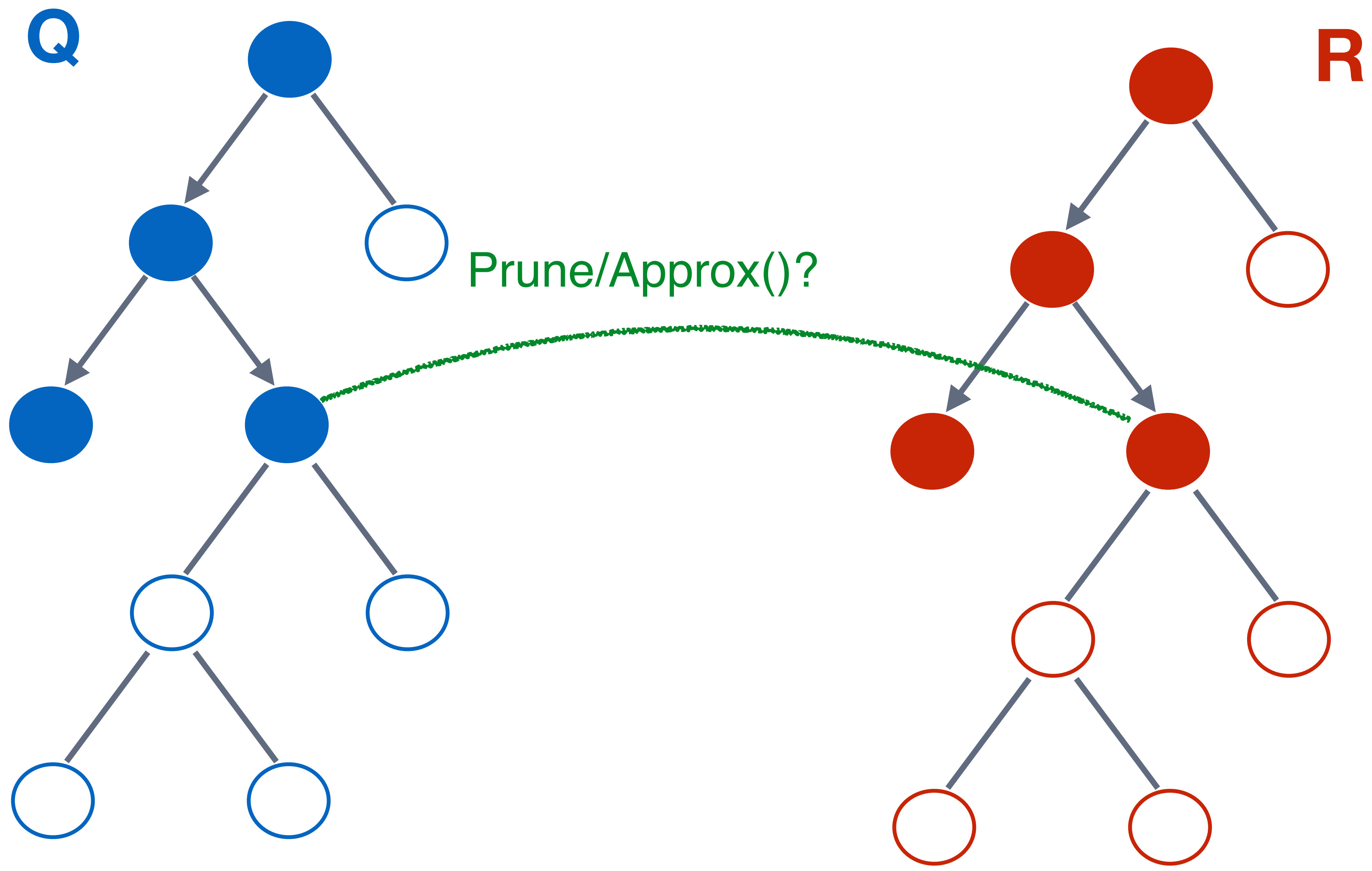
Direct $Q_L \otimes R_L \rightarrow O(q^2)$

Tree Traversal

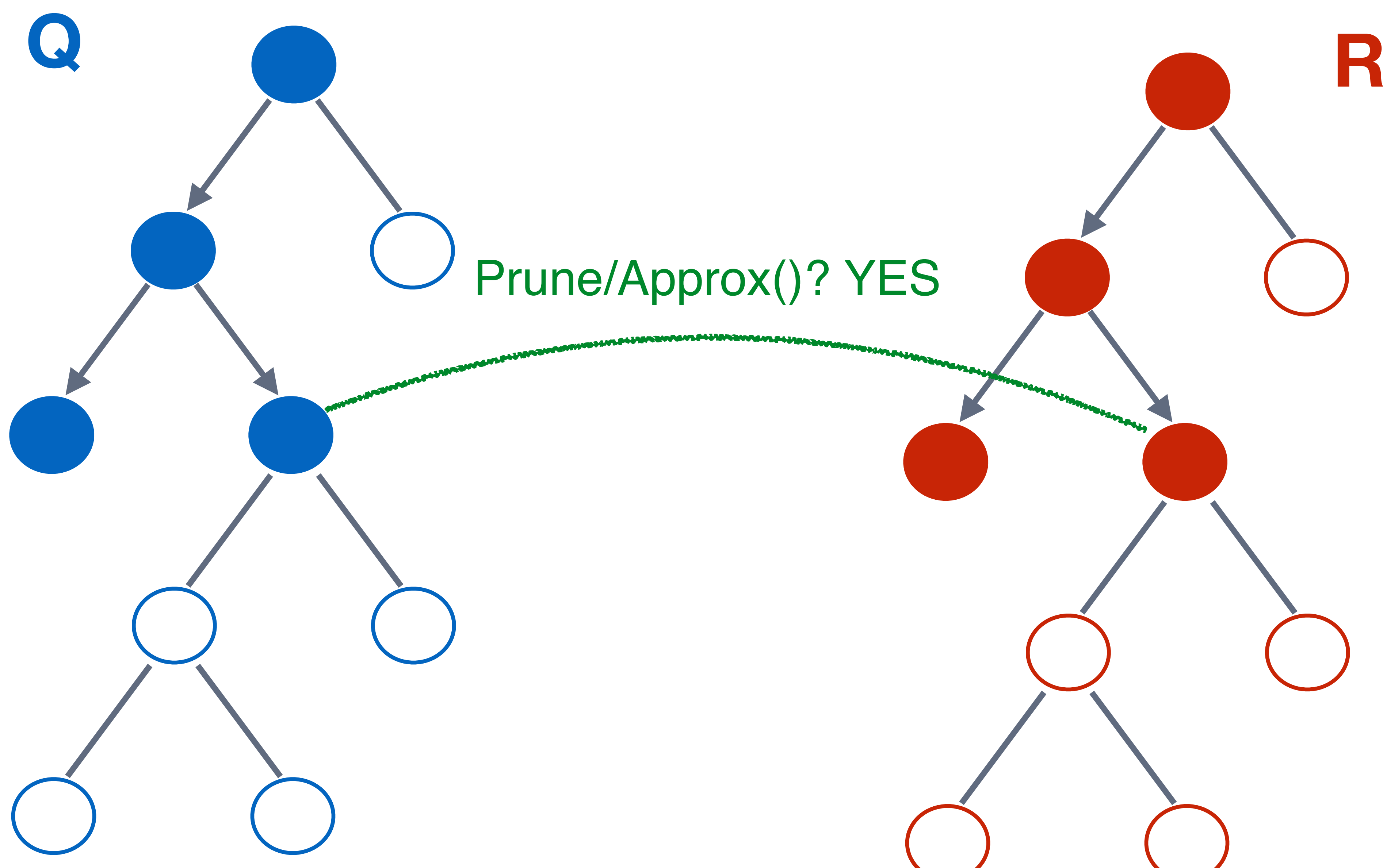




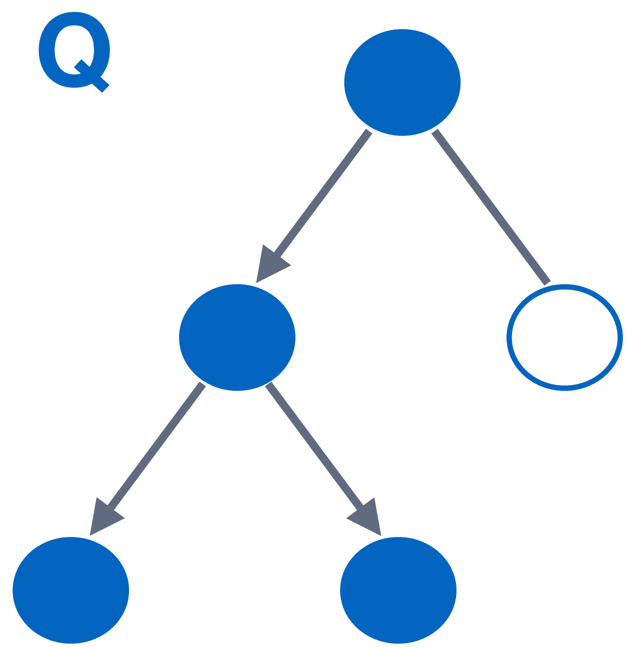
Tree Traversal

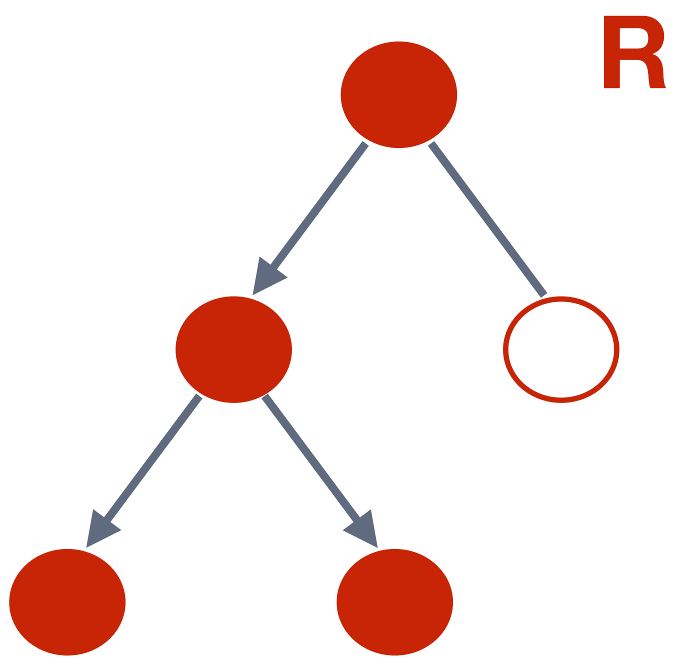


Tree Traversal



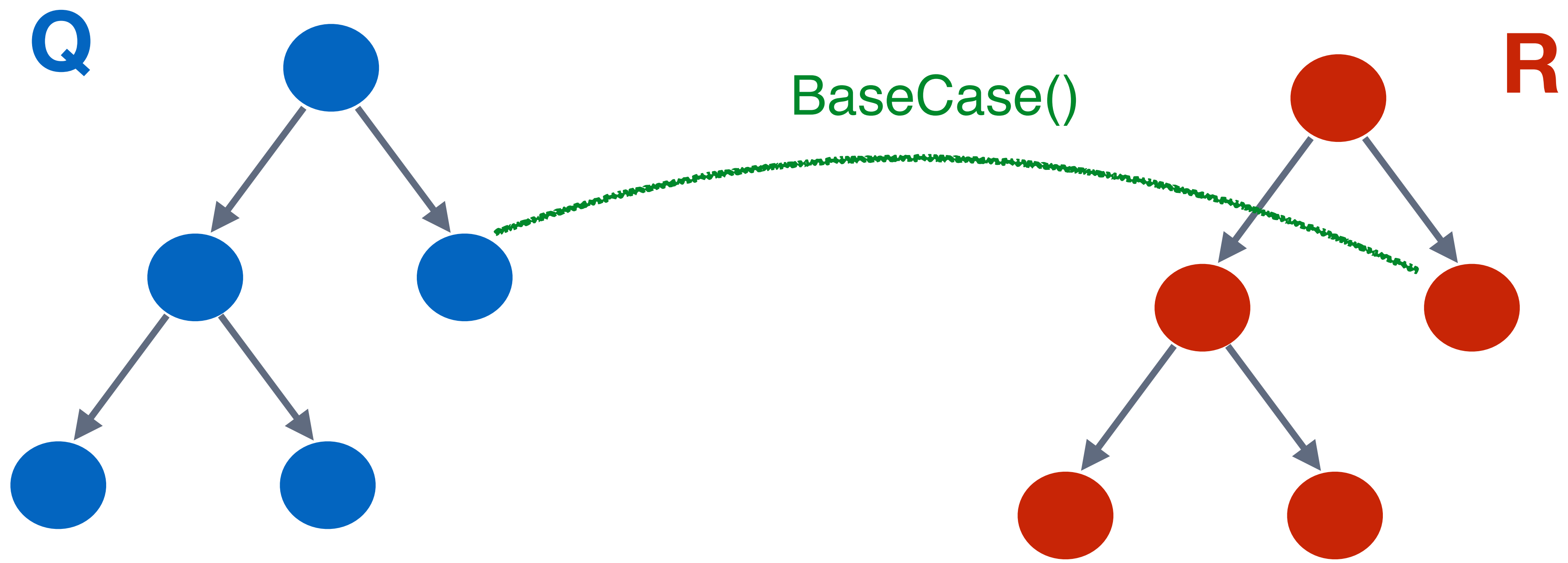
Tree Traversal



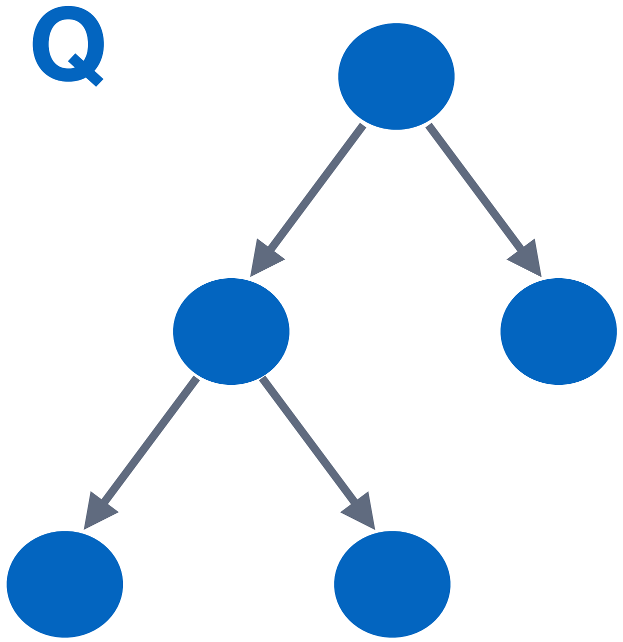


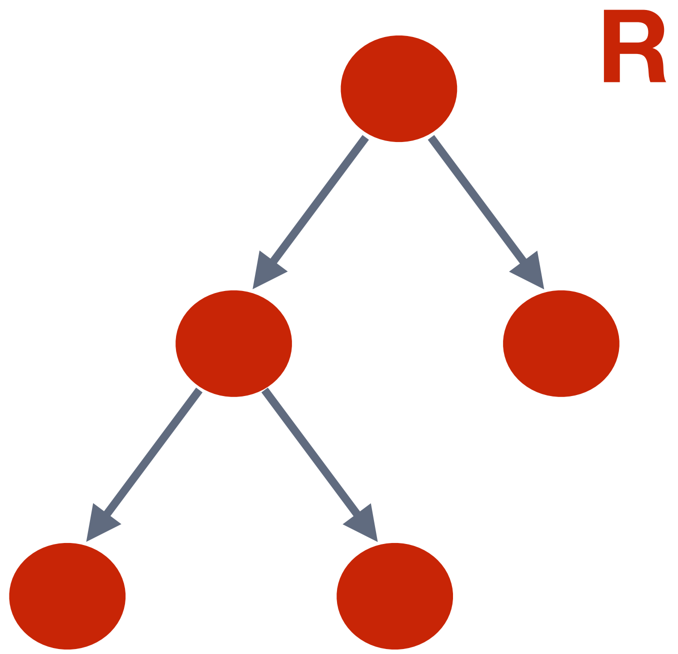
If Prune/Approx() is true, discard the entire subtree for pruning problems

Tree Traversal

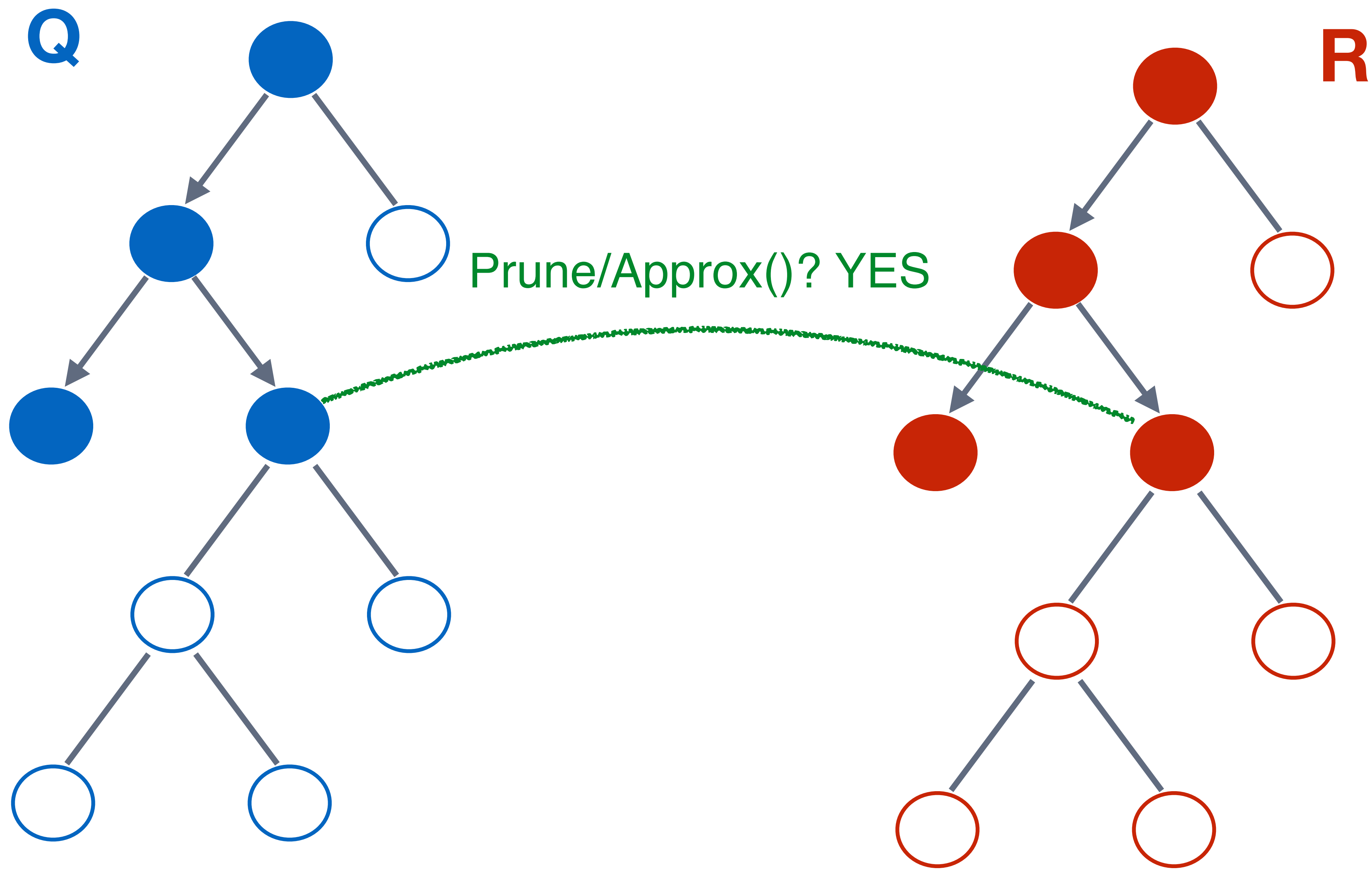


Tree Traversal





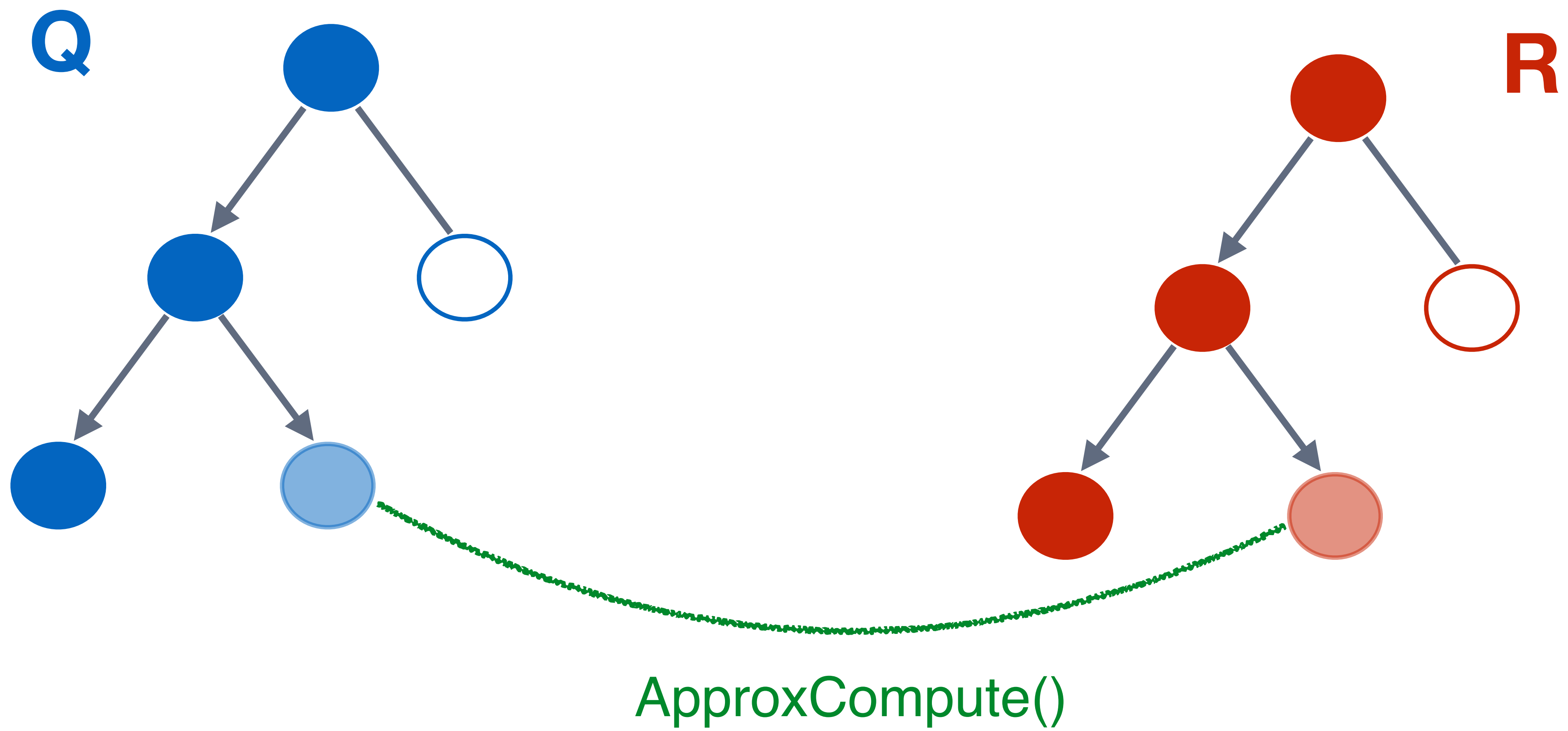
Tree Traversal



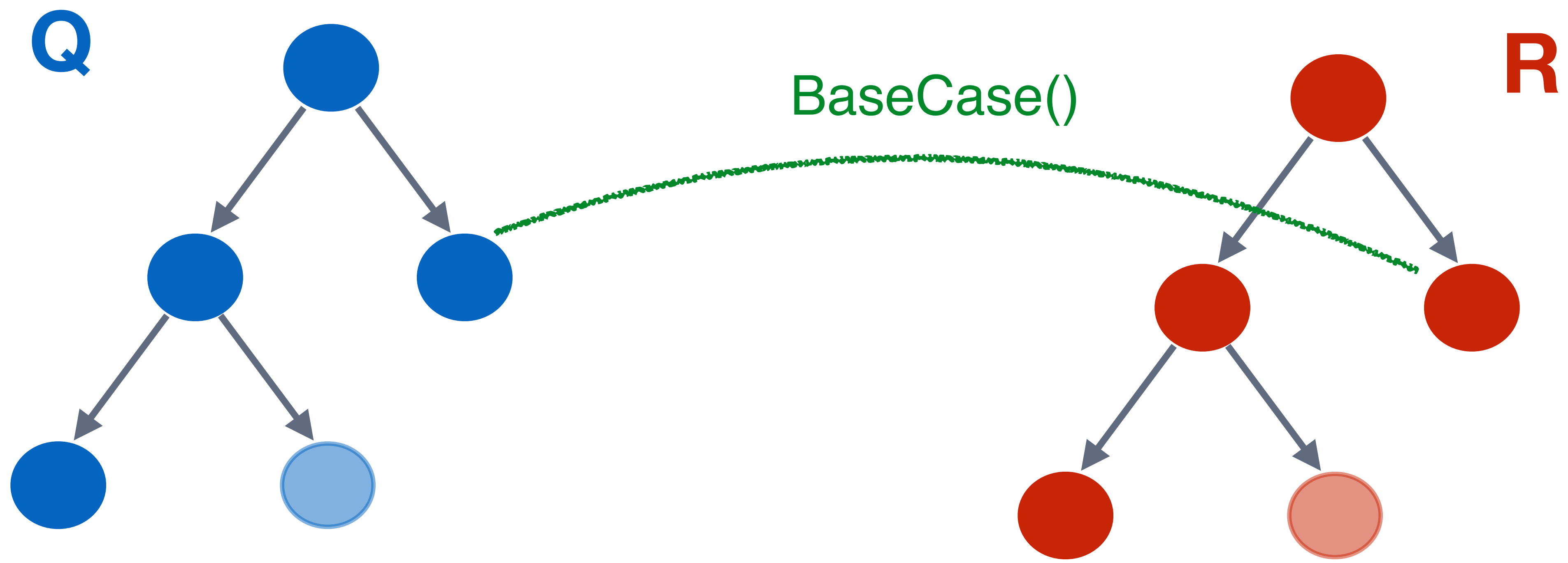
Tree Traversal

If Prune/Approx() is true, replace the subtree with the centroid for approximation problems

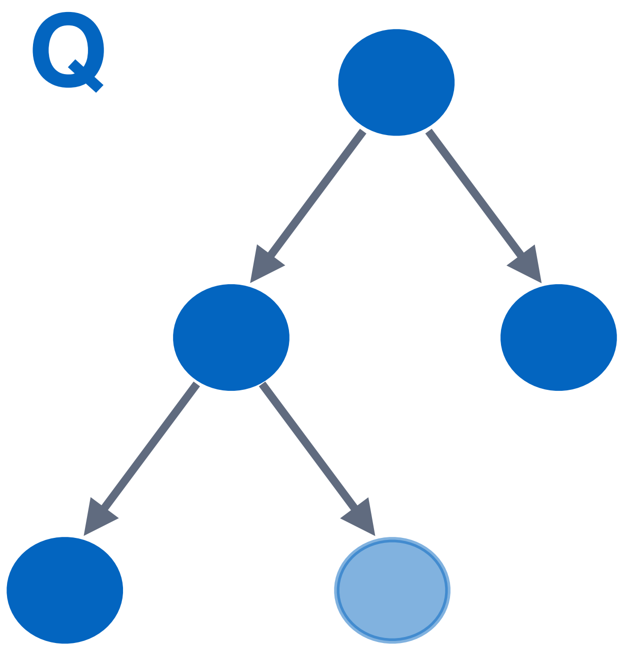
Tree Traversal

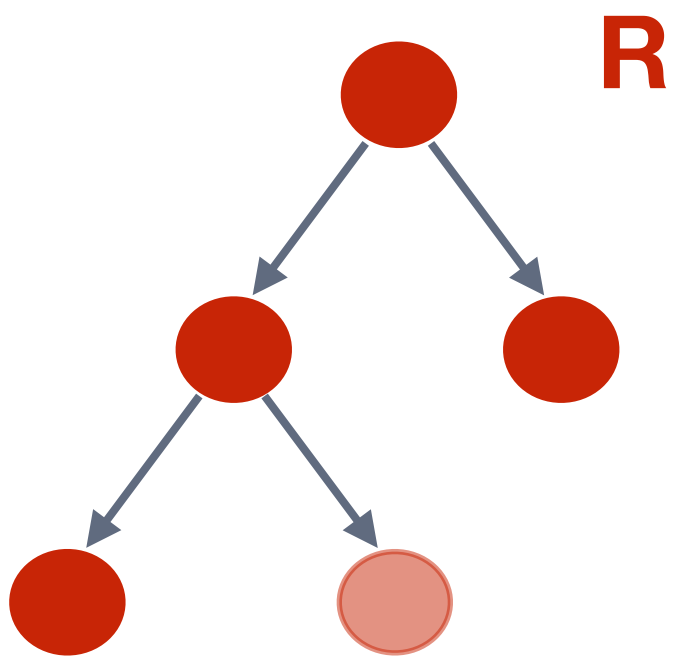


Tree Traversal



Tree Traversal





Prune/Approximate Condition Generator

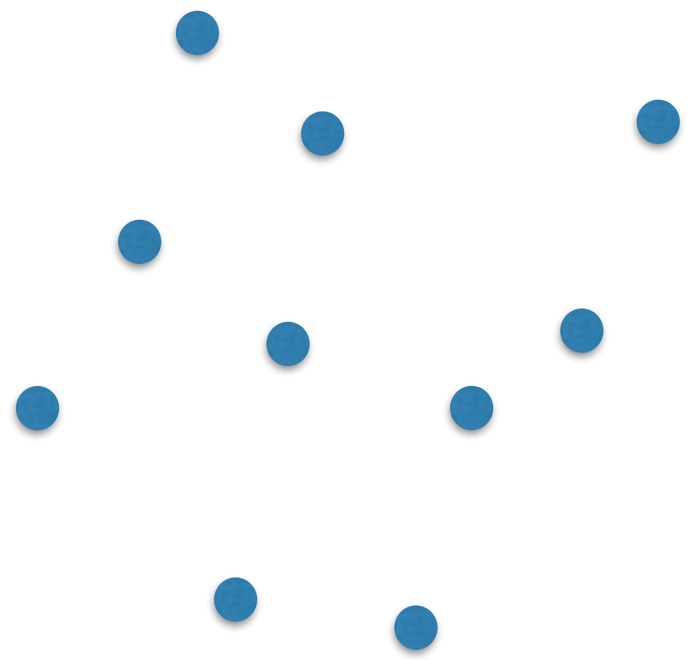
- Prune *e.g.*, *Hausdorff Distance* $\max_q, \min_r ||x_q - x_r||$

Hausdorff Distance

$$\max_q, \min_r ||x_q - x_r||$$

Hausdorff Distance

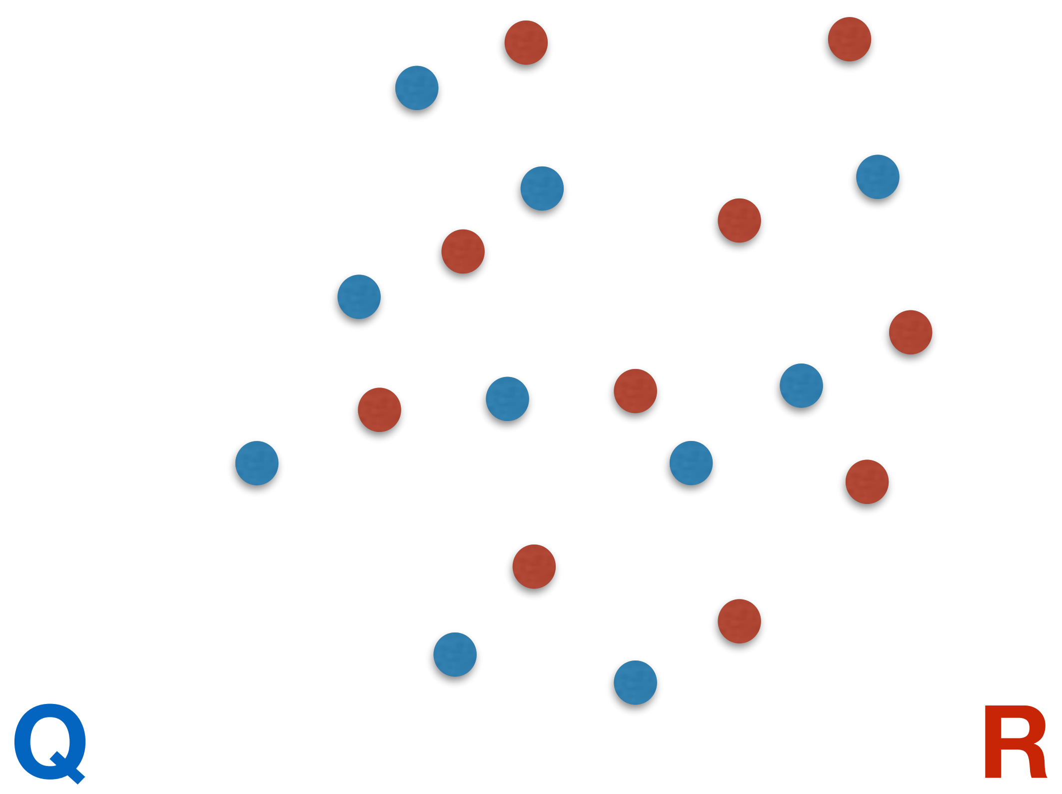
$$\max_q, \min_r ||x_q - x_r||$$



Q

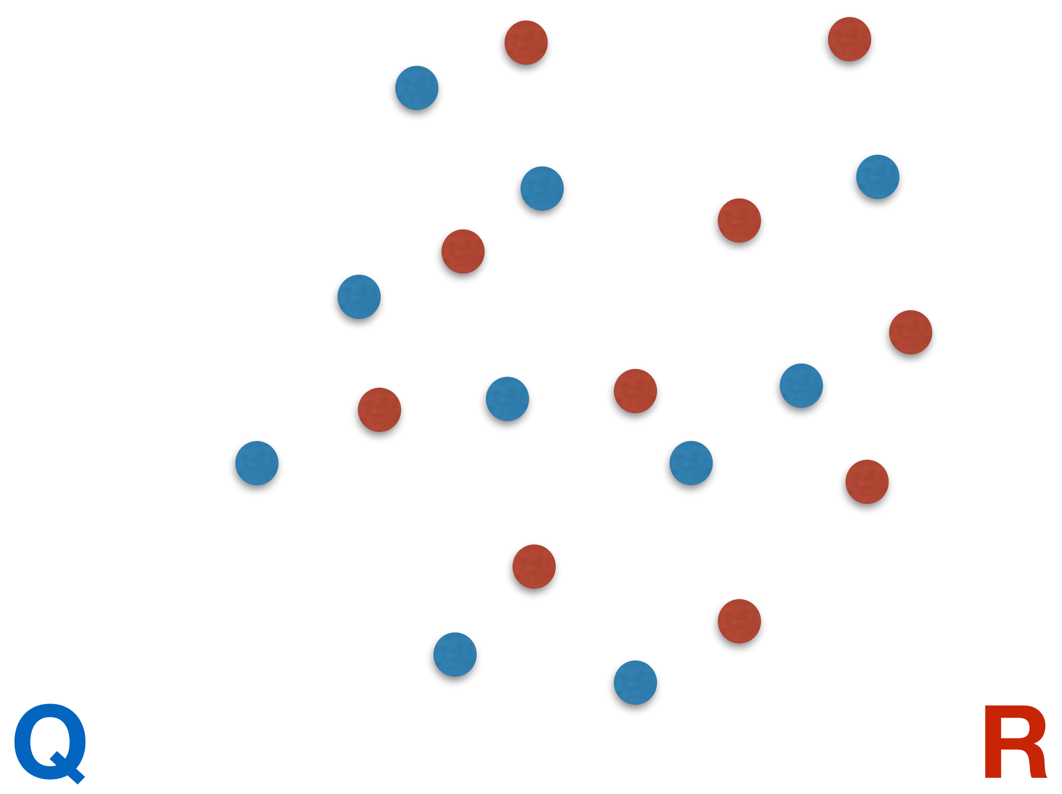
Hausdorff Distance

$$\max_q, \min_r ||x_q - x_r||$$



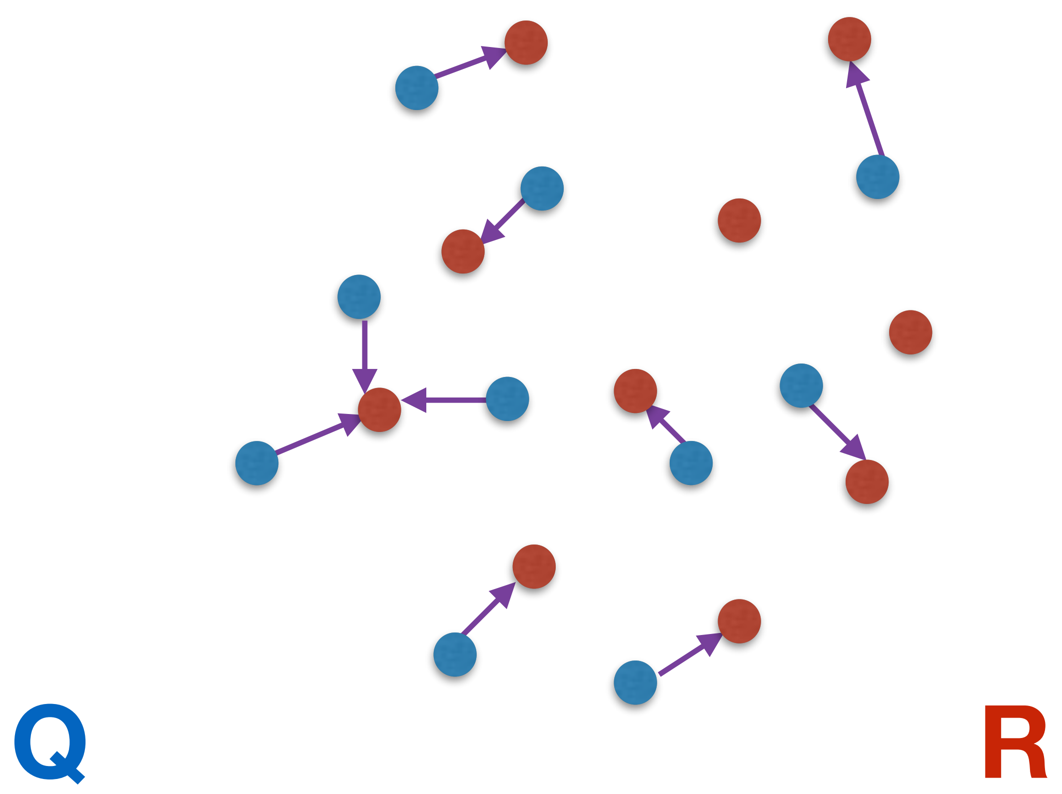
Hausdorff Distance

$$\max_q \min_r ||x_q - x_r||$$



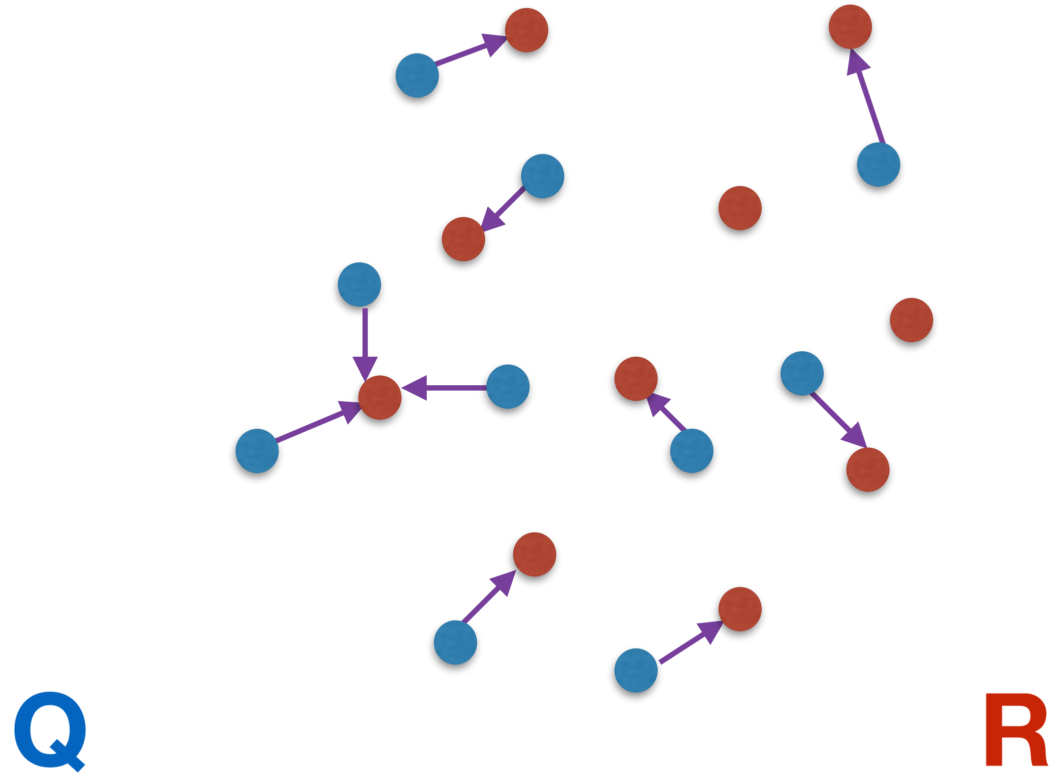
Hausdorff Distance

$$\max_q \min_r ||x_q - x_r||$$



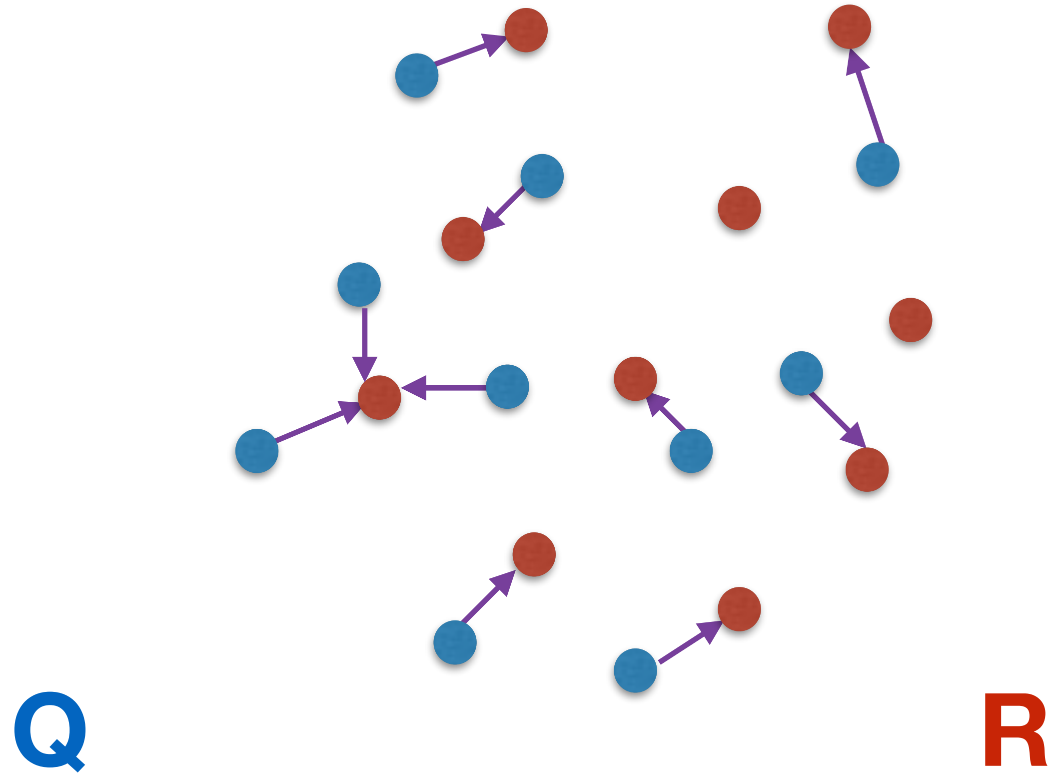
Hausdorff Distance

$$\max_q, \min_r ||x_q - x_r||$$



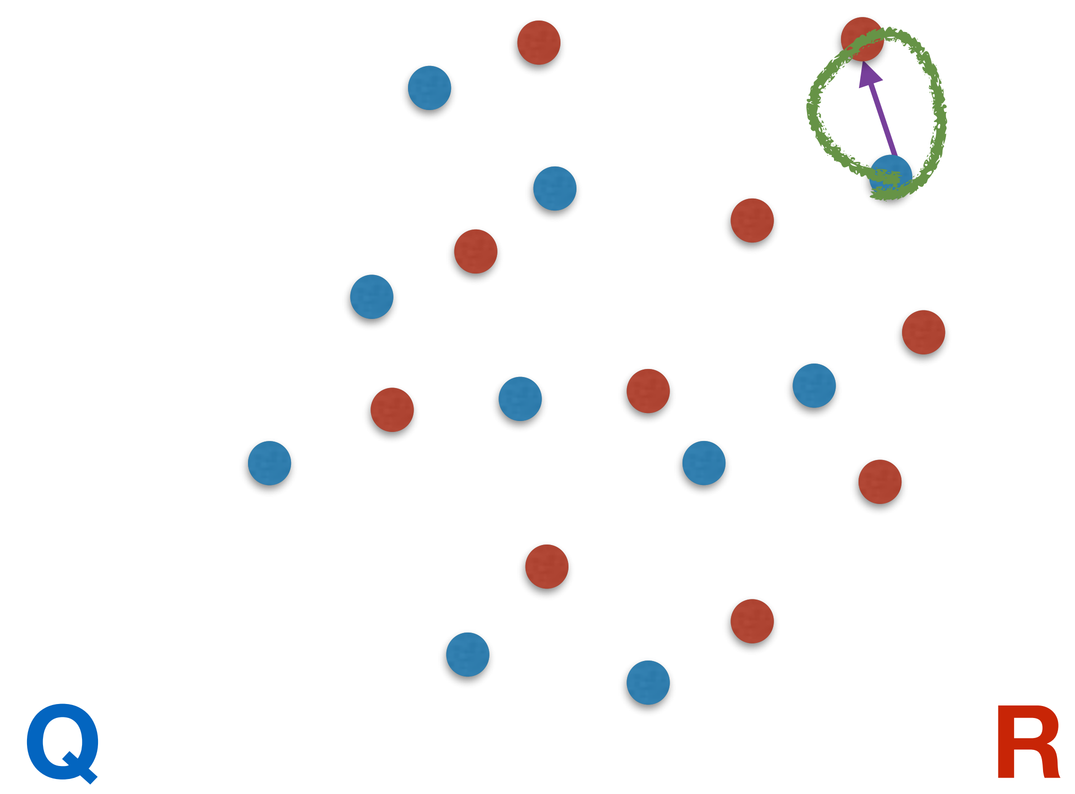
Hausdorff Distance

$$\max_q \min_r ||x_q - x_r||$$



Hausdorff Distance

$$\max_q \min_r ||x_q - x_r||$$



Prune/Approximate Condition Generator

- Prune *e.g.*, *Hausdorff Distance* $\max_q, \min_r ||x_q - x_r||$
- Approximation *e.g.*, *Expectation Maximization (EM)*

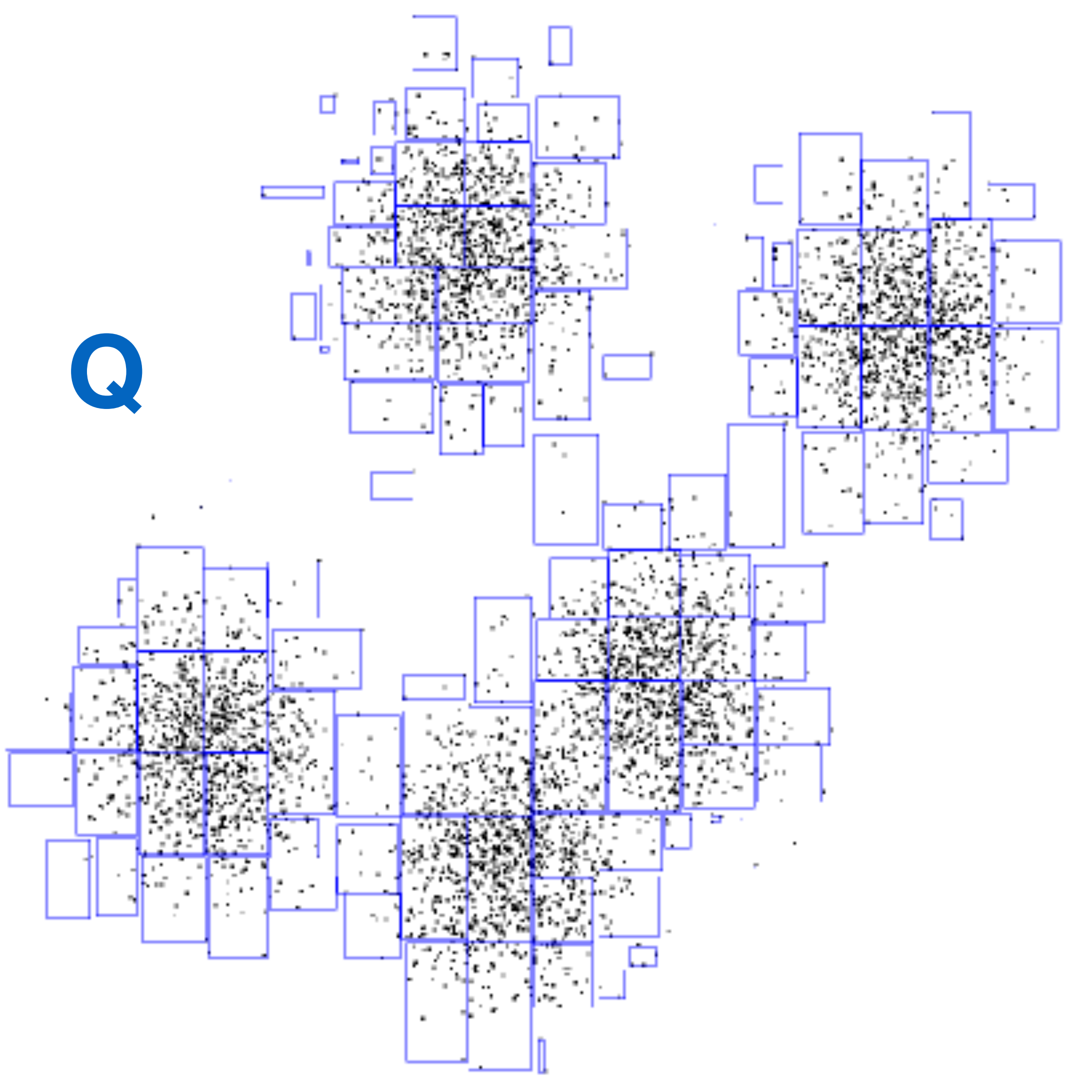
  E-step $\forall q, \forall r, \quad \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$

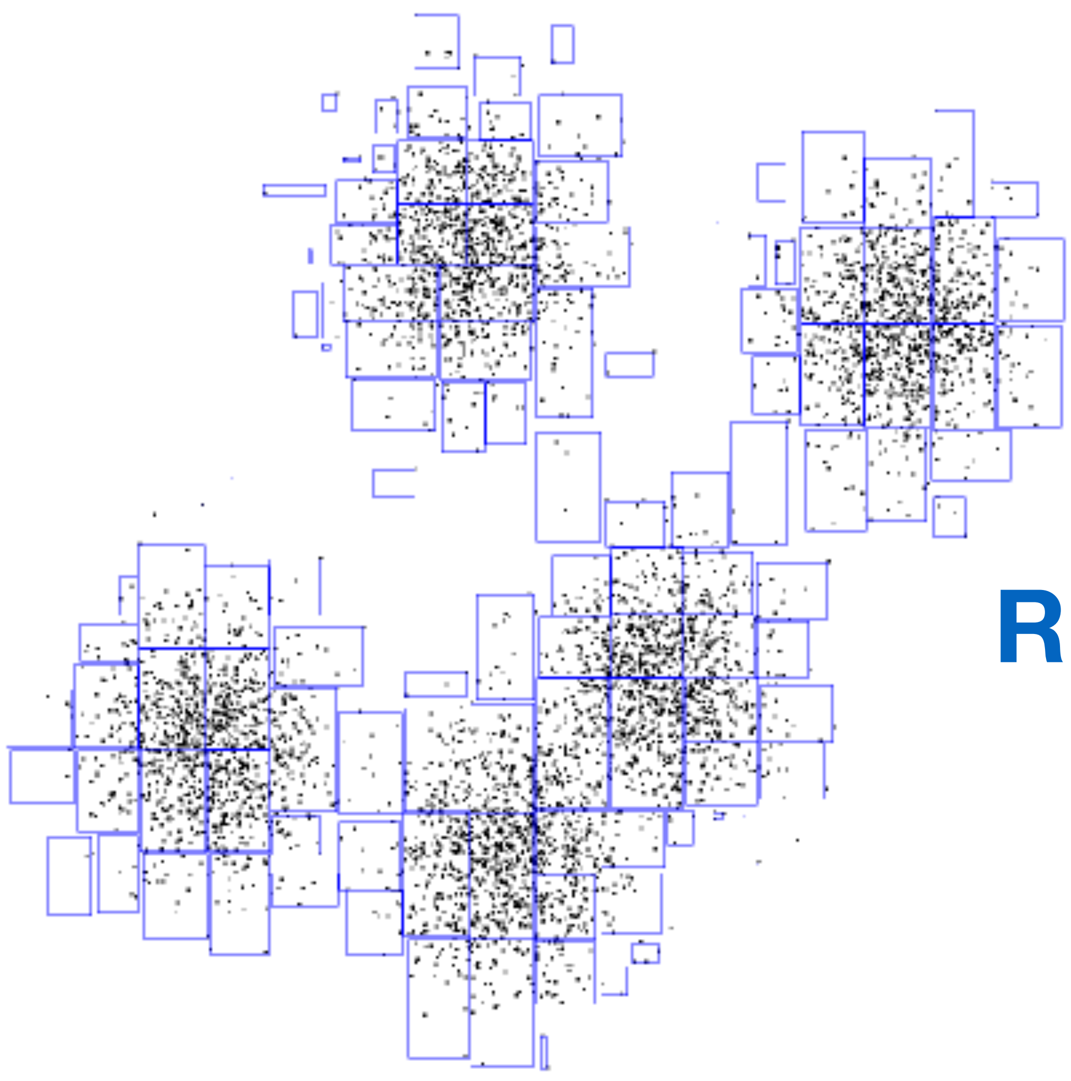
  M-step

  Log-likelihood $\sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)$

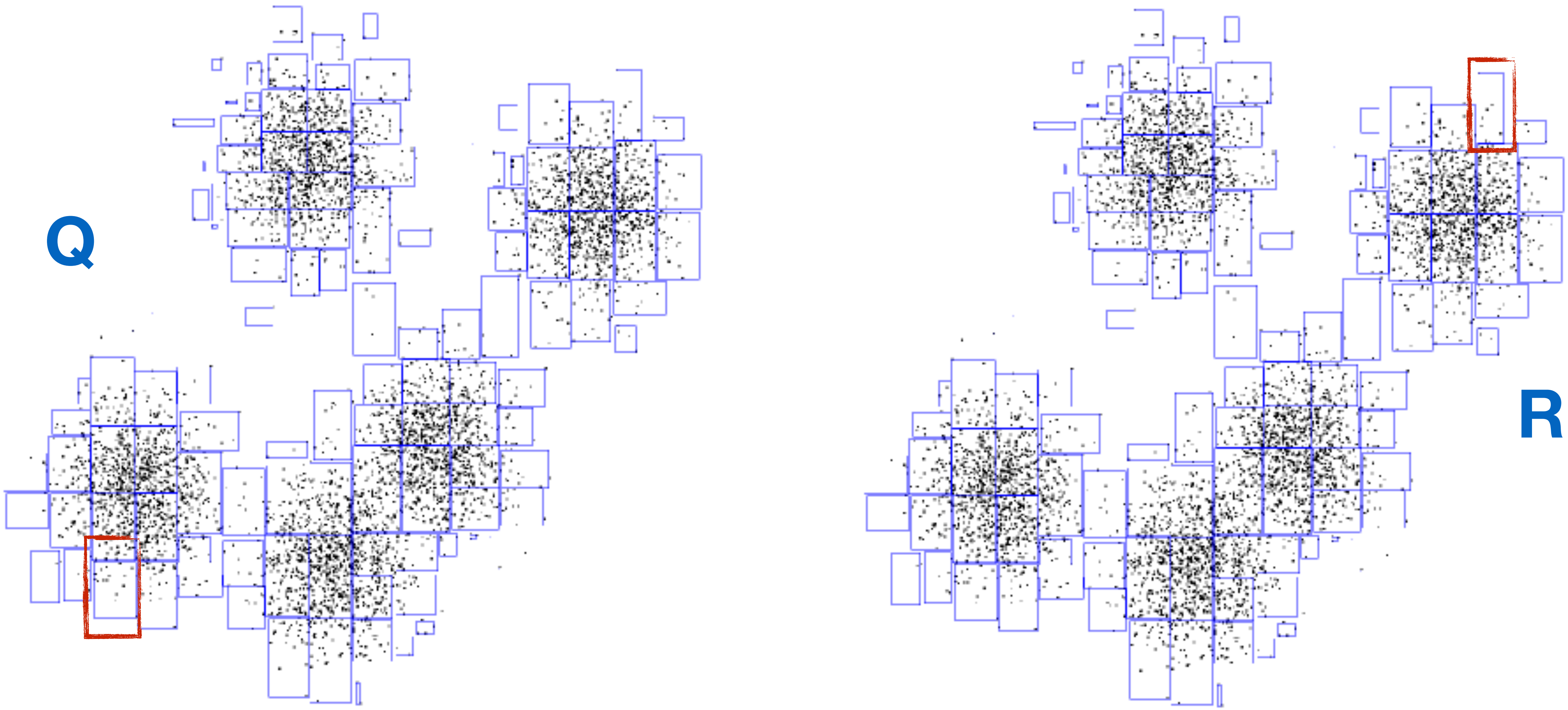
Approximate Condition for EM

Approximate Condition for EM

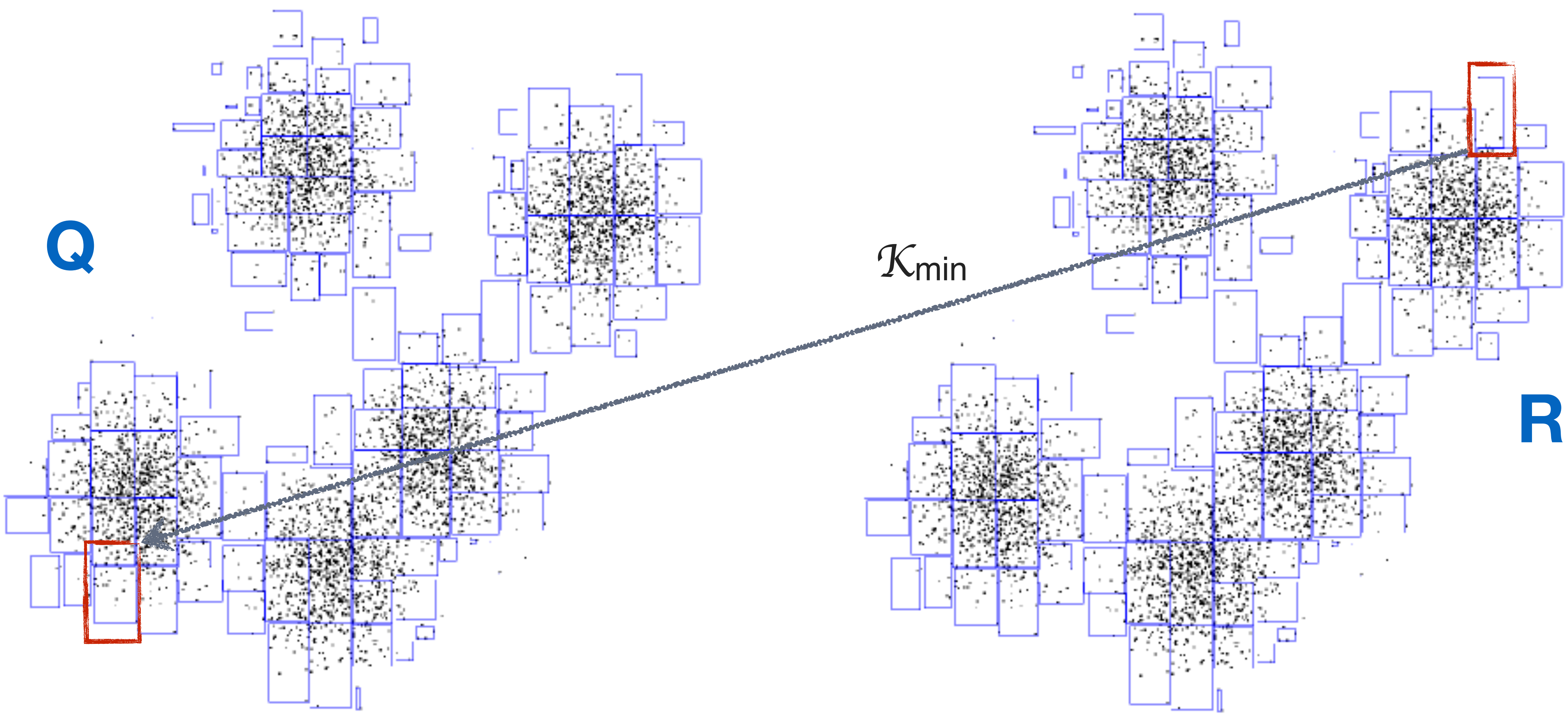




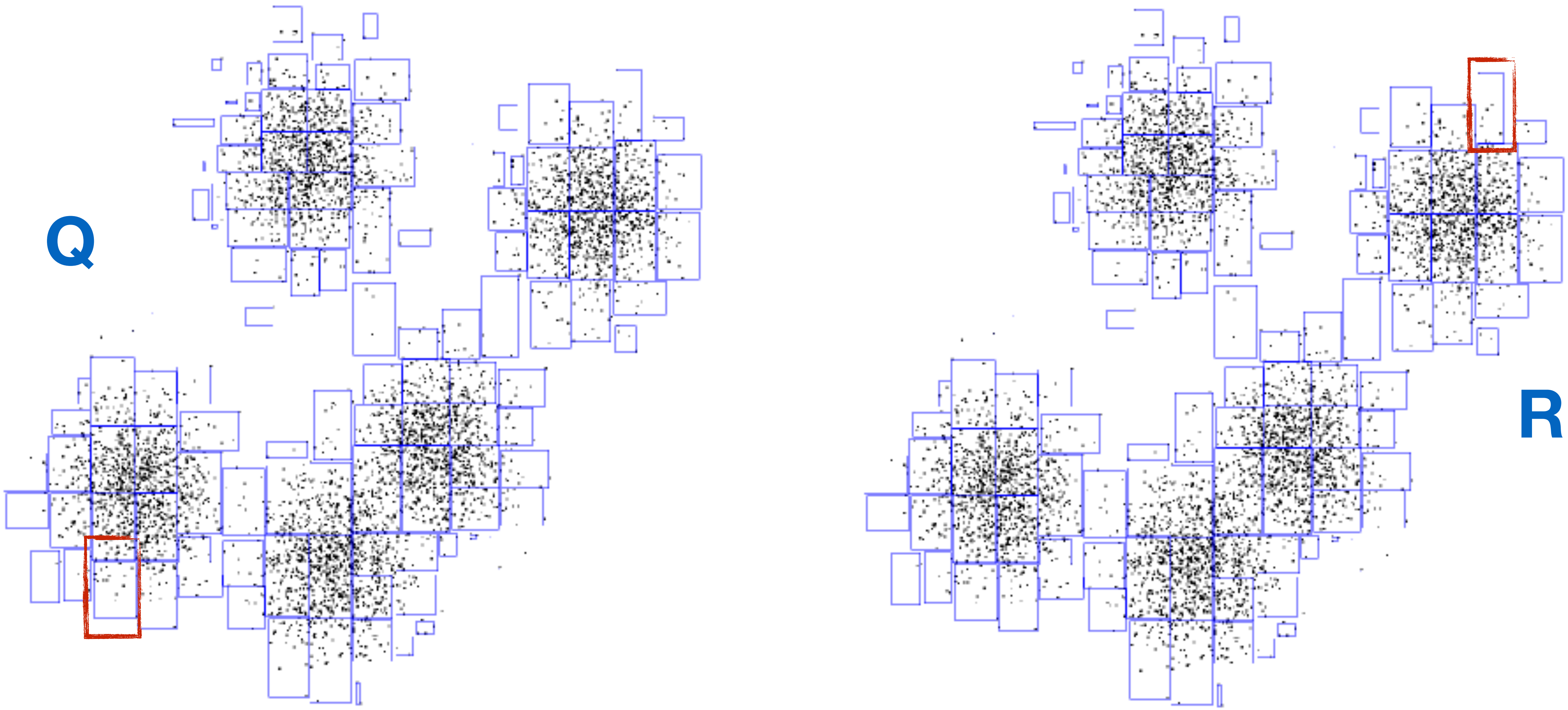
Approximate Condition for EM



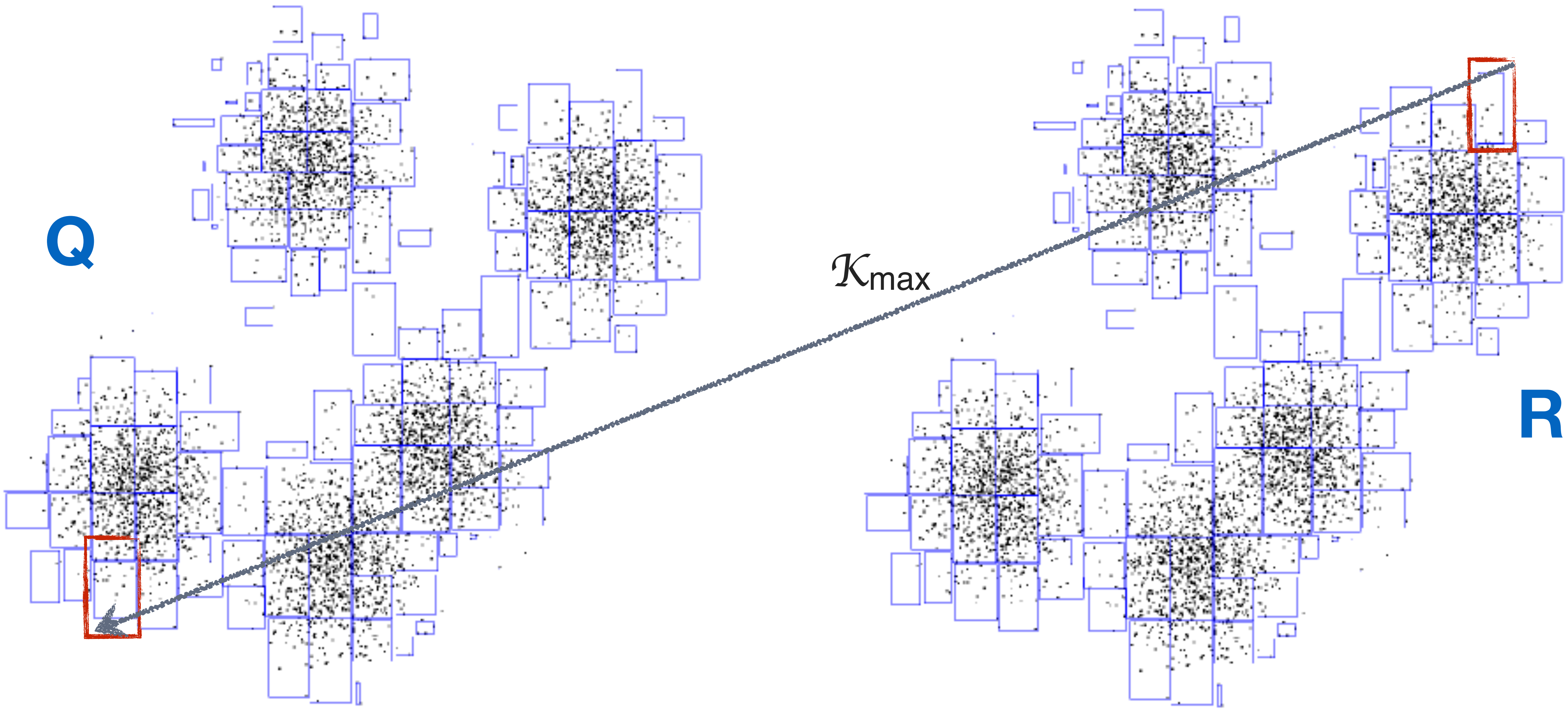
Approximate Condition for EM



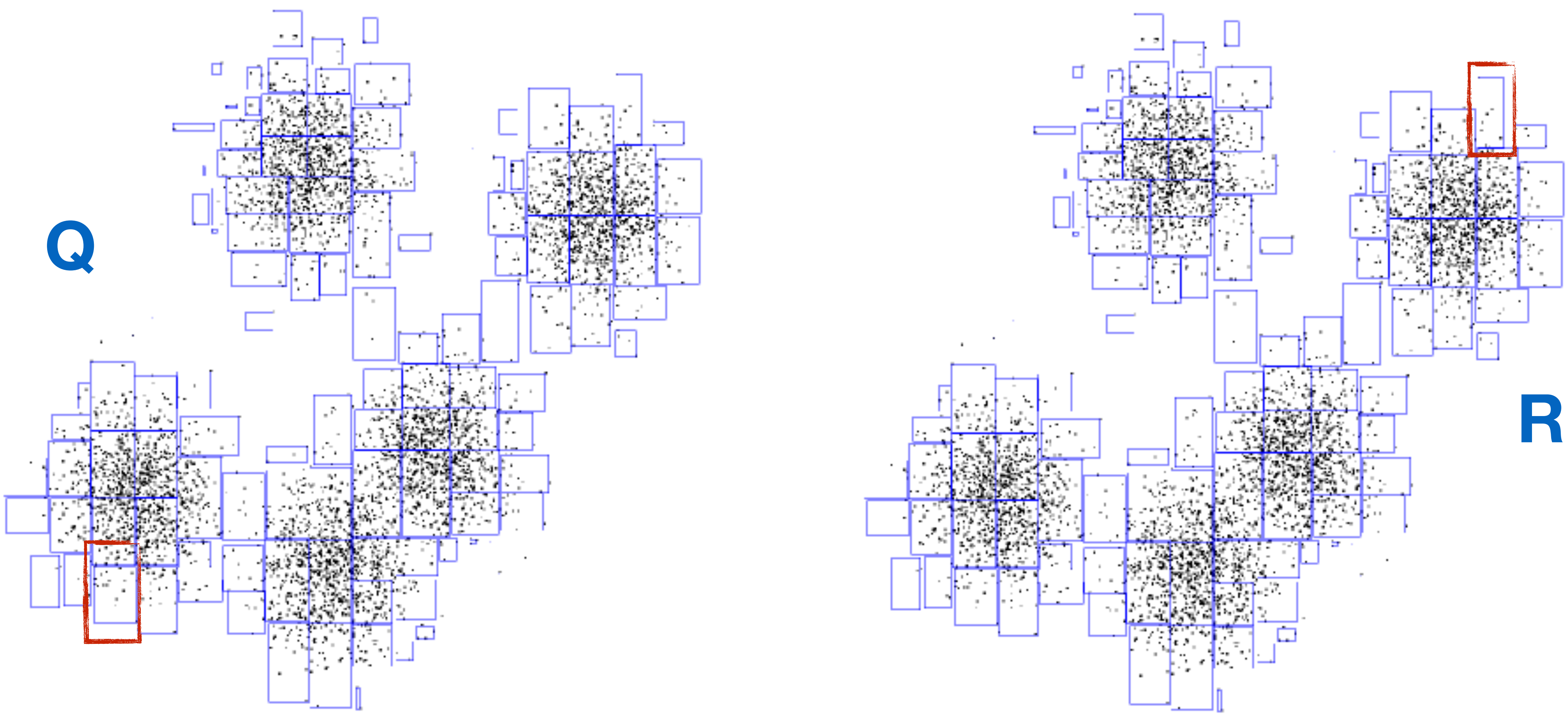
Approximate Condition for EM



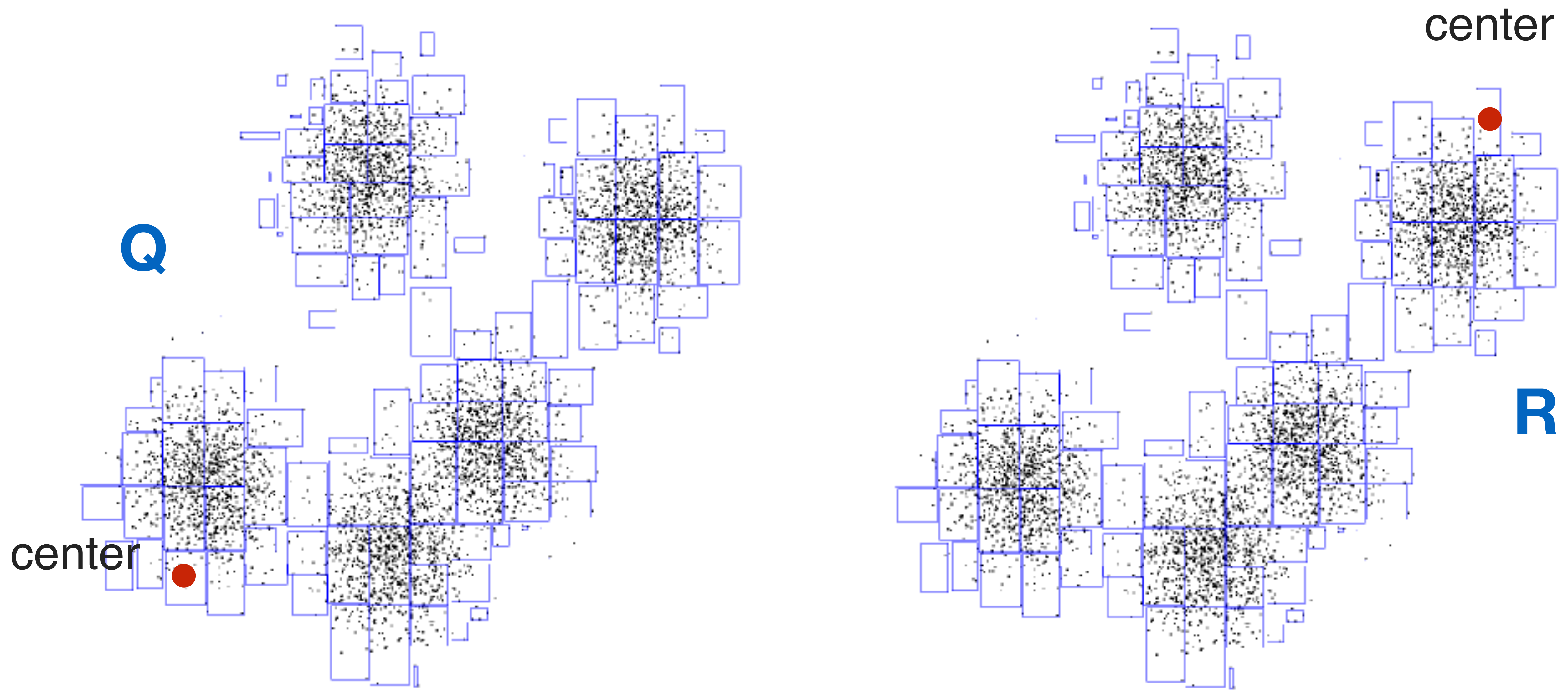
Approximate Condition for EM



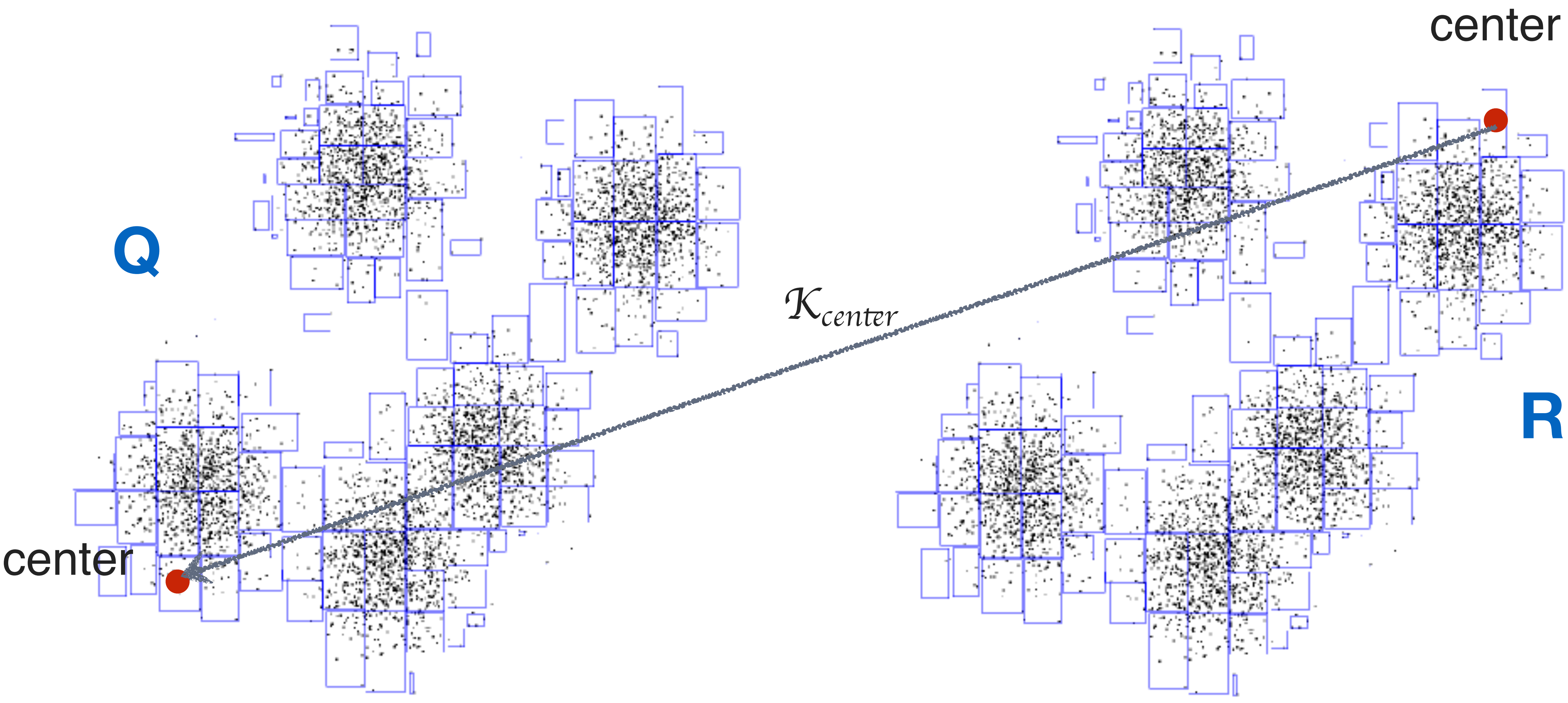
Approximate Condition for EM



Approximate Condition for EM

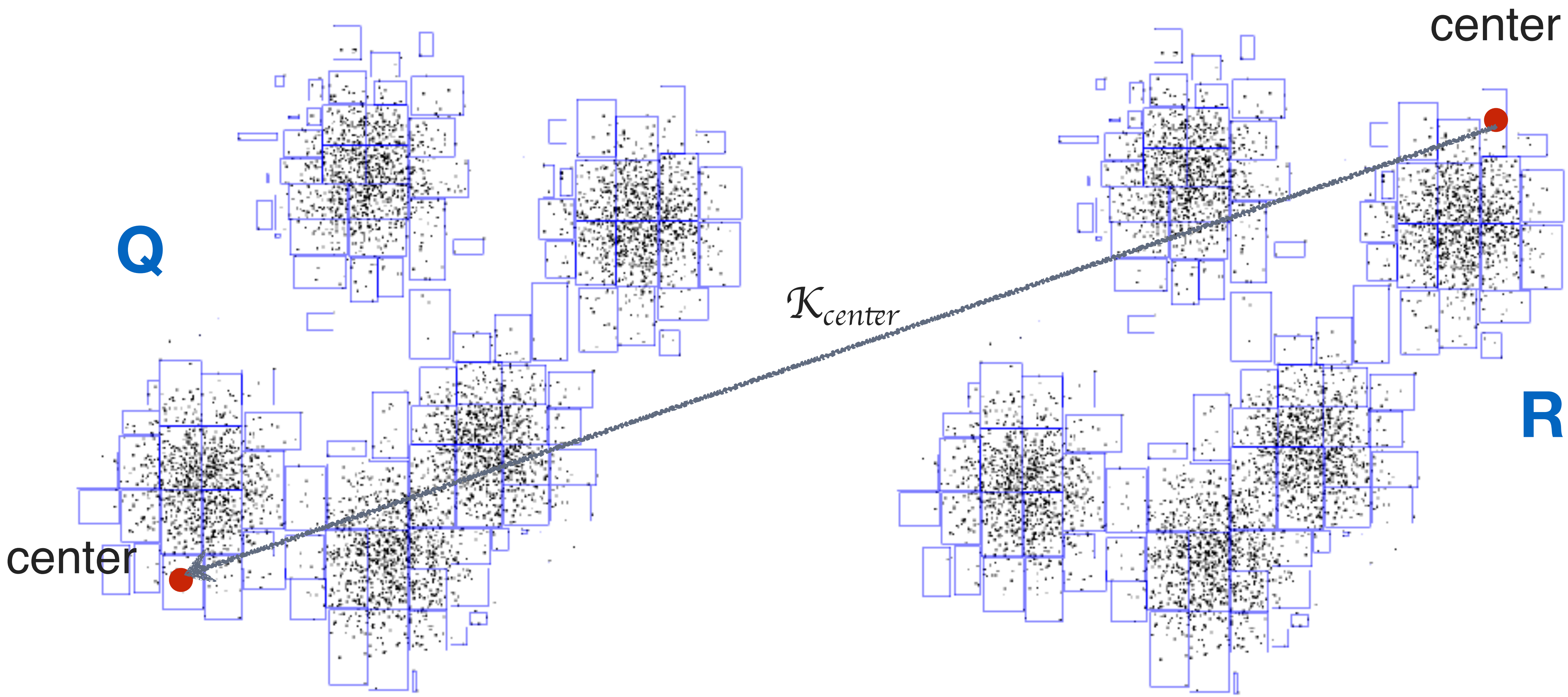


Approximate Condition for EM



Approximate Condition for EM

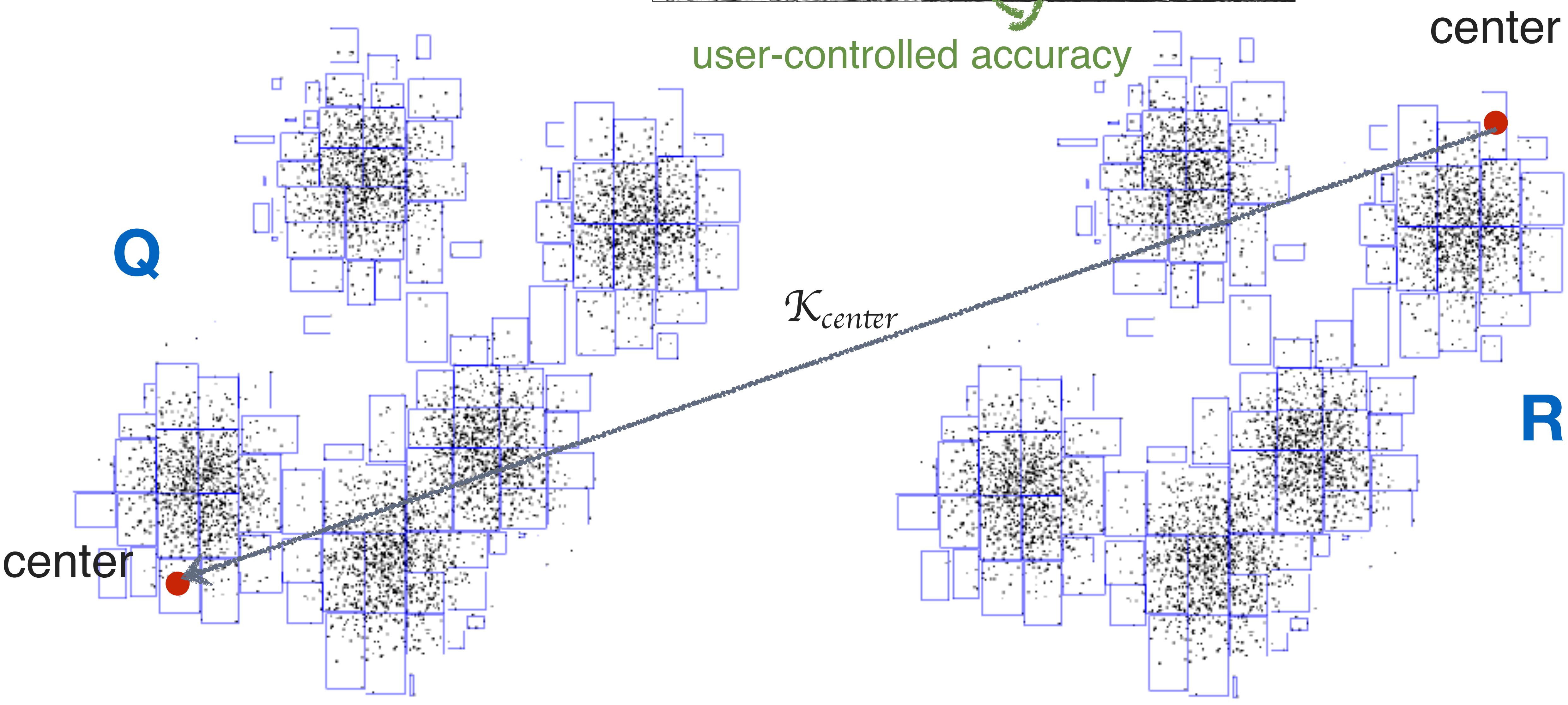
$$\mathcal{K}_{max} - \mathcal{K}_{min} < \beta \times \mathcal{K}_{center}$$



Approximate Condition for EM

$$\mathcal{K}_{max} - \mathcal{K}_{min} < \beta \times \mathcal{K}_{center}$$

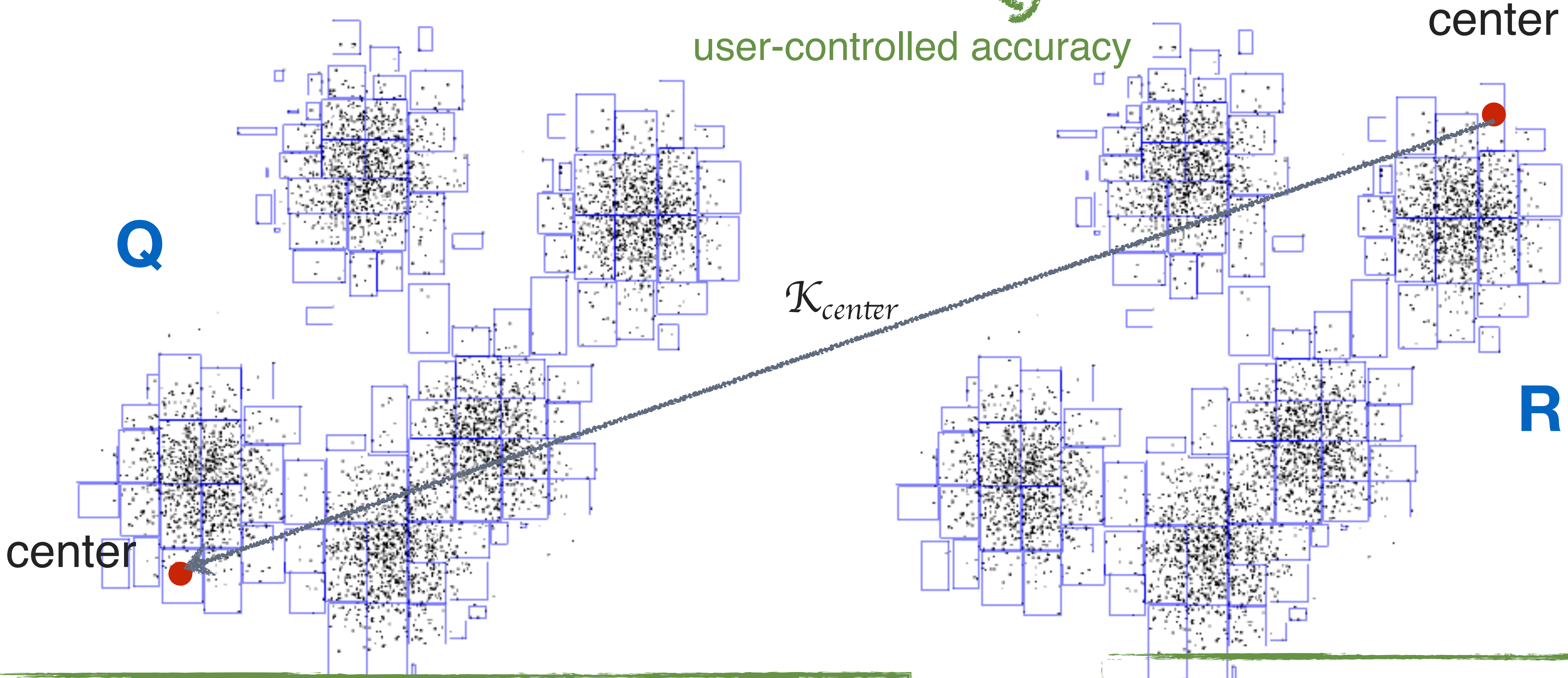
user-controlled accuracy



Approximate Condition for EM

$$\mathcal{K}_{max} - \mathcal{K}_{min} < \beta \times \mathcal{K}_{center}$$

user-controlled accuracy



log liklihood:

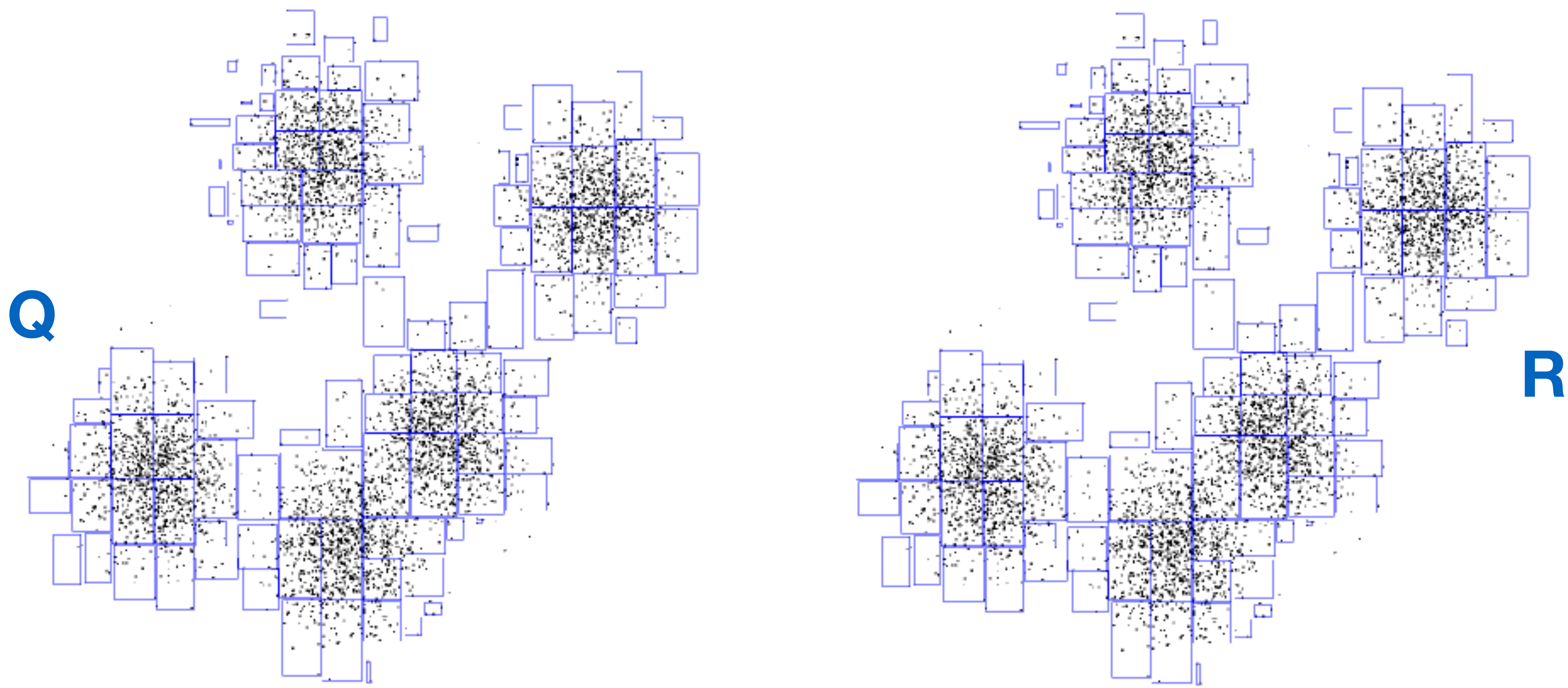
$$\log \sum_{i=1}^{K} \pi_i \mathcal{N}(x_{max}|\theta_i) - \log \sum_{i=1}^{K} \pi_i \mathcal{N}(x_{min}|\theta_i) < \alpha \log(\sum_{i=1}^{K} \pi_i \mathcal{N}(x_{mean}|\theta_i))$$

E-step:

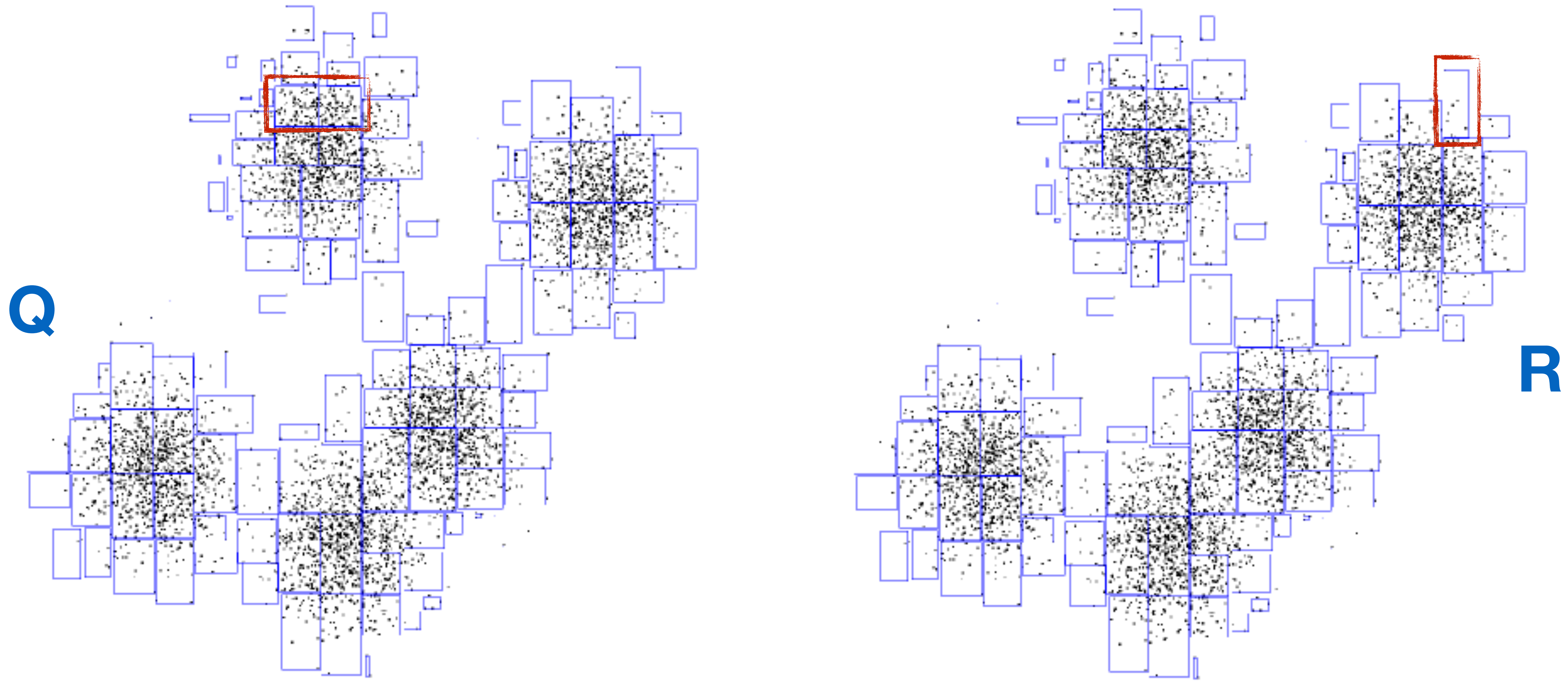
$$(r_{i,max} - r_{i,min}) < \beta\, r_{i,mean} (i = 1, ..., K)$$

Prune Condition for Hausdorff distance: $\max_q, \min_r ||x_q - x_r||$

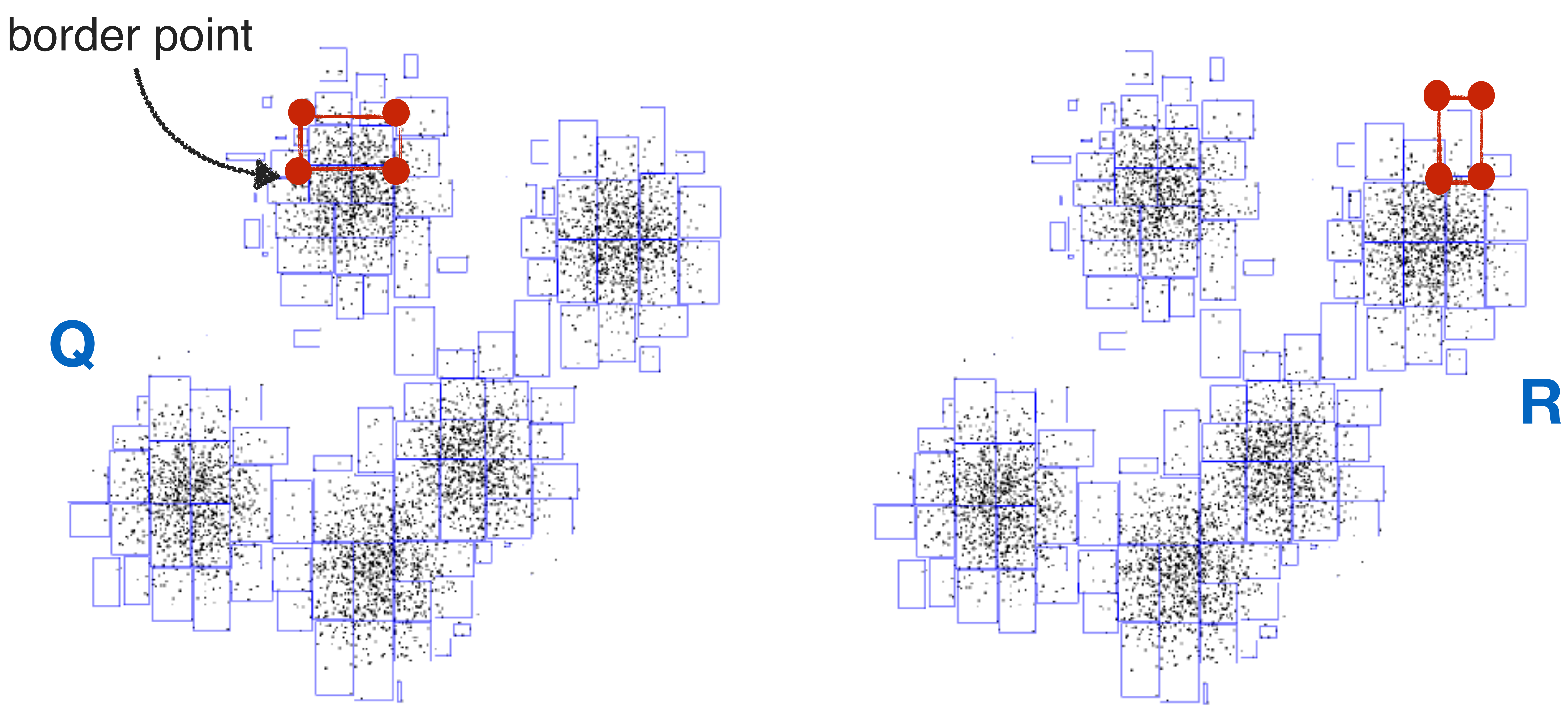
Prune Condition for Hausdorff distance: $\max_q, \min_r ||x_q - x_r||$



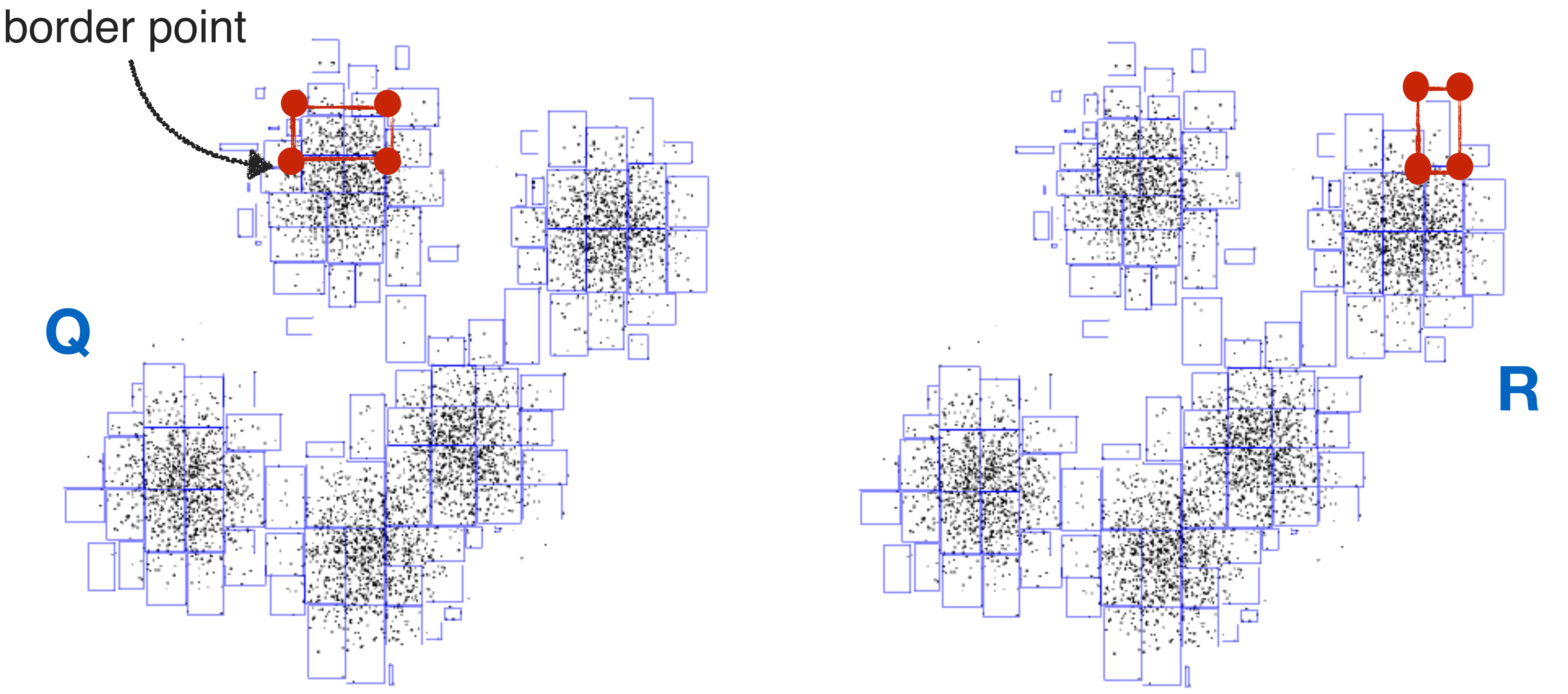
Prune Condition for Hausdorff distance: $\max_{q}, \min_{r} ||x_q - x_r||$



Prune Condition for Hausdorff distance: $\max_q, \min_r ||x_q - x_r||$



Prune Condition for Hausdorff distance: $\max_q, \min_r ||x_q - x_r||$



$$op_{\oplus_1}(\tau_1, \mathcal{K}(x_q, x_r)|op_{\oplus_2}(\tau_2, \mathcal{K}(x_q, x_r))) \;\; s.t. \quad \forall x_q \in \mathcal{N}_q^{\text{border}}, \forall x_r \in \mathcal{N}_r^{\text{border}}$$

Outline

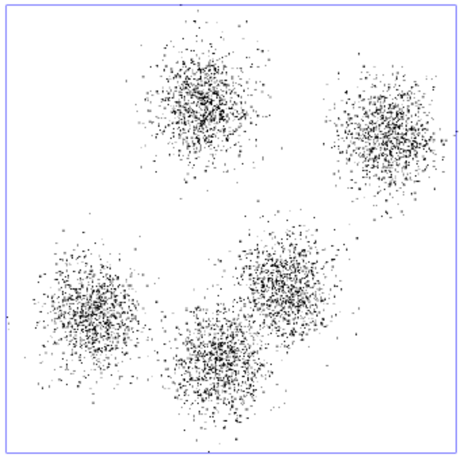
- Introduction
- PASCAL Framework
  - Space Partitioning Trees
  - Tree Traversal
  - Prune/Approximate Generators
- Optimizations & Parallelization
- Experiments & Results
- Conclusions & Future Work

Optimizations

- Incremental bounding box calculation

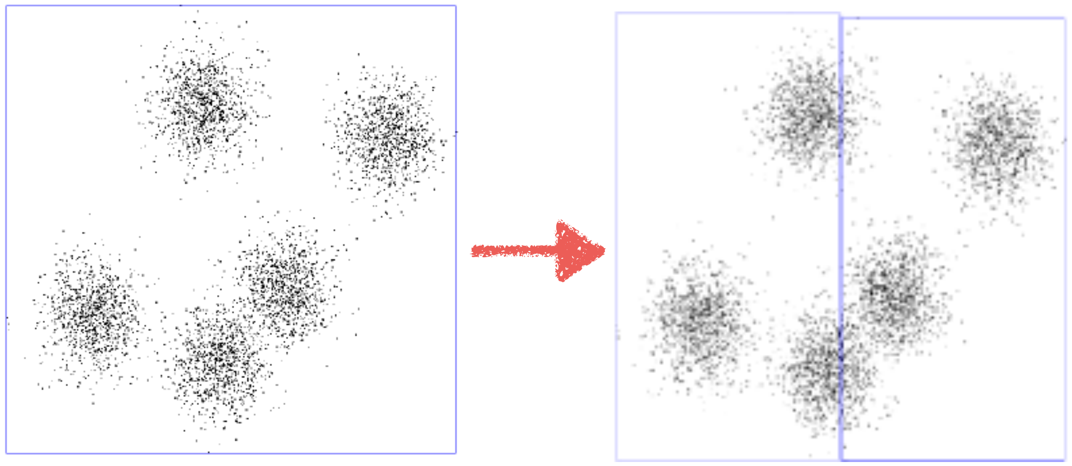
Optimizations

- Incremental bounding box calculation



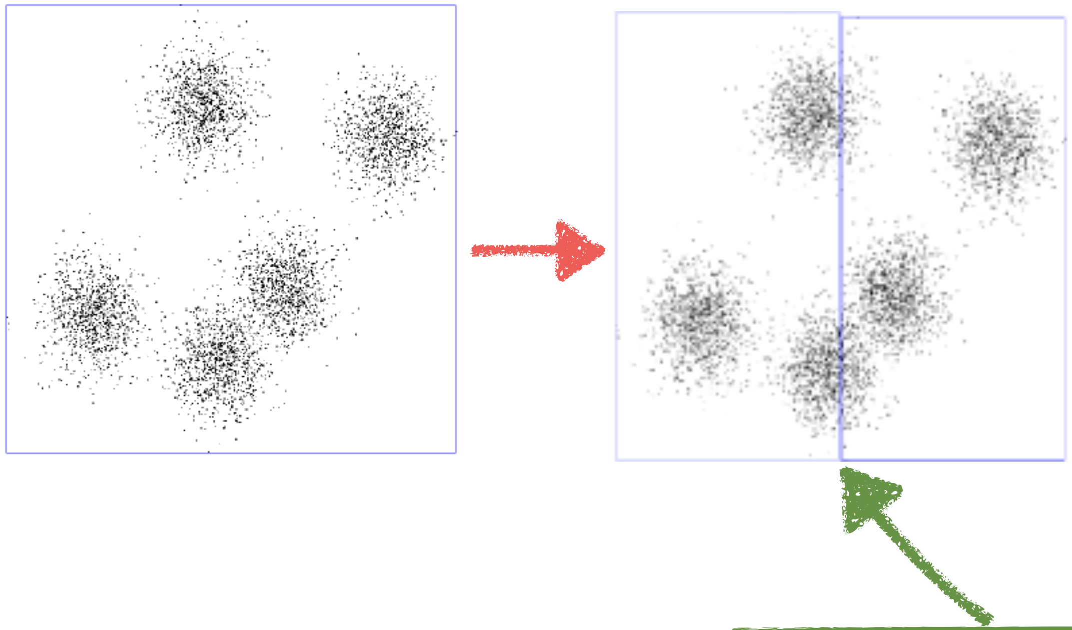
Optimizations

- Incremental bounding box calculation



Optimizations

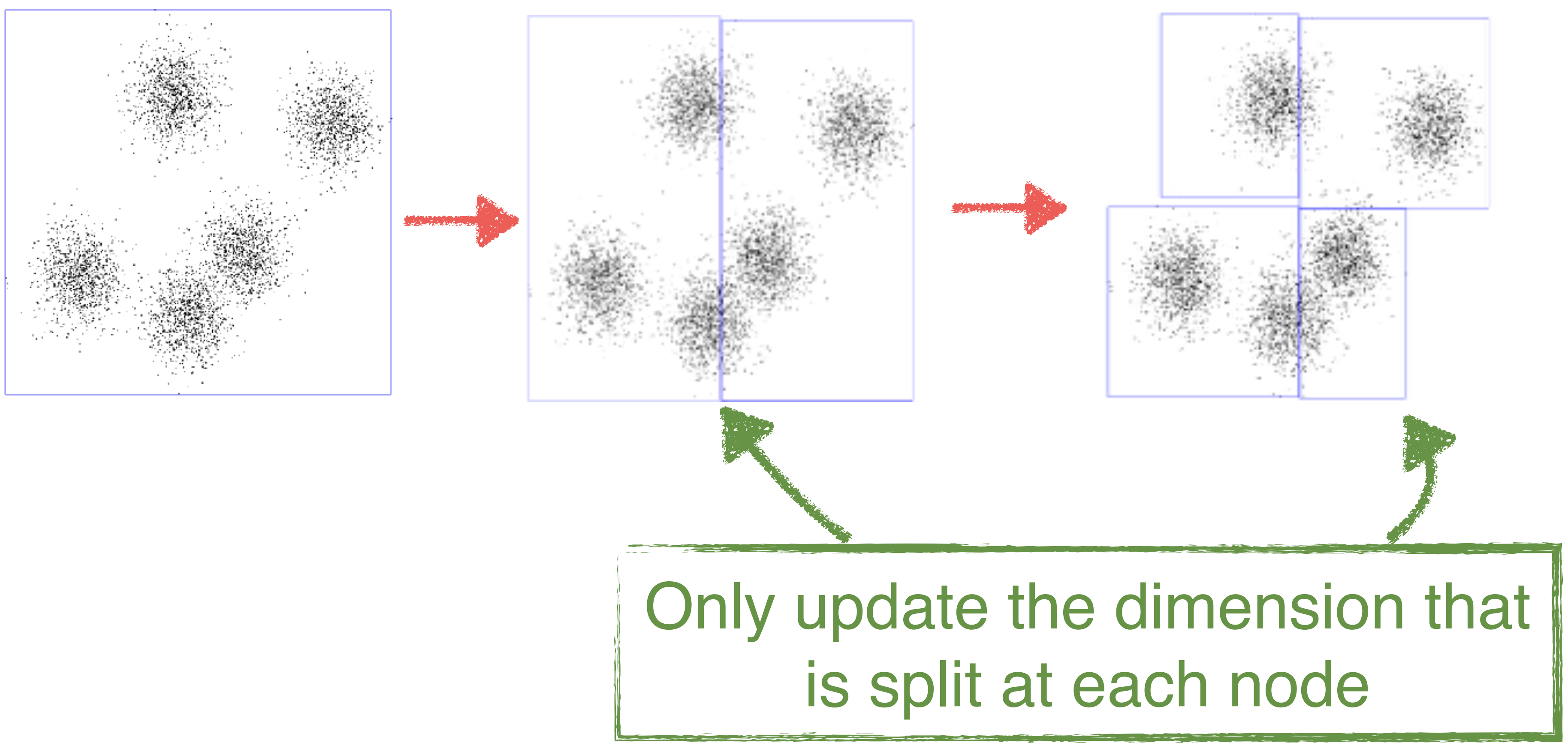
- Incremental bounding box calculation



Only update the dimension that is split at each node

Optimizations

- Incremental bounding box calculation



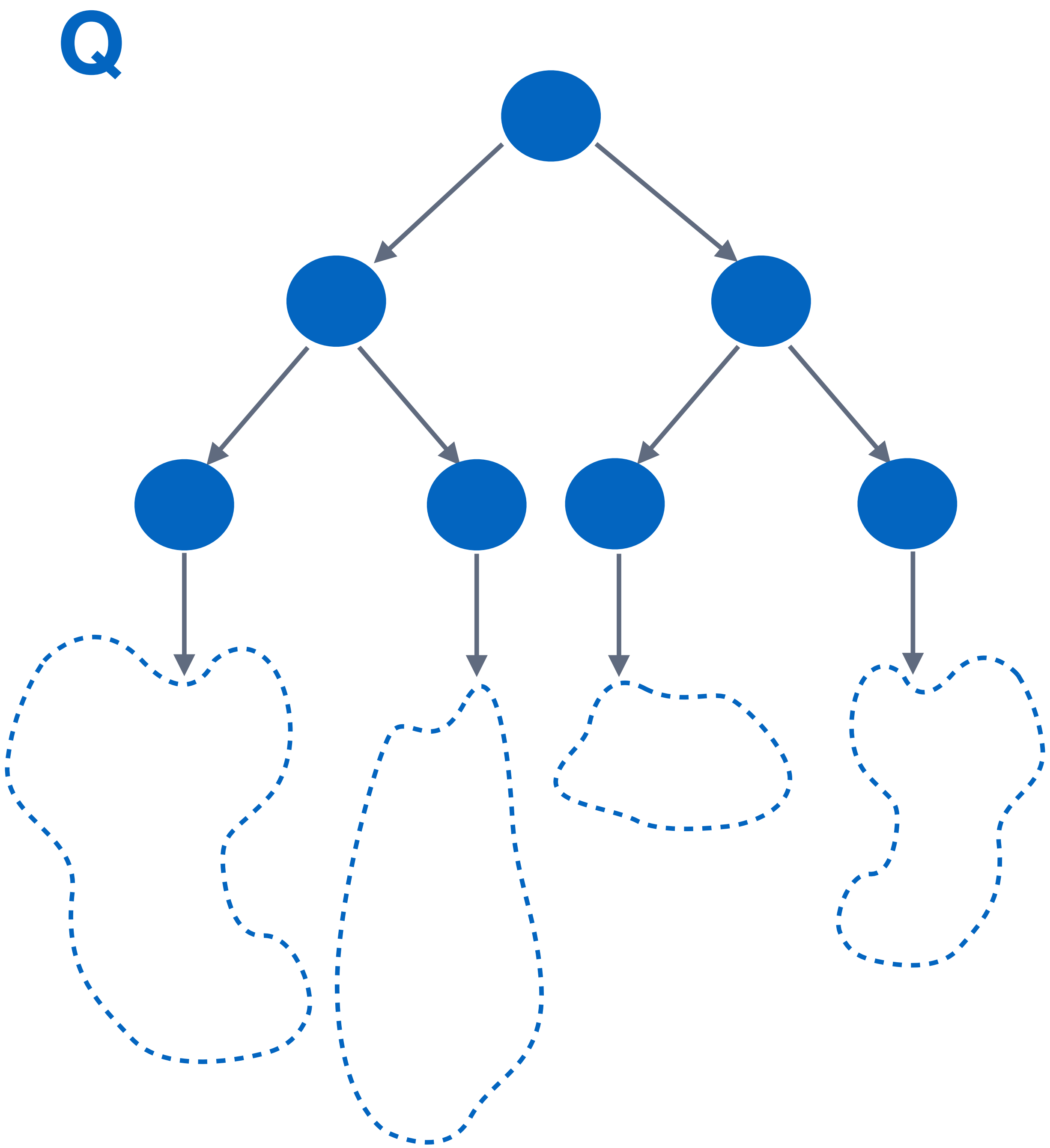
Optimizations

- Incremental bounding box calculation
- Optimal Metric Calculation
  - Reduced distance
    - *e.g.*, squared Euclidean distance
    - Eliminates expensive **sqrt** instruction with long latencies
  - Partial distance
    - Big payoff for large dimensional datasets

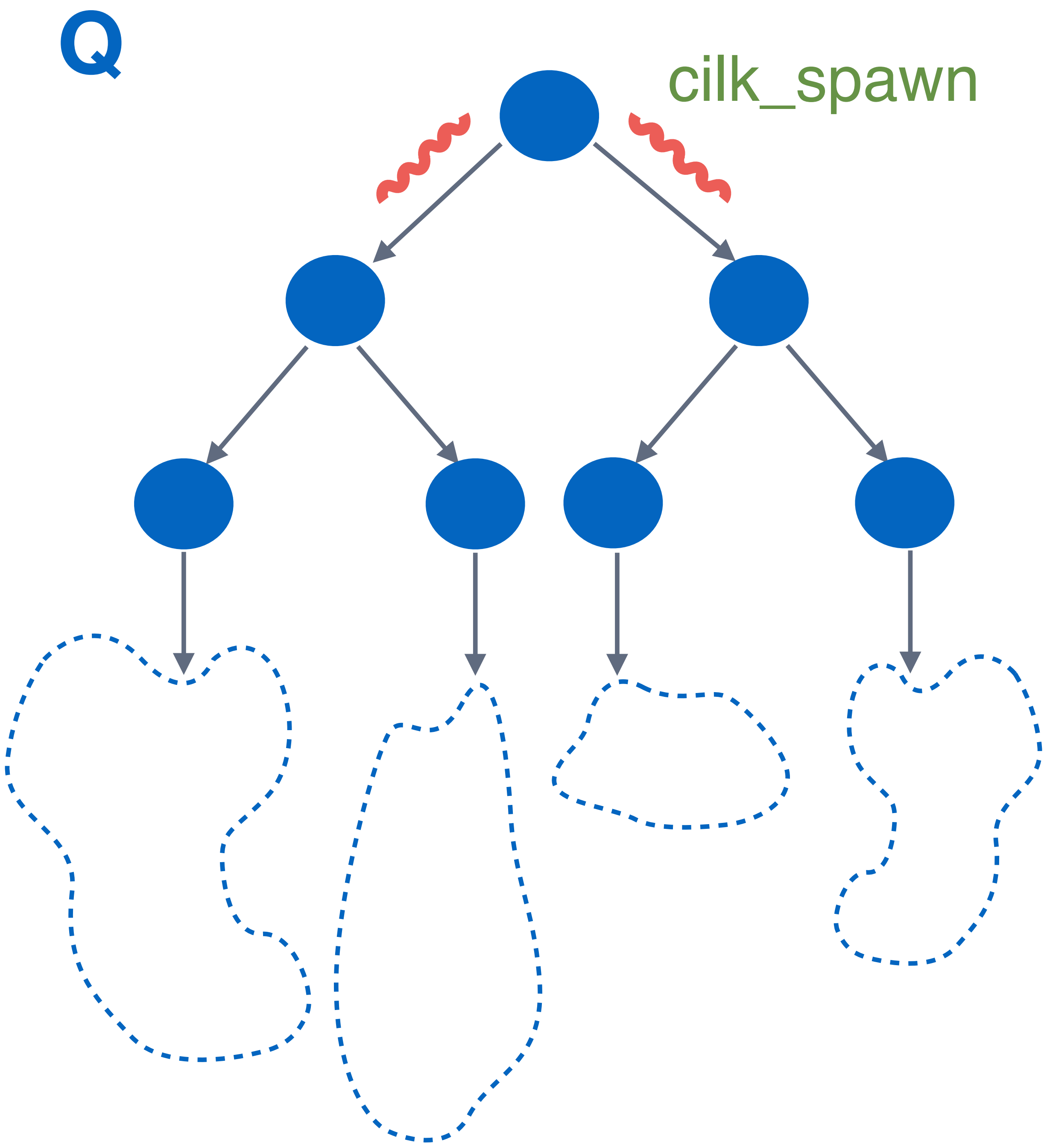
Optimizations

- Incremental bounding box calculation
- Optimal Metric Calculation
  - Reduced distance
    - *e.g.*, squared Euclidean distance
    - Eliminates expensive **sqrt** instruction with long latencies
  - Partial distance
    - Big payoff for large dimensional datasets
- Incremental distance calculation
  - Node-to-node distance computed incrementally from parent's distance in constant time

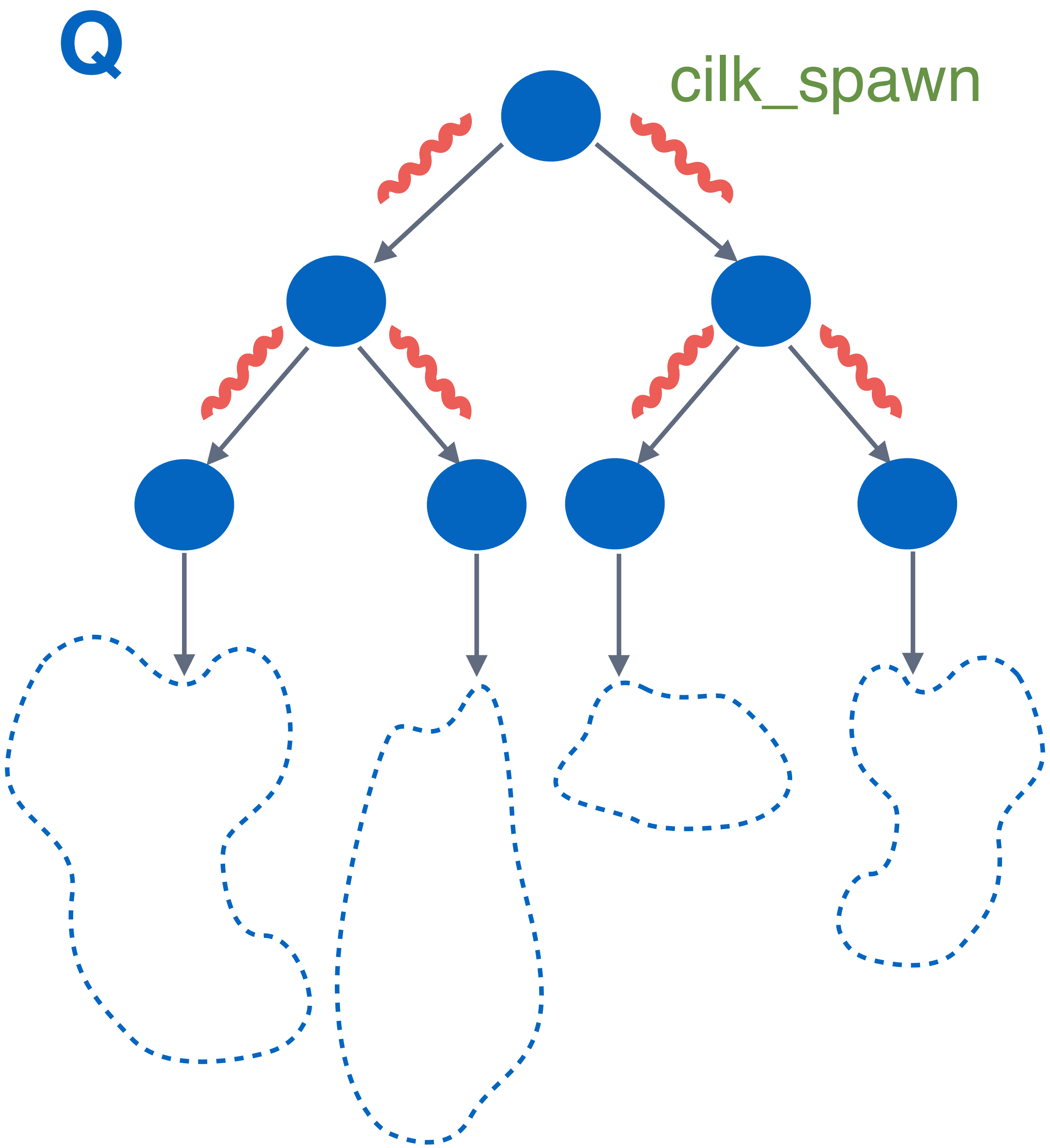
Parallelization



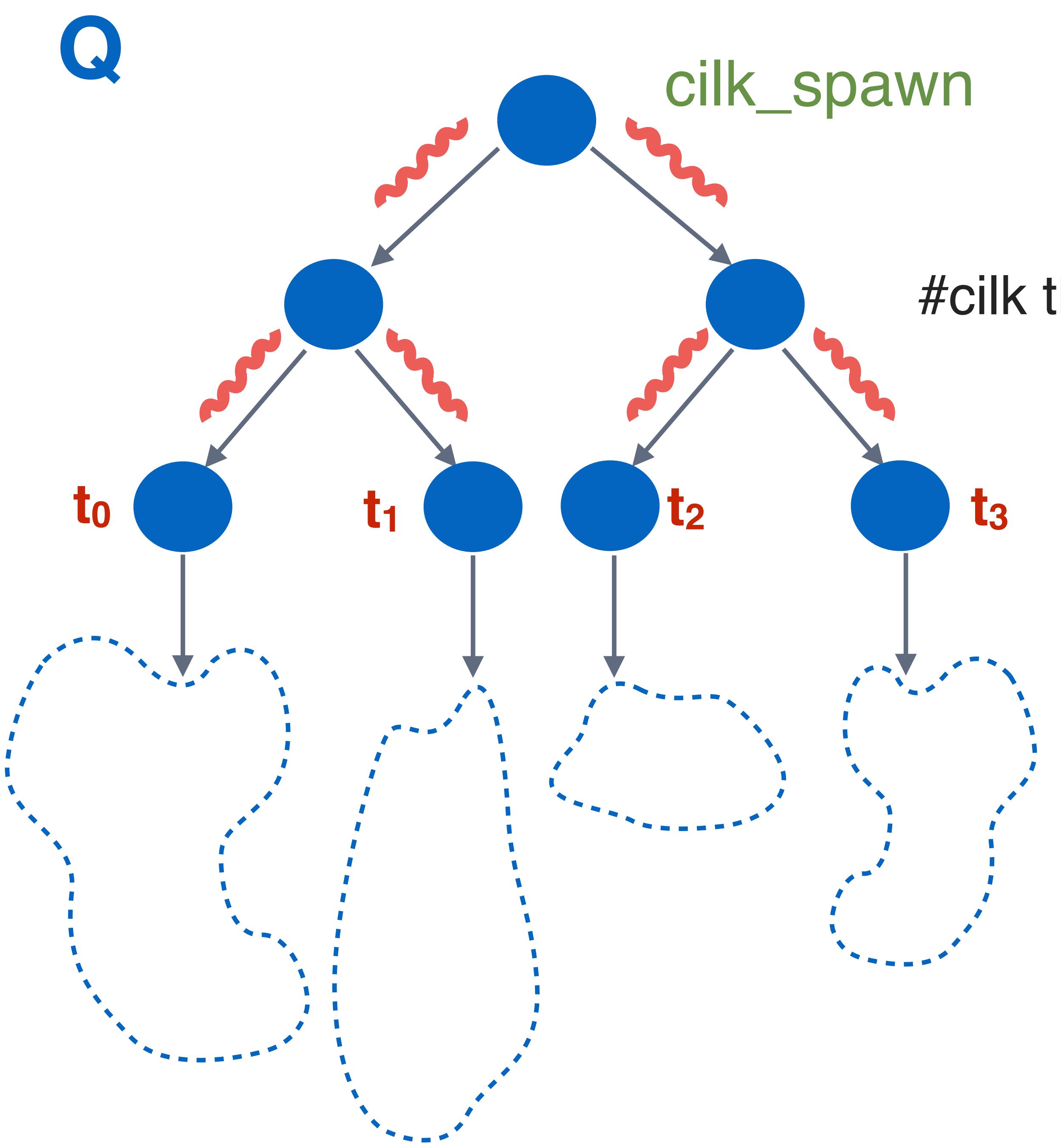
Parallelization



Parallelization

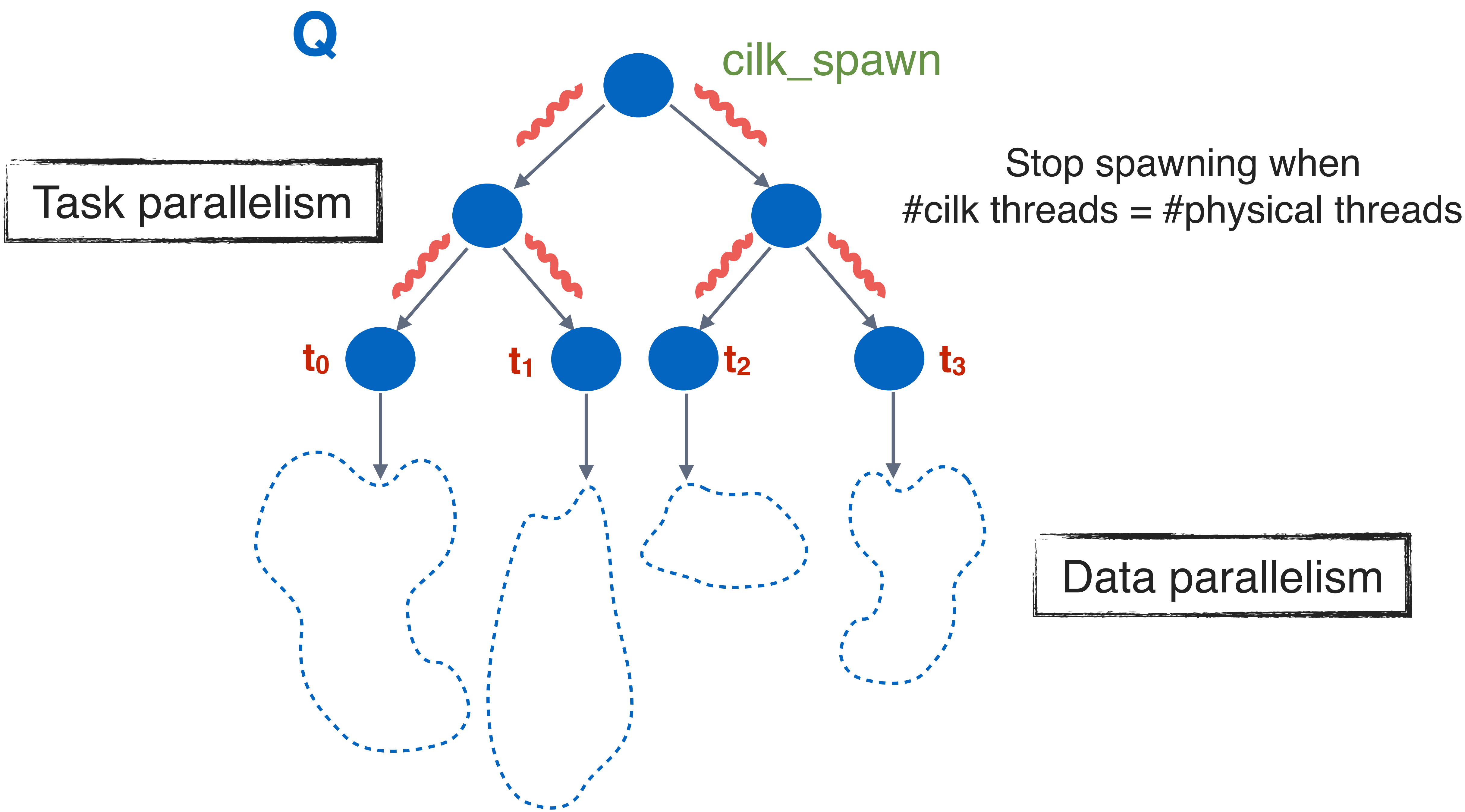


Parallelization

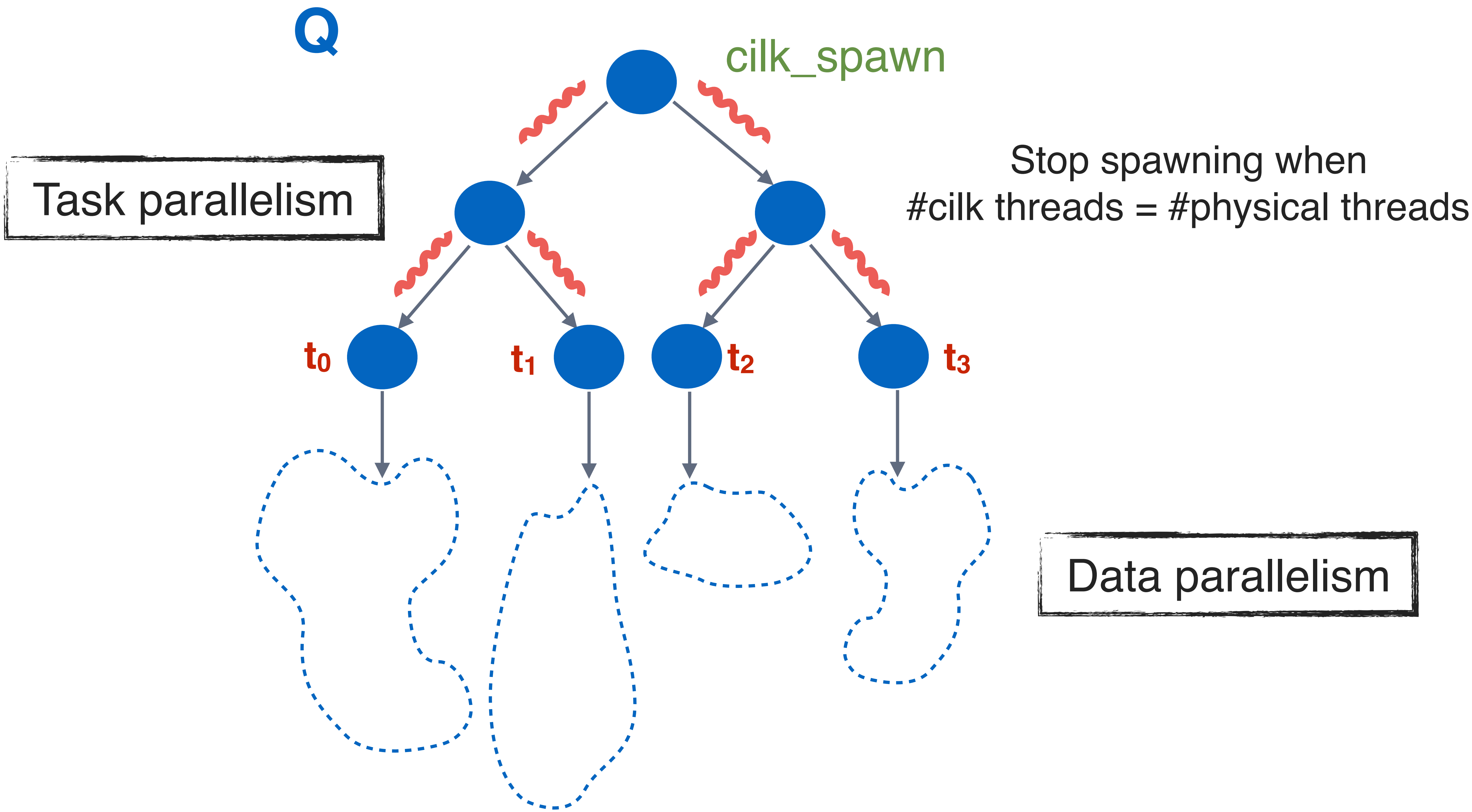


Stop spawning when
#cilk threads = #physical threads

Parallelization



Parallelization



Pruning/Approximation causes load imbalance

Outline

- Introduction
- PASCAL Framework
  - Space Partitioning Trees
  - Tree Traversal
  - Prune/Approximate Generators
- Optimizations & Parallelization
- Experiments & Results
- Conclusions & Future Work

Experimental Setup

- Architecture
  - Dual-socket Intel Xeon E5-2630 v3 processor (Haswell-EP)
  - Each socket has 8 cores
  - Theoretical peak performance of 614.4 GFlops
- Compiler
  - Intel C++ complier (icpc v15.0.2)
  - Python v2.7.6 (Scikit-learn)
  - Java v1.8.0 (Weka)

Dataset	N	d
Yahoo!	41904293	11
IHEPC	2075259	9
HIGGS	11000000	28
Census	2458285	68
KDD	4898431	42

Case Studies (Direct)

- Nearest Neighbors $\forall q, \ \arg\min_r ||x_q - x_r||$
- Range-Search $\forall q, \ \bigcup \arg_r \ I\left(||x_q - x_r|| \leq h\right)$
- Kernel Density Estimation $\forall q, \frac{1}{N_r} \sum_r K\left(\frac{||x_q - x_r||}{\sigma}\right)$
- Hausdorff Distance $\max_q, \ \min_r \ ||x_q - x_r||$

Case Studies (Iterative)

- Expectation Maximization (EM)

  E-step $\forall q, \forall r, \quad \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$

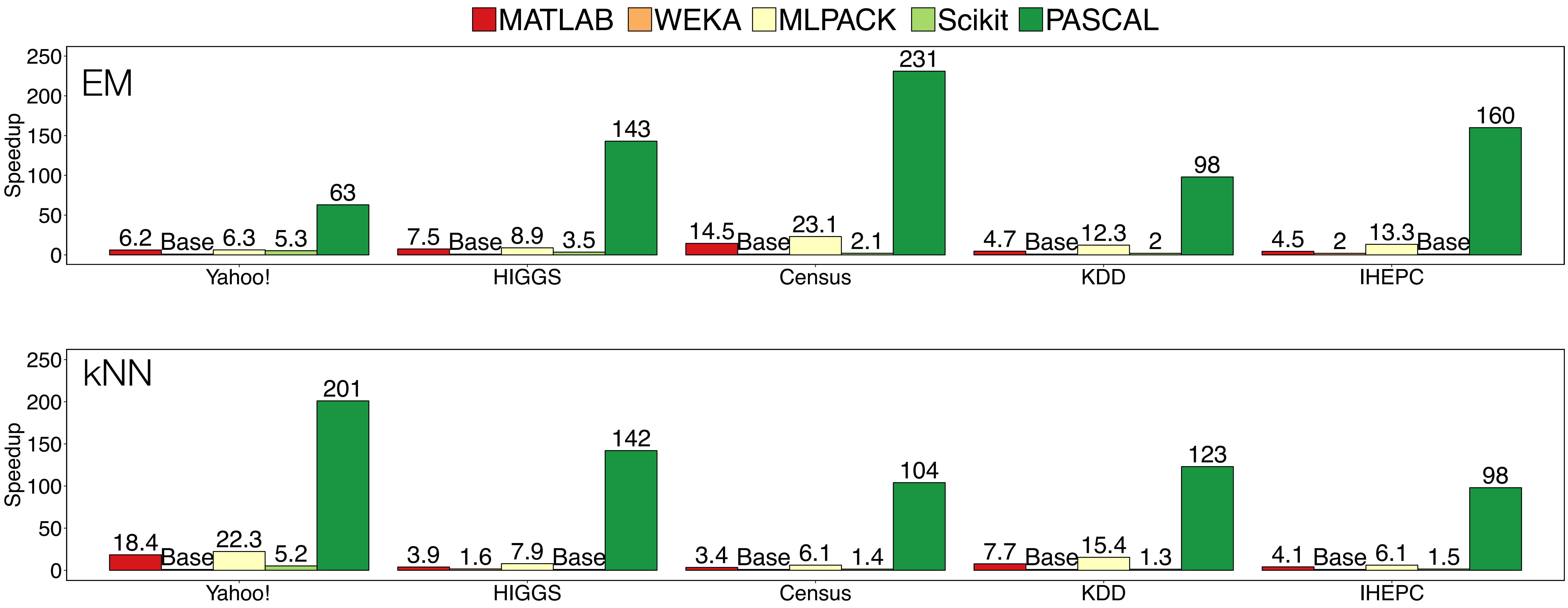
  M-step

  Log-likelihood $\sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)$

- Euclidean Minimum Spanning Tree $\forall q, \ \arg\min_r ||x_q - x_r||$

Library Comparison

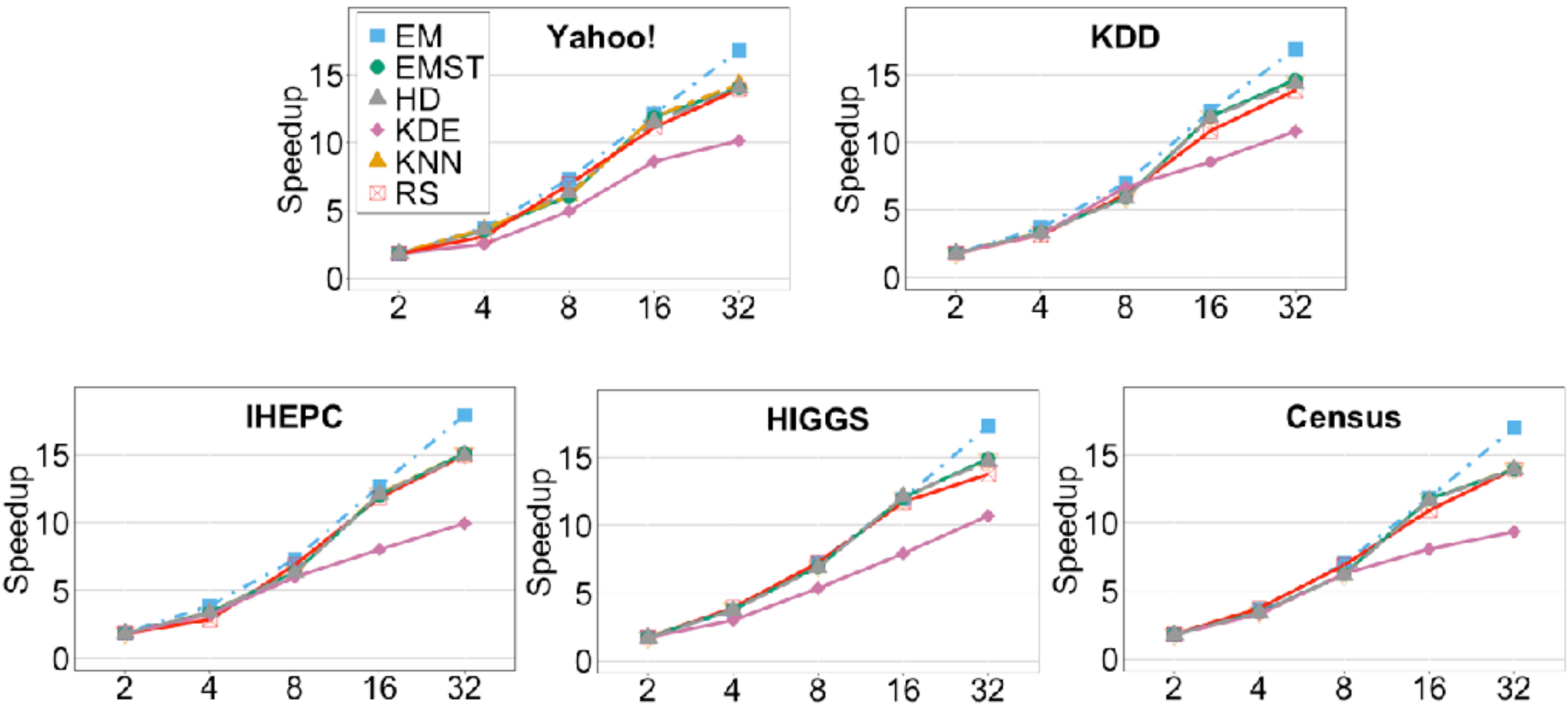
- **Weka**: 6,677,053 downloads, written in Java
- **Scikit-learn**: 121,841 downloads, written in Python
- **MATLAB**: over 1,000,000 licensed users, uses C in backend
- **MLPACK:** exploits C++ language features to provide maximum performance



Speedup Breakdown

	KNN			EM			KDE			HD			RS			EMST		
	Alg	+Opt	+Par	Alg	+Opt	+Par	Alg	+Opt	+Par	Alg	+Opt	+Par	Alg	+Opt	+Par	Alg	+Opt	+Par
Yahoo!	3.1	12.1	173.1	1.6	3.2	53.7	2.1	9.1	92.1	2.5	11.5	161.1	2.2	9.1	126.8	2.9	11.9	166.7
HIGGS	2.1	7.3	108.1	1.5	6.8	117.6	1.7	4.7	50.1	1.9	6.1	89.6	1.9	6.3	86.5	2.0	6.9	102.8
Census	1.4	6.5	90.8	1.3	11.2	190.0	1.4	8.1	75.6	1.3	10.2	141.8	1.3	10.4	144.9	1.4	10.9	151.6
KDD	1.6	6.8	100.7	1.4	4.1	70.9	1.5	3.1	33.5	1.4	3.8	54.4	1.4	5.1	70.5	1.5	3.8	55.5
IHEPC	3.0	4.3	61.5	1.5	7.6	127.6	2.0	5.4	53.6	2.5	6.8	101.3	2.1	6.3	94.1	2.9	7.1	107.1

Scalability



Summary and Status

- First generalized algorithmic framework for N-body problems
  - Out-of-the-box new optimal algorithms
    - O(N log N) EM algorithm
    - O(N) Hausdorff distance algorithm
  - Generalizes to more than two operators
  - 10-230x speedup from optimal tree algorithm, domain-specific optimizations and parallelization
- Short-term: DSL + code generator for base-case, optimizations and parallelization
- Long-term: Extend to GPUs and distributed memory systems