The Johnson-Lindenstrauss Lemma for Clustering and Subspace Approximation

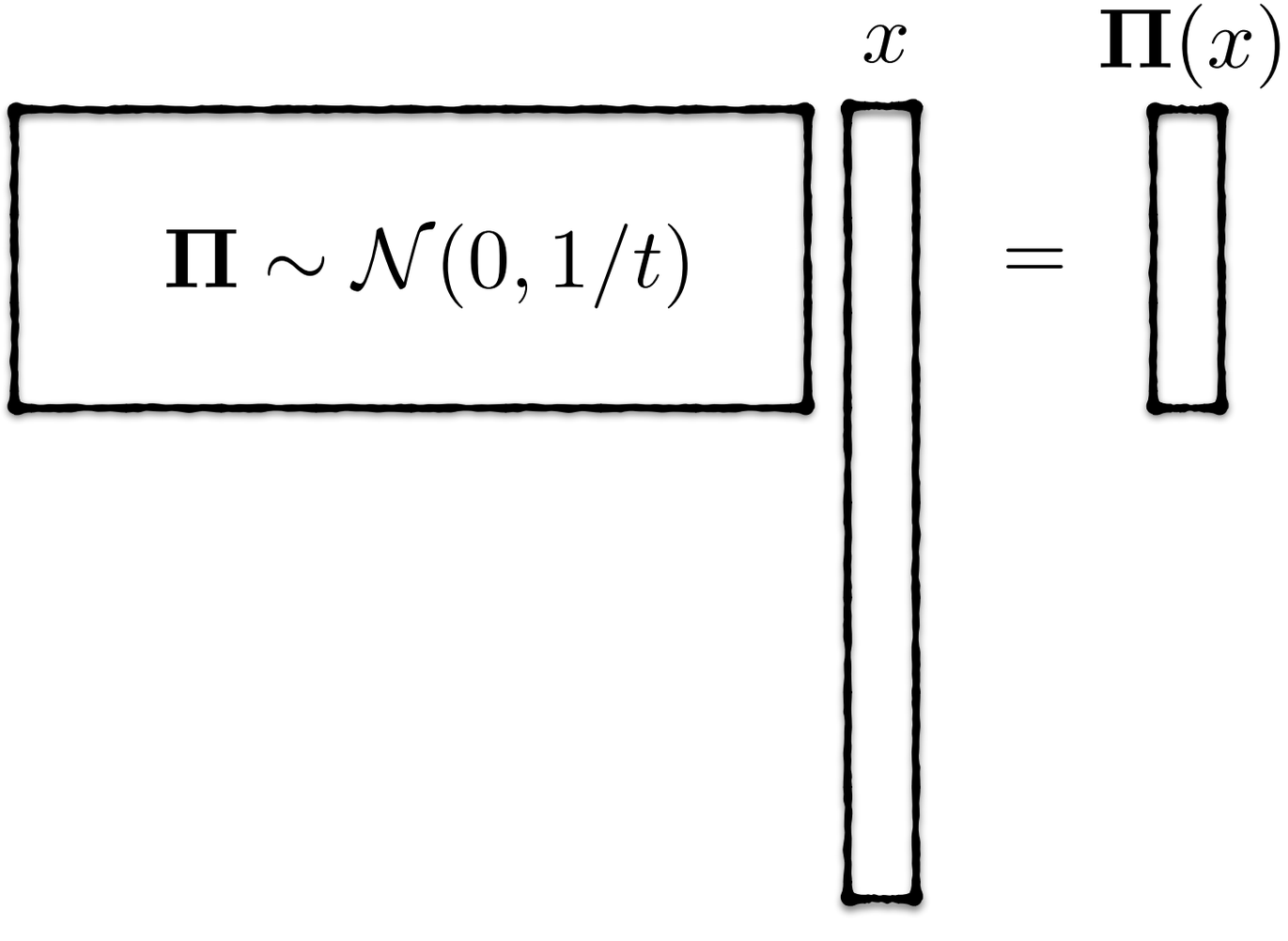
Erik Waingarten and Moses Charikar

Johnson-Lindenstrauss (JL) Lemma '84:

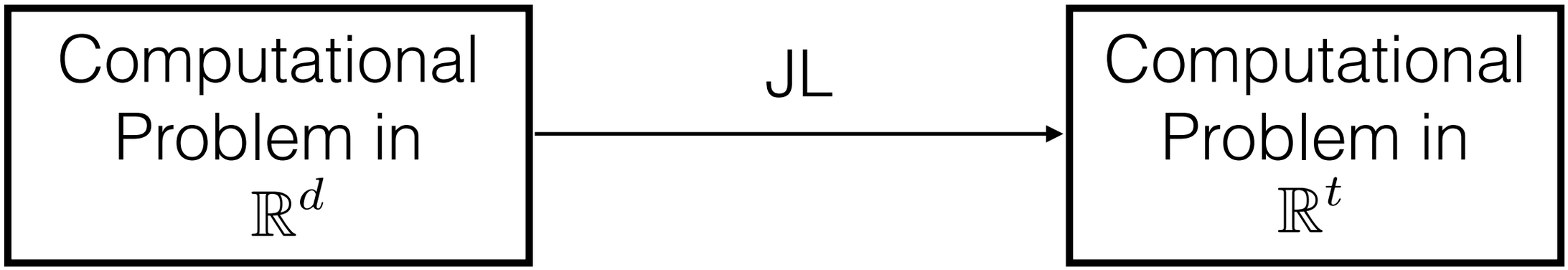
Let $X \subset \mathbb{R}^d$ be any subset of n points. There exists a map $\Pi \colon \mathbb{R}^d \to \mathbb{R}^t$, with

$$t = O(\log n/\epsilon^2)$$

such that for any $x, y \in X$, $\quad \|x - y\|_2 \approx_{1\pm\epsilon} \|\Pi(x) - \Pi(y)\|_2$



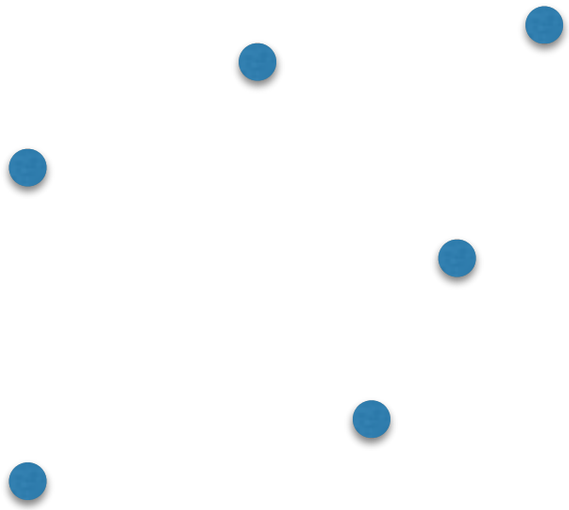
Our Motivating Question:



How large does t need to be
to preserve solution up to $(1+\epsilon)$?

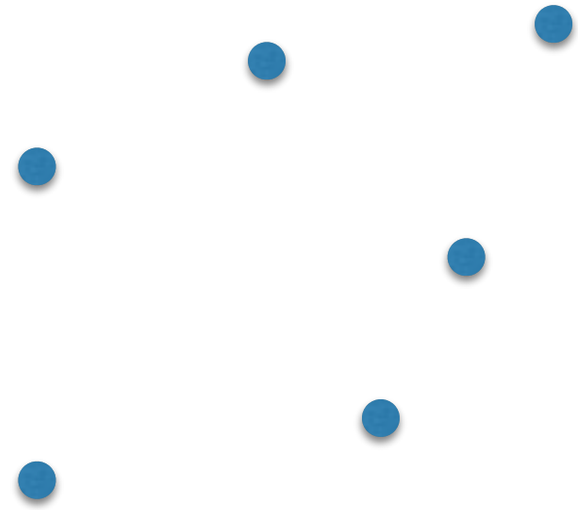
1-Median

$$\min_{c \in \mathbb{R}^d} \sum_{i=1}^{n} \|x_i - c\|_2$$



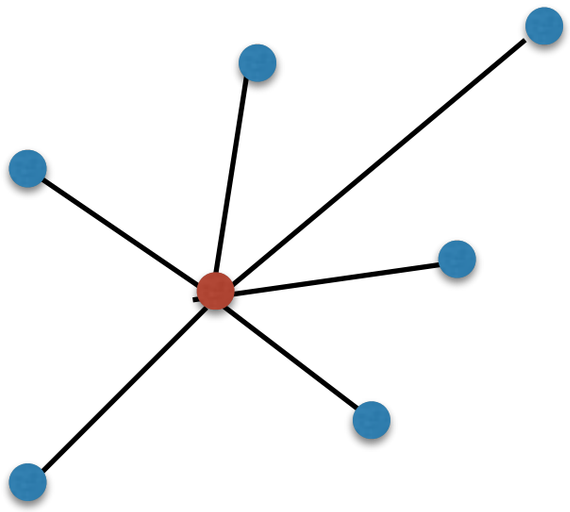
1-Medoid

$$\min_{c \in \{x_1, \ldots, x_n\}} \sum_{i=1}^{n} \|x_i - c\|_2$$



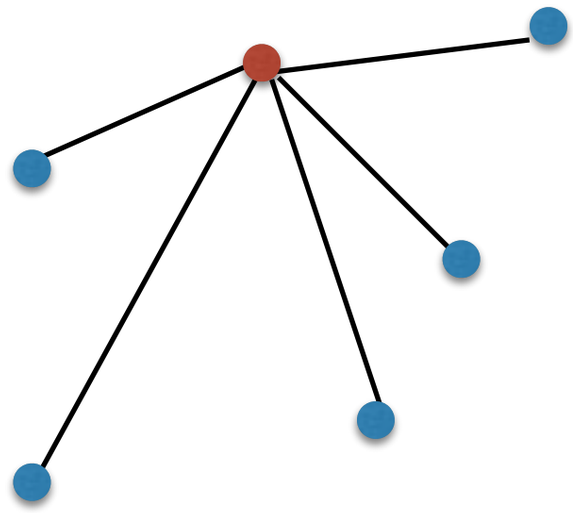
1-Median

$$\min_{c \in \mathbb{R}^d} \sum_{i=1}^{n} \|x_i - c\|_2$$



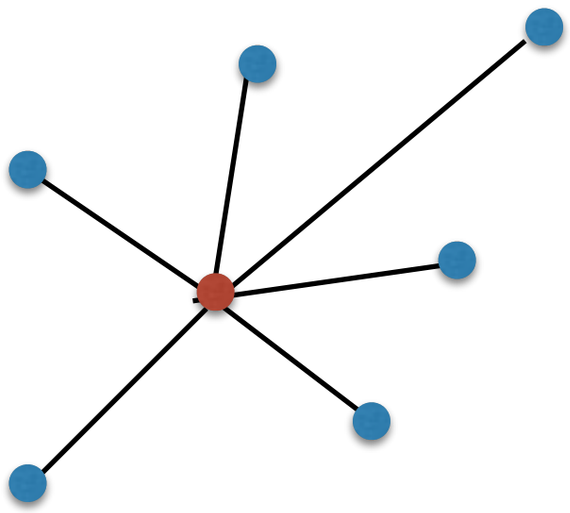
1-Medoid

$$\min_{c \in \{x_1, \ldots, x_n\}} \sum_{i=1}^{n} \|x_i - c\|_2$$



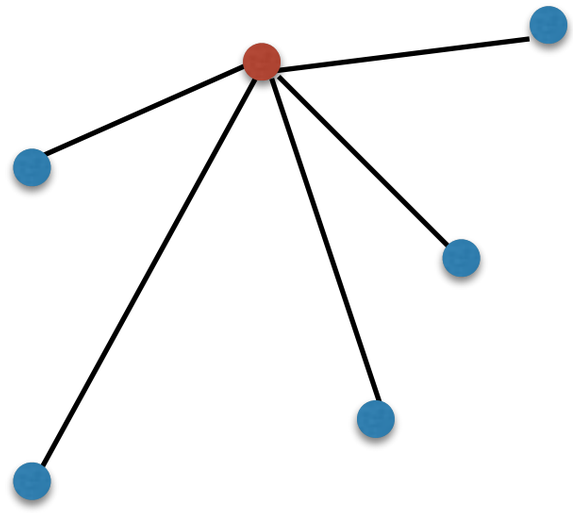
1-Median

$$\min_{c \in \mathbb{R}^d} \sum_{i=1}^{n} \|x_i - c\|_2$$



1-Medoid

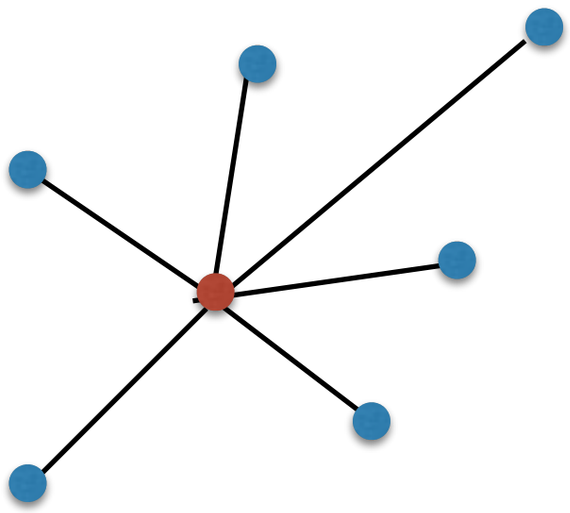
$$\min_{c \in \{x_1, \ldots, x_n\}} \sum_{i=1}^{n} \|x_i - c\|_2$$



Can we apply JL at all?

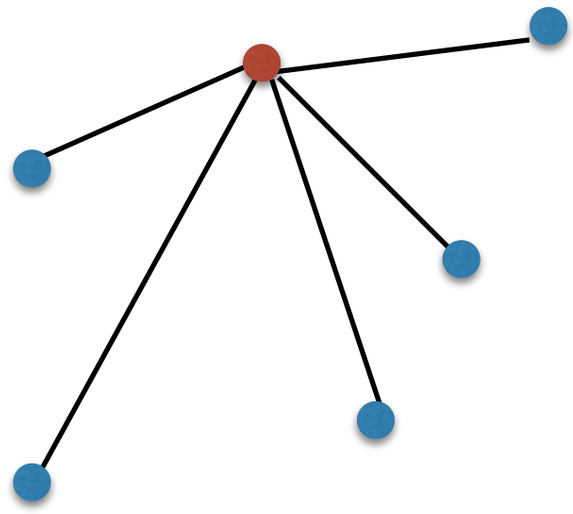
1-Median

$$\min_{c \in \mathbb{R}^d} \sum_{i=1}^{n} \|x_i - c\|_2$$



1-Medoid

$$\min_{c \in \{x_1, \ldots, x_n\}} \sum_{i=1}^{n} \|x_i - c\|_2$$

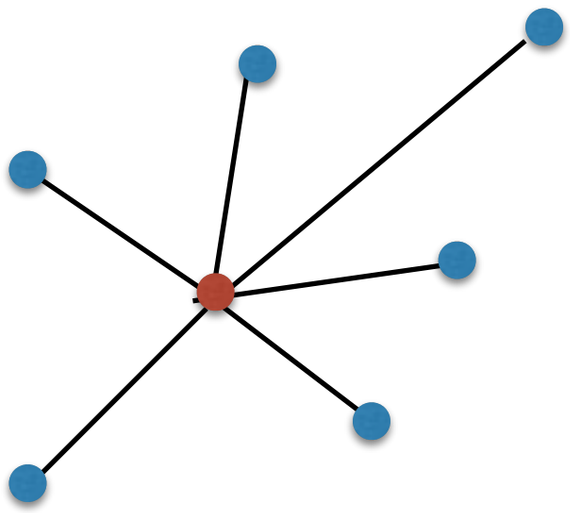


Can we apply JL at all?

$$\Theta(\log n / \epsilon^2)$$

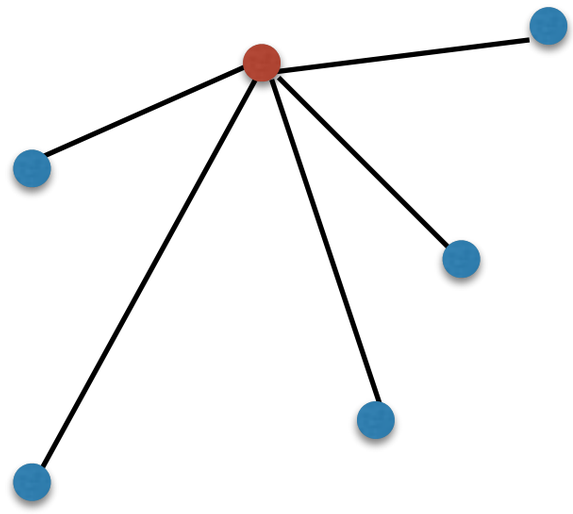
1-Median

$$\min_{c \in \mathbb{R}^d} \sum_{i=1}^{n} \|x_i - c\|_2$$



1-Medoid

$$\min_{c \in \{x_1, \ldots, x_n\}} \sum_{i=1}^{n} \|x_i - c\|_2$$



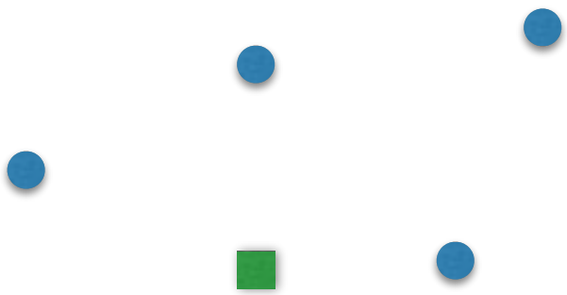
Can we apply JL at all?

$\tilde{\Theta}(1/\epsilon^2)$

$\Theta(\log n/\epsilon^2)$

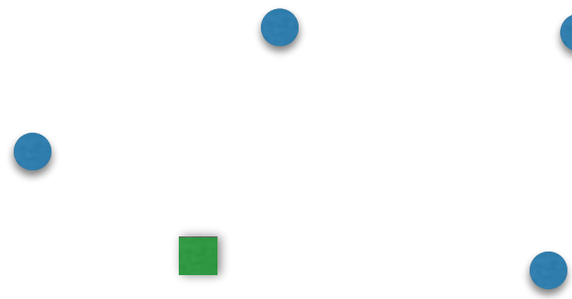
1-Subspace Approximation

$$\min_{\substack{S\subset\mathbb{R}^d \\ \dim S\leq 1}} \sum_{i=1}^{n} \|x_i - \rho_S(x_i)\|_2$$



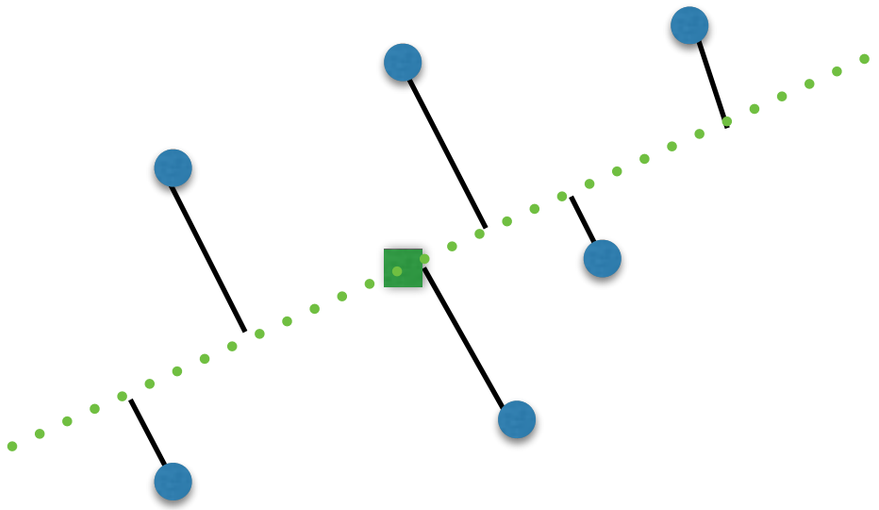
1-Column Subset Selection

$$\min_{\substack{S\subset\mathbb{R}^d \\ S=\mathrm{span}(x_j)}} \sum_{i=1}^{n} \|x_i - \rho_S(x_i)\|_2$$



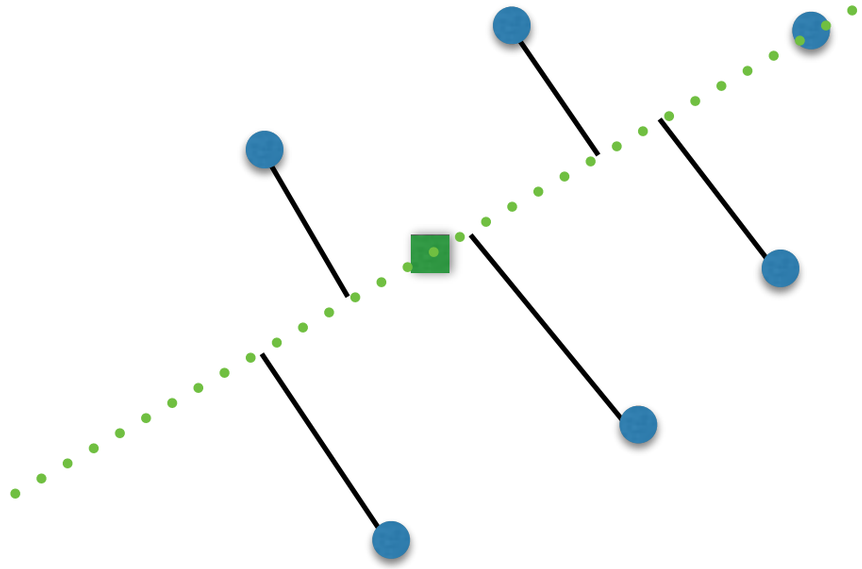
1-Subspace Approximation

$$\min_{\substack{S\subset\mathbb{R}^d \\ \dim S\leq 1}} \sum_{i=1}^{n} \|x_i - \rho_S(x_i)\|_2$$



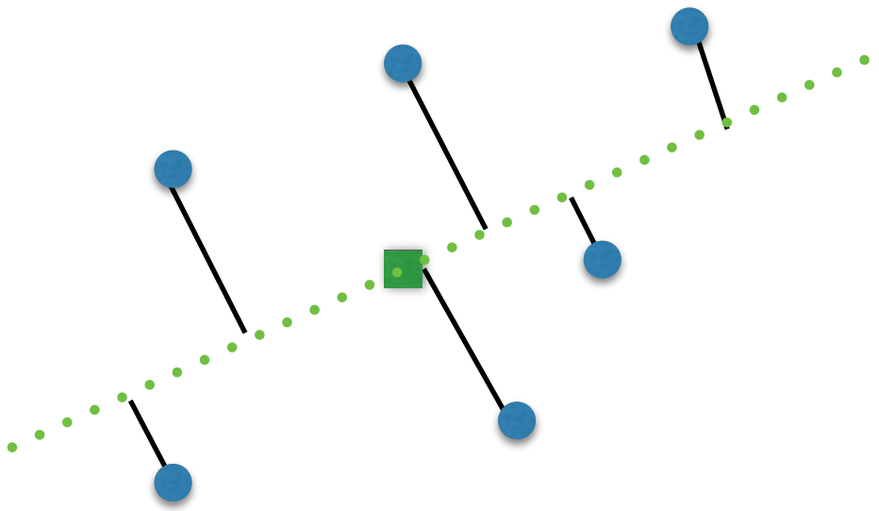
1-Column Subset Selection

$$\min_{\substack{S\subset\mathbb{R}^d \\ S=\mathrm{span}(x_j)}} \sum_{i=1}^{n} \|x_i - \rho_S(x_i)\|_2$$



1-Subspace Approximation

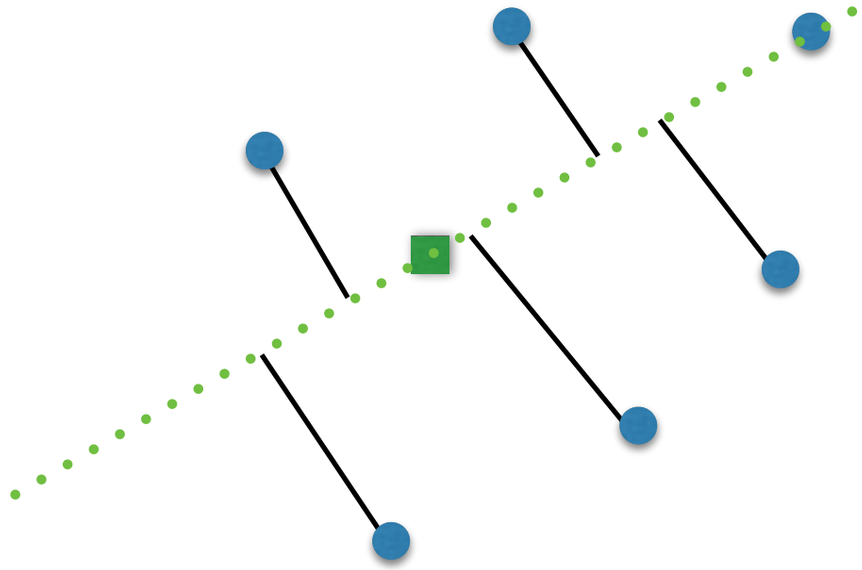
$$\min_{\substack{S\subset\mathbb{R}^d \\ \dim S\leq 1}} \sum_{i=1}^{n} \|x_i - \rho_S(x_i)\|_2$$



$$\tilde{O}(1/\epsilon^3)$$

1-Column Subset Selection

$$\min_{\substack{S\subset\mathbb{R}^d \\ S=\mathrm{span}(x_j)}} \sum_{i=1}^{n} \|x_i - \rho_S(x_i)\|_2$$



$$\Theta(\log n/\epsilon^2)$$

Talk Outline:

- Problems we will study
  - Prior work

- Techniques

- Open Problems

Talk Outline:

- Problems we will study
  - Prior work

- Techniques

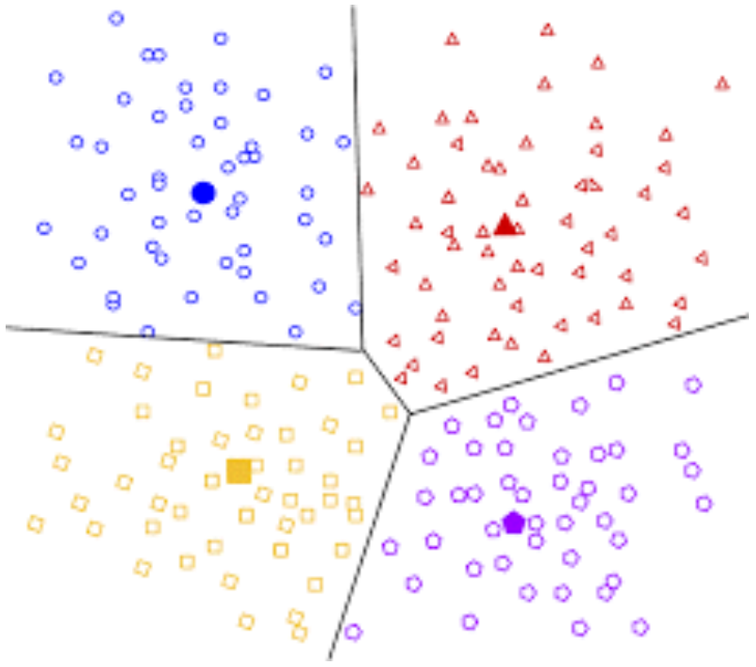
- Open Problems

Conceptual Take-away:
Use algorithms for constructing *coresets* in order to analyze the JL transform.

(k, z)-Clustering :

Input: $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d, k \in \mathbb{N}, z \geq 1$

Objective: $\min_{c_1,\ldots,c_k \in \mathbb{R}^d} \left(\sum_{i=1}^{n} \min_{j \in [k]} \|x_i - c_j\|_2^z \right)^{1/z}$



z = 1: k-median

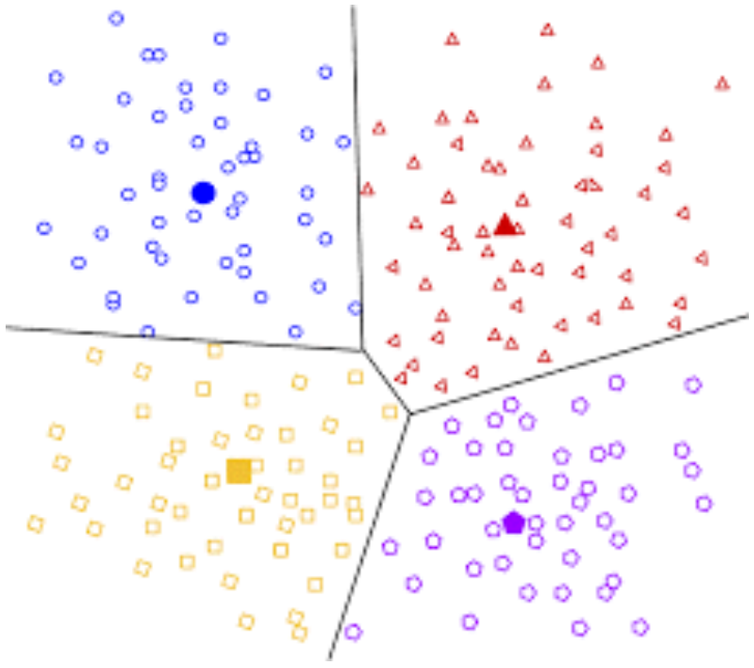
z = 2: k-mean

z = ∞ : k-center

(k, z)-Clustering :

Input: $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d, k \in \mathbb{N}, z \geq 1$

Objective: $$\min_{c_1,\ldots,c_k \in \mathbb{R}^d} \left(\sum_{i=1}^{n} \min_{j \in [k]} \|x_i - c_j\|_2^z \right)^{1/z}$$



z = 1: k-median
z = 2: k-mean
z = ∞ : k-center

$$\min_{y_1,\ldots,y_k \in \mathbb{R}^t} \left(\sum_{i=1}^{n} \min_{j \in [k]} \|\Pi(x_i) - y_j\|_2^z \right)^{1/z}$$

(k, z)-Subspace Approximation:

$$\text{Input: } X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d, k \in \mathbb{N}, z \geq 1$$

$$\text{Objective: } \min_{\substack{S \subset \mathbb{R}^d \\ \dim S \leq k}} \left(\sum_{i=1}^{n} \|x_i - \rho_S(x_i)\|_2^z \right)^{1/z}$$

z = 2: low-rank approx

z = ∞: radii/width of points

(k, z)-Subspace Approximation:

$$\text{Input: } X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d, k \in \mathbb{N}, z \geq 1$$

$$\text{Objective: } \min_{\substack{S \subset \mathbb{R}^d \\ \dim S \leq k}} \left(\sum_{i=1}^{n} \|x_i - \rho_S(x_i)\|_2^z \right)^{1/z}$$

z = 2: low-rank approx

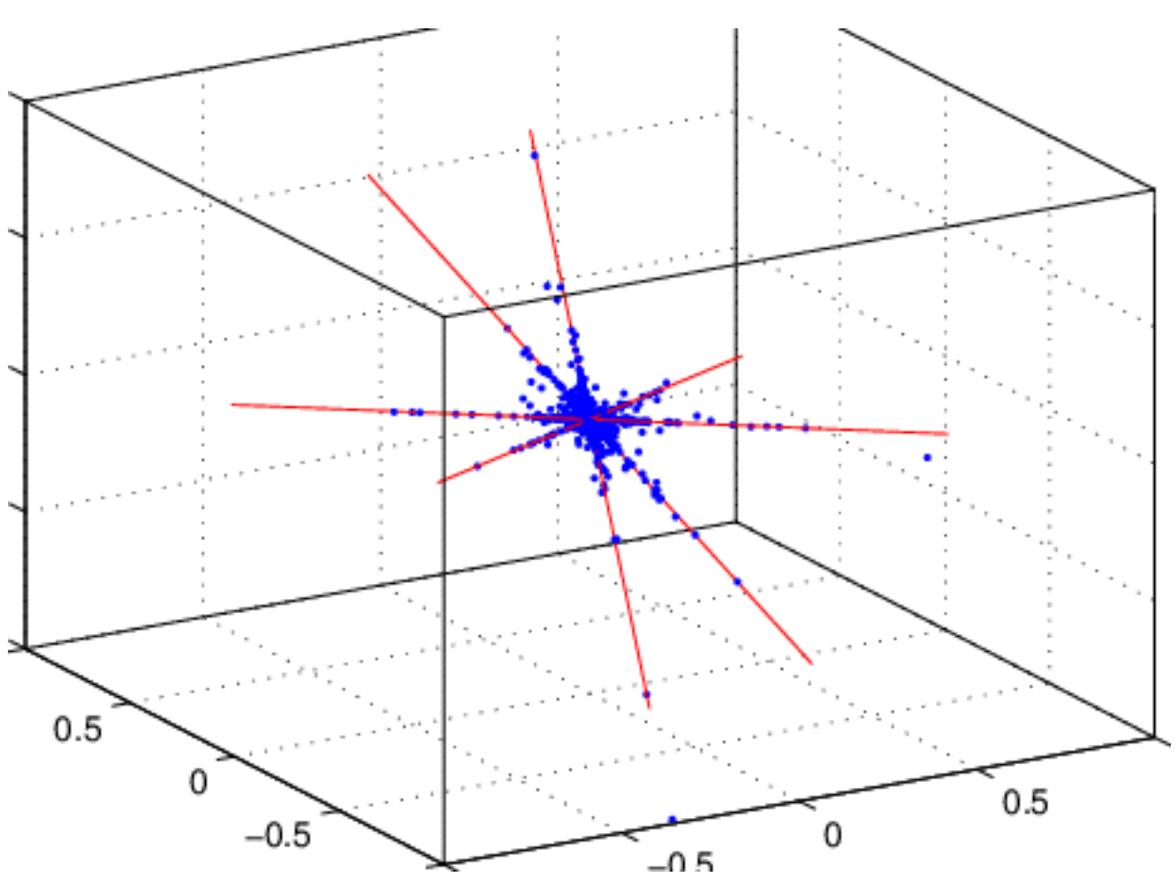
z = ∞: radii/width of points

$$\min_{\substack{T \subset \mathbb{R}^t \\ \dim T \leq k}} \left(\sum_{i=1}^{n} \|\Pi(x_i) - \rho_T(\Pi(x_i))\|_2^z \right)^{1/z}$$

(k, z)-Line Approximation:

$$\text{Input: } X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d, k \in \mathbb{N}, z \geq 1$$

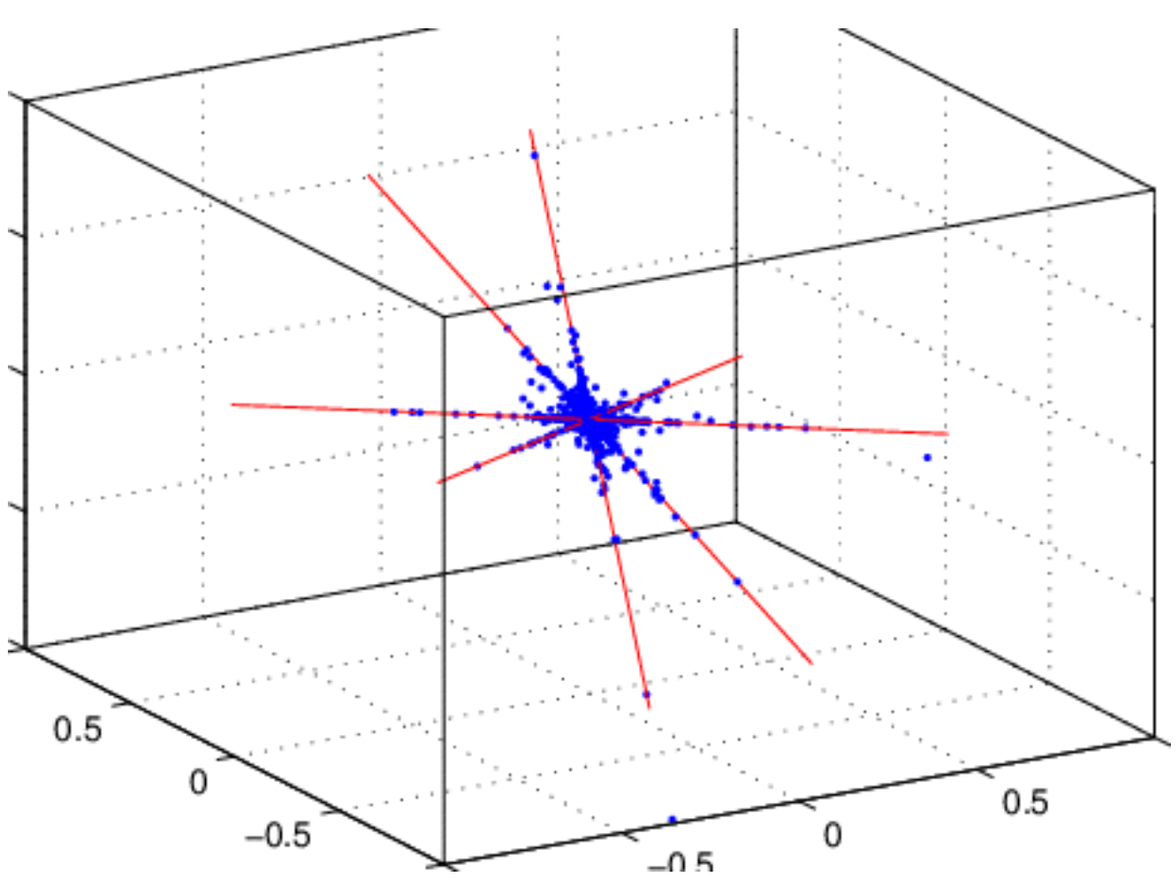
$$\text{Objective: } \min_{\substack{\ell_1,\ldots,\ell_k \\ \text{lines in } \mathbb{R}^t}} \left(\sum_{i=1}^{n} \min_{j \in [k]} \|x_i - \rho_{\ell_j}(x_i)\|_2^z \right)^{1/z}$$



(k, z)-Line Approximation:

$$\text{Input: } X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d, k \in \mathbb{N}, z \geq 1$$

$$\text{Objective: } \min_{\substack{\ell_1,\ldots,\ell_k \\ \text{lines in } \mathbb{R}^t}} \left(\sum_{i=1}^{n} \min_{j \in [k]} \|x_i - \rho_{\ell_j}(x_i)\|_2^z \right)^{1/z}$$



Projective Clustering

Prior Work:

$z = 2$

General z

Prior Work:

$z = 2$	General z

k-Means:

- [Bousidis, Zouzias, Drineas '10]
- [Cohen, Elder, Musco, Musco, Persu '15]
- [Beccheti, Bury, Cohen-Added, Grandoni, Schwiegelshohn '19]
- [Makarychev, Makarychev, Razenshteyn '19] $O(\log(k/\epsilon)/\epsilon^2)$

Low-Rank Approx:

- [Cohen, Elder, Musco, Musco, Persu '15] $O(k/\epsilon^2)$

Prior Work:

$z = 2$

k-Means:

- [Bousidis, Zouzias, Drineas '10]
- [Cohen, Elder, Musco, Musco, Persu '15]
- [Beccheti, Bury, Cohen-Added, Grandoni, Schwiegelshohn '19]
- [Makarychev, Makarychev, Razenshteyn '19] $O(\log(k/\epsilon)/\epsilon^2)$

Low-Rank Approx:

- [Cohen, Elder, Musco, Musco, Persu '15] $O(k/\epsilon^2)$

General z

(k,z)-Clustering:

- [Makarychev, Makarychev, Razenstheyn '19]

$$O\left(\frac{\log k + z\log(1/\epsilon) + z^2}{\epsilon^2}\right)$$

(k,z)-Subspace Approx:

- [Kerber, Raghvendra '15] $O(zk^2 \log n/\epsilon^3)$

Prior Work:

$z = 2$

k-Means:

- [Bousidis, Zouzias, Drineas '10]
- [Cohen, Elder, Musco, Musco, Persu '15]
- [Beccheti, Bury, Cohen-Added, Grandoni, Schwiegelshohn '19]
- [Makarychev, Makarychev, Razenshteyn '19] $O(\log(k/\epsilon)/\epsilon^2)$

Low-Rank Approx:

- [Cohen, Elder, Musco, Musco, Persu '15] $O(k/\epsilon^2)$

General z

(k,z)-Clustering:

- [Makarychev, Makarychev, Razenstheyn '19]

$$O\left(\frac{\log k + z\log(1/\epsilon) + z^2}{\epsilon^2}\right)$$

(k,z)-Subspace Approx:

- [Kerber, Raghvendra '15] $O(zk^2 \log n/\epsilon^3)$

$$(k,z)\text{-Clustering}: \quad t = O\left(\frac{\log k + z\log(1/\epsilon)}{\epsilon^2}\right)$$

$$(k,z)\text{-Subspace Approximation}: \quad t = \tilde{O}\left(\frac{zk^2}{\epsilon^3}\right)$$

$$(k,z)\text{-Line Approximation}: \quad t = \tilde{O}\left(\frac{k\log\log n + z + \log(1/\epsilon)}{\epsilon^3}\right)$$

Talk Outline:

- Problems we will study
  - Prior work

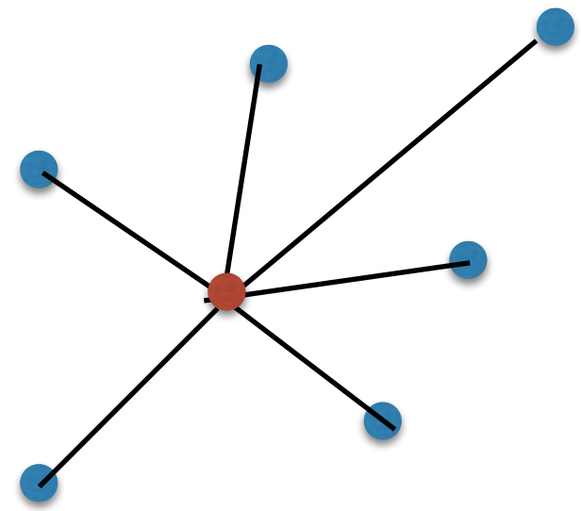
- **Techniques**
  - **Coresets to JL**

- Open Problems

Example: 1-Median

Input: $X = \{x_1, \ldots, x_n\}$ in $\mathbb{R}^d$

Objective: $\min_{c \in \mathbb{R}^d} \sum_{i=1}^{n} \|x_i - c\|_2$



Question:

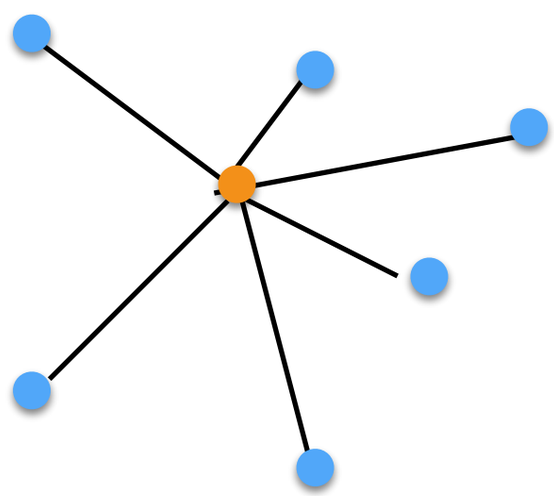
~~$\tilde{\Theta}(1/\epsilon^2)$~~

$\mathrm{poly}(1/\epsilon)$

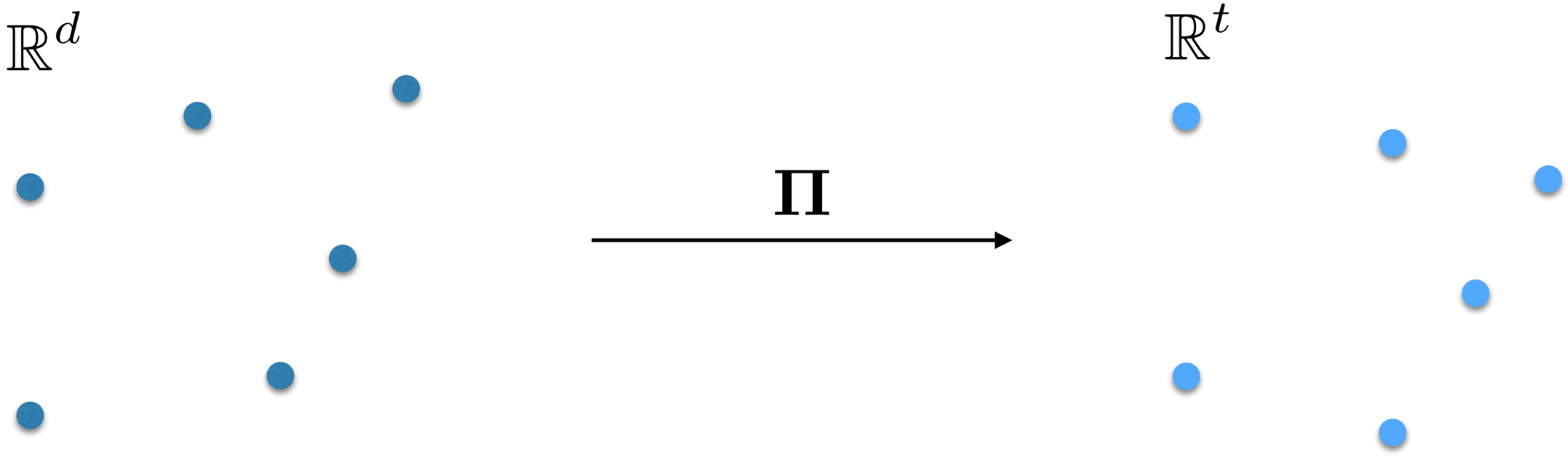
$\mathbf{\Pi}$ ≈ ?

Sample JL Map $\mathbf{\Pi}$

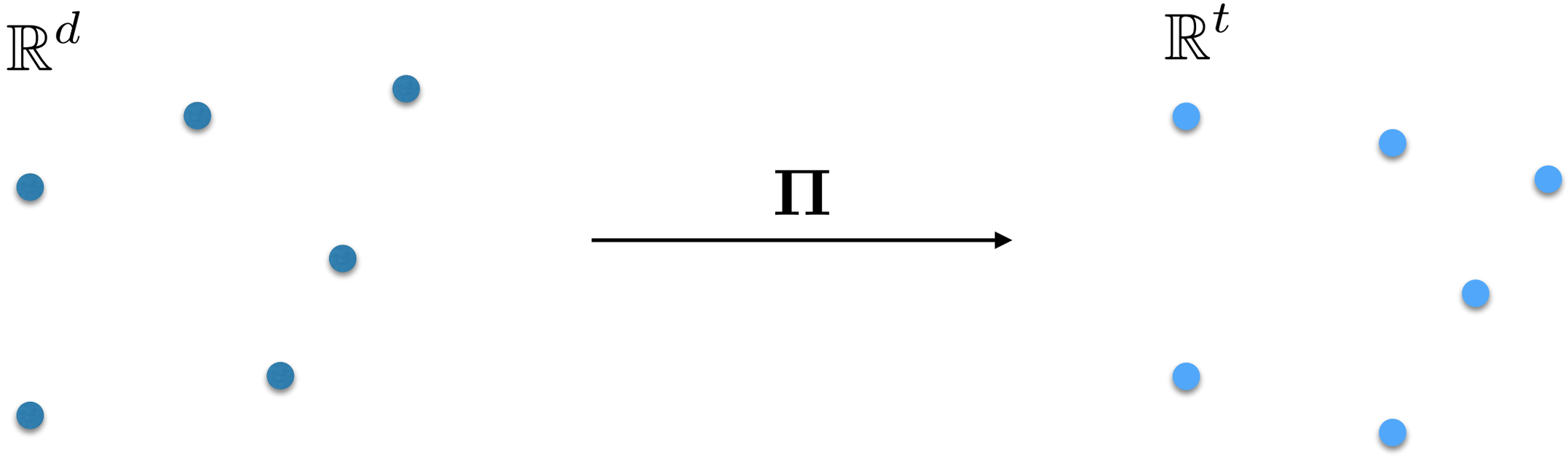
$$\min_{y \in \mathbb{R}^t} \sum_{i=1}^{n} \|\mathbf{\Pi}(x_i) - y\|_2$$



Example: 1-Median — Easy Direction



Example: 1-Median — Easy Direction



<u>Claim:</u> As long as $t \geq 1/\epsilon^2$, *cost does not increase:*

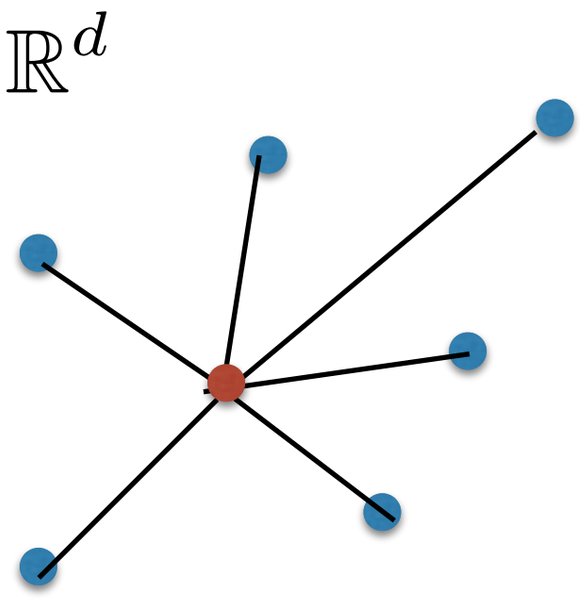
Cost in original space	$\times(1+\epsilon)$	$\geq$	Cost in projected space

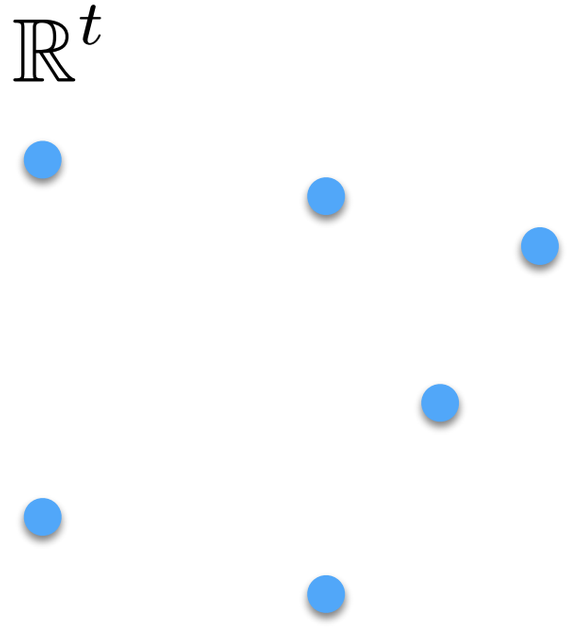
Example: 1-Median — Easy Direction

<u>Claim:</u> As long as $t \geq 1/\epsilon^2$, *cost does not increase:*

Cost in original space $\times(1+\epsilon)$ $\geq$ Cost in projected space

Example: 1-Median — Easy Direction

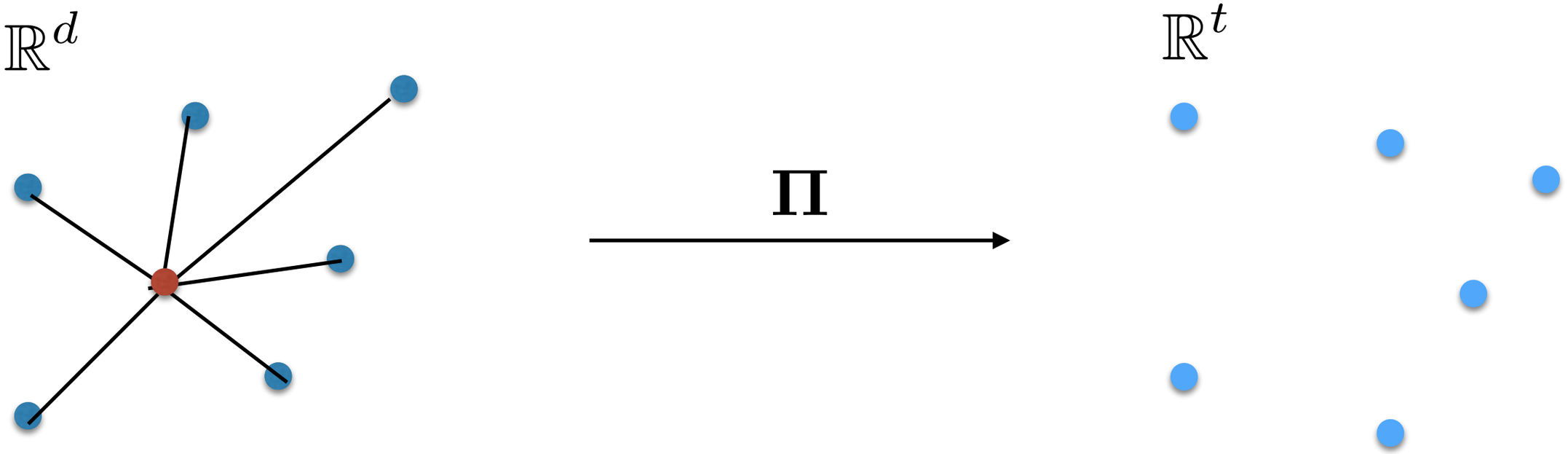




Claim: As long as $t \geq 1/\epsilon^2$, *cost does not increase:*

Cost in original space	$\times(1+\epsilon)$	$\geq$	Cost in projected space
$\sum_{i=1}^{n} \|x_i - c\|_2$			

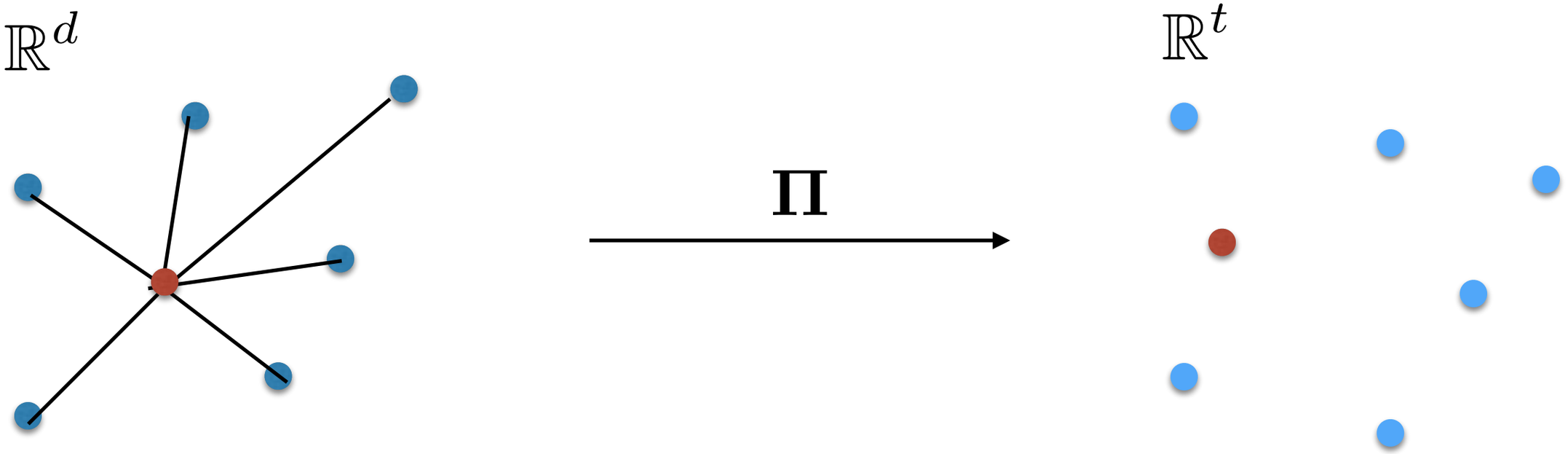
Example: 1-Median — Easy Direction



<u>Claim:</u> As long as $t \geq 1/\epsilon^2$, *cost does not increase:*

$$\underbrace{\sum_{i=1}^{n} \|x_i - c\|_2}_{\text{Cost in original space}} \times(1+\epsilon) \geq \underbrace{\qquad\qquad}_{\text{Cost in projected space}}$$

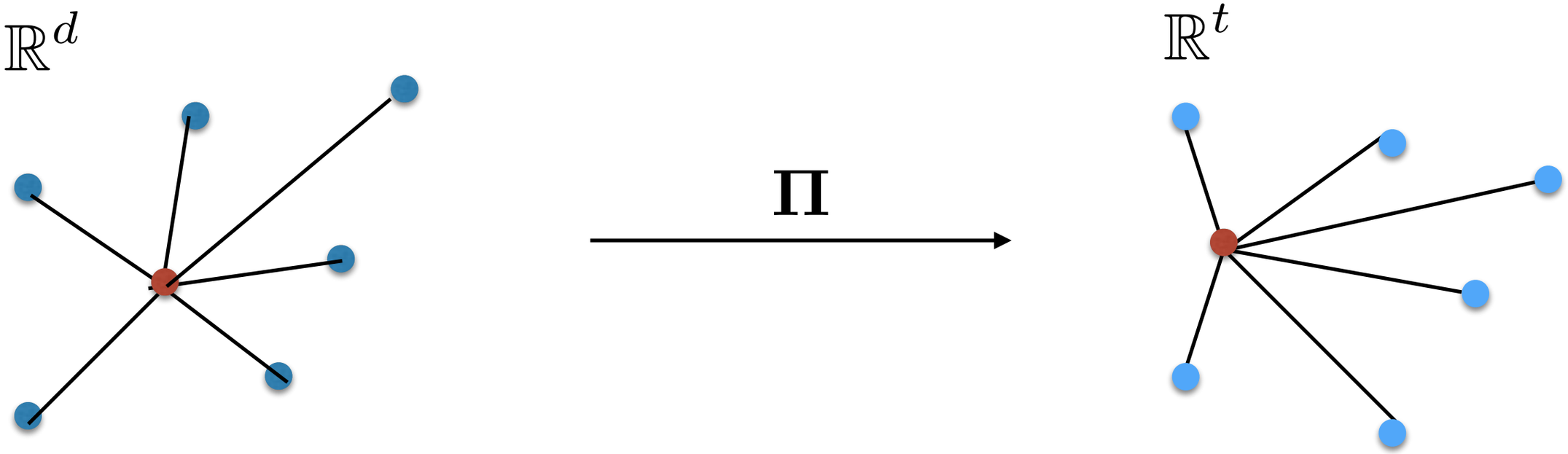
Example: 1-Median — Easy Direction



<u>Claim:</u> As long as $t \geq 1/\epsilon^2$, *cost does not increase:*

Cost in original space	$\times(1+\epsilon)$	$\geq$	Cost in projected space
$\sum_{i=1}^{n} \|x_i - c\|_2$			

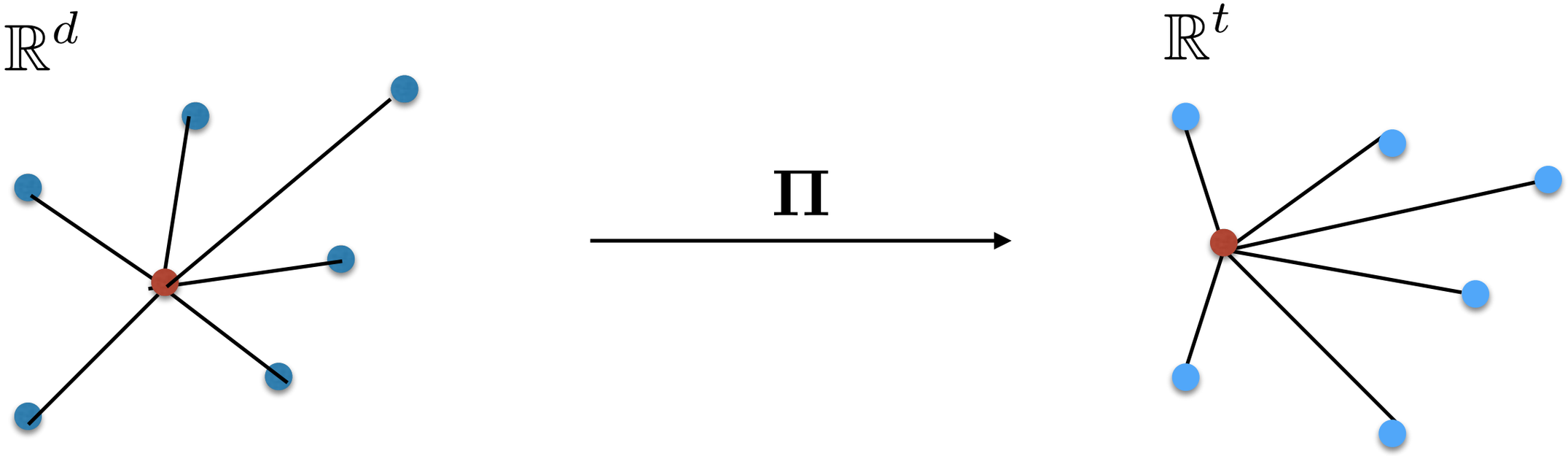
Example: 1-Median — Easy Direction



<u>Claim:</u> As long as $t \geq 1/\epsilon^2$, *cost does not increase:*

Cost in original space	$\times(1+\epsilon)$	$\geq$	Cost in projected space
$\sum_{i=1}^{n} \|x_i - c\|_2$			

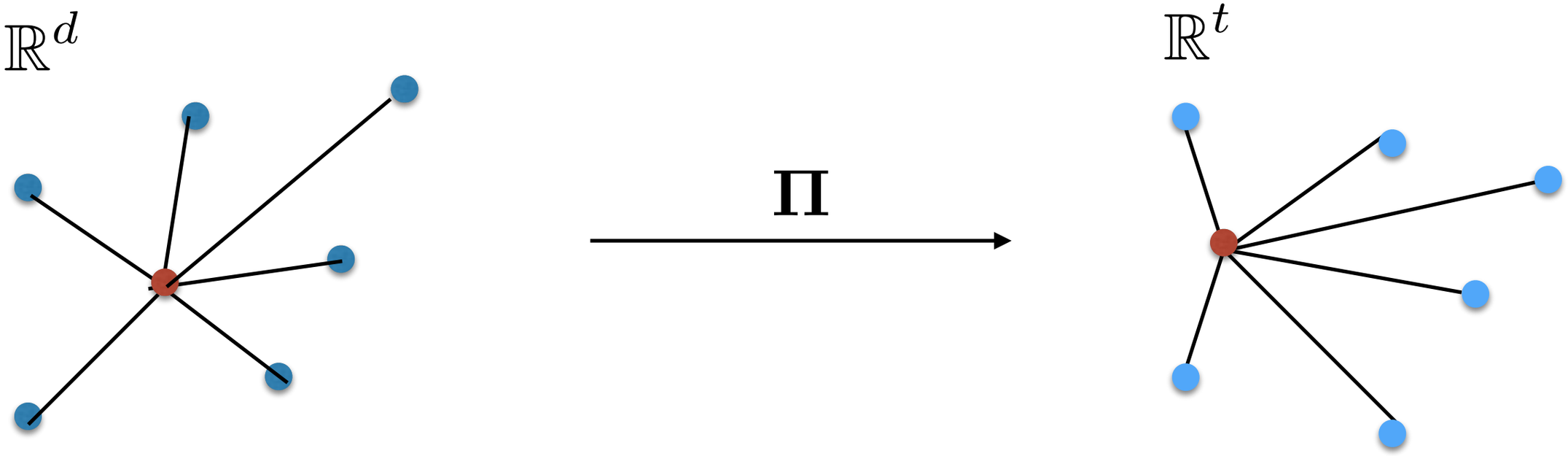
Example: 1-Median — Easy Direction



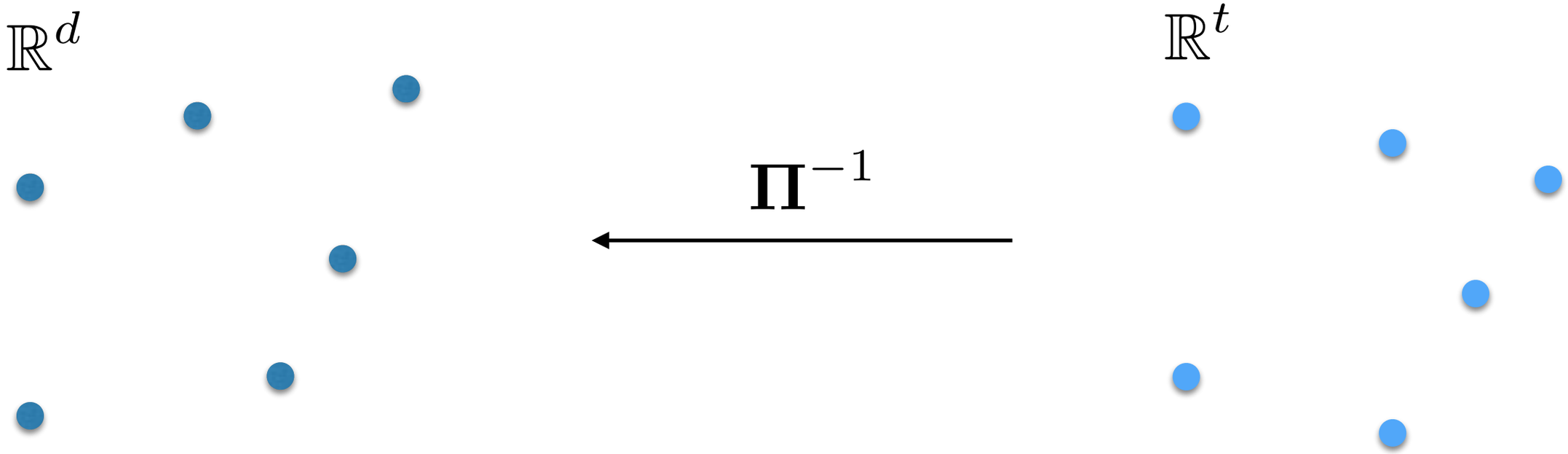
Claim: As long as $t \geq 1/\epsilon^2$, *cost does not increase:*

$$\underbrace{\sum_{i=1}^{n} \|x_i - c\|_2}_{\text{Cost in original space}} \times (1+\epsilon) \geq \underbrace{\sum_{i=1}^{n} \|\mathbf{\Pi}(x_i) - \mathbf{\Pi}(c)\|_2}_{\text{Cost in projected space}}$$

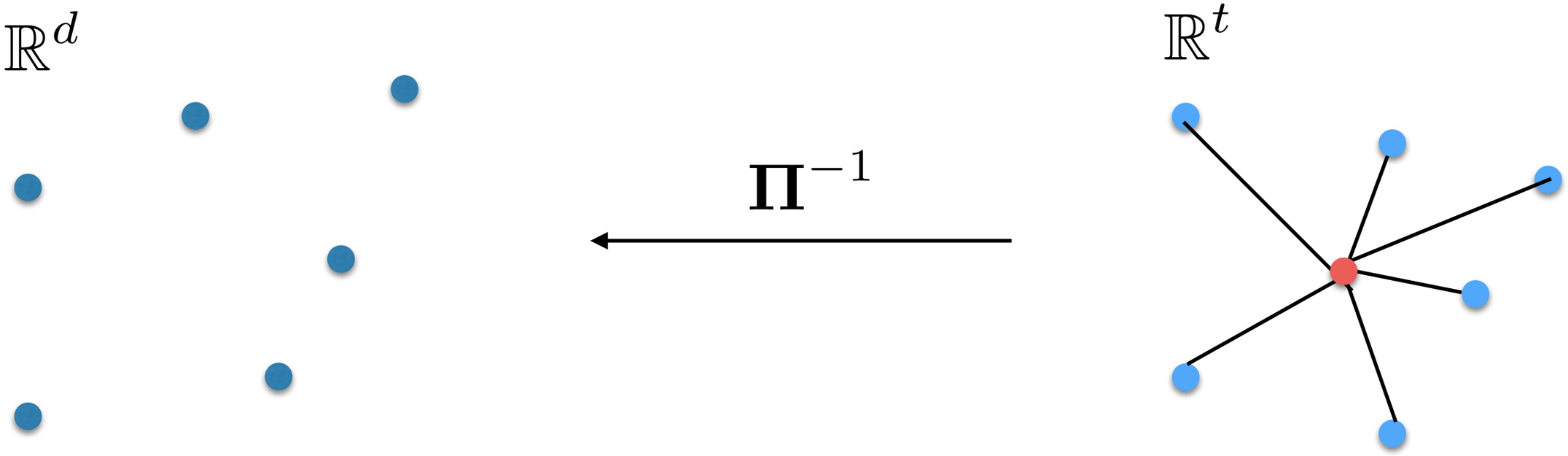
Example: 1-Median — Hard Direction



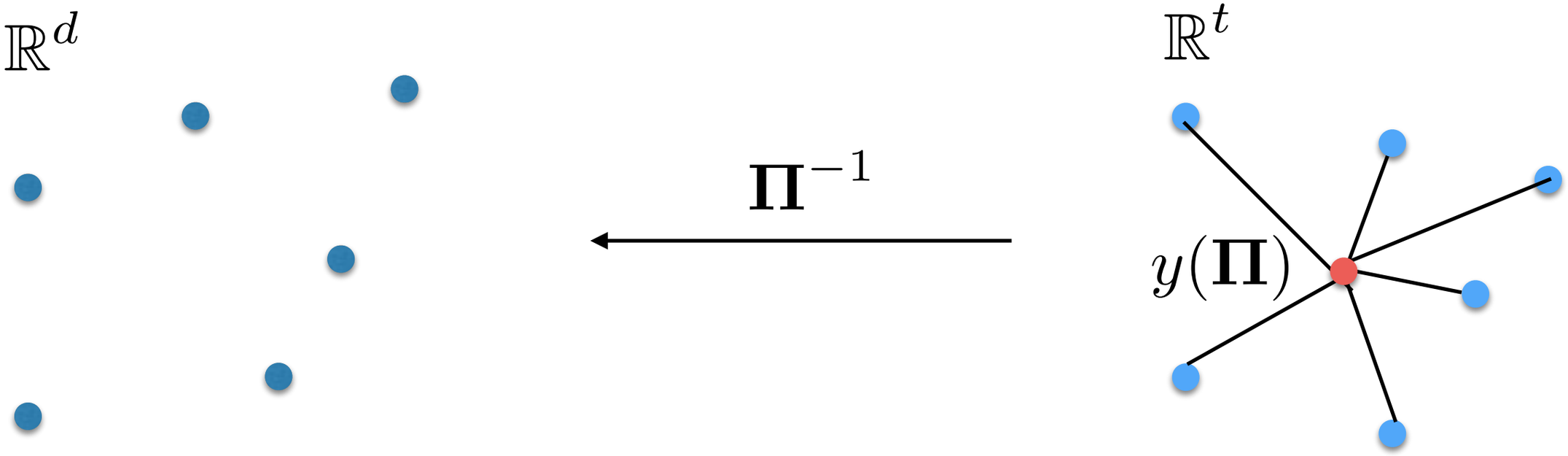
Example: 1-Median — Hard Direction



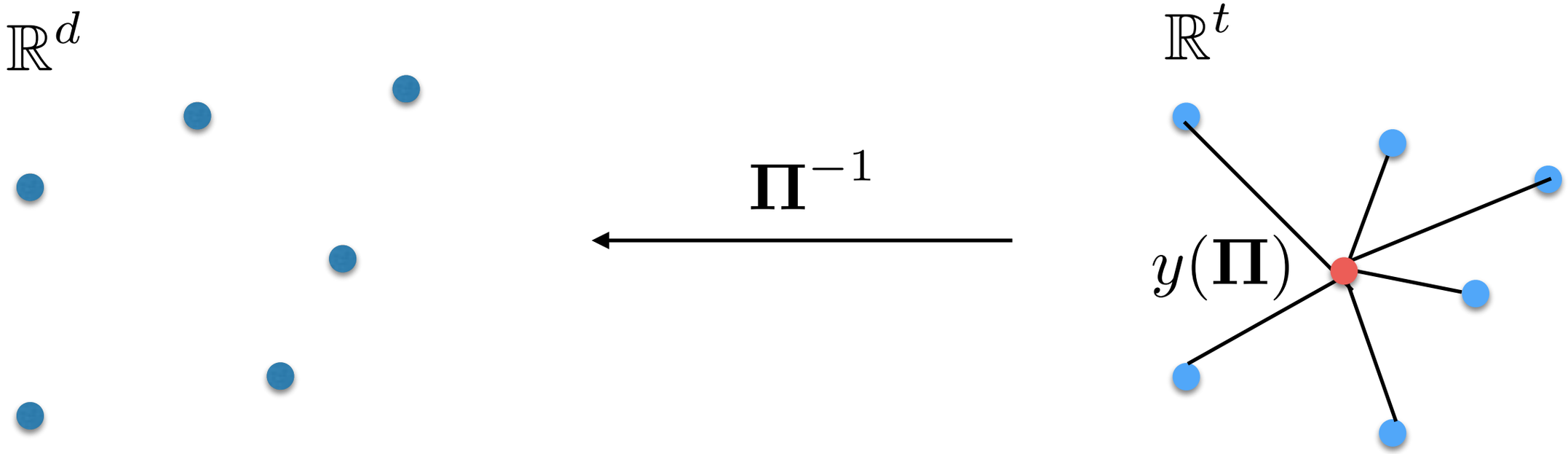
Example: 1-Median — Hard Direction



Example: 1-Median — Hard Direction



Example: 1-Median — Hard Direction



Can’t use distributional properties of $\mathbf{\Pi}^{-1}$ on the dataset points!

Coreset: “Dataset Reduction” Technique

[Agarwal, Har-Peled, Varadarajan ‘03]

Coreset: “Dataset Reduction” Technique

[Agarwal, Har-Peled, Varadarajan ‘03]

Original dataset

Smaller *weighted* dataset

1-Median

$$\min_{c \in \mathbb{R}^d} \sum_{i=1}^{n} \|x_i - c\|_2$$

Size: $\mathrm{poly}(1/\epsilon)$

Coreset

(S, w)

1-Median

$$\min_{c \in \mathbb{R}^d} \sum_{i=1}^{n} \|x_i - c\|_2$$

Size: $\mathrm{poly}(1/\epsilon)$

Coreset

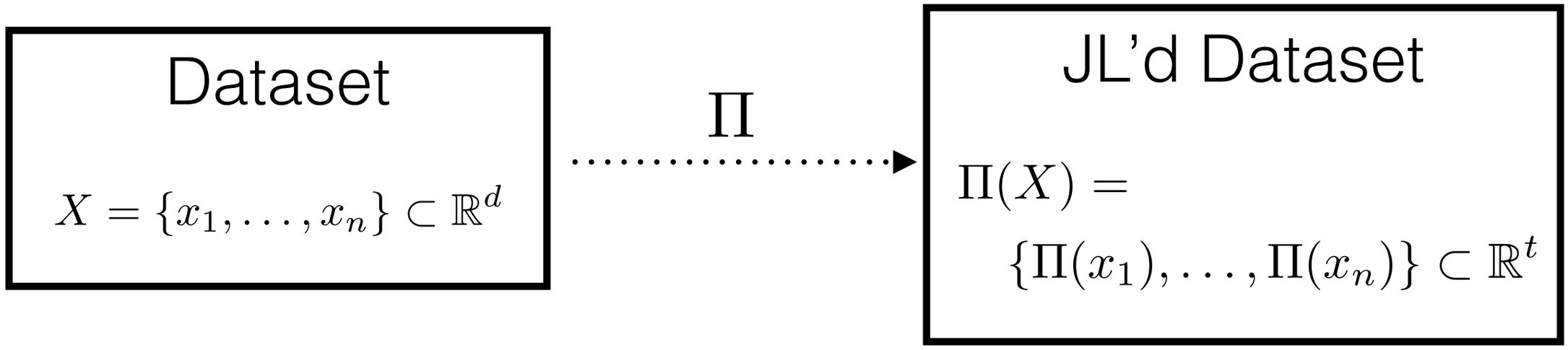
(S, w)

Message:
Looks like it depends on all data,
but really, it doesn't.

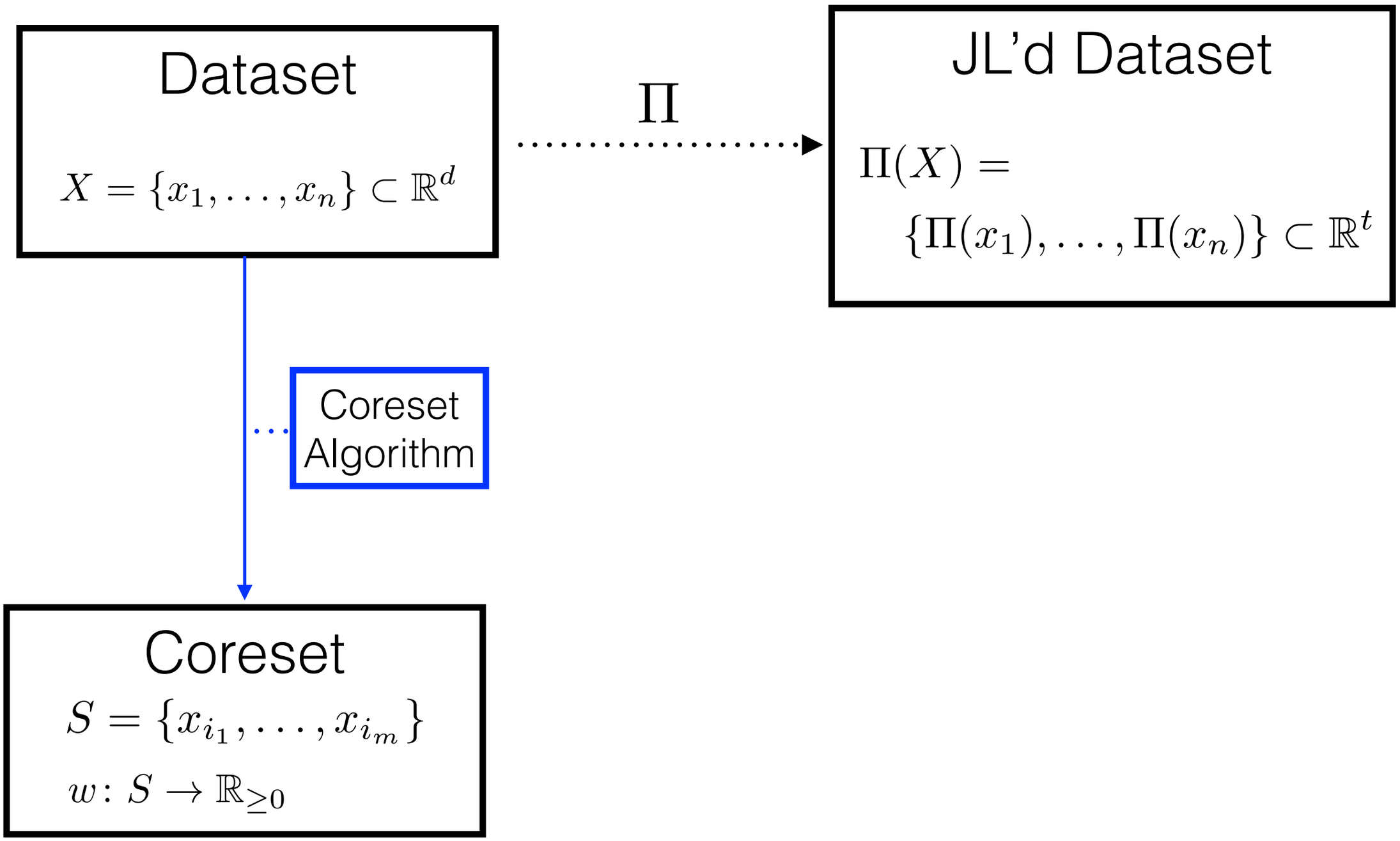
Proof Diagram for using

Coresets for **Dimension Reduction**

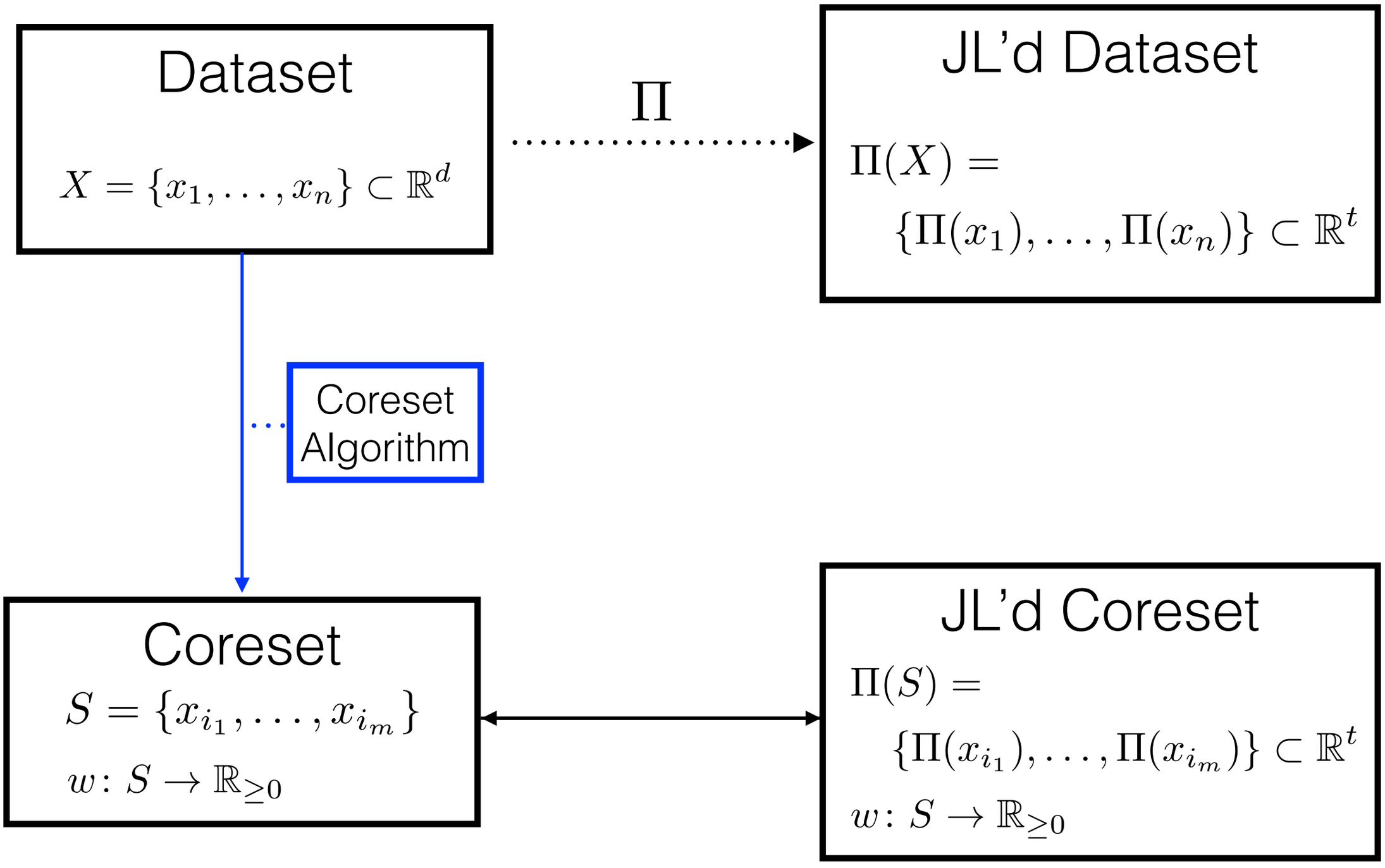
Proof Diagram for using **Coresets** for **Dimension Reduction**



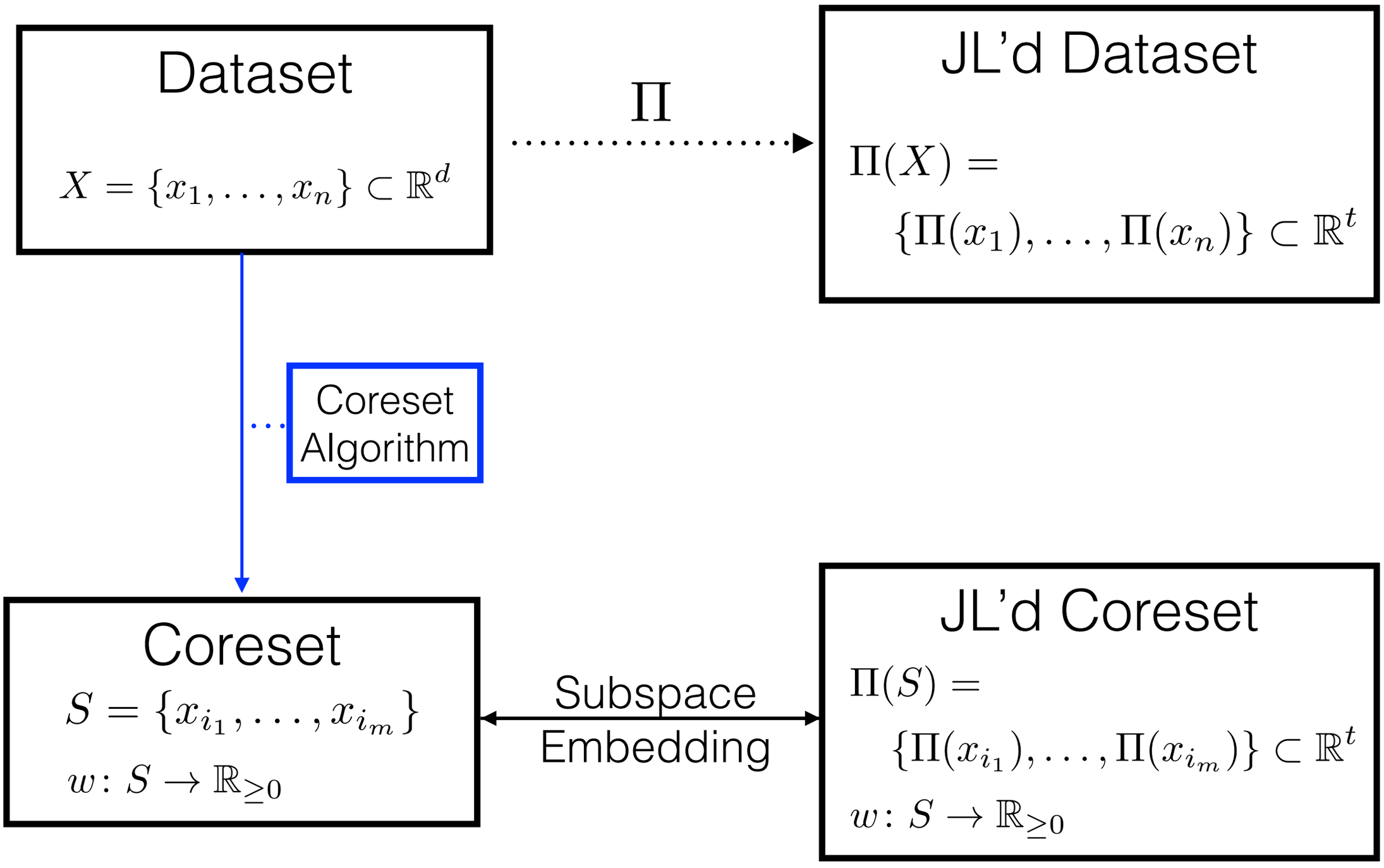
Proof Diagram for using **Coresets** for **Dimension Reduction**



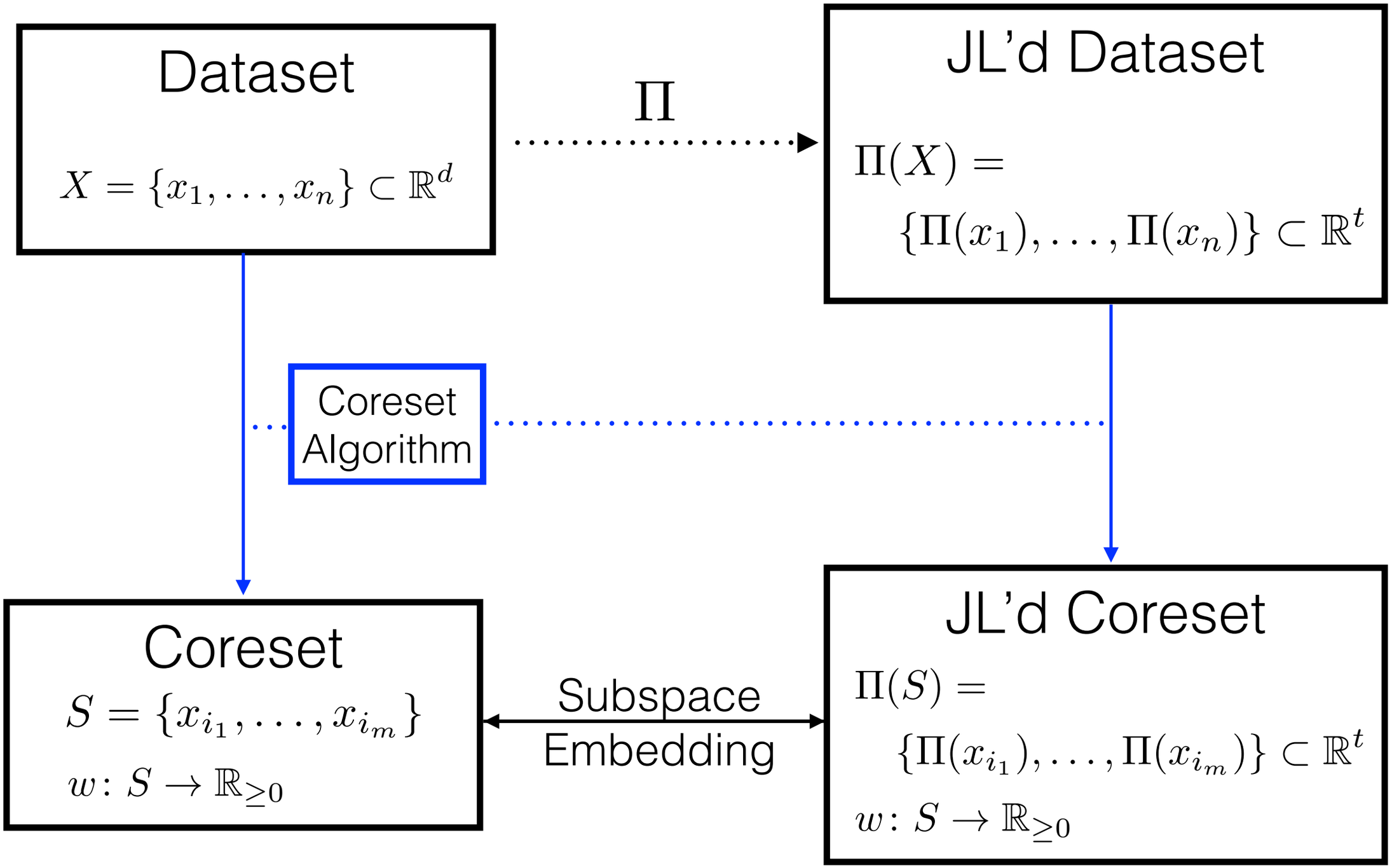
Proof Diagram for using **Coresets** for **Dimension Reduction**



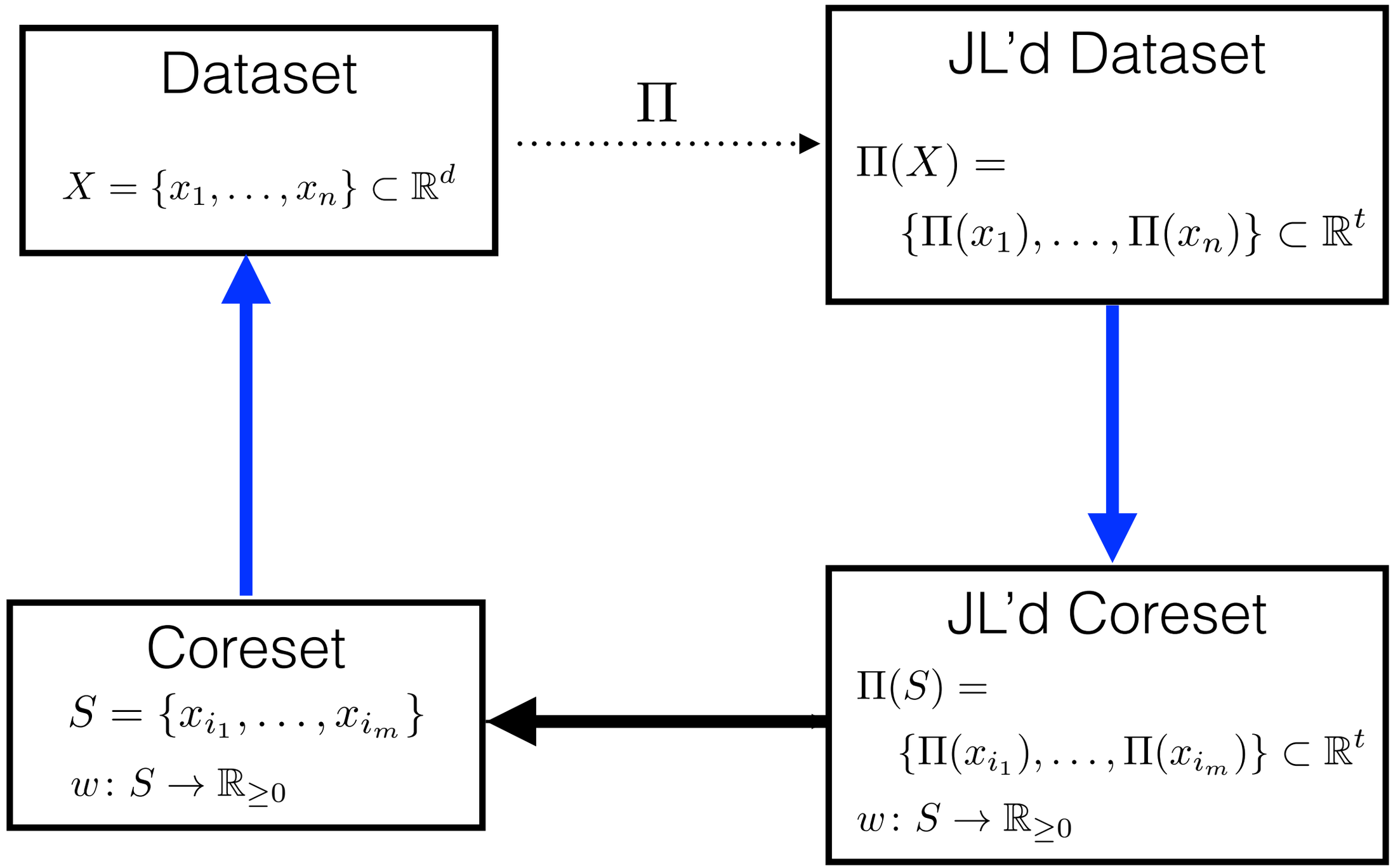
Proof Diagram for using **Coresets** for **Dimension Reduction**



Proof Diagram for using **Coresets** for **Dimension Reduction**

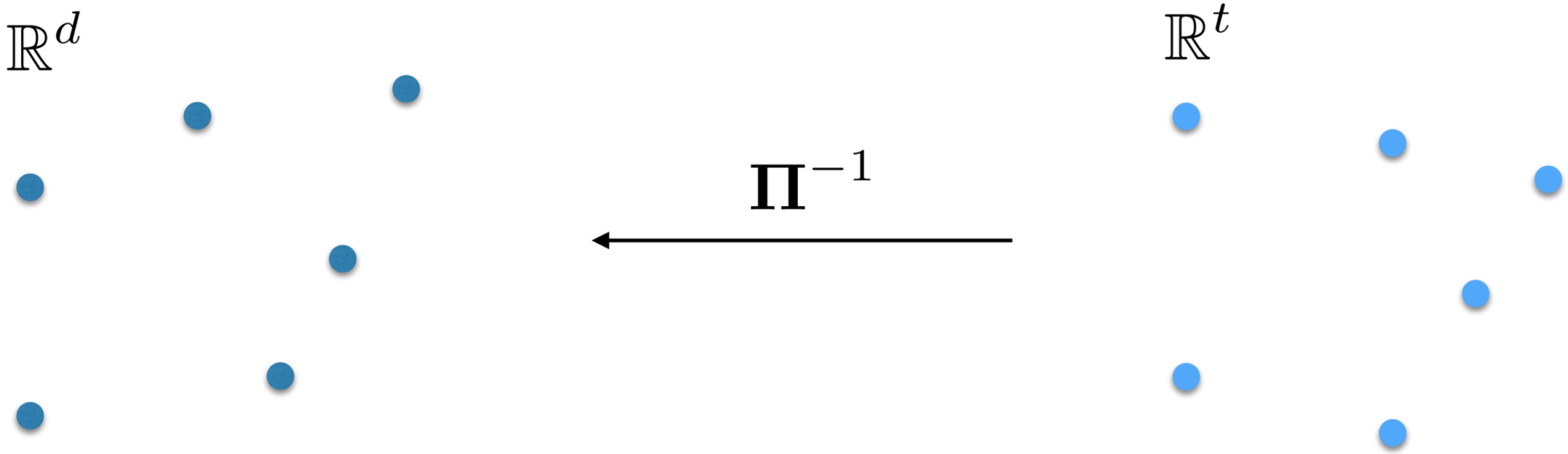


Proof Diagram for using **Coresets** for **Dimension Reduction**

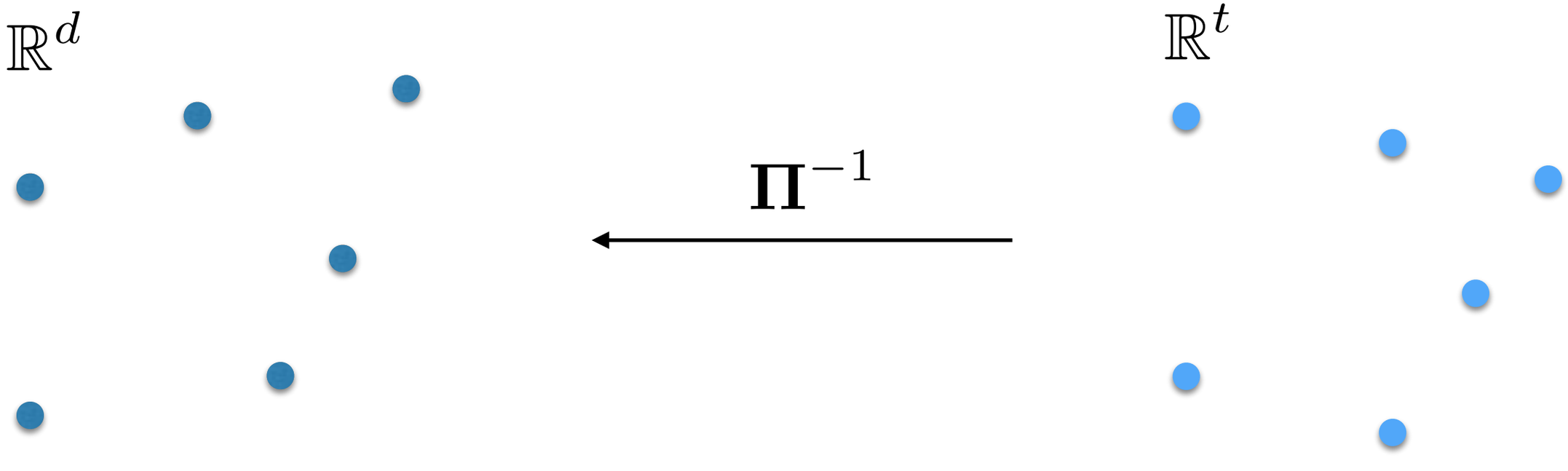


Goal: Follow arrows while only *decreasing* cost

Example: 1-Median — Hard Direction

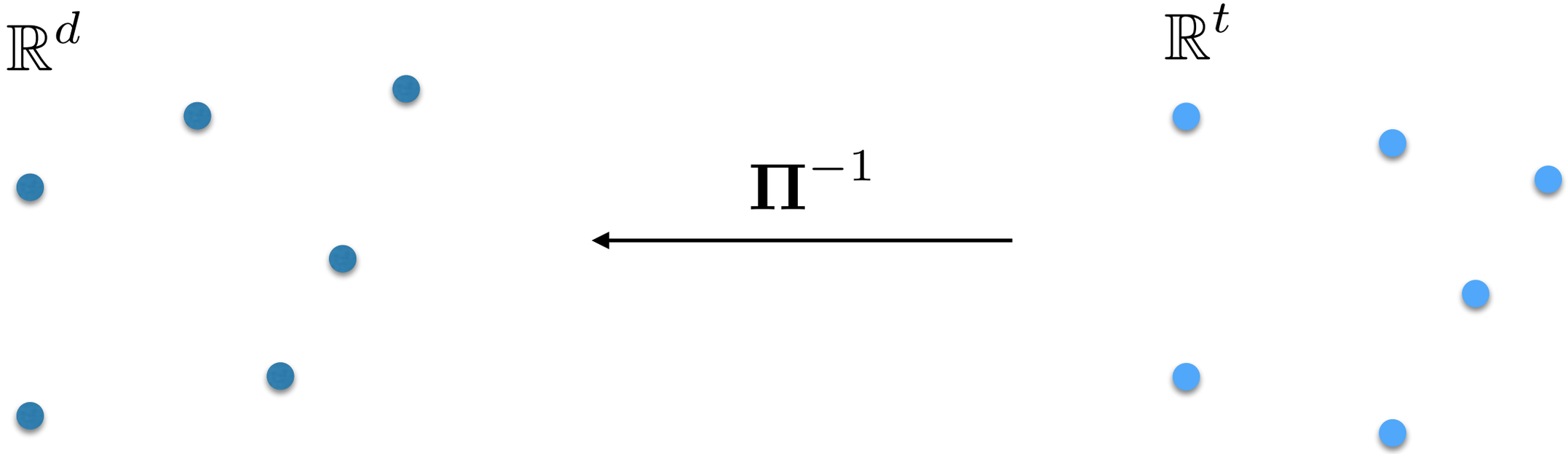


Example: 1-Median — Hard Direction



Success Event:

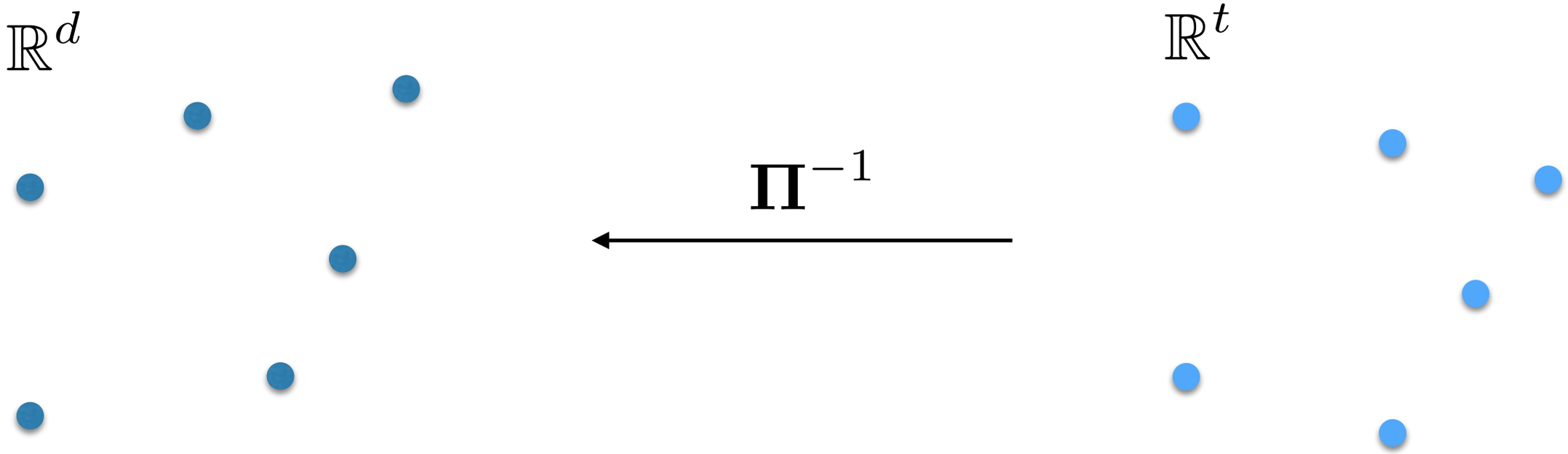
Example: 1-Median — Hard Direction



Success Event:

1. Coreset algorithm produces:
   - (S, w) for $\mathbb{R}^d$
   - $(\mathbf{\Pi}(S), w)$ for $\mathbb{R}^t$

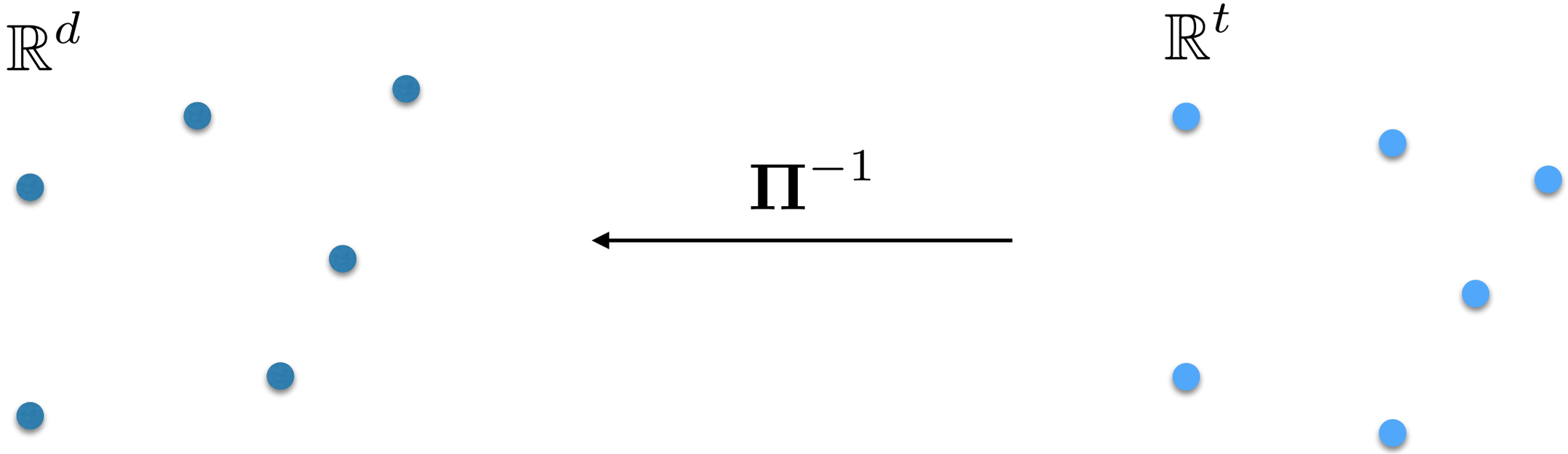
Example: 1-Median — Hard Direction



Success Event:

1. Coreset algorithm produces:
   - (S, w) for $\mathbb{R}^d$
   - $(\mathbf{\Pi}(S), w)$ for $\mathbb{R}^t$
2. JL was a subspace embedding $\mathbf{\Pi}$: $\mathrm{span}(S) \to \mathbb{R}^t$

Example: 1-Median — Hard Direction

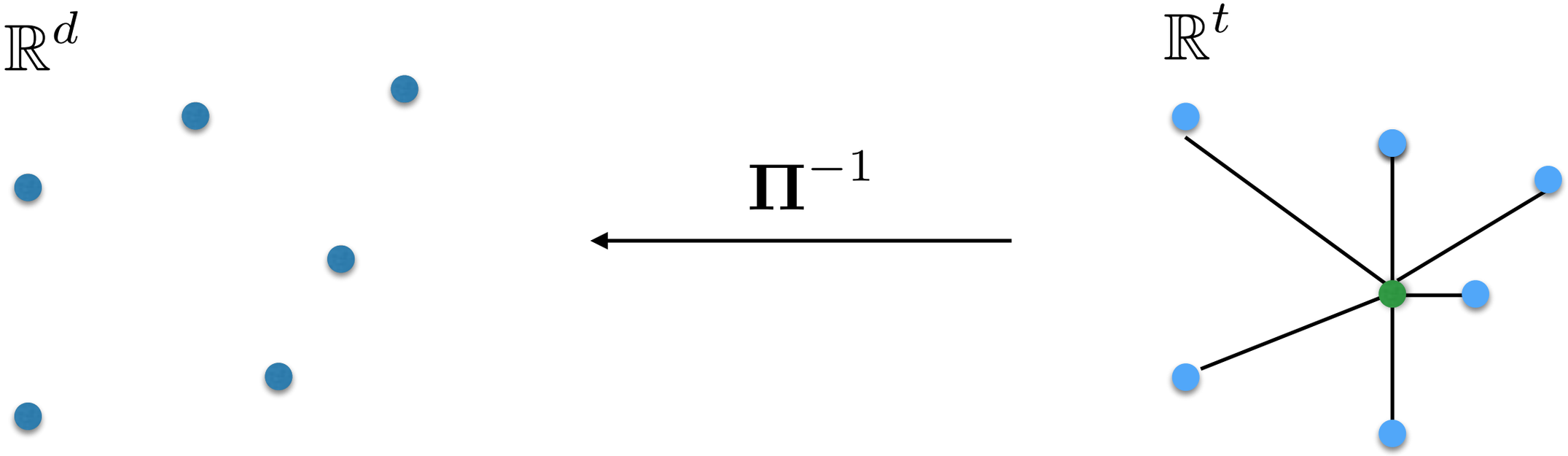


Success Event:

1. Coreset algorithm produces:
   - (S, w) for $\mathbb{R}^d$
   - $(\mathbf{\Pi}(S), w)$ for $\mathbb{R}^t$
2. JL was a subspace embedding $\mathbf{\Pi}$: $\mathrm{span}(S) \to \mathbb{R}^t$

   $t = O(|S|/\epsilon^2)$

Example: 1-Median — Hard Direction

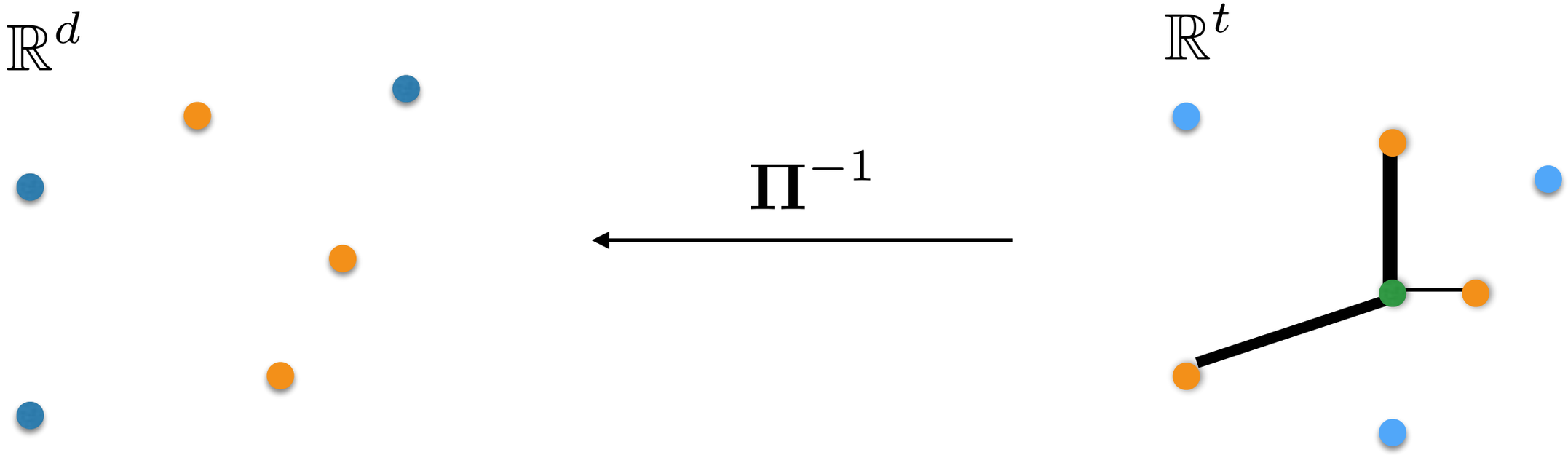


Success Event:

1. Coreset algorithm produces:
   - (S, w) for $\mathbb{R}^d$
   - $(\mathbf{\Pi}(S), w)$ for $\mathbb{R}^t$
2. JL was a subspace embedding $\mathbf{\Pi}$: $\mathrm{span}(S) \to \mathbb{R}^t$

   $t = O(|S|/\epsilon^2)$

Example: 1-Median — Hard Direction

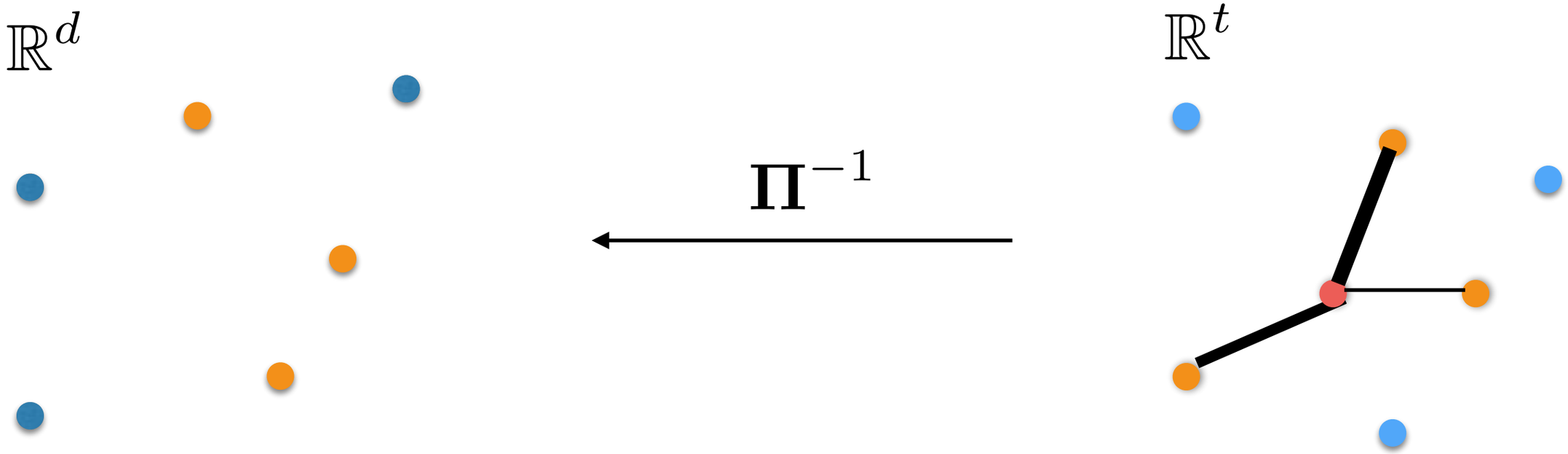


Success Event:

1. Coreset algorithm produces:
   - (S, w) for $\mathbb{R}^d$
   - $(\mathbf{\Pi}(S), w)$ for $\mathbb{R}^t$
2. JL was a subspace embedding $\mathbf{\Pi}$: $\mathrm{span}(S) \to \mathbb{R}^t$

   $t = O(|S|/\epsilon^2)$

Example: 1-Median — Hard Direction

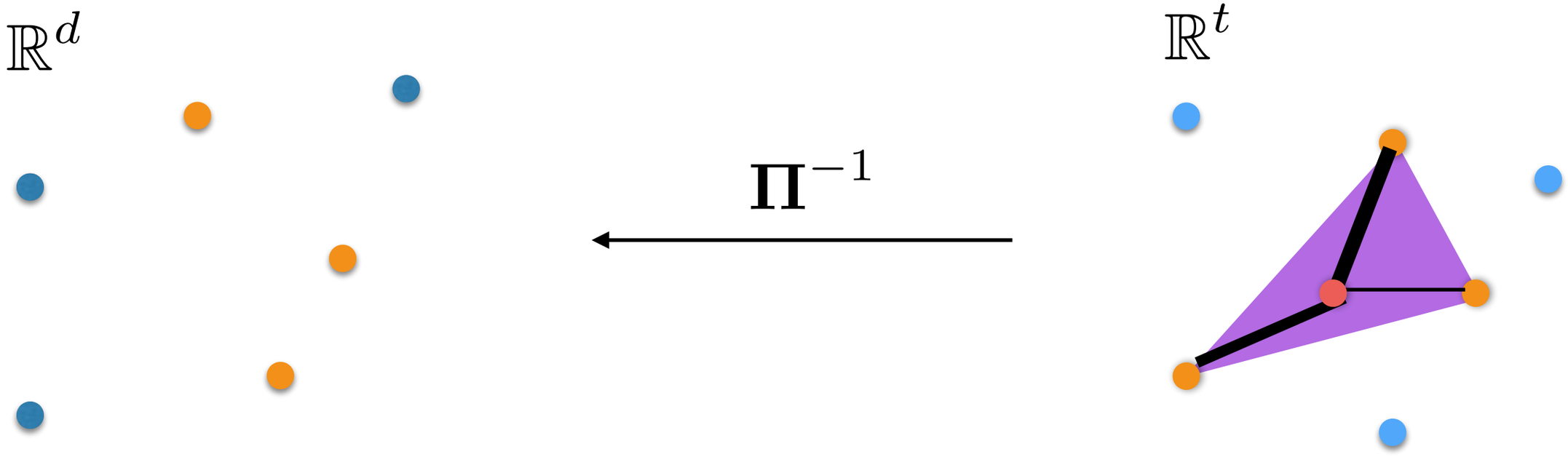


Success Event:

1. Coreset algorithm produces:
   - (S, w) for $\mathbb{R}^d$
   - $(\mathbf{\Pi}(S), w)$ for $\mathbb{R}^t$
2. JL was a subspace embedding $\mathbf{\Pi}$: $\mathrm{span}(S) \to \mathbb{R}^t$

   $t = O(|S|/\epsilon^2)$

Example: 1-Median — Hard Direction

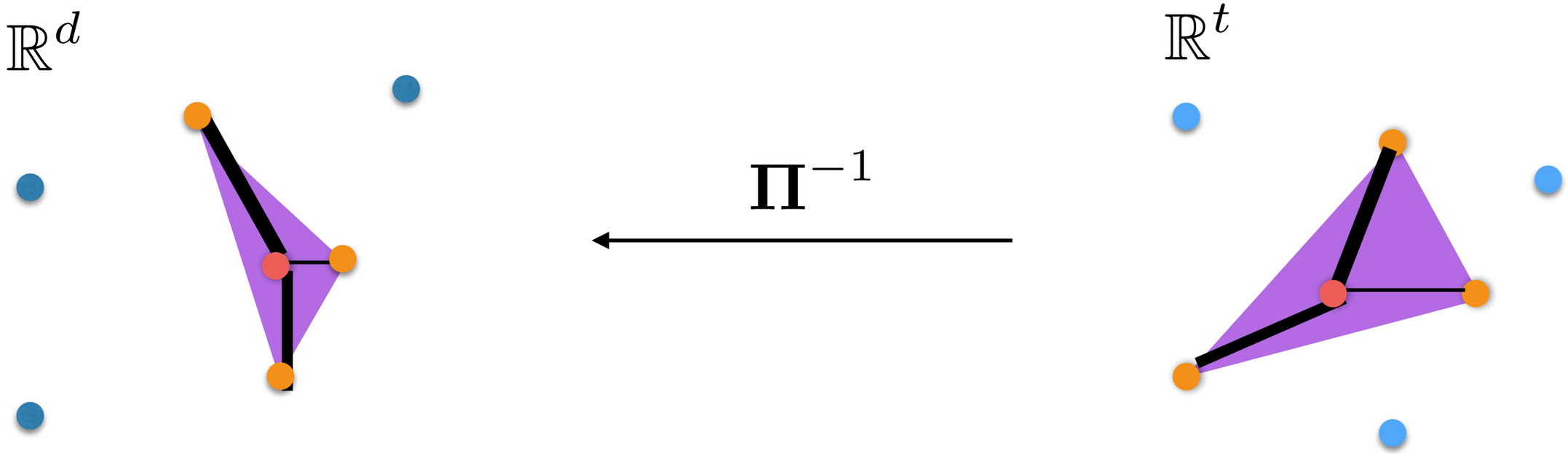


Success Event:

1. Coreset algorithm produces:
   - (S, w) for $\mathbb{R}^d$
   - $(\mathbf{\Pi}(S), w)$ for $\mathbb{R}^t$
2. JL was a subspace embedding $\mathbf{\Pi}$: $\mathrm{span}(S) \to \mathbb{R}^t$

   $t = O(|S|/\epsilon^2)$

Example: 1-Median — Hard Direction

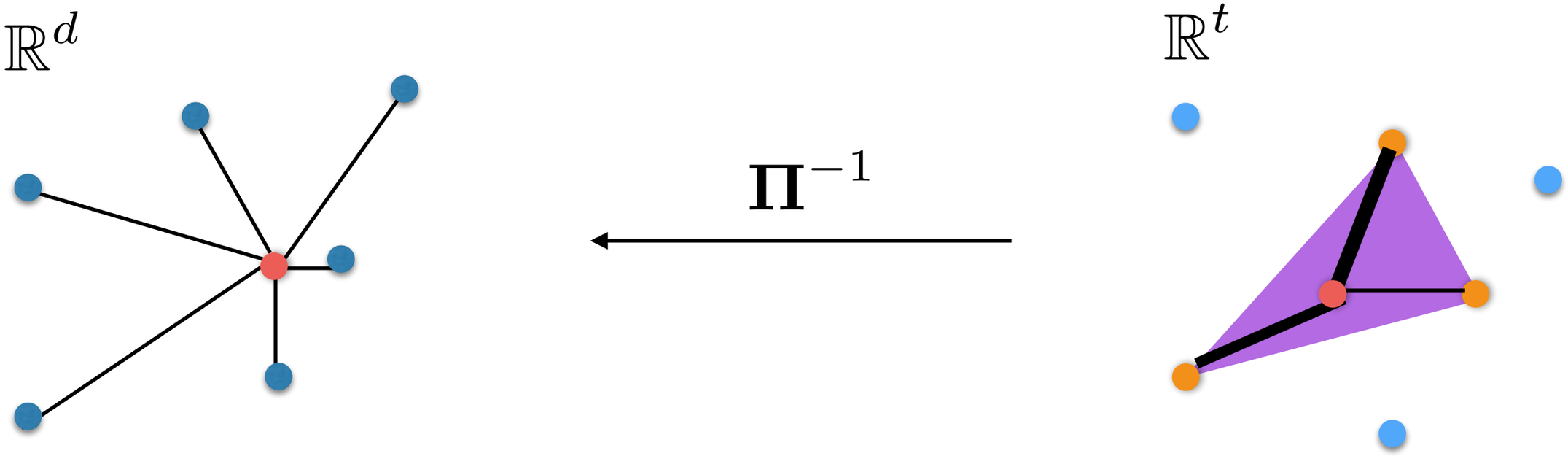


Success Event:

1. Coreset algorithm produces:
   - (S, w) for $\mathbb{R}^d$
   - $(\mathbf{\Pi}(S), w)$ for $\mathbb{R}^t$
2. JL was a subspace embedding $\mathbf{\Pi}$: $\mathrm{span}(S) \to \mathbb{R}^t$

   $t = O(|S|/\epsilon^2)$

Example: 1-Median — Hard Direction

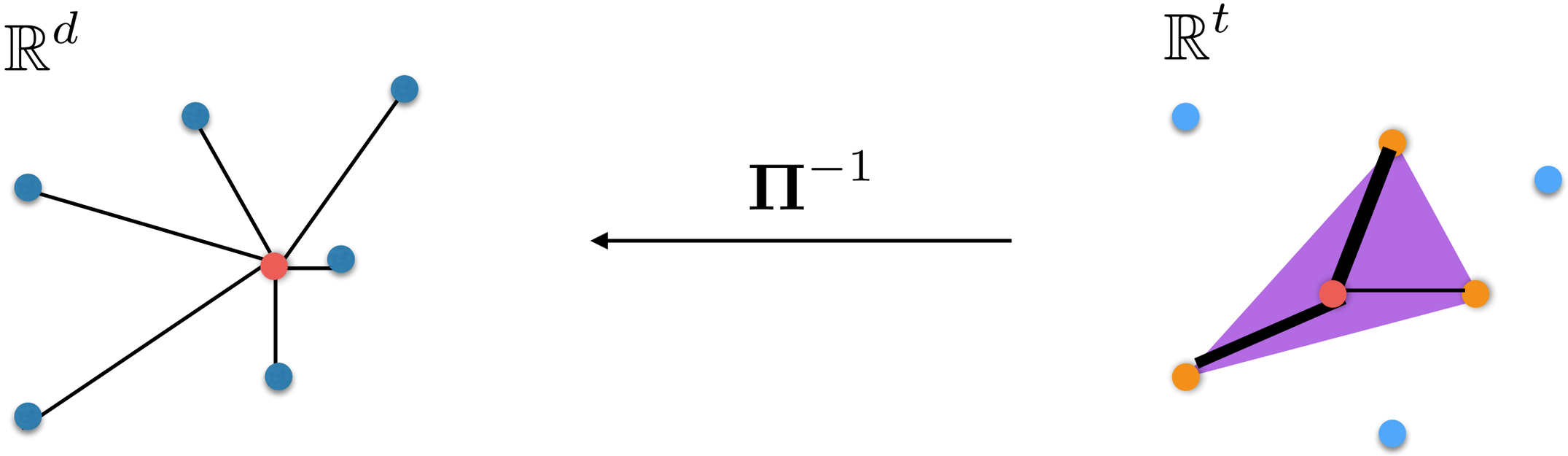


Success Event:

1. Coreset algorithm produces:
   - (S, w) for $\mathbb{R}^d$
   - $(\mathbf{\Pi}(S), w)$ for $\mathbb{R}^t$
2. JL was a subspace embedding $\mathbf{\Pi}$: $\mathrm{span}(S) \to \mathbb{R}^t$

   $t = O(|S|/\epsilon^2)$

Example: 1-Median — Hard Direction

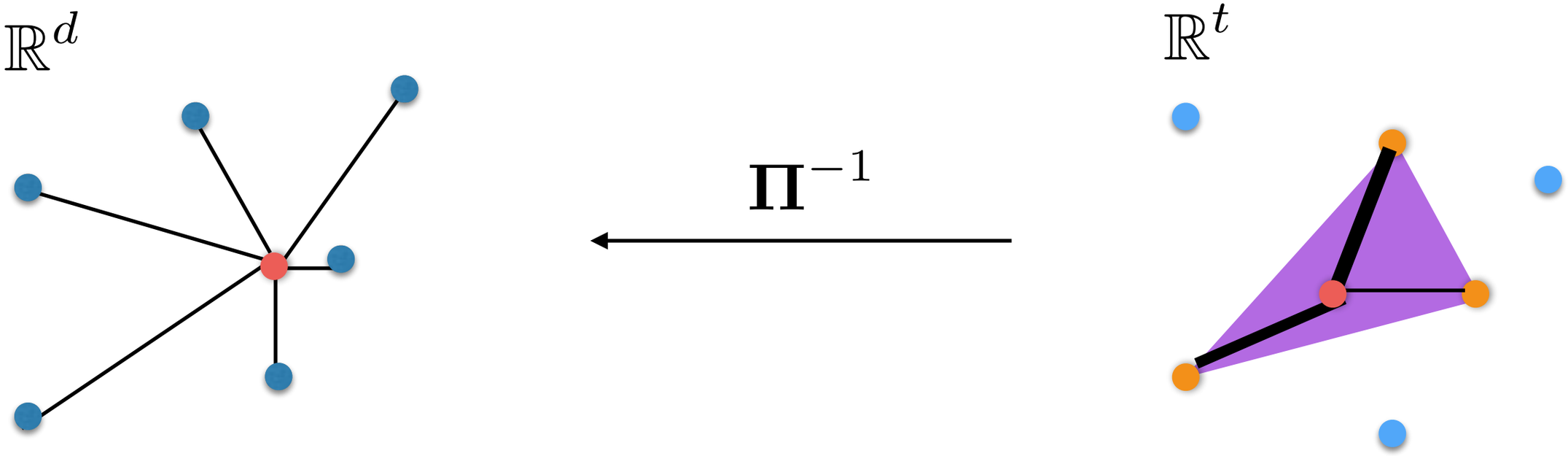


Success Event:

1. Coreset algorithm produces:
   - (S, w) for $\mathbb{R}^d$
   - $(\mathbf{\Pi}(S), w)$ for $\mathbb{R}^t$
2. JL was a subspace embedding $\mathbf{\Pi}$: $\mathrm{span}(S) \to \mathbb{R}^t$

   $t = O(|S|/\epsilon^2)$

Example: 1-Median — Hard Direction



$$\sum_{x \in X} \|\mathbf{\Pi}(x) - y\|_2 \approx \sum_{x \in S} w(x) \|\mathbf{\Pi}(x) - y\|_2$$

for opt y

Success Event:

1. Coreset algorithm produces:
   - (S, w) for $\mathbb{R}^d$
   - ★ $(\mathbf{\Pi}(S), w)$ for $\mathbb{R}^t$
2. JL was a subspace embedding $\mathbf{\Pi}$: $\mathrm{span}(S) \to \mathbb{R}^t$

   $t = O(|S|/\epsilon^2)$

Final Step: Coreset Algorithm which “Commutes”

Final Step: Coreset Algorithm which “Commutes”

Importance Sampling

1. Define sampling distribution.
2. For each fixed center, low-variance estimator.
3. “Union-bound” over all centers.

Final Step: Coreset Algorithm which “Commutes”

Final Step: Coreset Algorithm which “Commutes”

$$\sigma(x) = \sup_{c \in \mathbb{R}^d} \frac{\|x - c\|_2}{\sum_{x' \in X} \|x' - c\|_2}$$

Final Step: Coreset Algorithm which "Commutes"

$$\sigma(x) = \sup_{c \in \mathbb{R}^d} \frac{\|x - c\|_2}{\sum_{x' \in X} \|x' - c\|_2}$$

Sampling distribution $\mathcal{D}$

$$\Pr_{\mathcal{D}}[\text{sample } x] = \frac{\sigma(x)}{T} \quad w(x) = \frac{T}{m\sigma(x)}$$

Final Step: Coreset Algorithm which “Commutes”

$\mathbb{R}^d$

$\mathbb{R}^t$

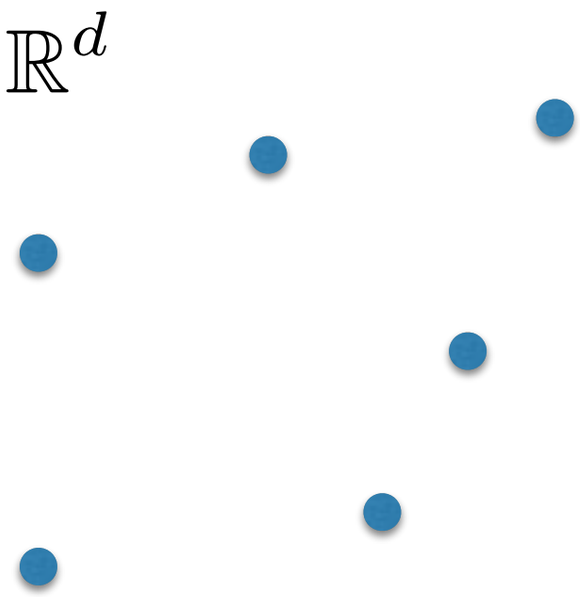
$$\sigma(x) = \sup_{c \in \mathbb{R}^d} \frac{\|x - c\|_2}{\sum_{x' \in X} \|x' - c\|_2}$$

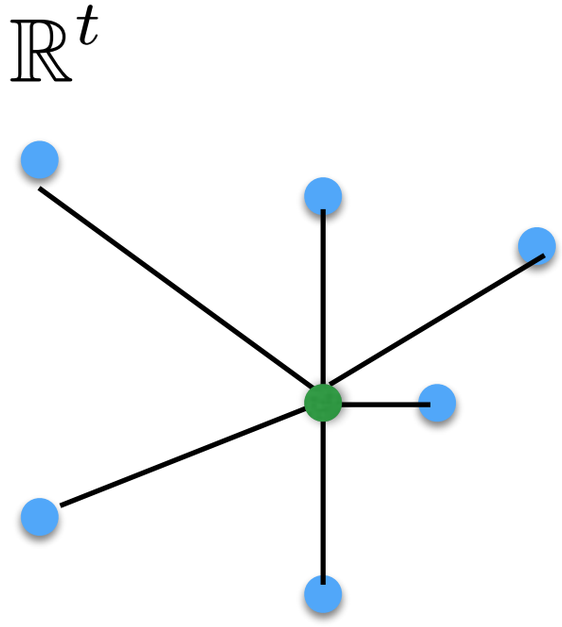
Coreset

Sampling distribution $\mathcal{D}$

$$\Pr_{\mathcal{D}}[\text{sample } x] = \frac{\sigma(x)}{T} \quad w(x) = \frac{T}{m\sigma(x)}$$

Final Step: Coreset Algorithm which “Commutes”





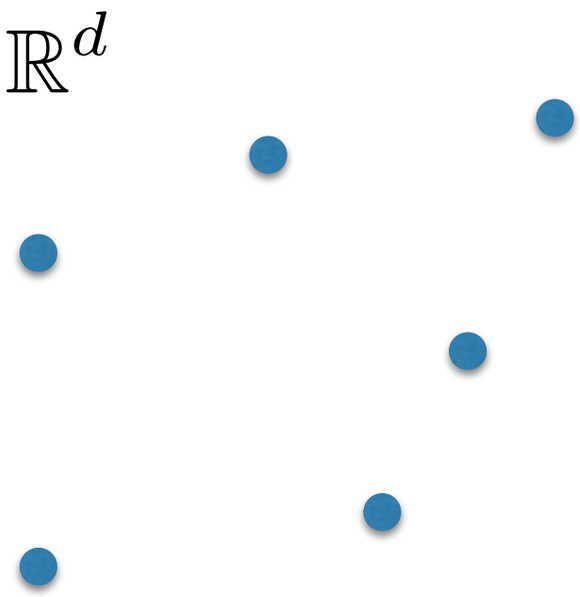
$$\sigma(x) = \sup_{c \in \mathbb{R}^d} \frac{\|x - c\|_2}{\sum_{x' \in X} \|x' - c\|_2}$$

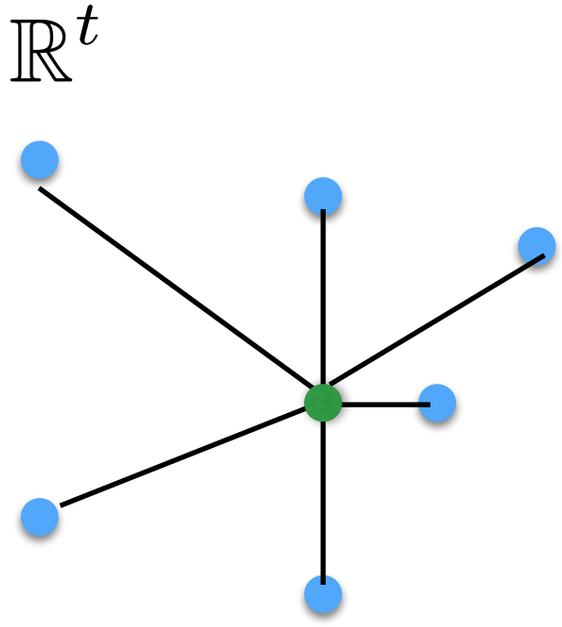
Coreset

Sampling distribution $\mathcal{D}$

$$\Pr_{\mathcal{D}}[\text{sample } x] = \frac{\sigma(x)}{T} \quad w(x) = \frac{T}{m\sigma(x)}$$

Final Step: Coreset Algorithm which “Commutes”



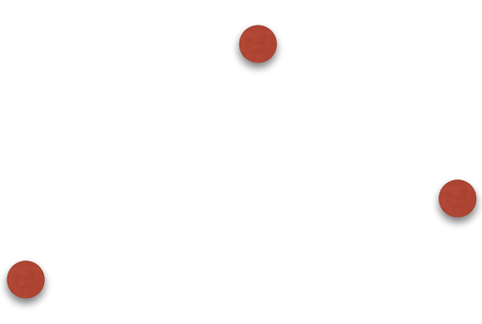


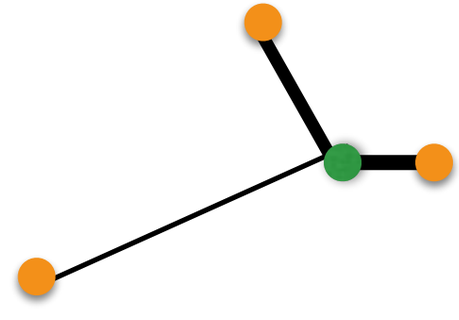
$$\sigma(x) = \sup_{c \in \mathbb{R}^d} \frac{\|x - c\|_2}{\sum_{x' \in X} \|x' - c\|_2}$$

Coreset

Sampling distribution $\mathcal{D}$

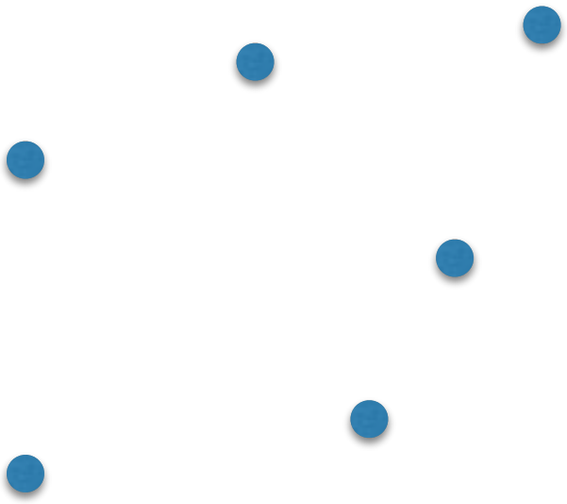
$$\Pr_{\mathcal{D}}[\text{sample } x] = \frac{\sigma(x)}{T} \quad w(x) = \frac{T}{m\sigma(x)}$$



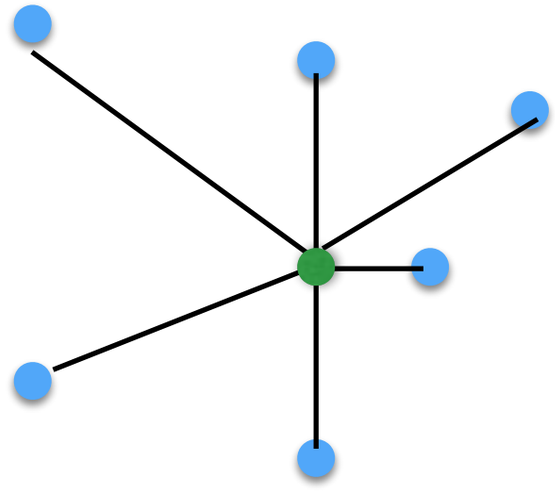


Final Step: Coreset Algorithm which “Commutes”

$\mathbb{R}^d$



$\mathbb{R}^t$



$$\sigma(x) = \sup_{c \in \mathbb{R}^d} \frac{\|x - c\|_2}{\sum_{x' \in X} \|x' - c\|_2}$$

$$\sum_{x \in X} \|\mathbf{\Pi}(x) - y\|_2$$

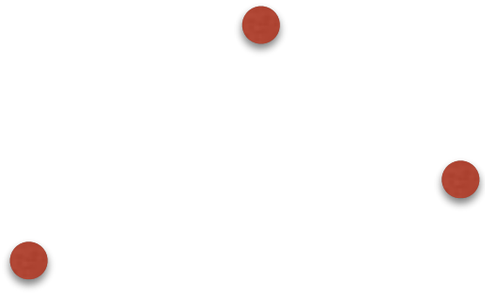
Coreset

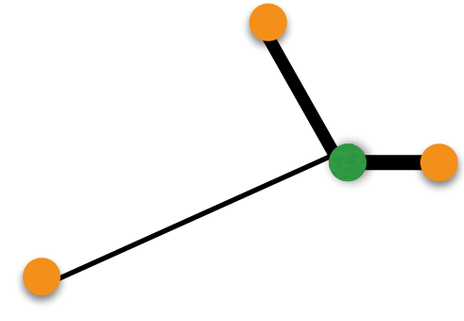
Sampling distribution $\mathcal{D}$

$$\Pr_{\mathcal{D}}[\text{sample } x] = \frac{\sigma(x)}{T} \quad w(x) = \frac{T}{m\sigma(x)}$$

||

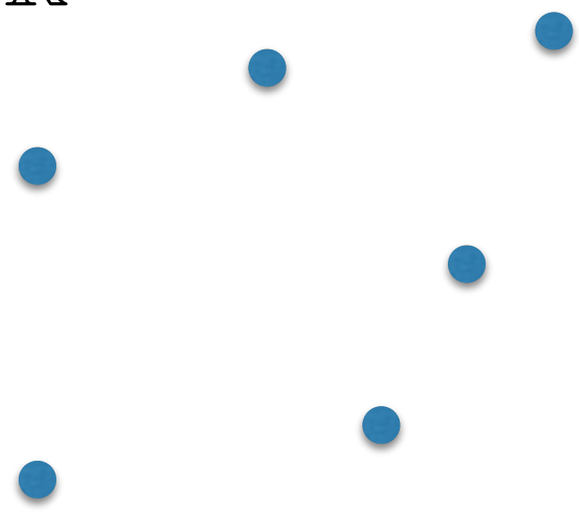
$$\mathbb{E}\left[\sum_{x \in \text{coreset}} w(x)\|\mathbf{\Pi}(x) - y\|_2\right]$$



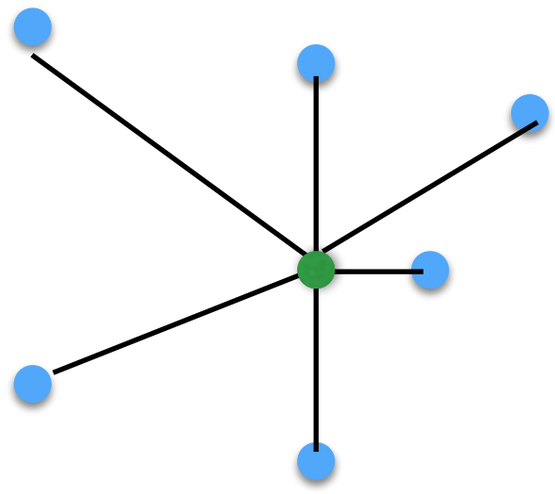


Final Step: Coreset Algorithm which “Commutes”

$\mathbb{R}^d$



$\mathbb{R}^t$



$$\sigma(x) = \sup_{c \in \mathbb{R}^d} \frac{\|x - c\|_2}{\sum_{x' \in X} \|x' - c\|_2}$$

$$\sum_{x \in X} \|\mathbf{\Pi}(x) - y\|_2$$

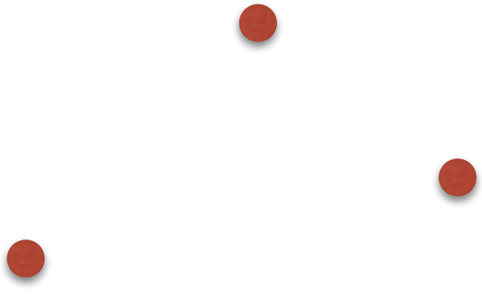
Coreset

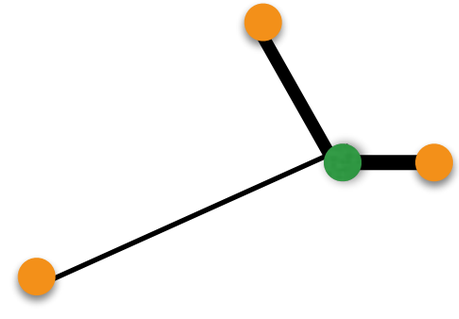
Sampling distribution $\mathcal{D}$

$$\Pr_{\mathcal{D}}[\text{sample } x] = \frac{\sigma(x)}{T} \quad w(x) = \frac{T}{m\sigma(x)}$$

$\vee\wr$

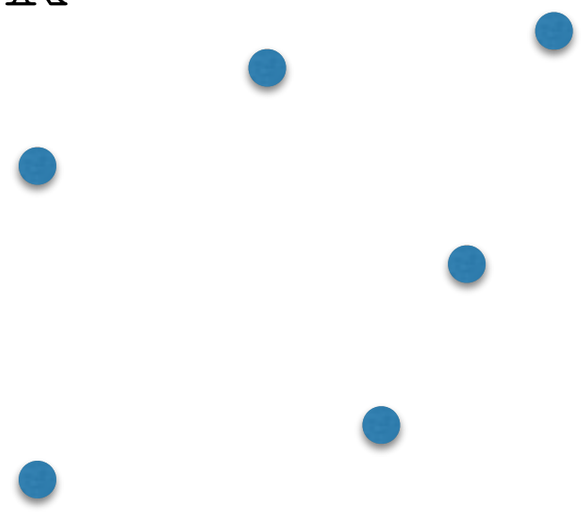
$$\sum_{x \in \text{coreset}} w(x)\|\mathbf{\Pi}(x) - y\|_2$$





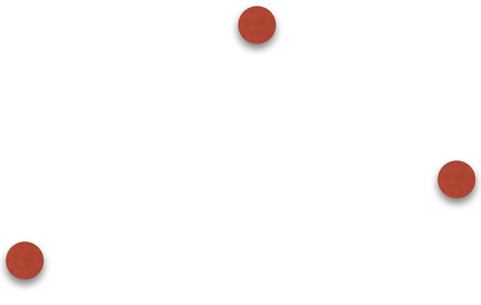
Final Step: Coreset Algorithm which "Commutes"

$\mathbb{R}^d$

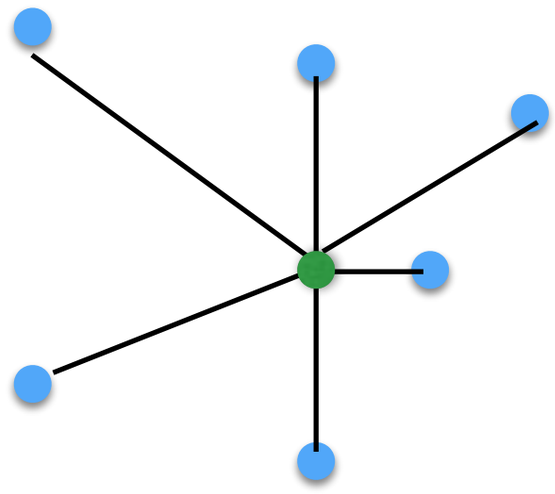


$$\sigma(x) = \sup_{c \in \mathbb{R}^d} \frac{\|x - c\|_2}{\sum_{x' \in X} \|x' - c\|_2}$$

Coreset



$\mathbb{R}^t$



$$\sum_{x \in X} \|\mathbf{\Pi}(x) - y\|_2$$

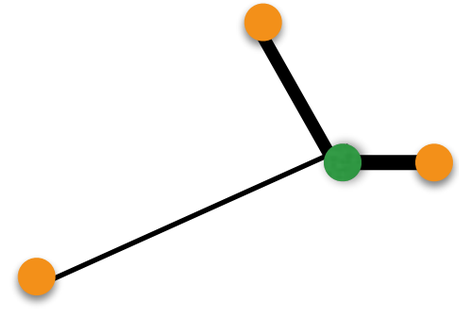
$$\wr\vee$$

$$\sum_{x \in \text{coreset}} w(x)\|\mathbf{\Pi}(x) - y\|_2$$

Sampling distribution $\mathcal{D}$

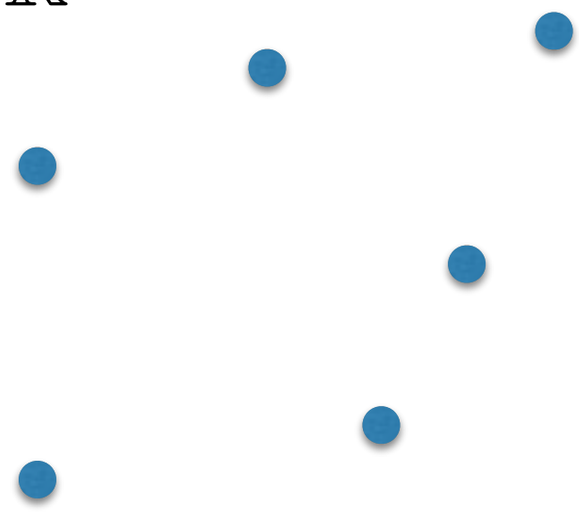
$$\Pr_{\mathcal{D}}[\text{sample } x] = \frac{\sigma(x)}{T} \quad w(x) = \frac{T}{m\sigma(x)}$$

JL preserves cost up to $O(1)$ w.p 0.9



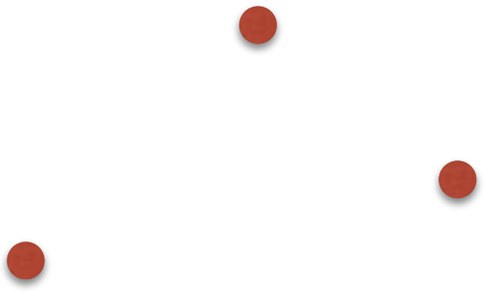
Final Step: Coreset Algorithm which "Commutes"

$\mathbb{R}^d$



$$\sigma(x) = \sup_{c \in \mathbb{R}^d} \frac{\|x - c\|_2}{\sum_{x' \in X} \|x' - c\|_2}$$

Coreset

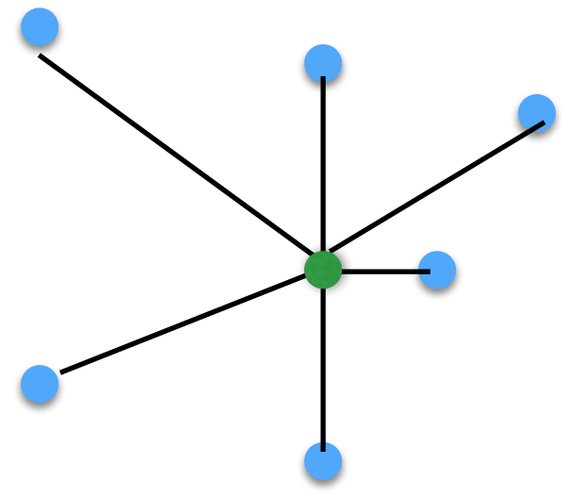


Sampling distribution $\mathcal{D}$

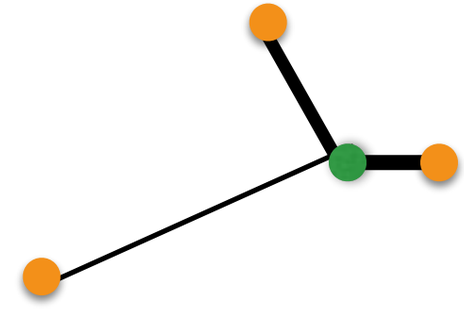
$$\Pr_{\mathcal{D}}[\text{sample } x] = \frac{\sigma(x)}{T} \quad w(x) = \frac{T}{m\sigma(x)}$$

JL preserves cost up to $O(1)$ w.p 0.9

$\mathbb{R}^t$

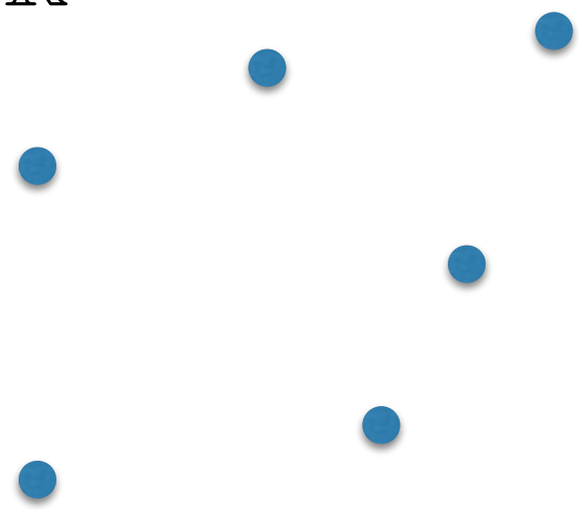


$$\sum_{x \in X} \|\mathbf{\Pi}(x) - y\|_2 \;\wr\; \sum_{x \in \text{coreset}} w(x)\|\mathbf{\Pi}(x) - y\|_2$$



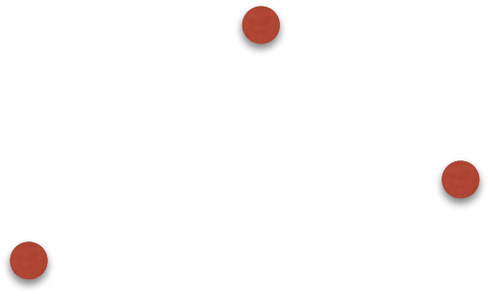
Final Step: Coreset Algorithm which "Commutes"

$\mathbb{R}^d$



$$\sigma(x) = \sup_{c \in \mathbb{R}^d} \frac{\|x - c\|_2}{\sum_{x' \in X} \|x' - c\|_2}$$

Coreset

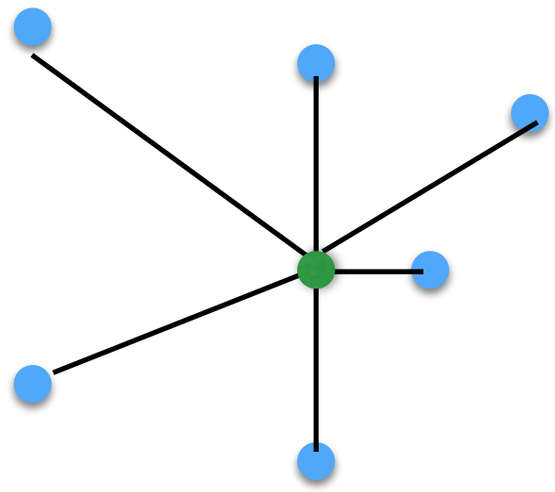


Sampling distribution $\mathcal{D}$

$$\Pr_{\mathcal{D}}[\text{sample } x] = \frac{\sigma(x)}{T} \quad w(x) = \frac{T}{m\sigma(x)}$$

JL preserves cost up to $O(1)$ w.p 0.9

$\mathbb{R}^t$

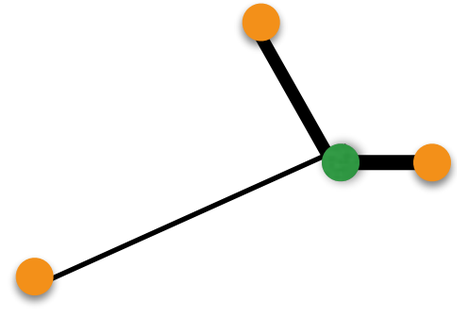


$$\sum_{x \in X} \|\mathbf{\Pi}(x) - y\|_2$$

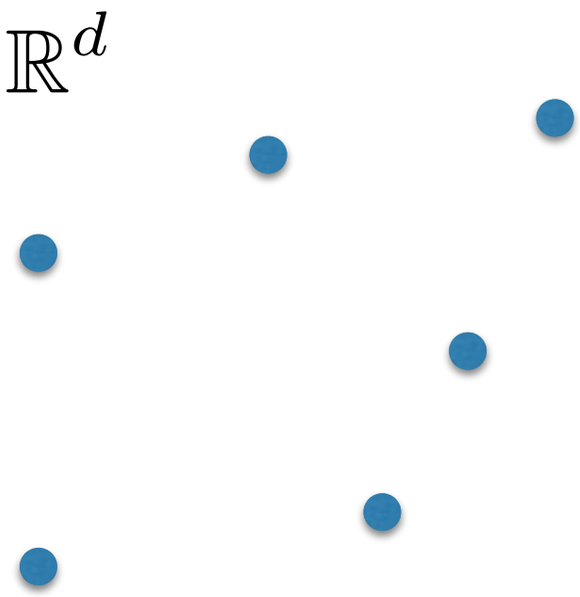
$$\times(1+\epsilon)$$

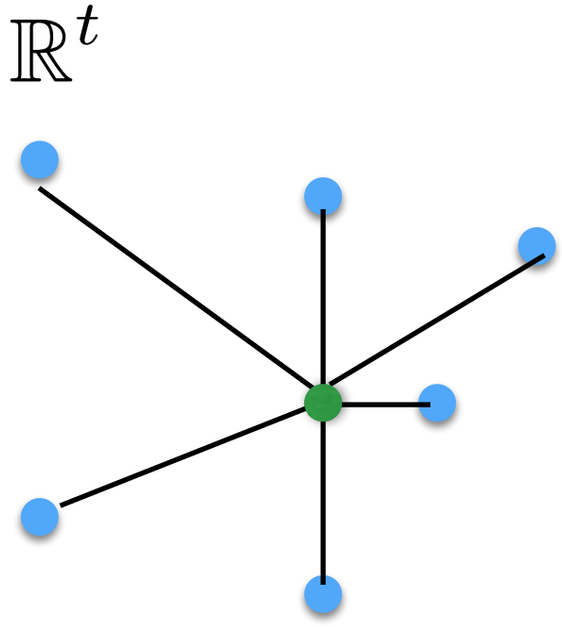
$$\vee|$$

$$\sum_{x \in \text{coreset}} w(x)\|\mathbf{\Pi}(x) - y\|_2$$



Final Step: Coreset Algorithm which "Commutes"



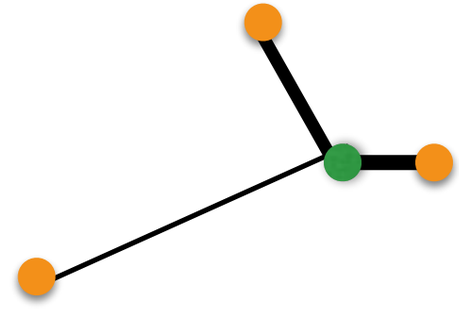


$$\sigma(x) = \sup_{c \in \mathbb{R}^d} \frac{\|x - c\|_2}{\sum_{x' \in X} \|x' - c\|_2}$$

Sampling distribution $\mathcal{D}$

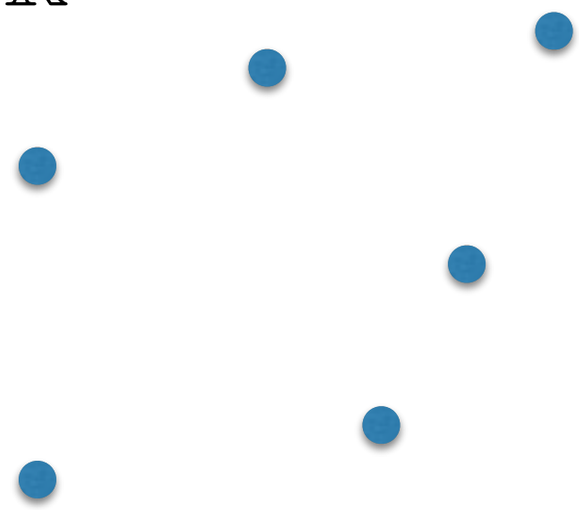
$$\Pr_{\mathcal{D}}[\text{sample } x] = \frac{\sigma(x)}{T} \quad w(x) = \frac{T}{m\sigma(x)}$$

JL preserves cost up to $O(1)$ w.p 0.9



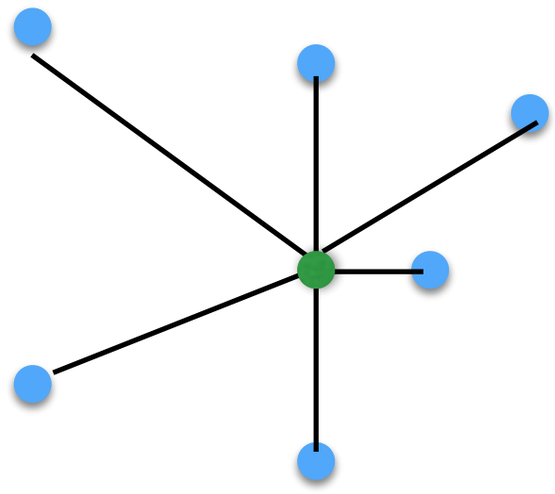
Final Step: Coreset Algorithm which "Commutes"

$\mathbb{R}^d$



$$\sigma(x) = \sup_{c \in \mathbb{R}^d} \frac{\|x - c\|_2}{\sum_{x' \in X} \|x' - c\|_2}$$

$\mathbb{R}^t$

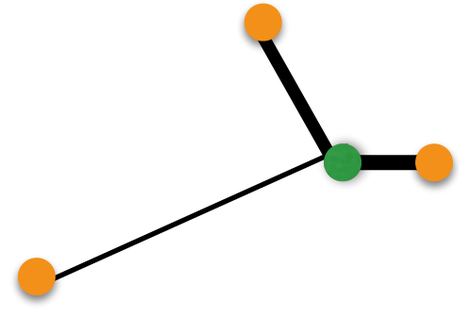


$$\sigma'(\mathbf{\Pi}(x)) = \sup_{y \in \mathbb{R}^t} \frac{\|\mathbf{\Pi}(x) - y\|_2}{\sum_{x' \in X} \|\mathbf{\Pi}(x') - y\|_2}$$

Sampling distribution $\mathcal{D}$

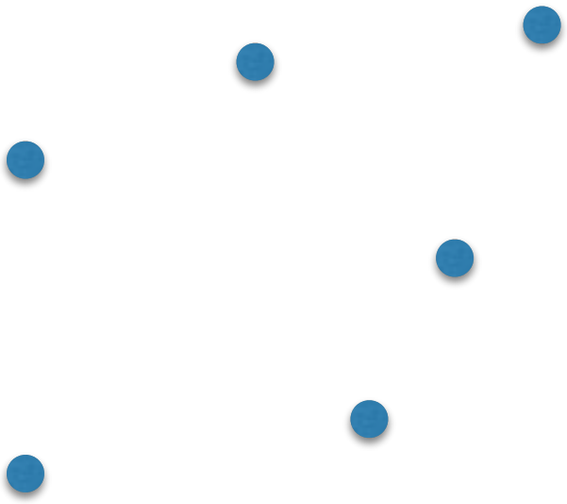
$$\Pr_{\mathcal{D}}[\text{sample } x] = \frac{\sigma(x)}{T} \quad w(x) = \frac{T}{m\sigma(x)}$$

JL preserves cost up to $O(1)$ w.p 0.9



Final Step: Coreset Algorithm which "Commutes"

$\mathbb{R}^d$



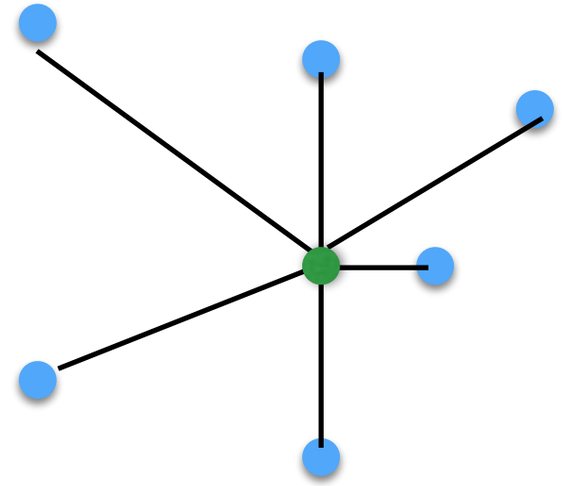
$$\sigma(x) = \sup_{c \in \mathbb{R}^d} \frac{\|x - c\|_2}{\sum_{x' \in X} \|x' - c\|_2}$$

Sampling distribution $\mathcal{D}$

$$\Pr_{\mathcal{D}}[\text{sample } x] = \frac{\sigma(x)}{T} \quad w(x) = \frac{T}{m\sigma(x)}$$

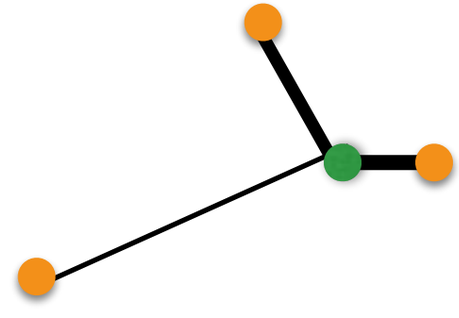
JL preserves cost up to $O(1)$ w.p 0.9

$\mathbb{R}^t$

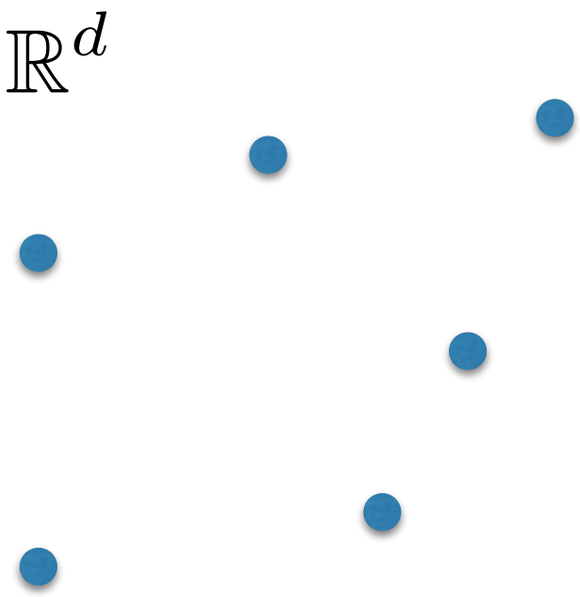


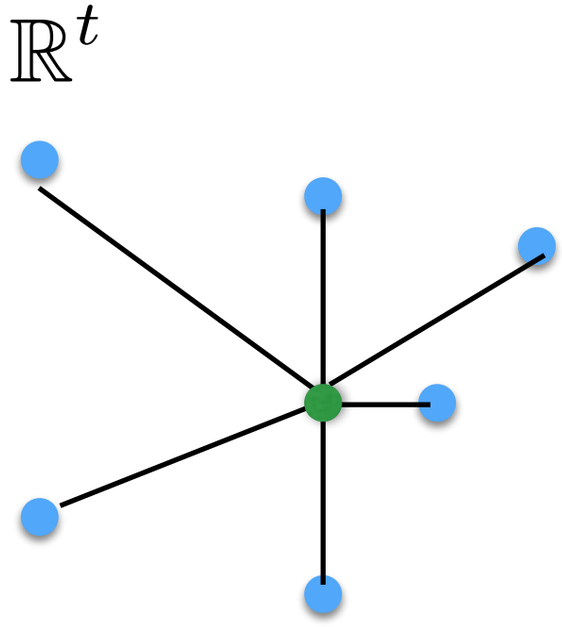
$$\sigma'(\mathbf{\Pi}(x)) = \sup_{y \in \mathbb{R}^t} \frac{\|\mathbf{\Pi}(x) - y\|_2}{\sum_{x' \in X} \|\mathbf{\Pi}(x') - y\|_2}$$

$$\mathop{\mathbb{E}}_{\mathbf{\Pi}, x}\left[\frac{\sigma'(x)}{\sigma(x)}\right] \leq O(1).$$



Final Step: Coreset Algorithm which "Commutes"





$$\sigma(x) = \sup_{c \in \mathbb{R}^d} \frac{\|x - c\|_2}{\sum_{x' \in X} \|x' - c\|_2}$$

$$\sigma'(\mathbf{\Pi}(x)) = \sup_{y \in \mathbb{R}^t} \frac{\|\mathbf{\Pi}(x) - y\|_2}{\sum_{x' \in X} \|\mathbf{\Pi}(x') - y\|_2}$$

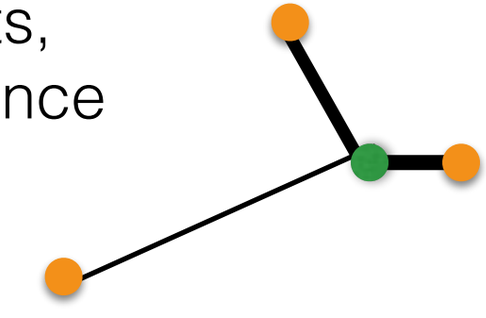
$$\mathop{\mathbb{E}}_{\mathbf{\Pi}, x} \left[\frac{\sigma'(x)}{\sigma(x)} \right] \leq O(1).$$

Sampling distribution $\mathcal{D}$

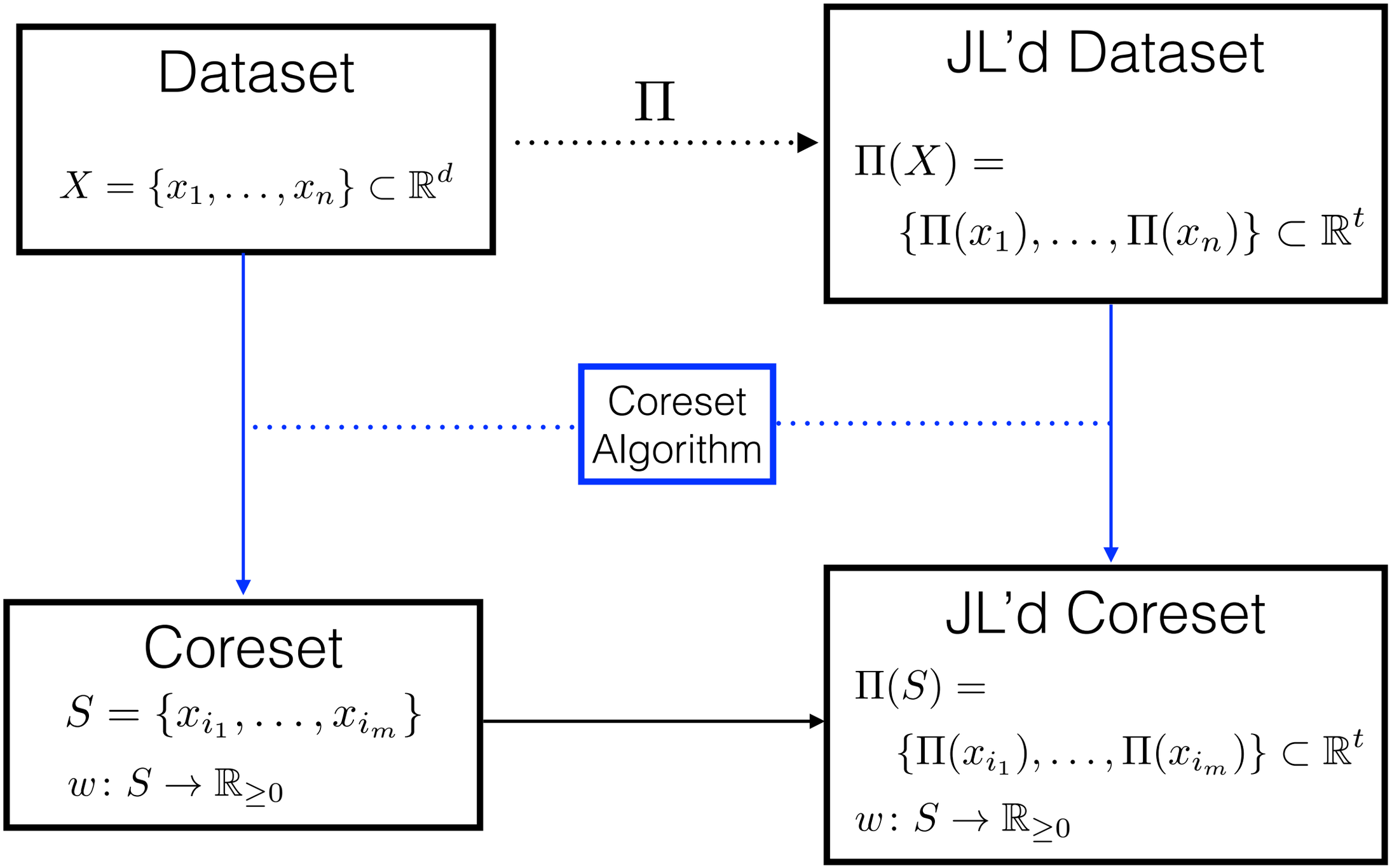
$$\Pr_{\mathcal{D}}[\text{sample } x] = \frac{\sigma(x)}{T} \quad w(x) = \frac{T}{m\sigma(x)}$$

JL preserves cost up to $O(1)$ w.p 0.9

Sample $O(1)$ more coreset points, decrease variance



Summary:
Coresets for **Dimension Reduction**



Some Open Problems

1. "For-all" vs "Optimal" Guarantees?

$$\min_{c_1,\ldots,c_k \in \mathbb{R}^d} \mathrm{cost}(X, \{c_1, \ldots, c_k\}) \approx \min_{c'_1,\ldots,c'_k \in \mathbb{R}^t} \mathrm{cost}(\mathbf{\Pi}(X), \{c'_1, \ldots, c'_k\})$$

[MMR '19, BBCGS'19]:

$$\forall (S_1, \ldots, S_k) \subset X : \mathrm{cost}(X; S_1, \ldots, S_k) \approx \mathrm{cost}(\mathbf{\Pi}(X), S_1, \ldots, S_k)$$

2. Avoiding the "curse of dimensionality:"

- clustering, subspace approximation,
- facility location, single-linkage clustering [Narayanan, Silwal, Indyk, Zamir '21]
- Wasserstein barycenters [Izzo, Silwal, Zhou '21]
- … ?