Data-Dependent LSH for the Earth Mover's Distance

Rajesh Jayaram
Google Research NYC

Erik Waingarten
Penn

Tian Zhang
Penn

Focus of This Talk

- EMD and Probabilistic Tree Embeddings
  - Will not define LSH in this talk

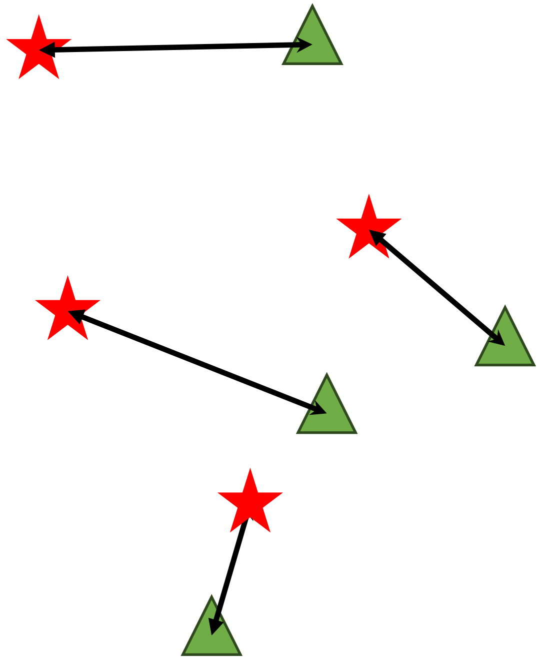
- "Data-Dependent" Probabilistic Trees [Chen-J-Levi-Waingarten STOC '22]
  - Suited for one EMD comparison.

- (New) Extension Lemma: controlling distortion over entire space.
  - Suited for *many* comparisons (NNS)

Earth Mover's Distance

Metric space: $(\mathbb{R}^d, \ell_1)$

Multisets: $A = \{a_1, \ldots, a_s\}, B = \{b_1, \ldots, b_s\} \subset \mathbb{R}^d$



$$\mathrm{EMD}(A,B) = \min_{\substack{\pi : [s]\to[s] \\ \text{bijection}}} \sum_{i=1}^{s} \|a_i - b_{\pi(i)}\|_1$$

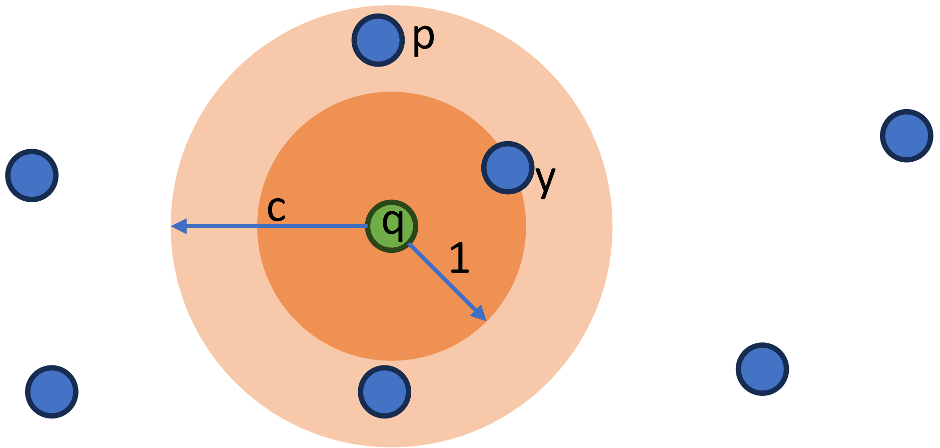
- Min cost perfect bipartite matching (/w triangle inequality)

Approximate Nearest Neighbor Search (ANN)

Fix a Metric Space X, approximation $c \geq 1$

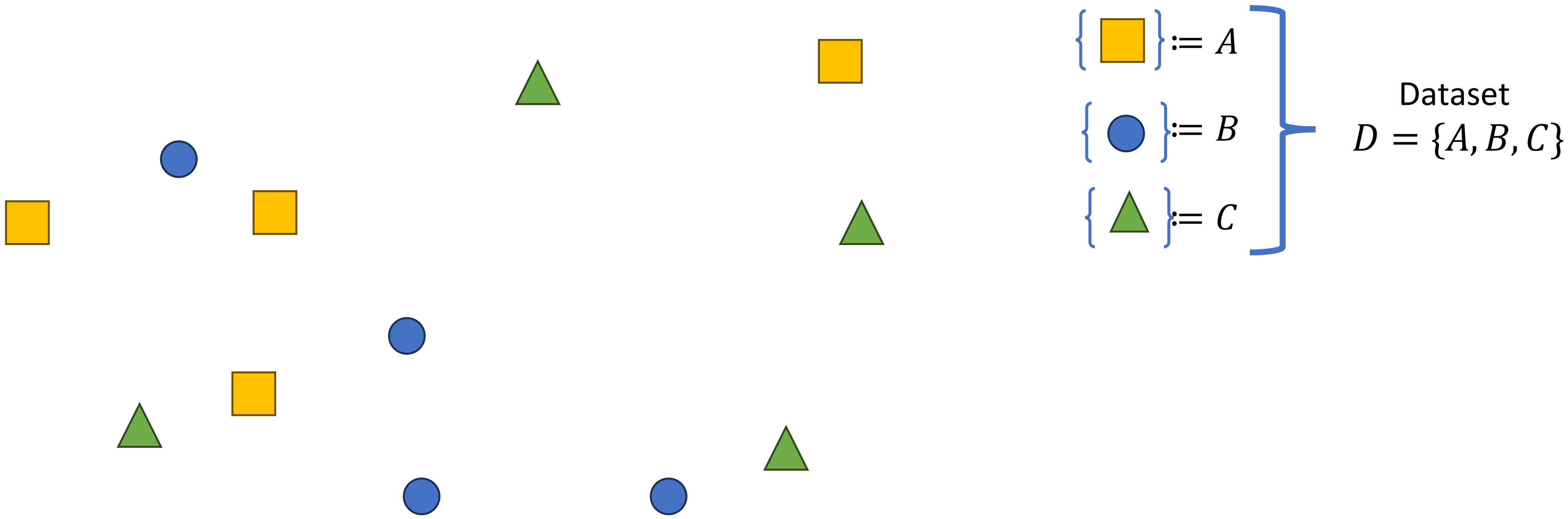
- **Preprocess**: a dataset $D \subset X$ of n points.
- **Query:** Given $q \in X$, output any $p \in D$ such that

$$d_X(q,p) \leq c \cdot \min_{y \in D} d_X(q,y)$$



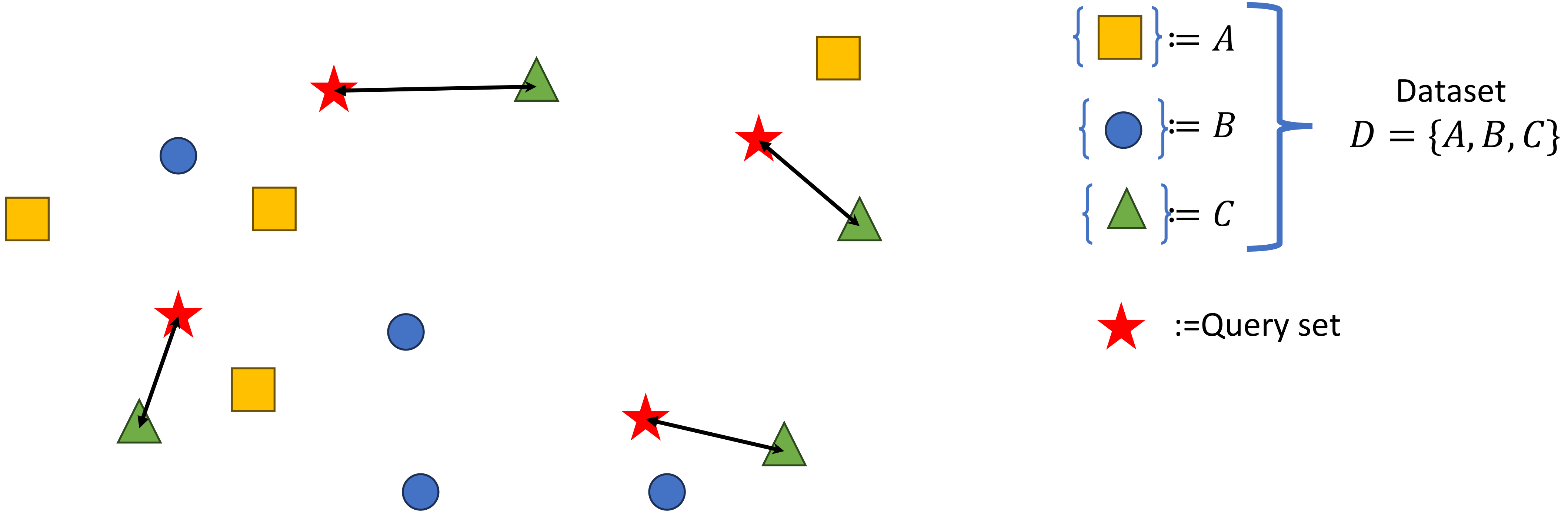
ANN for EMD

Reminder: A “point” in EMD is a size-s subset $A \subset \mathbb{R}^d$



ANN for EMD

Reminder: A "point" in EMD is a size-s subset $A \subset \mathbb{R}^d$



Parameterization

$\mathrm{EMD}_s(\mathbb{R}^d, \ell_1)$:= EMD over size s subsets of $(\mathbb{R}^d, \ell_1)$

Two Key Parameters

- n := size of dataset
- s := subset size

From now on:

- WLOG: $\Delta, d = \mathrm{poly}(s)$, ($\Delta$:= Aspect Ratio)

Think of $n \gg s$, since n is the size of the dataset, s is description size of single point

Ideal Trade-offs: Case of ℓ_1

Theorem (Indyk-Motwani STOC'98): For any $\epsilon > 0$, there is an ANN data structure for n points in $(\mathbb{R}^d, \|\cdot\|_1)$ which obtains:

- Approximation: $O\left(\frac{1}{\epsilon}\right)$
- Space & pre-processing time: $O(d\ ) \cdot n^{1+\epsilon}$
- Query time: $O(d) \cdot n^{\epsilon}$

Ideal Trade-offs: Case of ℓ_1

Theorem (Indyk-Motwani STOC'98): For any $\epsilon > 0$, there is an ANN data structure for n points in $\left(\mathbb{R}^d, \|\cdot\|_1\right)$ which obtains:

- Approximation: $O\left(\frac{1}{\epsilon}\right)$
- Space & pre-processing time: $O(d\ ) \cdot n^{1+\epsilon}$
- Query time: $O(d) \cdot n^{\epsilon}$

Gold standard trade-off

- Optimal for ℓ_1
  - (Andoni, Laarhoven, Razenshteyn and Waingarten SODA'17)

But EMD more complex than ℓ_1 ...

EMD ANN: Prior Work + Main Result

Theorem (Indyk STOC'04): For any $\epsilon > 0$, there is an ANN data structure for s $\mathrm{EMD_s}(\mathbb{R}^d, \ell_1)$ which obtains:

- Approximation: $O\left(\frac{1}{\epsilon} \cdot d \log s\right)$
- Space & pre-processing time: $O(d\ ) \cdot n^{1+\epsilon}$
- Query time: $O(d) \cdot n^{\epsilon}$

EMD ANN: Prior Work + Main Result

Theorem (Indyk STOC'04): For any $\epsilon > 0$, there is an ANN data structure for s $\mathrm{EMD}_s(\mathbb{R}^d, \ell_1)$ which obtains:

- Approximation: $O\left(\frac{1}{\epsilon} \cdot d \log s\right)$
- Space & pre-processing time: $O(d\ ) \cdot n^{1+\epsilon}$
- Query time: $O(d) \cdot n^{\epsilon}$

Theorem (Andoni-Indyk-Krauthgamer SODA'08):

- There is a data structure with approximation $O\left(\frac{1}{\epsilon} \cdot \log^2 s\right)$

EMD ANN: Prior Work + Main Result

Theorem (Indyk STOC'04): For any $\epsilon > 0$, there is an ANN data structure for s $\mathrm{EMD_s}(\mathbb{R}^d, \ell_1)$ which obtains:

- Approximation: $O\left(\frac{1}{\epsilon} \cdot d \log s\right)$
- Space & pre-processing time: $O(d\ ) \cdot n^{1+\epsilon}$
- Query time: $O(d) \cdot n^{\epsilon}$

Theorem (Andoni-Indyk-Krauthgamer SODA'08):

- There is a data structure with approximation $O\left(\frac{1}{\epsilon} \cdot \log^2 s\right)$

Theorem (J-Waingarten-Zhang STOC'24):

- There is a data structure with approximation $\tilde{O}\left(\frac{1}{\epsilon} \cdot \log s\right) \cdot$

EMD is a complex metric

EMD = min-cost geometric bipartite matching

- “Hungarian Algorithm”: $O(n^3)$ time
  - Kuhn-Munkres, Edmonds-Karp 1950s, Jacobi 1850s.
- Fast min cost flow solvers: $O(n^{2+o(1)})$ time
  - [Chen, Kyng, Liu, Peng, Gutenberg, Sachdeva FOCS '22] :

Greedy is bad (even in the line) --- $n^{.58496}$ approx [Reingold-Tarjan ’81]

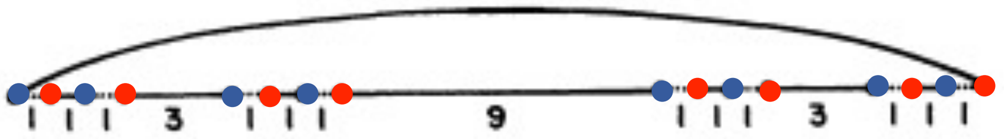
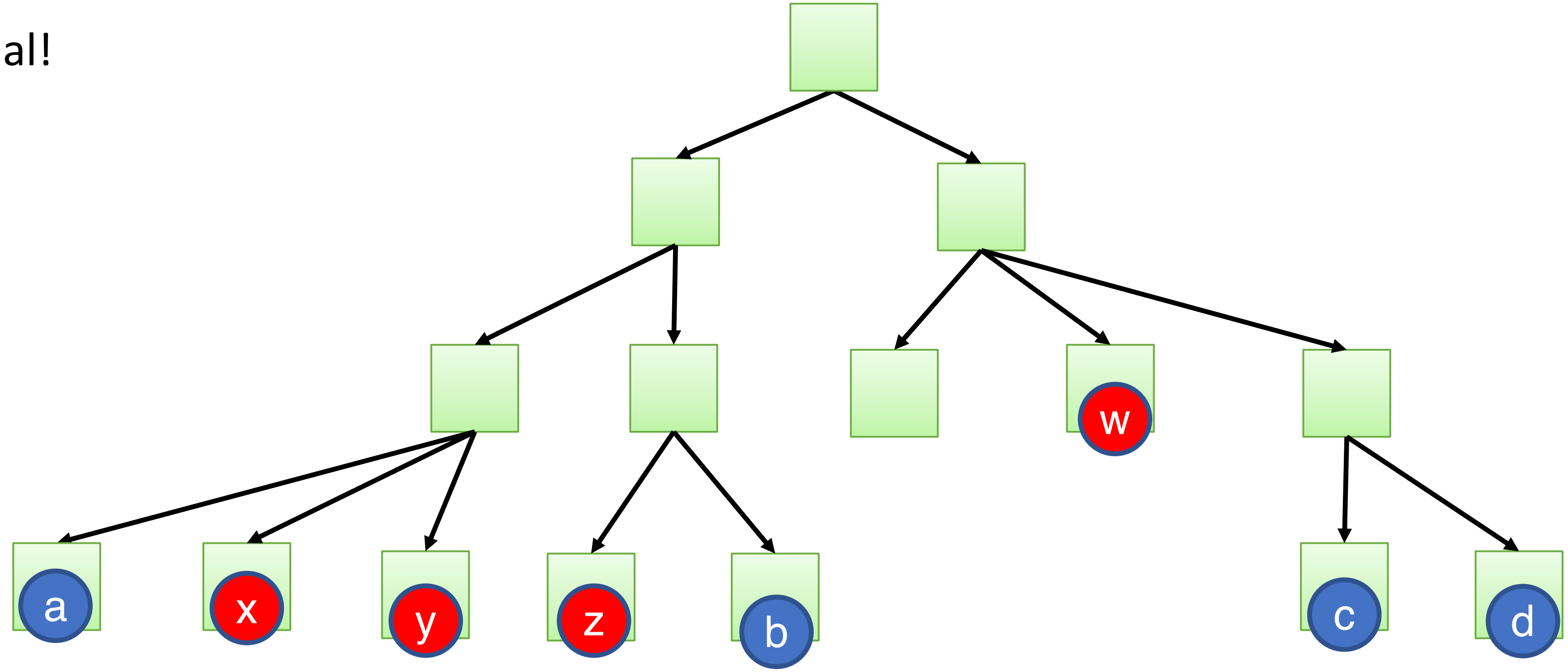


FIG. 1. *Examples in which the greedy heuristic produces matchings (shown in solid lines) costing* $\frac{4}{3}n^{\lg\frac{3}{2}}-1$ *times as much as the minimal matching (shown in dotted lines) for* $n = 2^t$. *Comparable examples are easy to construct for N even but not a power of 2.*

Probabilistic Tree Embeddings

EMD is not a simple metric.

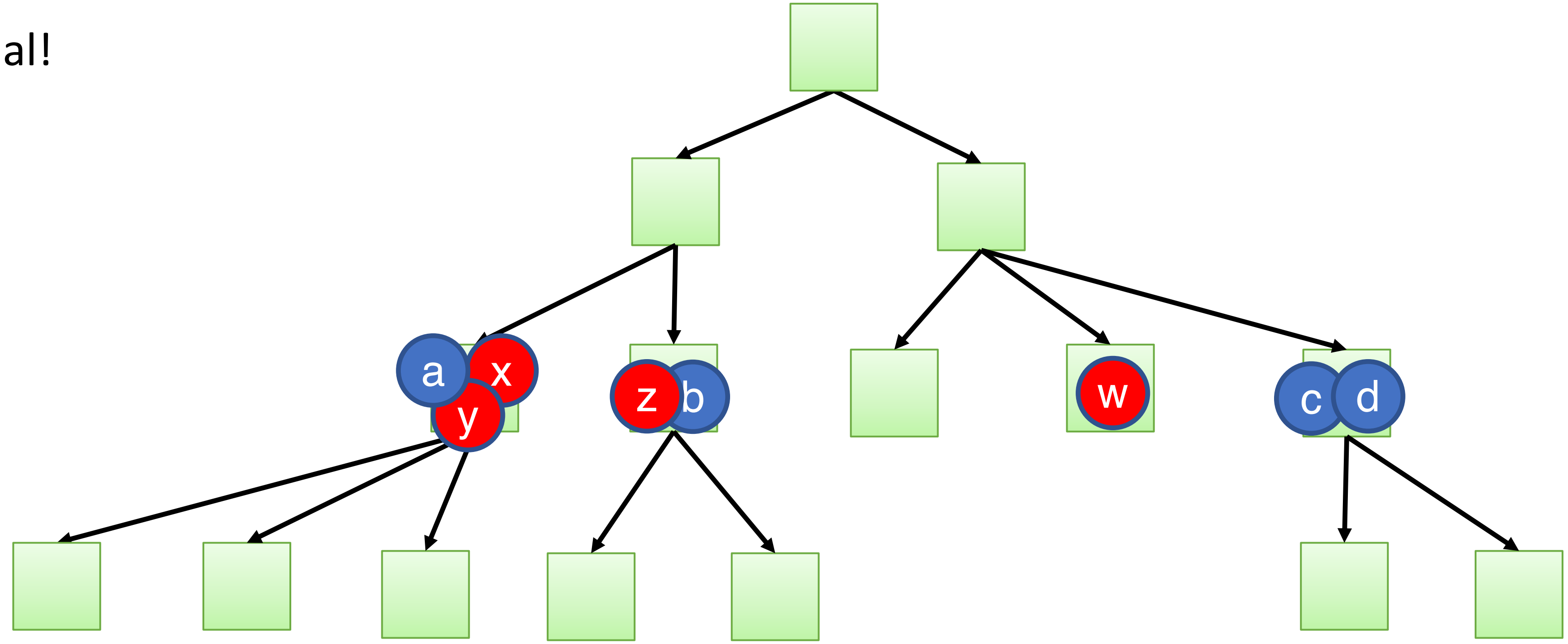
- Approach of all prior work: embed $\mathrm{EMD}_s(\mathbb{R}^d, \ell_1)$ into a simpler metric.
- EMD over tree metrics ($\mathrm{EMD}_s(T, d_T)$) is simpler!
    - Greedy is Optimal!



Probabilistic Tree Embeddings

EMD is not a simple metric.

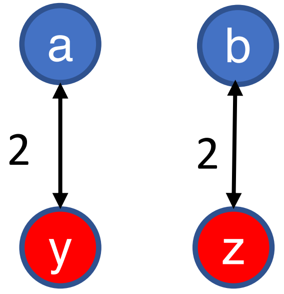
- Approach of all prior work: embed $\mathrm{EMD}_s(\mathbb{R}^d, \ell_1)$ into a simpler metric.
- EMD over tree metrics ($\mathrm{EMD}_s(T, d_T)$) is simpler!
  - Greedy is Optimal!

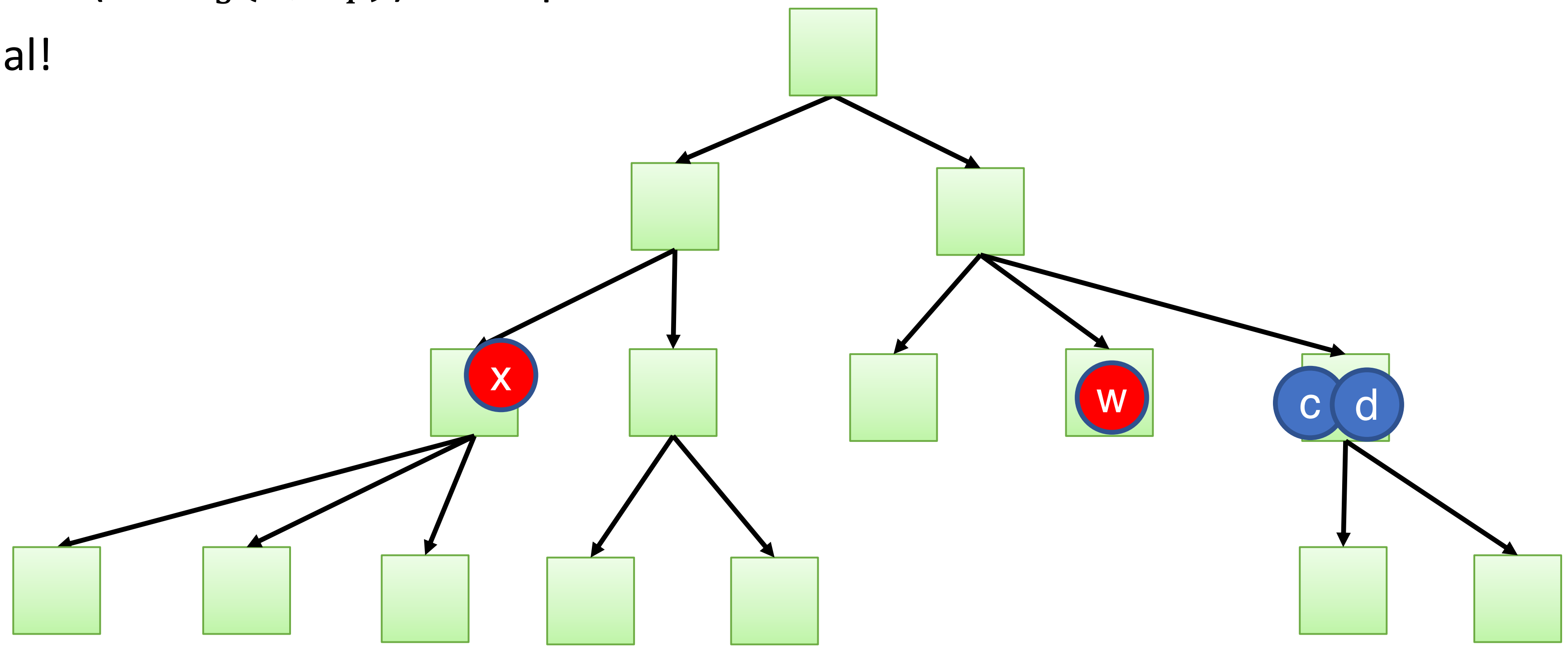


Probabilistic Tree Embeddings

EMD is not a simple metric.

- Approach of all prior work: embed $\mathrm{EMD}_s(\mathbb{R}^d, \ell_1)$ into a simpler metric.
- EMD over tree metrics ($\mathrm{EMD}_s(T, d_T)$) is simpler!
  - Greedy is Optimal!

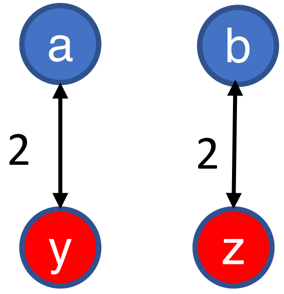


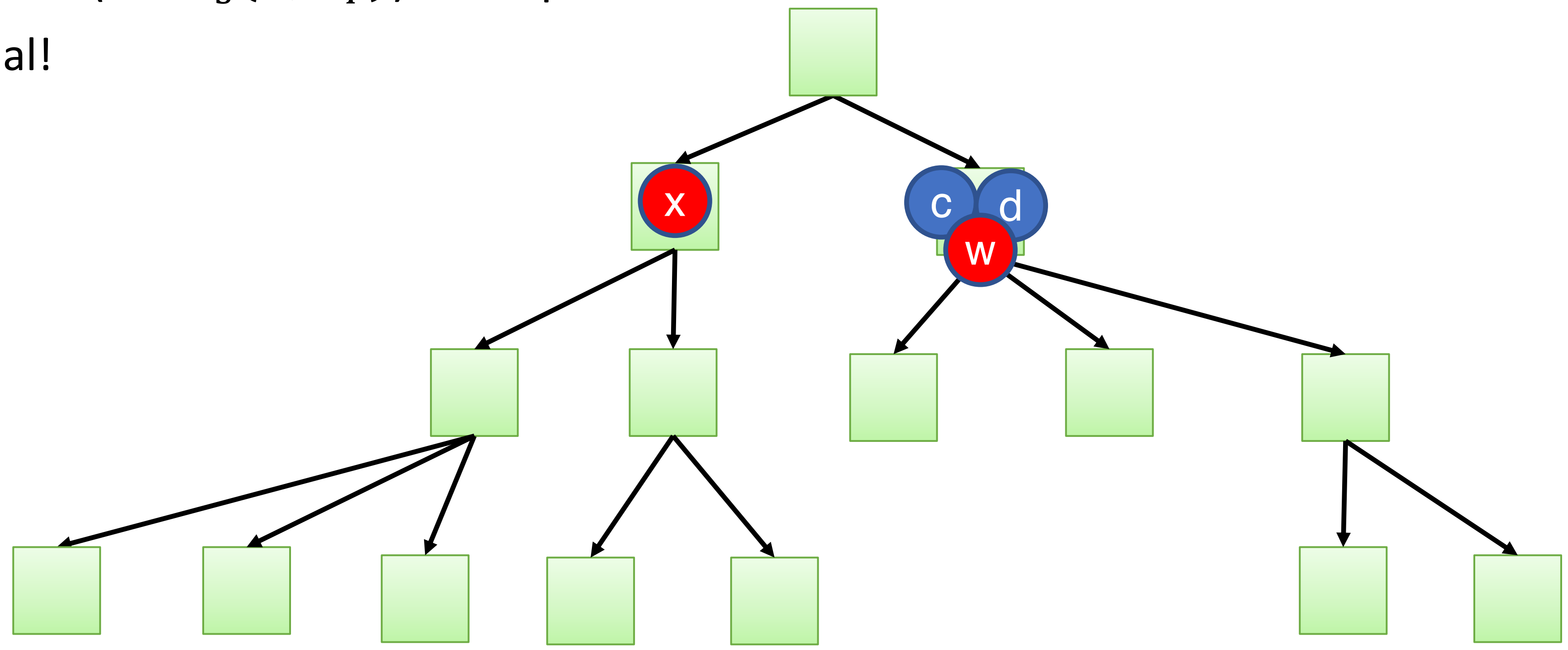


Probabilistic Tree Embeddings

EMD is not a simple metric.

- Approach of all prior work: embed $\mathrm{EMD}_s(\mathbb{R}^d, \ell_1)$ into a simpler metric.
- EMD over tree metrics ($\mathrm{EMD}_s(T, d_T)$) is simpler!
    - Greedy is Optimal!

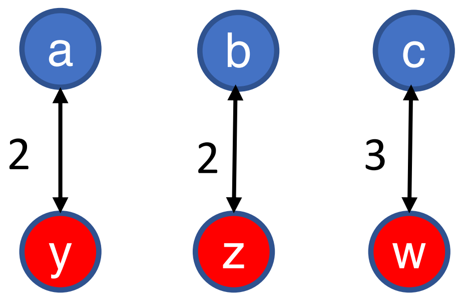


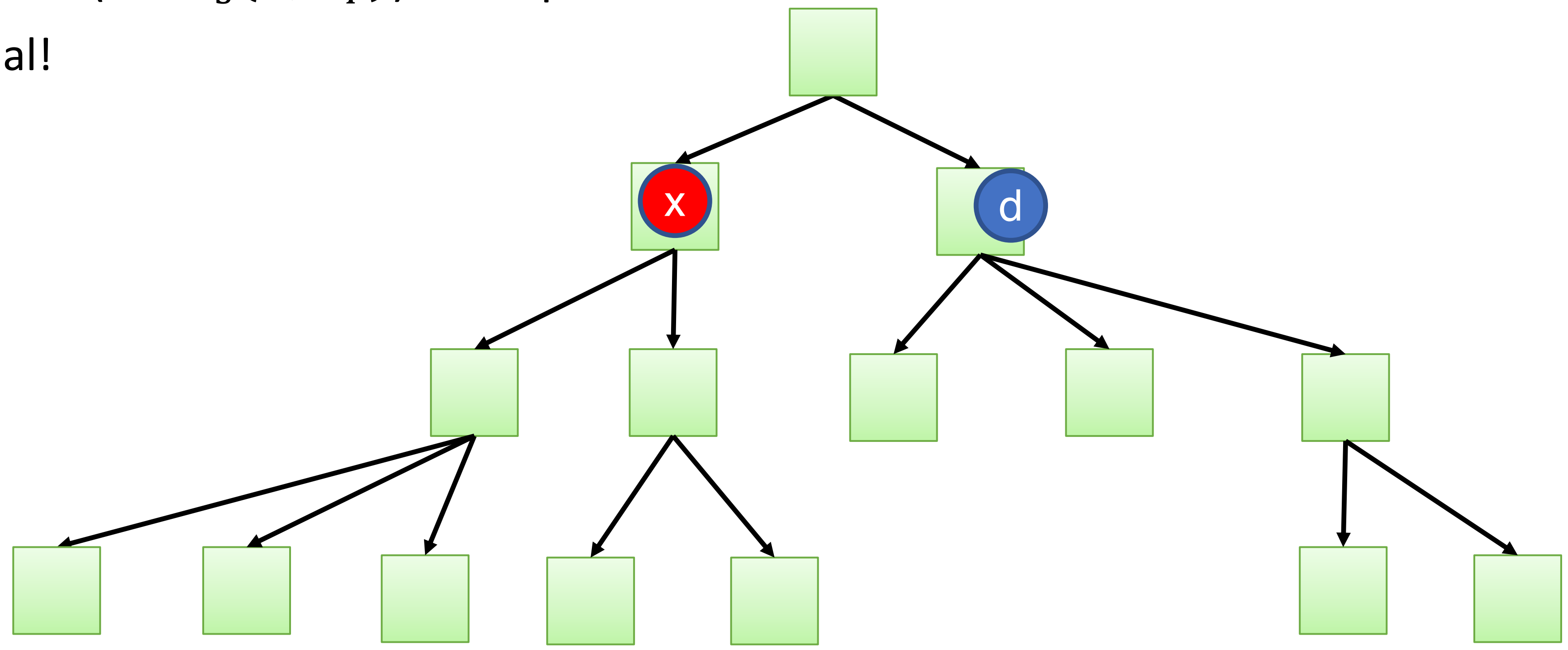


Probabilistic Tree Embeddings

EMD is not a simple metric.

- Approach of all prior work: embed $\mathrm{EMD}_s(\mathbb{R}^d, \ell_1)$ into a simpler metric.
- EMD over tree metrics ($\mathrm{EMD}_s(T, d_T)$) is simpler!
    - Greedy is Optimal!

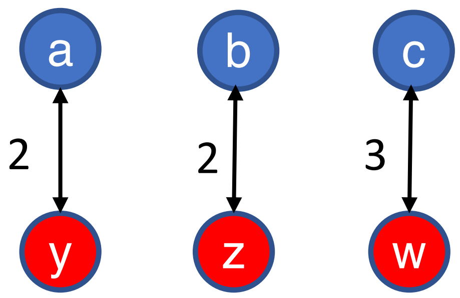


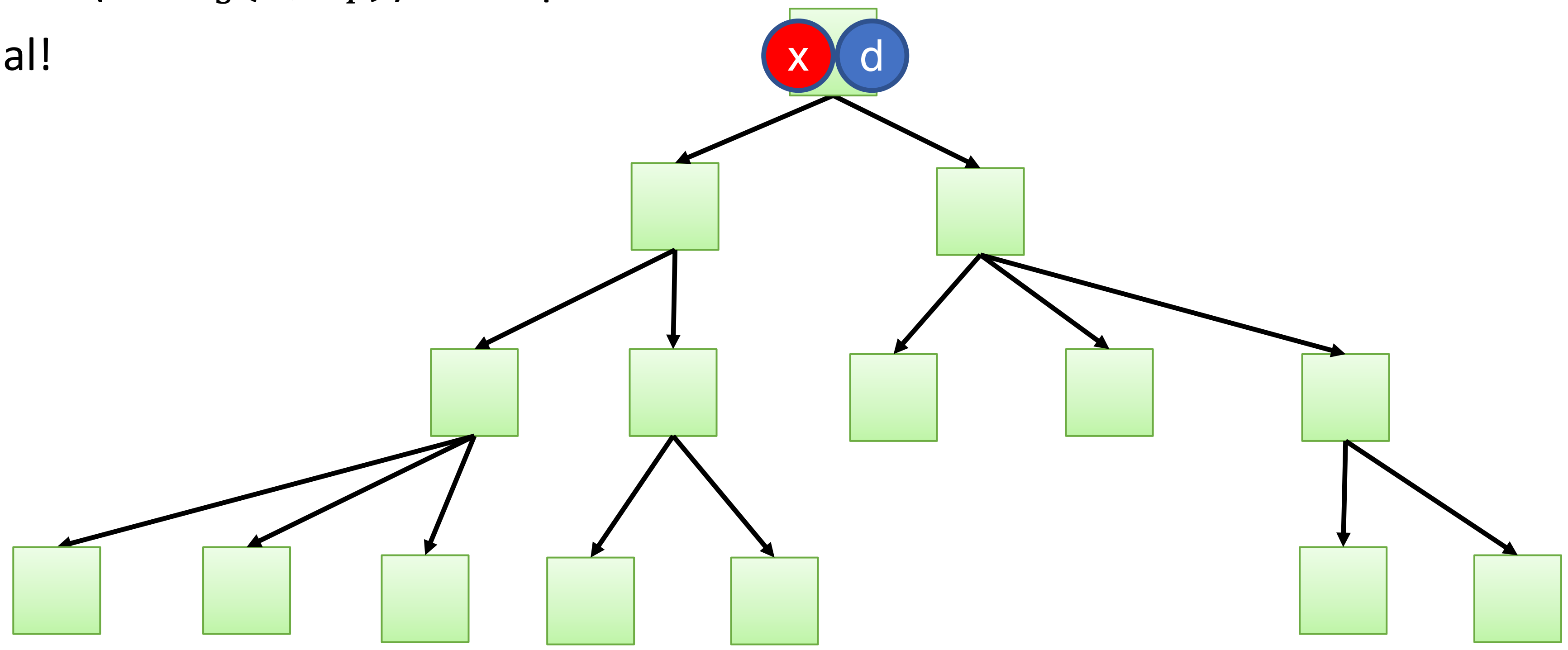


Probabilistic Tree Embeddings

EMD is not a simple metric.

- Approach of all prior work: embed $\mathrm{EMD}_s(\mathbb{R}^d, \ell_1)$ into a simpler metric.
- EMD over tree metrics ($\mathrm{EMD}_s(T, d_T)$) is simpler!
    - Greedy is Optimal!

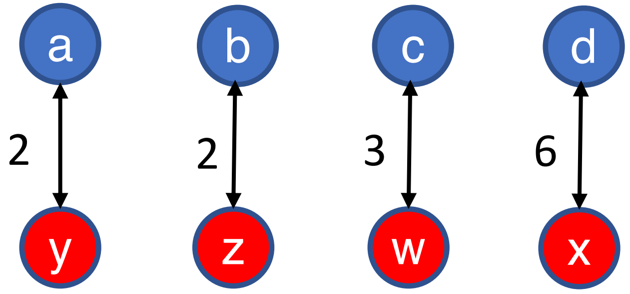


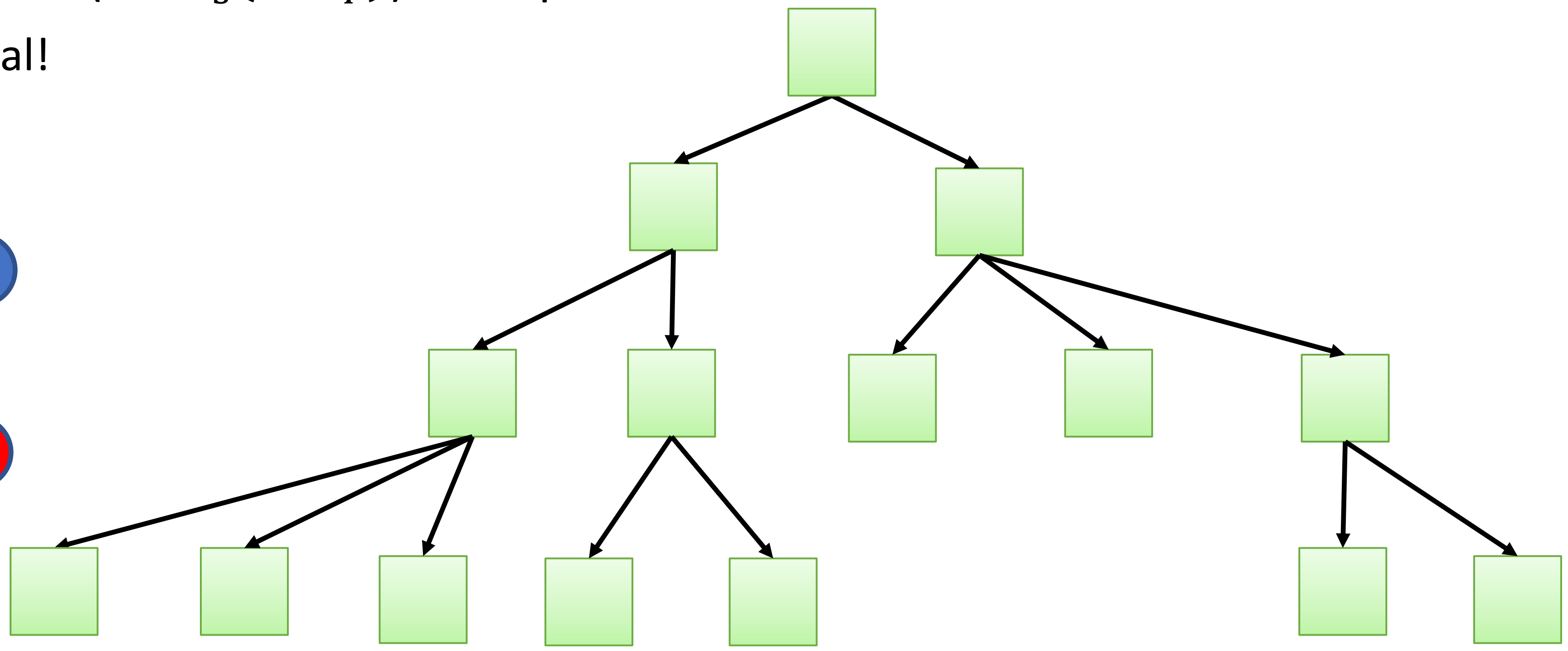


Probabilistic Tree Embeddings

EMD is not a simple metric.

- Approach of all prior work: embed $\mathrm{EMD}_s(\mathbb{R}^d, \ell_1)$ into a simpler metric.
- EMD over tree metrics ($\mathrm{EMD}_s(T, d_T)$) is simpler!
    - Greedy is Optimal!





Probabilistic Tree Embeddings

EMD is not a simple metric.

- Approach of all prior work: embed $\mathrm{EMD}_s(\mathbb{R}^d, \ell_1)$ into a simpler metric.
- EMD over tree metrics ($\mathrm{EMD}_s(T, d_T)$) is simpler!
  - Greedy is Optimal!

Isometric embedding into ℓ_1.

$$f : \mathrm{EMD}_s(T, d_T) \to (\mathbb{R}^k, \ell_1)$$

such that

$$\mathrm{EMD}(x, y) = \|f(x) - f(y)\|_1$$

Goal: Embed into a tree!

The Plan

Step 1 - Probabilistic Tree: $(\mathbb{R}^d, \ell_1) \longmapsto (\mathbf{T}, d_{\mathbf{T}})$

Step 2 - Reduce to ℓ_1: $\mathrm{EMD}_s(\mathbb{R}^d, \ell_1) \overset{?}{\longmapsto} \mathrm{EMD}_s(\mathbf{T}, d_{\mathbf{T}}) \overset{\text{(isometric)}}{\longmapsto} \ell_1$

Small note:

 * Computationally efficient (and succinct).
 * Once in ℓ_1, can use ANN for ℓ_1

Probabilistic Tree Embeddings

Theorem (Indyk '04): There is an embedding $(\mathbb{R}^d, \ell_1) \to (\mathbf{T}, d_{\mathbf{T}})$ satisfying

- Non-contraction: $\|a - b\|_1 \leq d_{\mathbf{T}}(a, b)$
- Bounded expansion: $\mathop{\mathbb{E}}_{\mathbf{T}}[d_{\mathbf{T}}(a, b)] \leq O(d \log s) \cdot \|a - b\|_1$

Probabilistic Tree Embeddings

Theorem (Indyk '04): There is an embedding $(\mathbb{R}^d, \ell_1) \to (\mathbf{T}, d_{\mathbf{T}})$ satisfying

- Non-contraction: $\|a - b\|_1 \leq d_{\mathbf{T}}(a, b)$
- Bounded expansion: $\underset{\mathbf{T}}{\mathbb{E}}[d_{\mathbf{T}}(a, b)] \leq O(d \log s) \cdot \|a - b\|_1$

Theorem (AIK '08): There is an embedding $(\mathbb{R}^d, \ell_1) \to (\mathbf{T}, d_{\mathbf{T}})$ satisfying for any subset $\Omega \subset \mathbb{R}^d$ of $2s$ points:

- Non-contraction: $\|a - b\|_1 \leq d_{\mathbf{T}}(a, b)$ w.h.p $\forall a, b \in \Omega$
- Bounded expansion: $\mathbb{E}_{\mathbf{T}}[d_{\mathbf{T}}(a, b)] \leq O(\log^2 s) \cdot \|a - b\|_1$

Probabilistic Tree Embeddings

Theorem (Indyk '04): There is an embedding $(\mathbb{R}^d, \ell_1) \to (\mathbf{T}, d_{\mathbf{T}})$ satisfying

- Non-contraction: $\|a-b\|_1 \le d_{\mathbf{T}}(a,b)$
- Bounded expansion: $\mathbb{E}_{\mathbf{T}}[d_{\mathbf{T}}(a,b)] \le O(d \log s) \cdot \|a-b\|_1$

Theorem (AIK '08): There is an embedding $(\mathbb{R}^d, \ell_1) \to (\mathbf{T}, d_{\mathbf{T}})$ satisfying for any subset $\Omega \subset \mathbb{R}^d$ of $2s$ points:

- Non-contraction: $\|a-b\|_1 \le d_{\mathbf{T}}(a,b)$ w.h.p $\forall a, b \in \Omega$
- Bounded expansion: $\mathbb{E}_{\mathbf{T}}[d_{\mathbf{T}}(a,b)] \le O(\log^2 s) \cdot \|a-b\|_1$

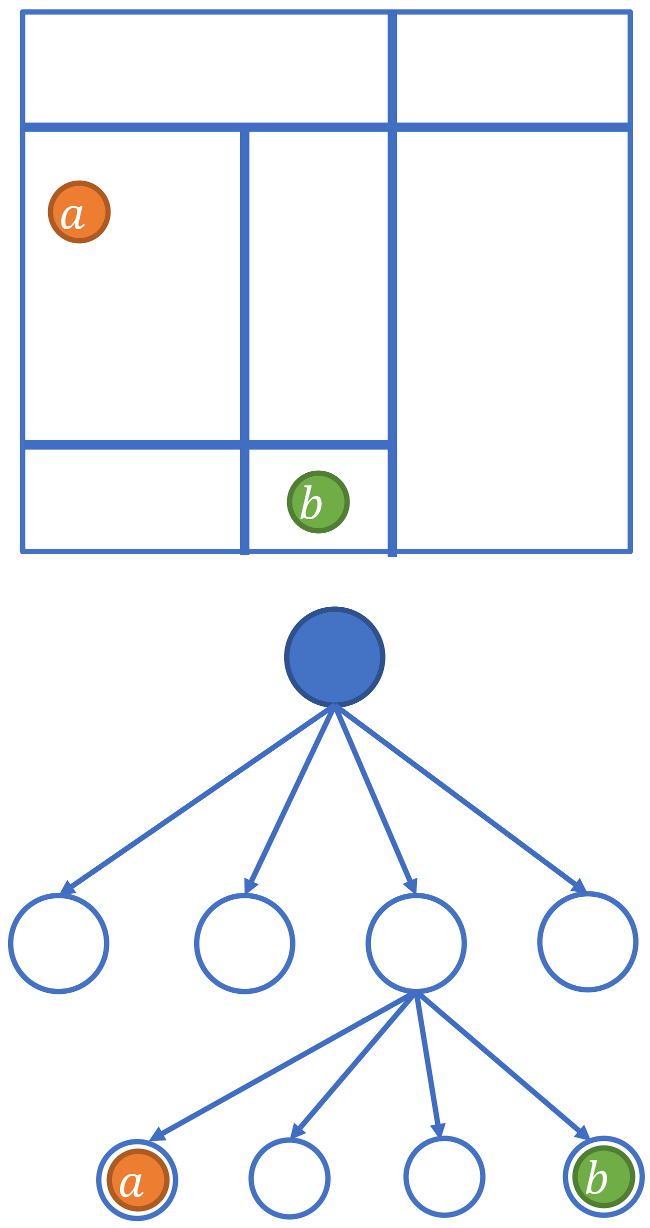
Theorem (CJLW '22): For any subset $\Omega \subset \mathbb{R}^d$, there is a (succinct and efficient) embedding $(\Omega, \ell_1) \longmapsto (\mathbf{T}, d_{\mathbf{T}})$ satisfying:

- Non-contraction: $\|a-b\|_1 \le d_{\mathbf{T}}(a,b)$ w.h.p $\forall a, b \in \Omega$
- Bounded expansion: $\mathbb{E}_{\mathbf{T}}[d_{\mathbf{T}}(a,b)] \le \tilde{O}(\log |\Omega|) \cdot \|a-b\|_1$ for all $a, b \in \Omega$

Quadtree Algorithm

Embedding $\mathbb{R}^d$ into a tree

1. Recursively subdivide $\mathbb{R}^d$, creating tree
2. Vertices of tree correspond to hypercubes in R^d
3. Map point p to the leaf containing it

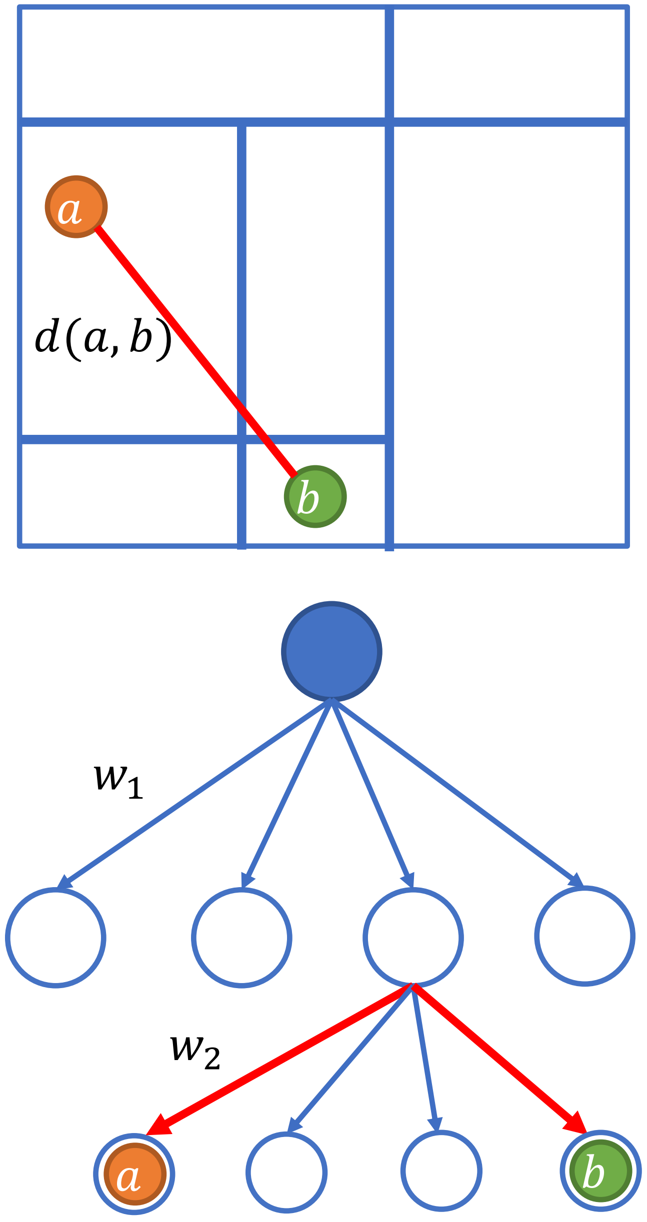


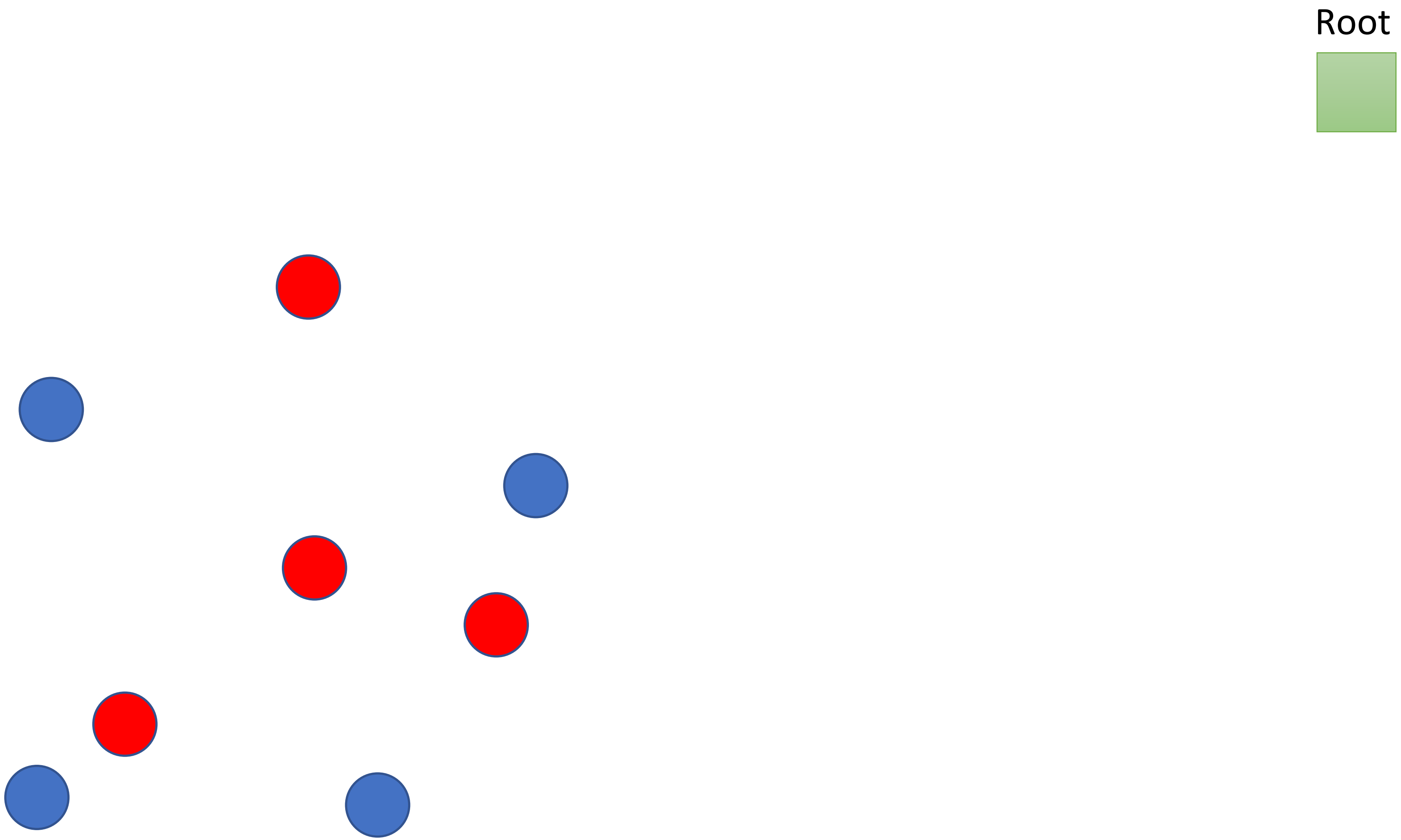
Quadtree Algorithm

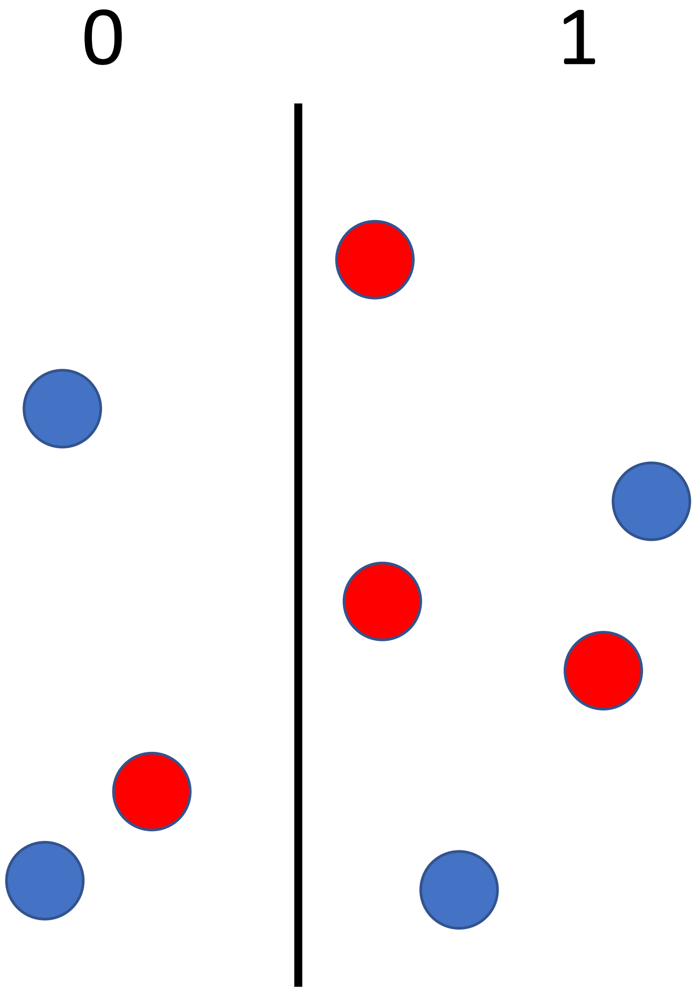
Embedding $\mathbb{R}^d$ into a tree

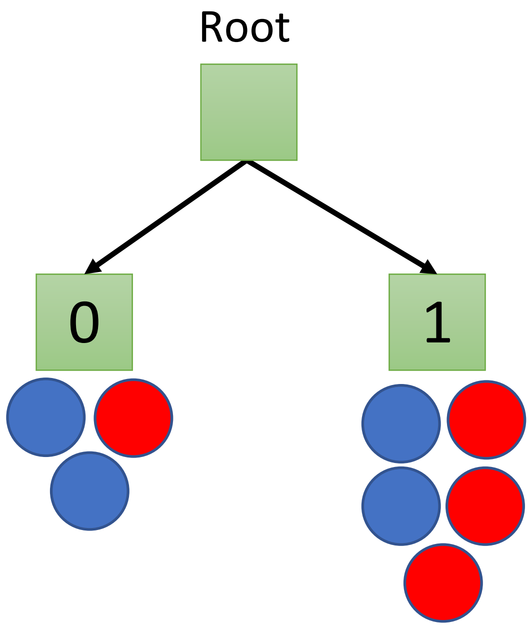
1. Recursively subdivide $\mathbb{R}^d$, creating tree
2. Vertices of tree correspond to hypercubes in R^d
3. Map point p to the leaf containing it

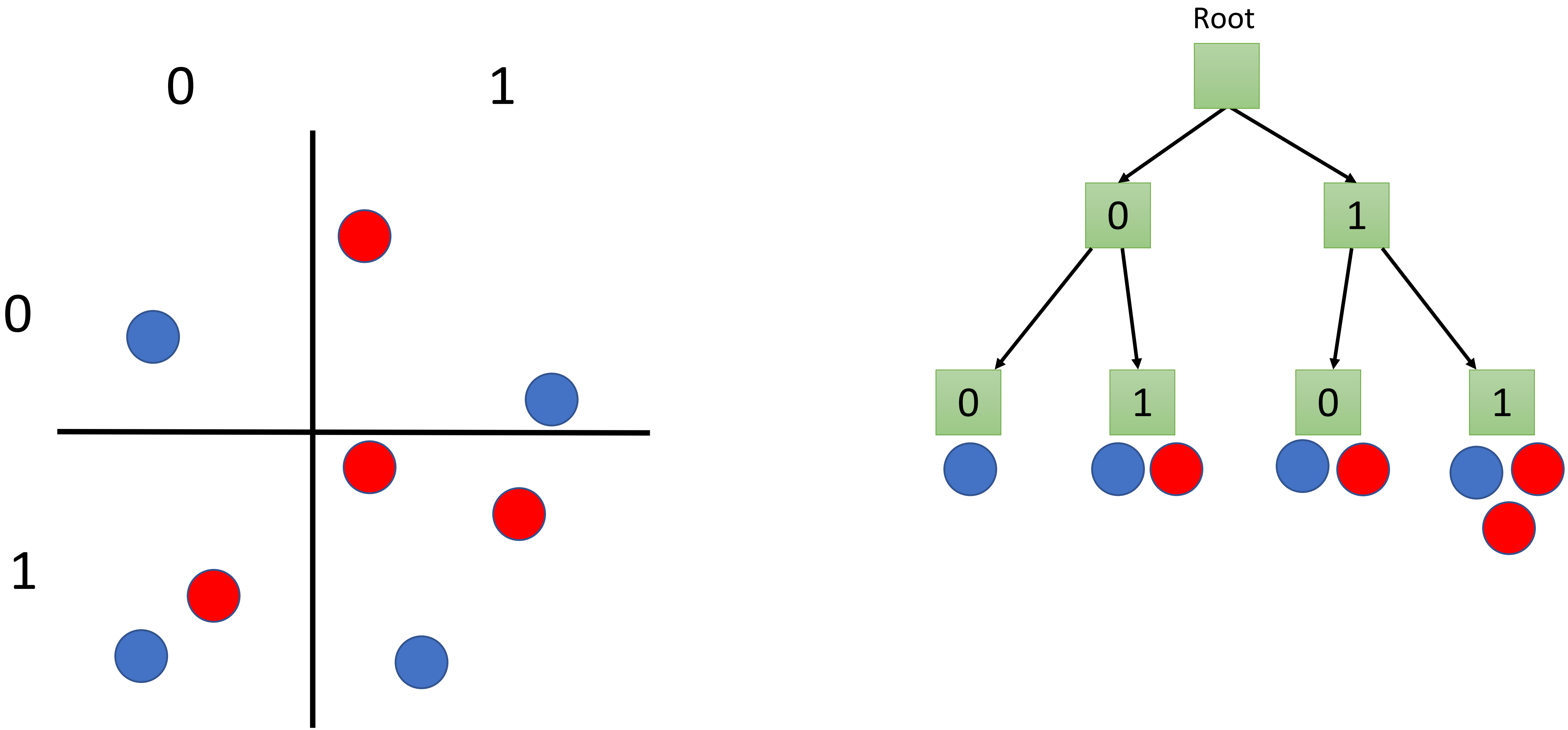
➢Set edge weights so tree distances approximate original.

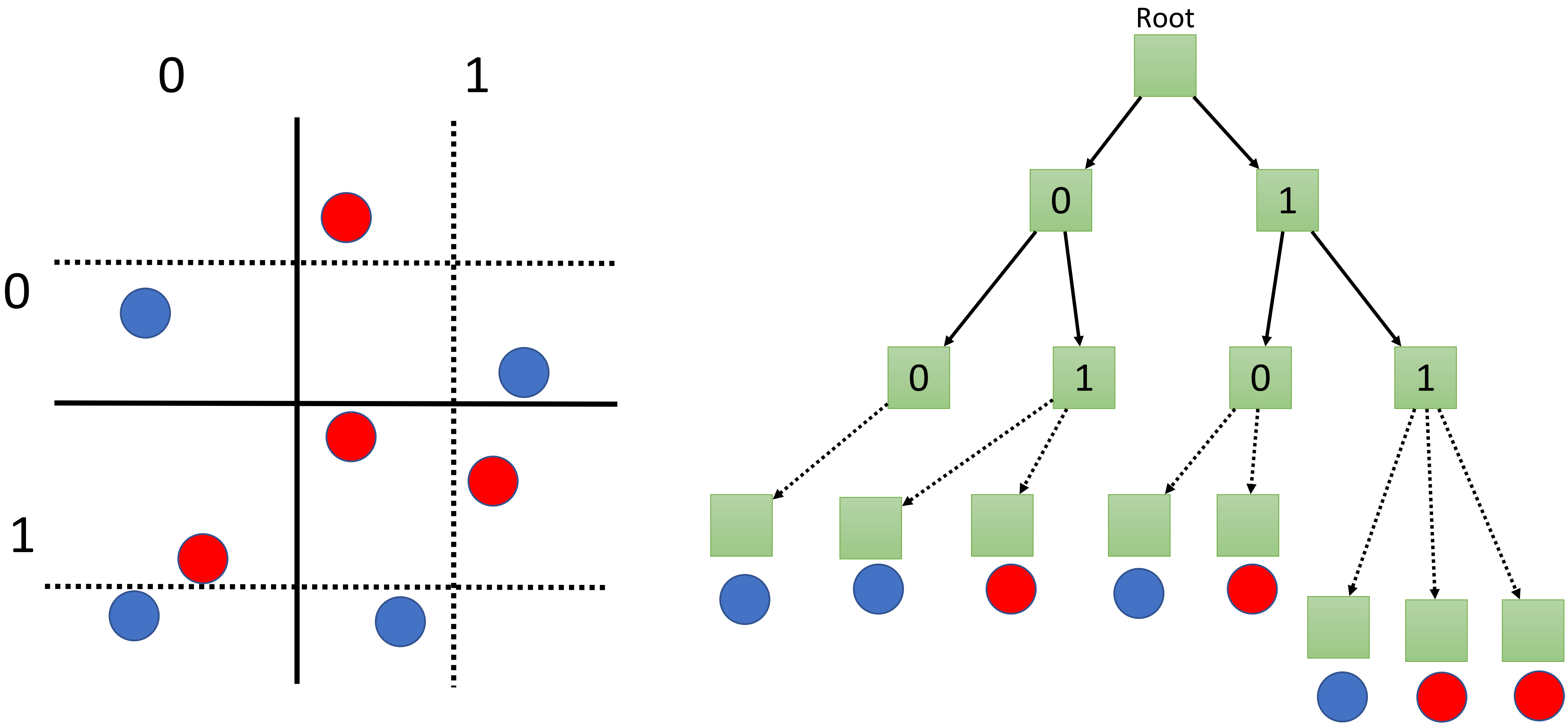



Root

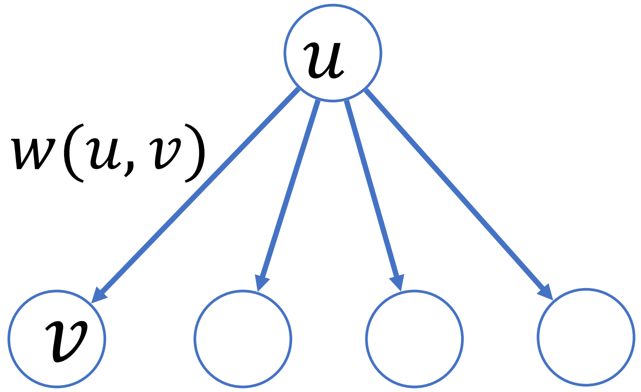

0
1


Root
0
1


0
1
0
1
Root
0
1
0
1
0
1

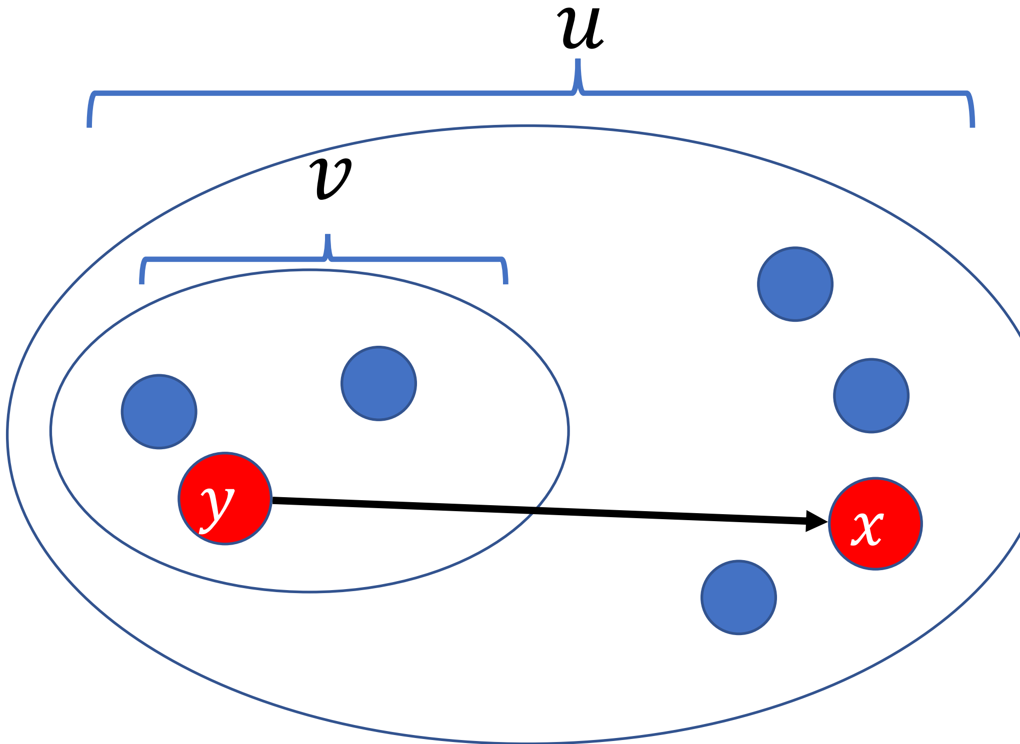

0
1
0
1
Root
0
1
0
1
0
1

The only change: edge weights

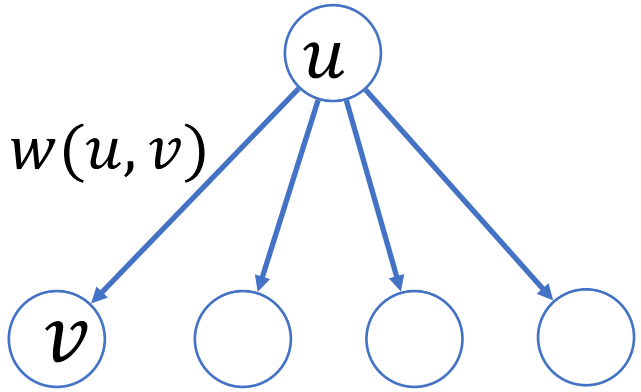


- Impose randomly shifted grid at $\log d\Delta$-scales
  - At depth $i \geq 0$, hyper-grid has side length $\Delta/2^i$
- T ≔ recursion tree
- For $\Omega \subset \mathbb{R}^d$, let $\Omega_v \subset \Omega$ be subset going through $v \in T$

	I'04:	**AIK '08:**	**CJLW '22:**
$w(u,v)$:	Diameter of grid cell	$\dfrac{\Delta \log s}{2^i}$	$\mathbb{E}_{\substack{x \sim \Omega_u \\ y \sim \Omega_v}}[\|x-y\|_1]$



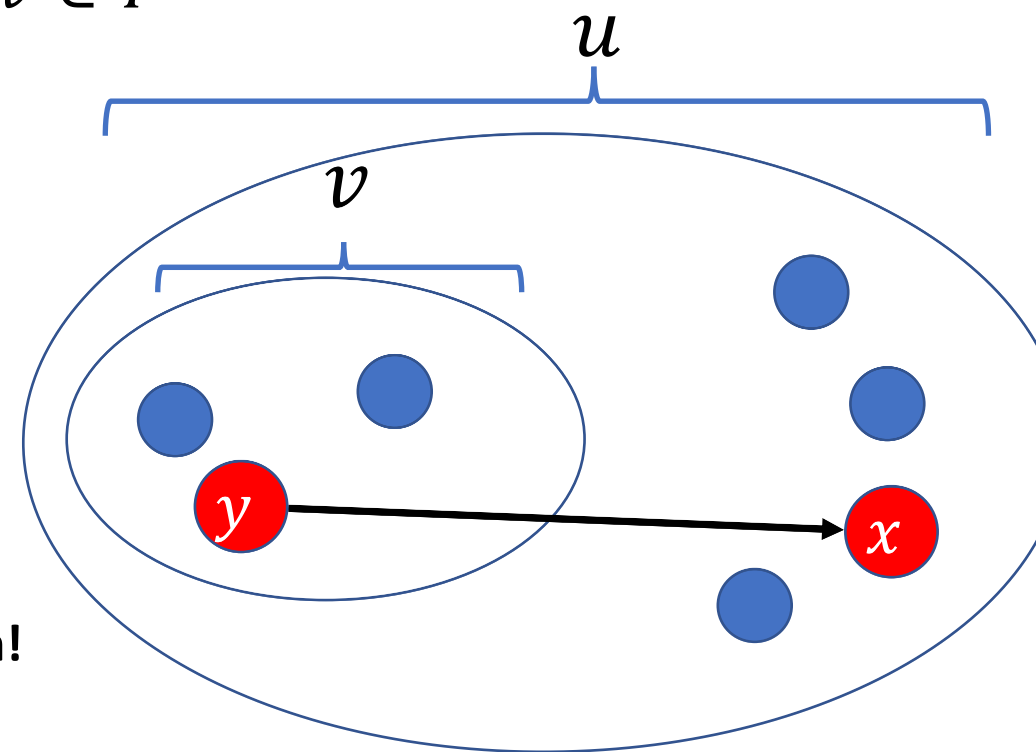
The only change: edge weights



- Impose randomly shifted grid at $\log d\Delta$-scales
  - At depth $i \geq 0$, hyper-grid has side length $\Delta/2^i$
- $T :=$ recursion tree
- For $\Omega \subset \mathbb{R}^d$, let $\Omega_v \subset \Omega$ be subset going through $v \in T$

	I'04:	**AIK '08:**	**CJLW '22:**
$w(u,v)$:	Diameter of grid cell	$\frac{\Delta \log s}{2^i}$	$\mathbb{E}_{\substack{x \sim \Omega_u \\ y \sim \Omega_v}}[\lVert x - y \rVert_1]$
Approximation	$O(d \log s)$	$O(\log^2 s)$	$\tilde{O}(\log \lvert\Omega\rvert)$

Data-dependent weights give improved approximation!



Using CJLW for ANN?

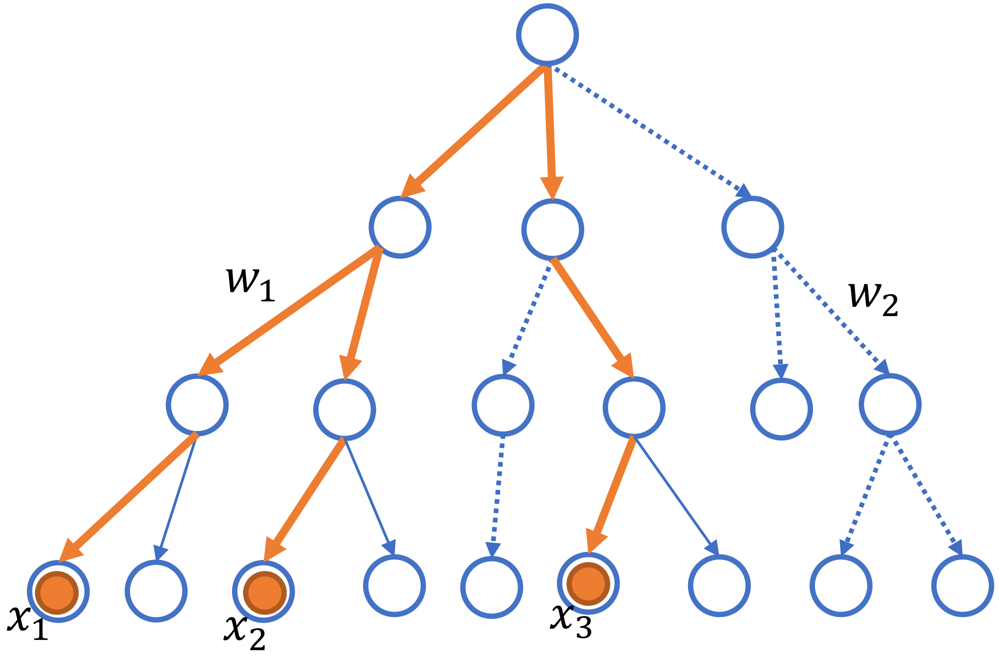
	I'04:	**AIK '08:**	**CJLW '22:**
$w(u,v)$:	Diameter of parts	$\frac{\Delta \log s}{2^i}$	$\mathbb{E}_{\substack{x \sim \Omega_u \\ y \sim \Omega_v}}[\|x-y\|_1]$

- CLJW'22 Pros:
  - Better distortion when $|\Omega| = O(\text{poly } s)$
  - Still concise and efficient
- Cons:
  - Only defined on Ω (what about query)?
  - Cannot define $\Omega :=$ all n sets of size s
    - Otherwise $\log ns$ distortion!

SampleTree

Cannot afford to use DD-Quadtree on all points in dataset for ANN.

- Instead, use random sample $\Omega \subset D$ of size $\mathrm{poly}(s)$.



SampleTree

Cannot afford to use DD-Quadtree on all points in dataset for ANN.

- Instead, use random sample $\Omega \subset D$ of size $\mathrm{poly}(s)$.

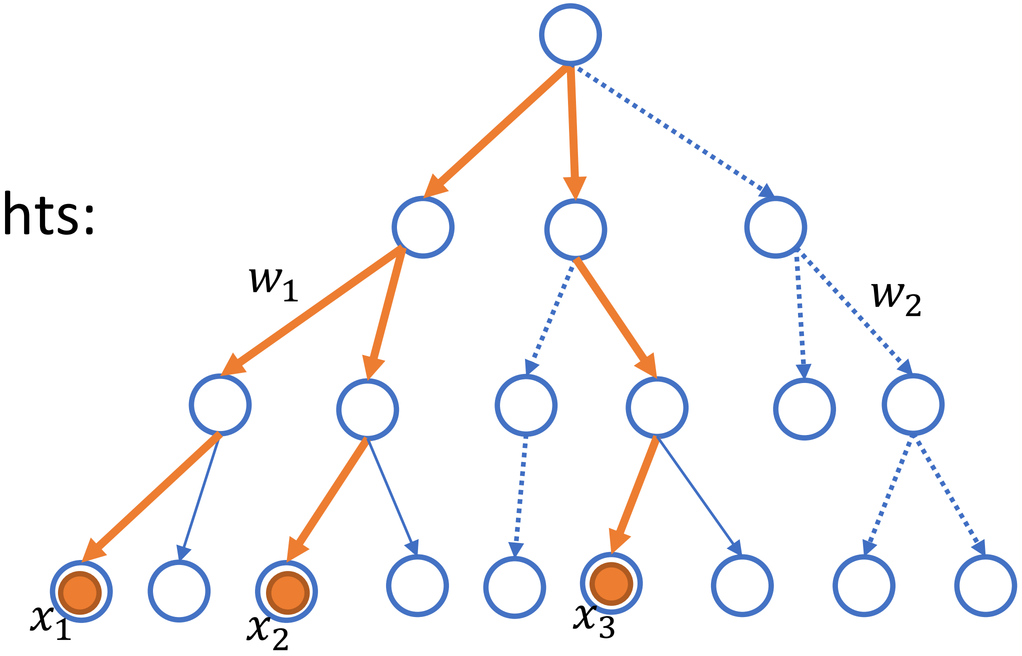
SampleTree:

1. Sample Quadtree T partition
2. $\Omega :=$ random $\mathrm{poly}(s)$ points from D
3. Define hybrid weights:

* If any $x \in \Omega$ goes through (u, v) use CJLW'22 weights:

$$w(u, v) = \mathbb{E}_{\substack{x \sim \Omega_u \\ y \sim \Omega_v}} [\|x - y\|_1]$$

* Otherwise, use AIK'08 weights: $\mathrm{w(u, v)} = \frac{\Delta \log s}{2^i}$



SampleTree

Cannot afford to use DD-Quadtree on all points in dataset for ANN.

- Instead, use random sample $\Omega \subset D$ of size $\mathrm{poly}(s)$.

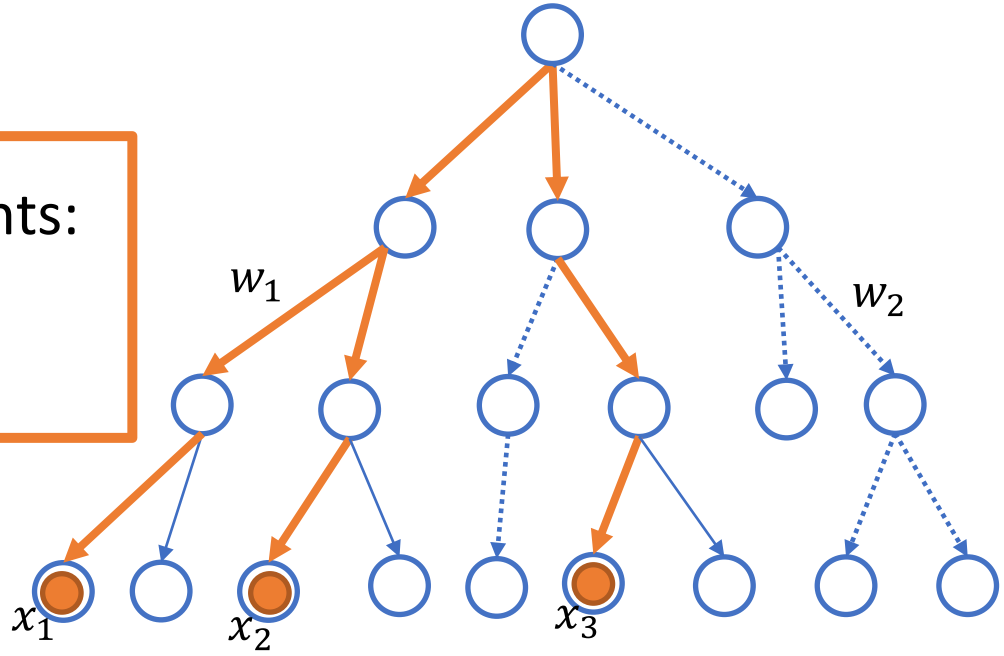
SampleTree:

1. Sample Quadtree T partition
2. $\Omega :=$ random $\mathrm{poly}(s)$ points from D
3. Define hybrid weights:

* If any $x \in \Omega$ goes through (u, v) use CJLW'22 weights:

$$w(u,v) = \mathbb{E}_{\substack{x \sim \Omega_u \\ y \sim \Omega_v}} [\|x - y\|_1]$$

* Otherwise, use AIK'08 weights: $\mathrm{w(u,v)} = \frac{\Delta \log s}{2^i}$



SampleTree

SampleTree now valid mapping: $\mathbb{R}^d \to T_\Omega$

Cannot afford to use DD-Quadtree on all points in dataset for ANN.

- Instead, use random sample $\Omega \subset D$ of size $\mathrm{poly}(s)$.

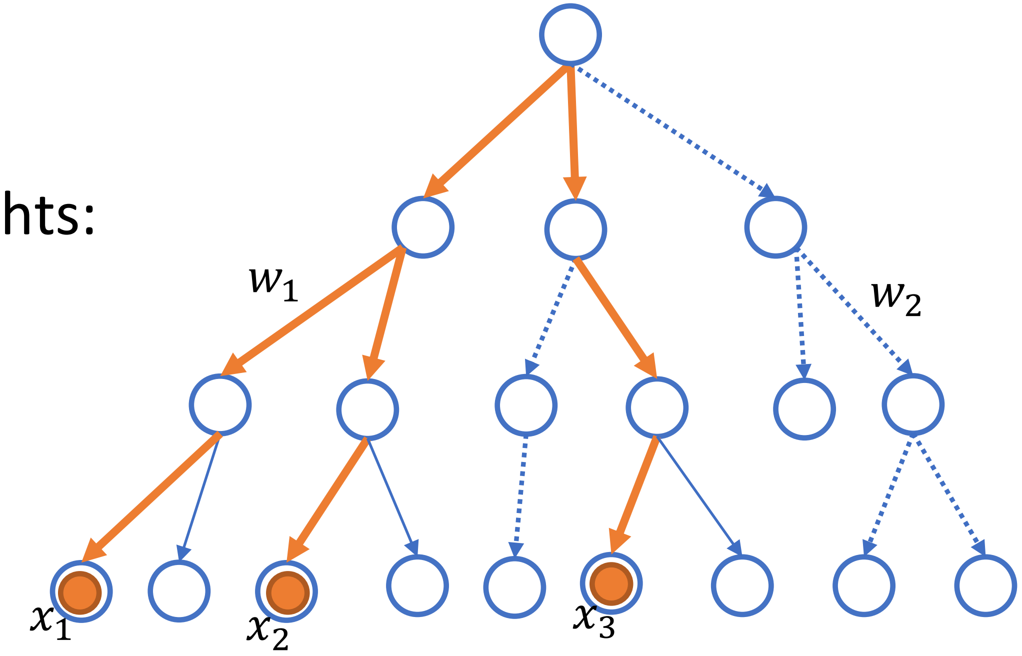
SampleTree:

1. Sample Quadtree T partition
2. $\Omega :=$ random $\mathrm{poly}(s)$ points from D
3. Define hybrid weights:

* If any $x \in \Omega$ goes through (u, v) use CJLW'22 weights:

$$w(u,v) = \mathbb{E}_{\substack{x\sim\Omega_u \\ y \sim \Omega_v}}[\|x-y\|_1]$$

* Otherwise, use AIK'08 weights: $\mathrm{w(u,v)} = \frac{\Delta \log s}{2^i}$



Distortion Bounds

$(T_\Omega, w_\Omega) \coloneqq \mathrm{SampleTree}.$

- If $x, y \in \Omega$, then $\|x-y\|_1 \leq d_{T_\Omega}(x,y) \leq \tilde{O}(\log s) \cdot \|x-y\|_1$
  - $d_{T_\Omega}(x,y)$ only uses data-dependent edge weights!

Distortion Bounds

$(T_\Omega, w_\Omega) \coloneqq \mathrm{SampleTree}.$

- If $x, y \in \Omega$, then $\|x - y\|_1 \leq d_{T_\Omega}(x, y) \leq \tilde{O}(\log s) \cdot \|x - y\|_1$
  - $d_{T_\Omega}(x, y)$ only uses data-dependent edge weights!

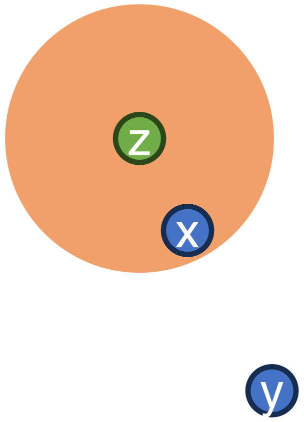
What about $\mathrm{x, y} \notin \Omega$?

- If x, y far from Ω, then $d_{T_\Omega}(x, y)$ only uses data-independent weights
  - $O(\log^2 s)$ approximation
- Since Ω was randomly sampled, most points are “close” to Ω!

Distortion Bounds

$(T_\Omega, w_\Omega) \coloneqq \text{SampleTree}.$

- If $x, y \in \Omega$, then $\|x-y\|_1 \leq d_{T_\Omega}(x,y) \leq \tilde{O}(\log s) \cdot \|x-y\|_1$
  - $d_{T_\Omega}(x,y)$ only uses data-dependent edge weights!



What about $\mathrm{x}, \mathrm{y} \notin \Omega$?

- If x, y far from Ω, then $d_{T_\Omega}(x,y)$ only uses data-independent weights
  - $O(\log^2 s)$ approximation
- Since Ω was randomly sampled, most points are “close” to Ω!

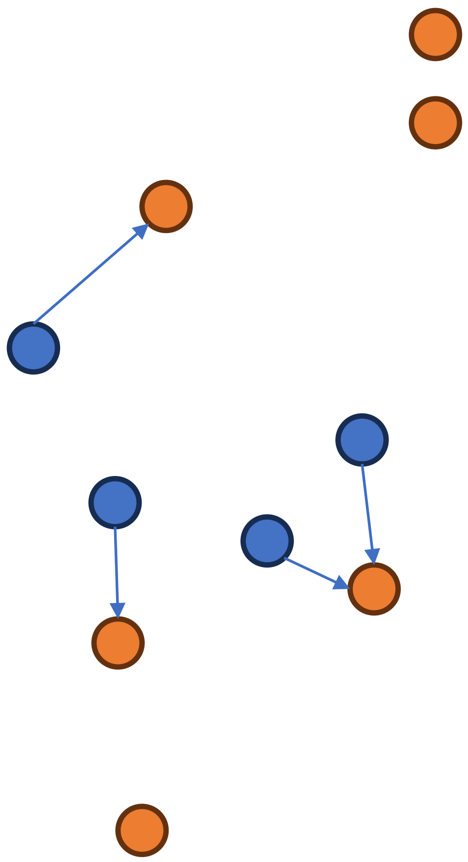
Hope: if $x \notin \Omega$ but is close to some $z \in \Omega$, can *extend* the DD-guarantees to x!

Extensions to Chamfer Neighborhood

Want to prove:

"If $A, B \in EMD_s(\mathbb{R}^d, \ell_1)$ are close to Ω, then $EMD_{T_\Omega}(A, B)$ is a $\tilde{O}(\log s)$ approx. of $EMD_{\mathbb{R}^d}(A, B)$"

What does it mean for $A \subset \mathbb{R}^d$ to be close to $\Omega \subset \mathbb{R}^d$?



Extensions to Chamfer Neighborhood

Want to prove:

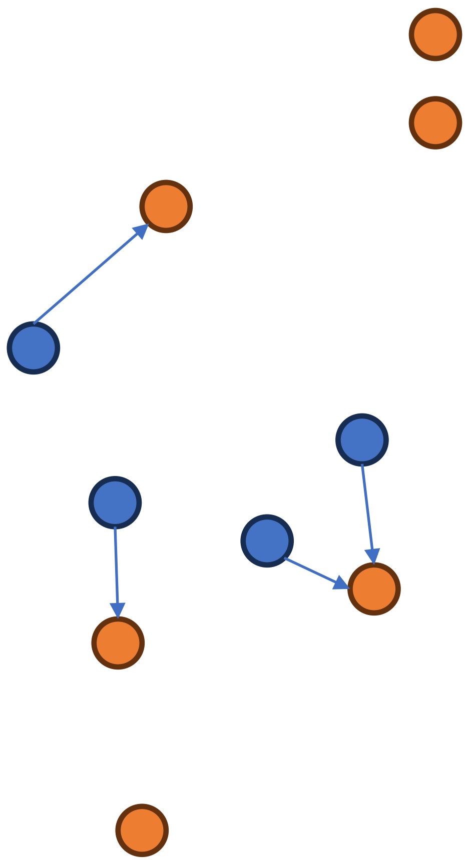
"If $A, B \in EMD_s(\mathbb{R}^d, \ell_1)$ are close to Ω, then $EMD_{T_\Omega}(A,B)$ is a $\tilde{O}(\log s)$ approx. of $EMD_{\mathbb{R}^d}(A,B)$"

What does it mean for $A \subset \mathbb{R}^d$ to be close to $\Omega \subset \mathbb{R}^d$?

Chamfer Distance:

$$\text{Chamfer}(A,\Omega) = \sum_{a \in A} \min_{x \in \Omega} \|a - x\|_1$$

"Cost of moving each point in A to nearest point in Ω"



Extensions to Chamfer Neighborhood

Chamfer Extension Lemma: Let $A, B \in EMD_s(\mathbb{R}^d, \ell_1)$ then:

$$\mathbb{E}_{T_\Omega}\left[EMD_{T_\Omega}(A,B)\right] \leq \tilde{O}(\log s) \cdot \mathrm{EMD}_{\mathbb{R}^d}(A,B) \cdot \log\left(\frac{\mathrm{Chamfer}_{\mathbb{R}^d}(A,\Omega)}{\mathrm{EMD}_{\mathbb{R}^d}(A,B)}\right)$$

"Extra" approximation factor is log-ratio:

- (How far is A from Ω) / (How far is A from B)

Extensions to Chamfer Neighborhood

Chamfer Extension Lemma: Let $A, B \in EMD_s(\mathbb{R}^d, \ell_1)$ then:

$$\mathbb{E}_{T_\Omega}\left[EMD_{T_\Omega}(A,B)\right] \leq \tilde{O}(\log s) \cdot \mathrm{EMD}_{\mathbb{R}^d}(A,B) \cdot \log\left(\frac{\mathrm{Chamfer}_{\mathbb{R}^d}(A,\Omega)}{\mathrm{EMD}_{\mathbb{R}^d}(A,B)}\right)$$

"Extra" approximation factor is log-ratio:

- (How far is A from Ω) / (How far is A from B)

If ratio = $O(\mathrm{poly}\log s)$, get a $\tilde{O}(\log s)$-approximation.

Extensions to Chamfer Neighborhood

Chamfer Extension Lemma: Let $A, B \in EMD_s(\mathbb{R}^d, \ell_1)$ then:

$$\mathbb{E}_{T_\Omega}\left[EMD_{T_\Omega}(A,B)\right] \leq \tilde{O}(\log s) \cdot \mathrm{EMD}_{\mathbb{R}^d}(A,B) \cdot \log\left(\frac{\mathrm{Chamfer}_{\mathbb{R}^d}(A,\Omega)}{\mathrm{EMD}_{\mathbb{R}^d}(A,B)}\right)$$

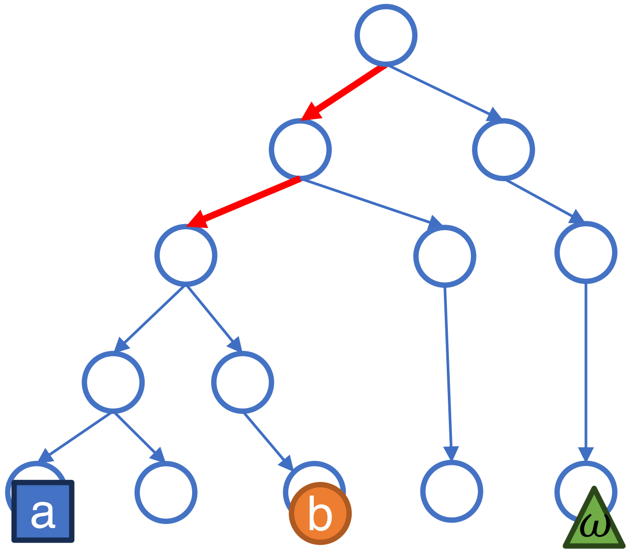
"Extra" approximation factor is log-ratio:

- (How far is A from Ω) / (How far is A from B)

If ratio = $O(\text{poly}\log s)$, get a $\tilde{O}(\log s)$-approximation.

Proof Steps: Three kinds of edges in T_Ω

- (1) Before a, b should meet – small distances
- (2) After a, b, ω all meet – DD edge weights → $\tilde{O}(\log s)$ approx.
- Between (1),(2) – Each edge overpays $\tilde{O}(\log s)$
  - # such edges bounded by ratio



Extensions to Chamfer Neighborhood

Chamfer Extension Lemma: Let $A, B \in EMD_s(\mathbb{R}^d, \ell_1)$ then:

$$\mathbb{E}_{T_\Omega}\left[EMD_{T_\Omega}(A,B)\right] \leq \tilde{O}(\log s) \cdot \mathrm{EMD}_{\mathbb{R}^d}(A,B) \cdot \log\left(\frac{\mathrm{Chamfer}_{\mathbb{R}^d}(A,\Omega)}{\mathrm{EMD}_{\mathbb{R}^d}(A,B)}\right)$$

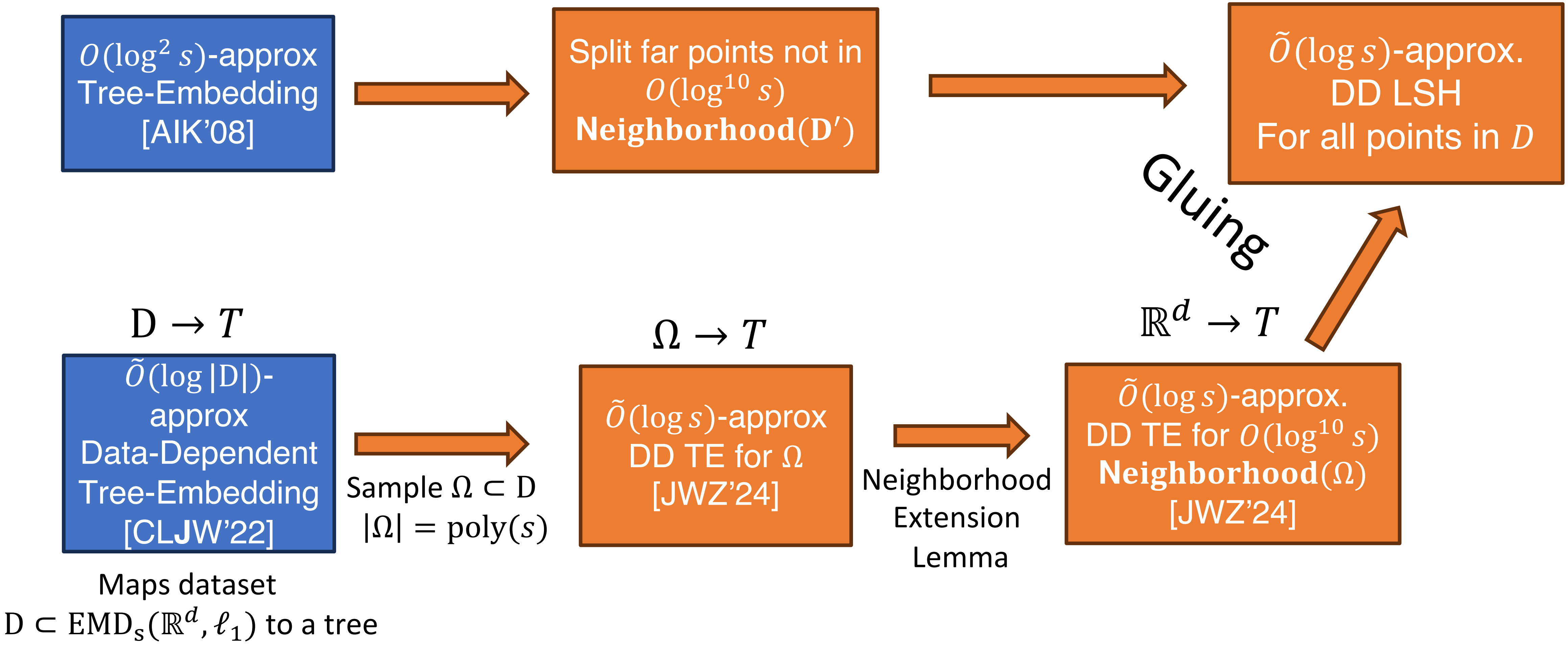
Case 1:

A is $\mathrm{O}(\log^{10} s)$ far from Ω, then [AIK '08] $\log^2 s$ approx is good enough.

Case 2:

Otherwise, Chamfer extension gives extra $\log \log s$ distortion, so new embedding handles it.

Summary:



$O(\log^2 s)$-approx Tree-Embedding [AIK'08]
Split far points not in $O(\log^{10} s)$ $\mathbf{Neighborhood}(\mathbf{D}')$
$\tilde{O}(\log s)$-approx. DD LSH For all points in D
Gluing
$\mathrm{D} \to T$
$\tilde{O}(\log |\mathrm{D}|)$-approx Data-Dependent Tree-Embedding [CLJW'22]
Maps dataset $\mathrm{D} \subset \mathrm{EMD_s}(\mathbb{R}^d, \ell_1)$ to a tree
Sample $\Omega \subset \mathrm{D}$
$|\Omega| = \mathrm{poly}(s)$
$\Omega \to T$
$\tilde{O}(\log s)$-approx DD TE for Ω [JWZ'24]
Neighborhood Extension Lemma
$\mathbb{R}^d \to T$
$\tilde{O}(\log s)$-approx. DD TE for $O(\log^{10} s)$ $\mathbf{Neighborhood}(\Omega)$ [JWZ'24]

Open Problems

1. Can we get a $O(1)$-approx. for EMD ANN in sublinear $O(n^{\epsilon})$ time?
   - We rule out a $O(1)$-approx. for any LSH where close points collide with $\Omega(1)$ probability
     - Nearly all ANN approaches satisfy this

2. $O(1)$-approximate sketching algorithm?
   - Best currently is $O(\log^2 s)$

Complexity of Sublinear EMD

	Sketching / Communication Complexity	Streaming	Nearest Neighbor Search
Approximation Upper Bound	$O(\log^2 s)$	$O(\log^2 s)$ - 1 pass $\tilde{O}(\log s)$ -2 pass	$\tilde{O}(\log s)$
Lower Bound	$\Omega(1)^{**}$	$\Omega(1)^{**}$	---

Thank You!