Fast Algorithms for a New Relaxation of OT

Erik Waingarten (UPenn)

Moses Charikar (Stanford) Beidi Chen (FAIR -> CMU) Chris Re (Stanford)

“Non-Locality” of Geometric Problems makes them difficult.

Input:

- Specify two sets of vectors: $\{x_1, \dots, x_n\}, \{y_1, \dots, y_n\} \subset \mathbb{R}^d$
- “High-dimensional regime:” $\Omega(\log n) \leq d \leq n^{o(1)}$

Closest Pair	Chamfer (Sum of NNs)	EMD (min-cost matching)
$\min_{i,j \in [n]} \|x_i - y_j\|_2$	$\min \sum_{i=1}^{n} \min_{j \in [n]} \|x_i - y_j\|_2$	$\min_{\pi} \sum_{i=1}^{n} \|x_i - y_{\pi(i)}\|_2$

“Non-Locality” of Geometric Problems makes them difficult.

$$\gamma \geq 0$$

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{ij} \|x_i - y_j\|_2$$

Closest Pair	Chamfer (Sum of NNs)	EMD (min-cost matching)
$\sum_{i,j} \gamma_{ij} = 1$	$\forall i : \sum_j \gamma_{ij} = 1$	$\forall i : \sum_j \gamma_{ij} = 1$ $\forall j : \sum_i \gamma_{ij} = 1$

“Non-Locality” of Geometric Problems makes them difficult.

$$\gamma \geq 0$$

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{ij} \|x_i - y_j\|_2$$

Approximate accurately in near-linear time?

Closest Pair	Chamfer (Sum of NNs)	EMD (min-cost matching)
$\sum_{i,j} \gamma_{ij} = 1$	$\forall i : \sum_{j} \gamma_{ij} = 1$	$\forall i : \sum_{j} \gamma_{ij} = 1$ $\forall j : \sum_{i} \gamma_{ij} = 1$

"Non-Locality" of Geometric Problems makes them difficult.

$$\gamma \geq 0$$

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{ij} \|x_i - y_j\|_2$$

Approximate accurately in near-linear time?

Closest Pair

$$\sum_{i,j} \gamma_{ij} = 1$$

Hard

Chamfer

(Sum of NNs)

$$\forall i : \sum_{j} \gamma_{ij} = 1$$

Hard and Easy

BIJSW'23

EMD

(min-cost matching)

$$\forall i : \sum_{j} \gamma_{ij} = 1$$

$$\forall j : \sum_{i} \gamma_{ij} = 1$$

Hard and ?

R'19

Talk Outline:

- “Spanner approach”
- Sinkhorn distances
- Main Result

The "Spanner Approach"

Theorem [Har-Peled, Indyk, Sidiropolous'13]:

c-approximate spanners in time $n^{1+O(1/c^2)}$

$$d_G(x,z) \overset{c}{\approx} \|x-z\|_2 \qquad \forall x,z \in X$$

Geometric Spanners ⟷ Approximate Nearest Neighbors

Time complexity $\approx$ $\tilde{O}(1)\,(n \cdot T(n) + S(n))$

Geometric Spanners ⟷ Approximate Nearest Neighbors

Time complexity $\approx$ $\tilde{O}(1)\,(n \cdot T(n) + S(n))$

Conceptual Challenge:
"Accurate" approximations essentially take quadratic time.

Optimal transport,
Euclidean MST,
Closest Pair, …

except for kernel density estimation.

EMD ⟷ Sinkhorn

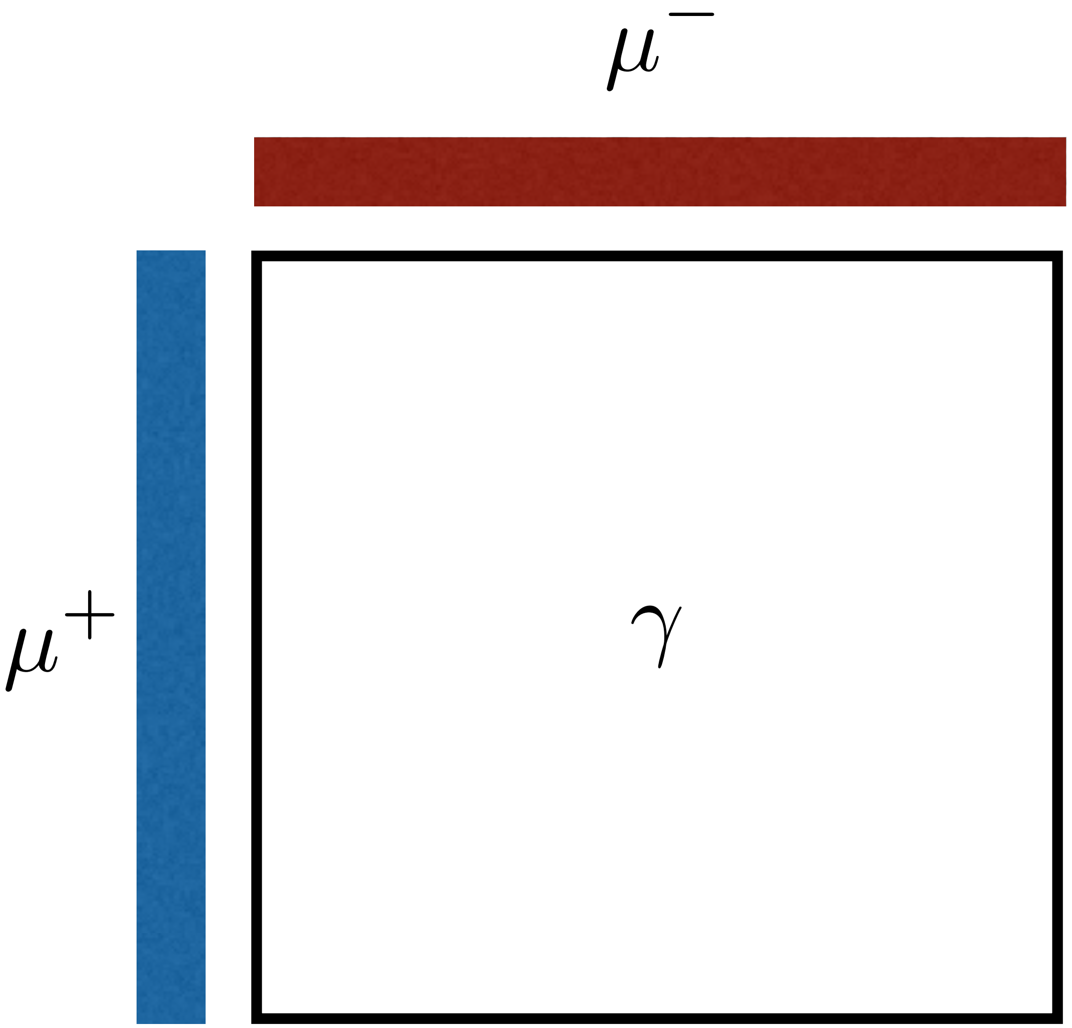
Fast algorithm
crude approx

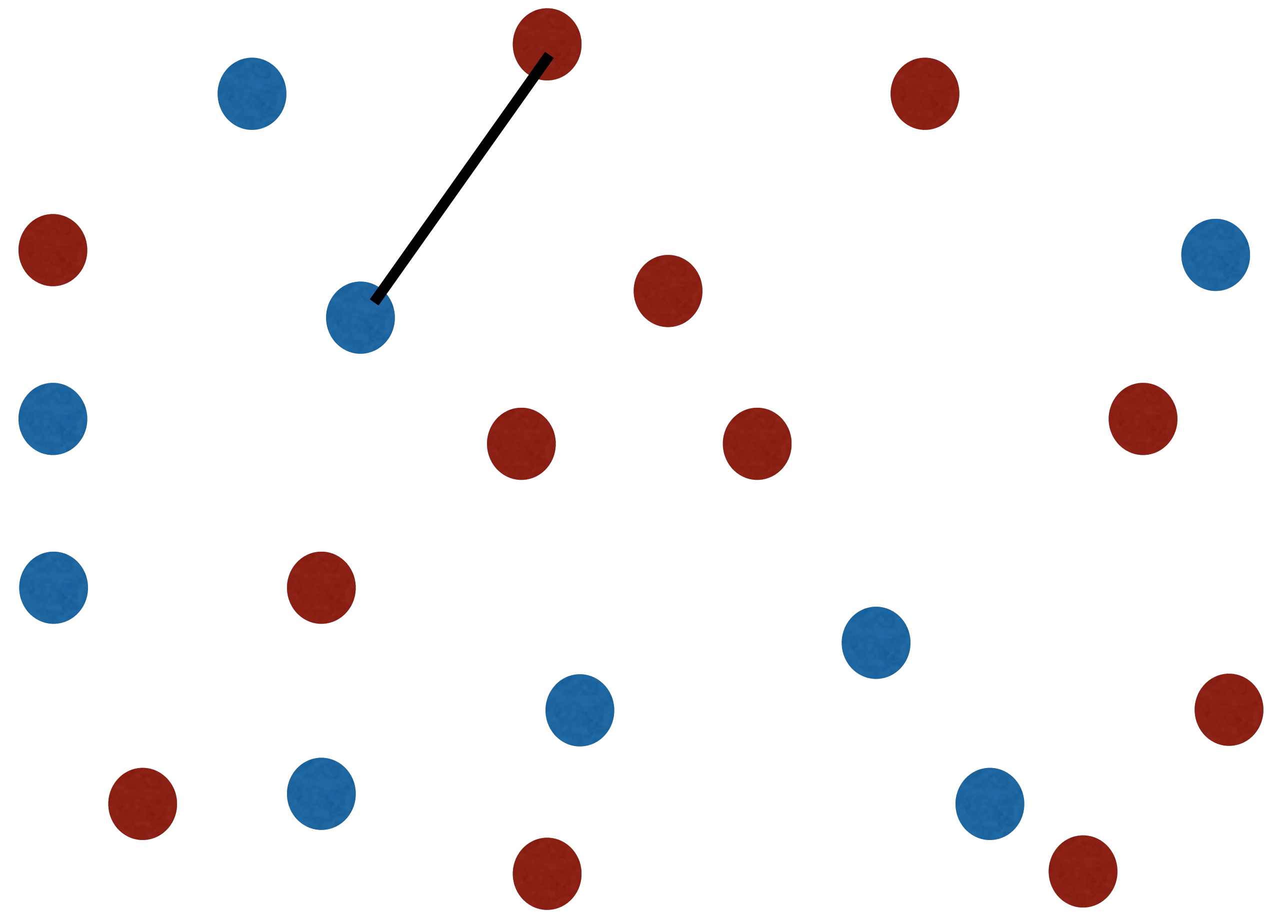
Slow algorithm
accurate additive
approx

Motivating Questions:

1. Can we break away from spanner approach?
2. Which transports approximate accurately?

Main Idea: Focus on a slight perturbation to EMD.

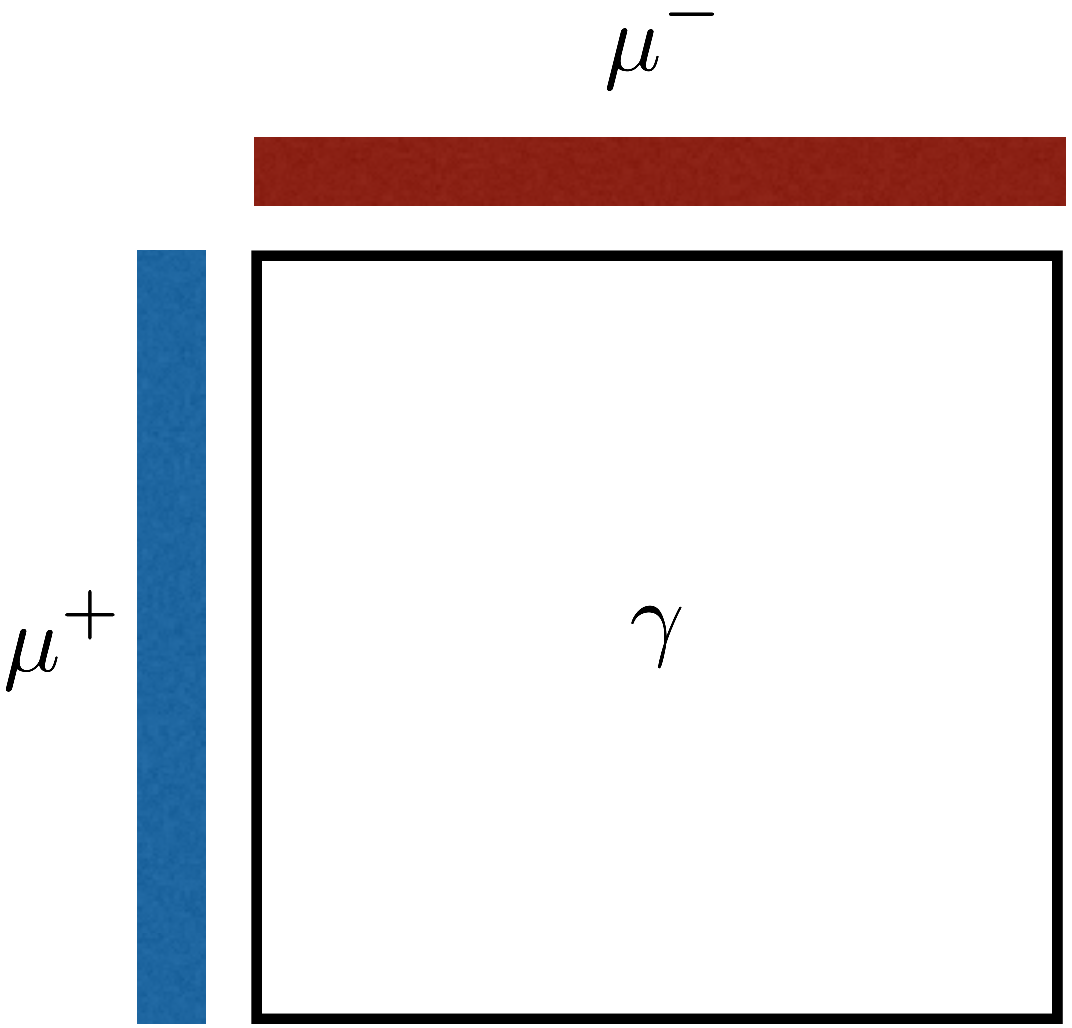


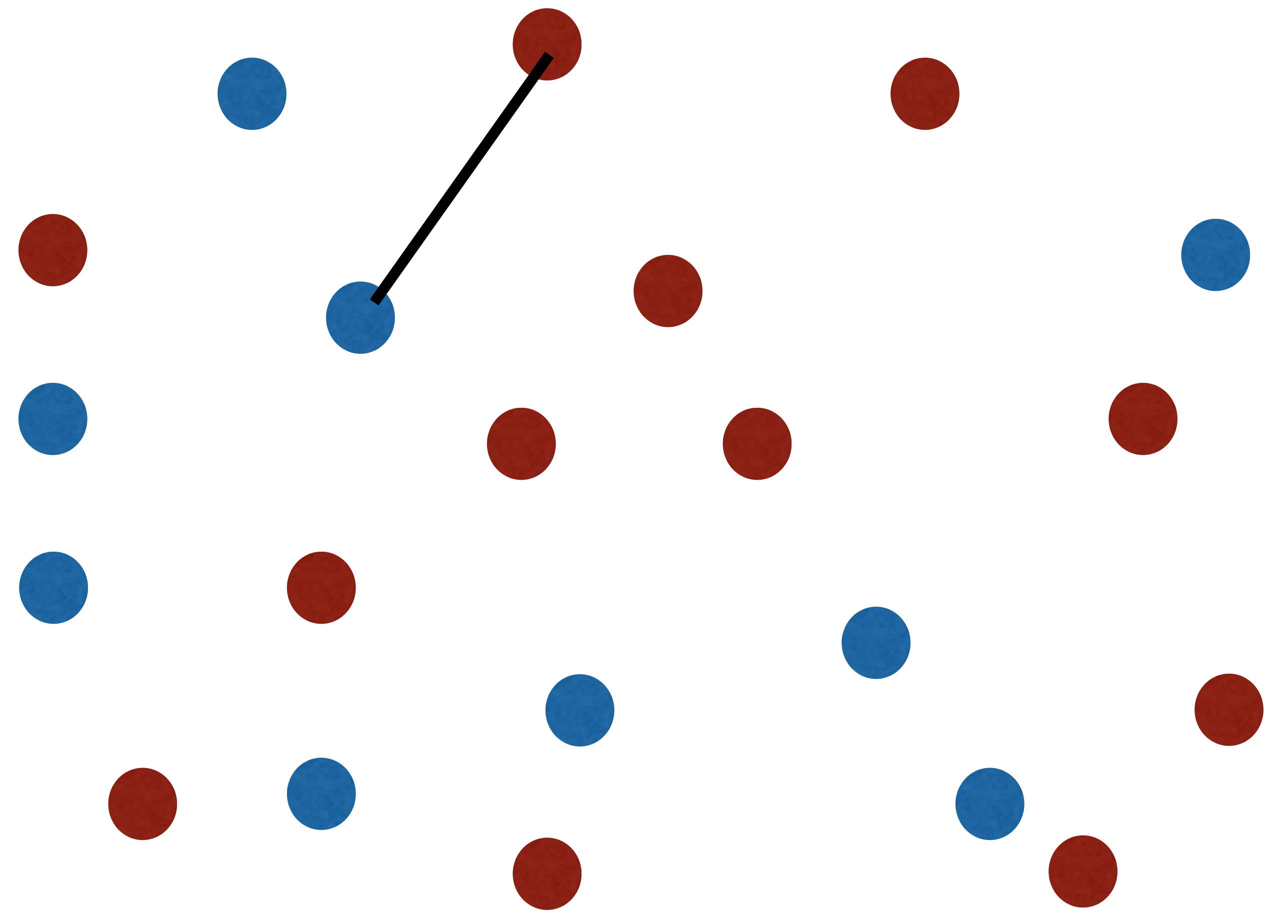


$$\mathrm{cost}(\gamma)_{ij} = \gamma_{ij}\|x_i - x_j\|_2$$

$$\mathtt{EMD}(\mu) = \min_{\gamma} \|\mathrm{cost}(\gamma)\|_1$$

Main Idea: Focus on a slight perturbation to EMD.





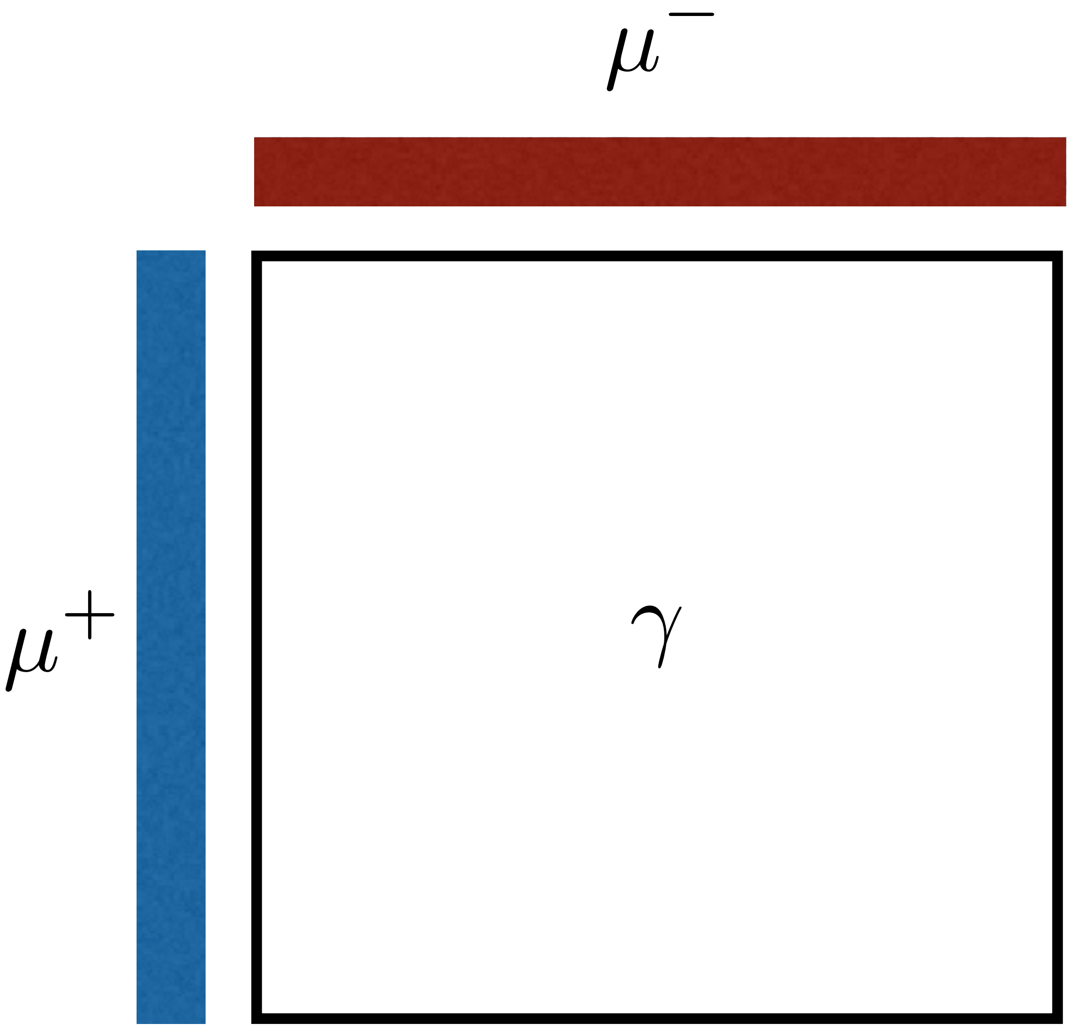
$$\mathrm{cost}(\gamma)_{ij} = \gamma_{ij} \|x_i - x_j\|_2$$

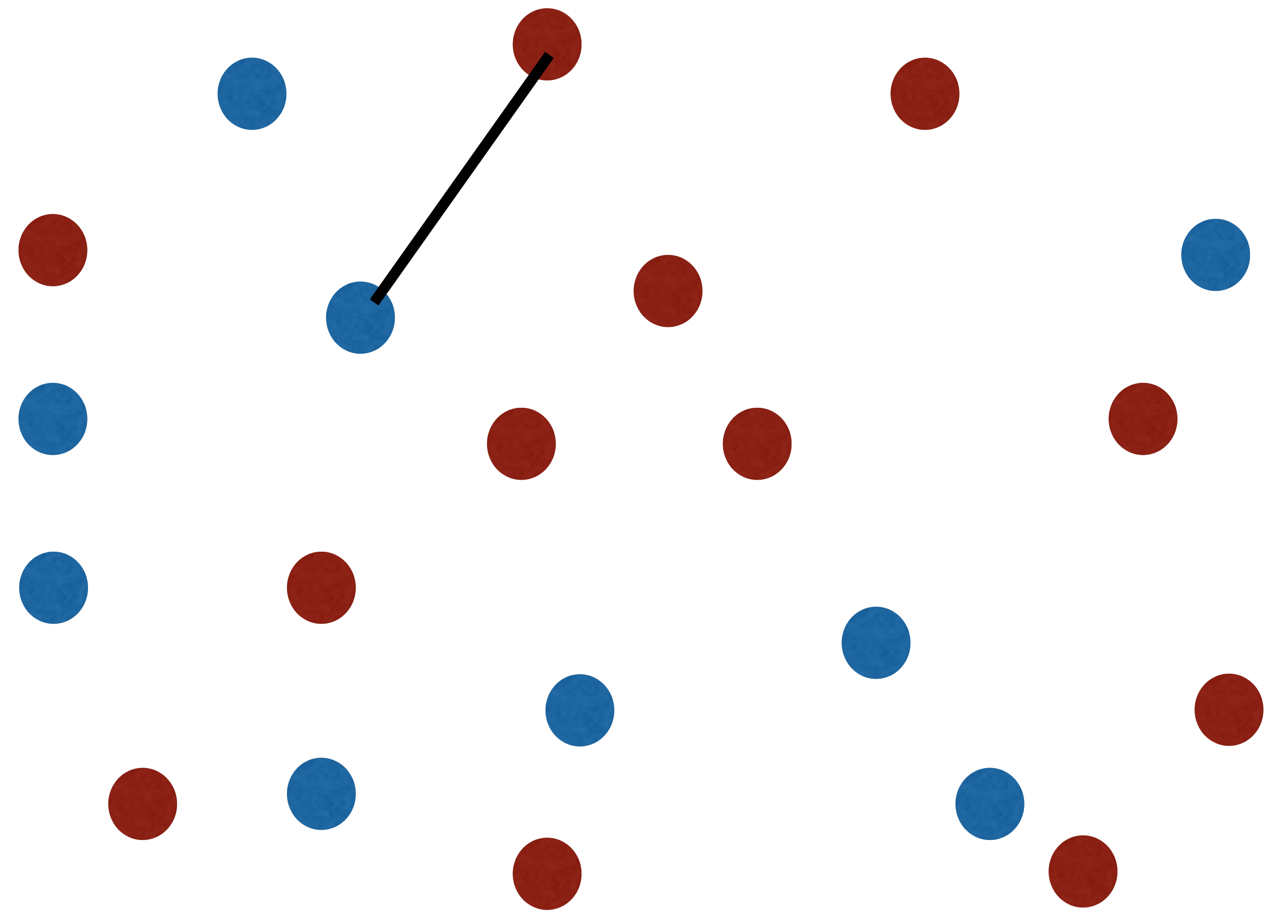
$$\texttt{EMD}(\mu) = \min_{\gamma} \|\mathrm{cost}(\gamma)\|_1$$

Our perturbation:

$$\mathcal{R}_\rho(\mu) = \min_{\gamma} \|\mathrm{cost}(\gamma)\|_\rho$$

Main Idea: Focus on a slight perturbation to EMD.





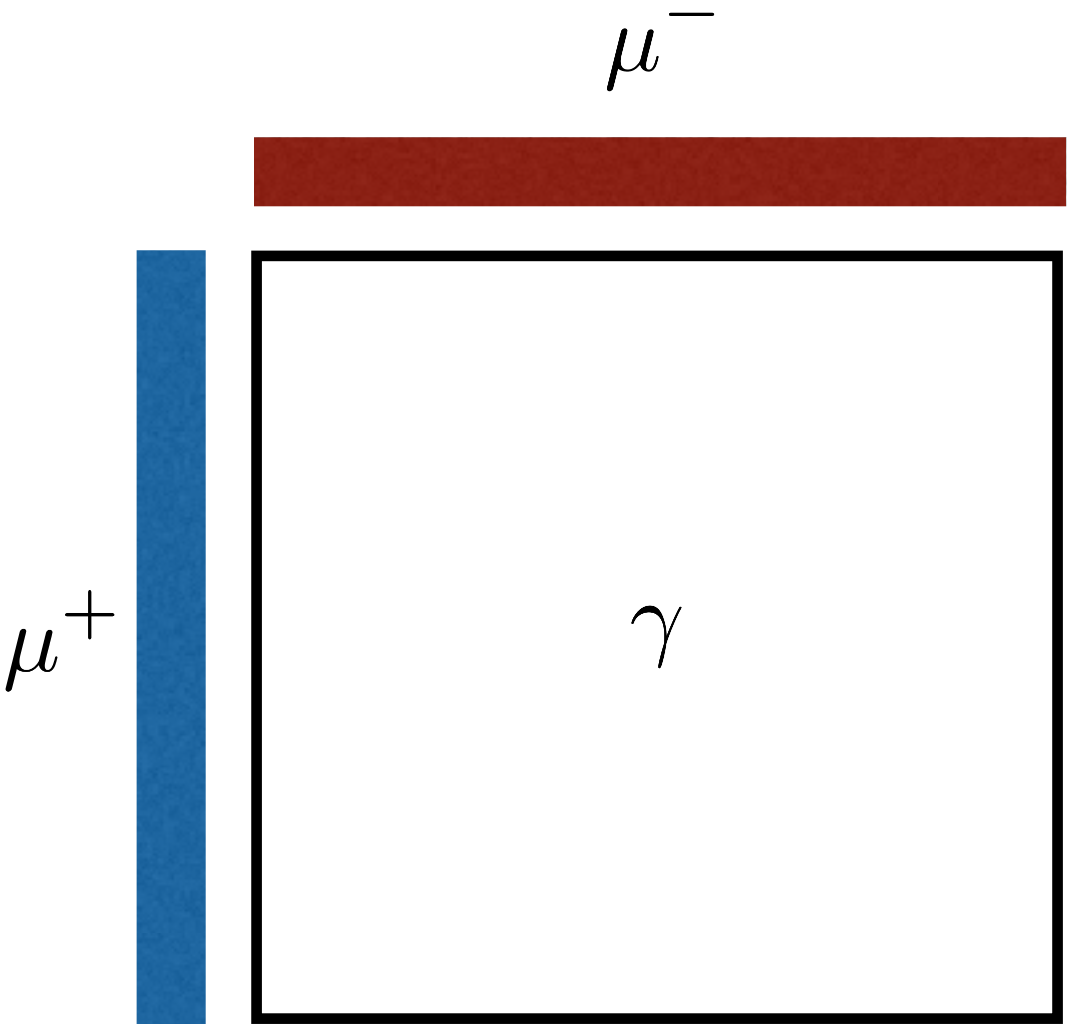
$\mathtt{EMD}(\mu)$

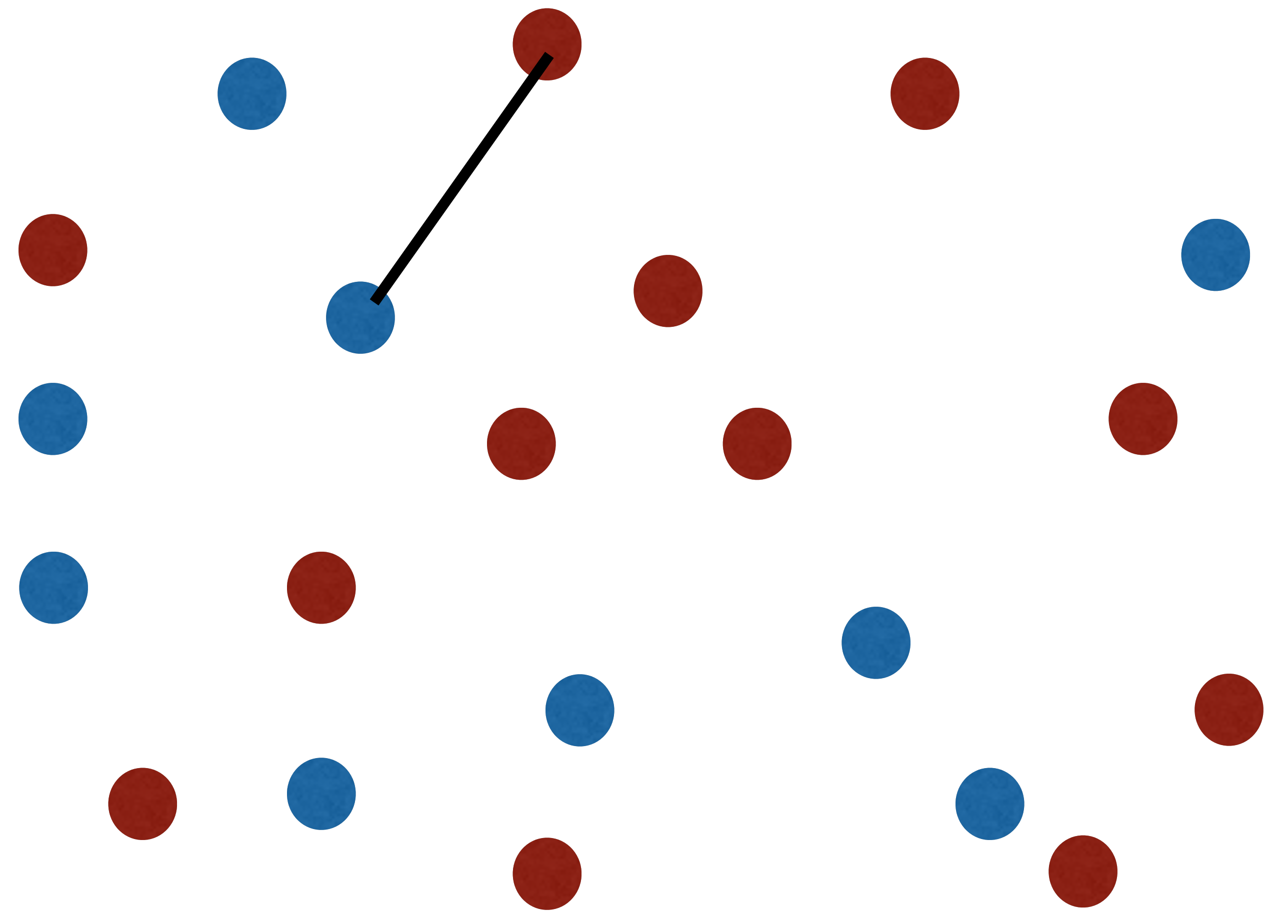
$$\min_{\gamma} \sum \sum \gamma_{ij} \|x_i - x_j\|_2$$

$\mathcal{R}_\rho(\mu)$

$$\min_{\gamma} \left(\sum \sum \gamma_{ij}^{\rho} \|x_i - x_j\|_2^{\rho} \right)^{1/\rho}$$

Main Idea: Focus on a slight perturbation to EMD.





$$\mathtt{EMD}(\mu) \qquad 1 \longleftarrow \rho \qquad \mathcal{R}_\rho(\mu)$$

$$\min_\gamma \sum \sum \gamma_{ij} \|x_i - x_j\|_2 \qquad \min_\gamma \left(\sum \sum \gamma_{ij}^\rho \|x_i - x_j\|_2^\rho \right)^{1/\rho}$$

Main Theorem: An algorithm for $\mathcal{R}_\rho(\mu)$

New Theorem:

For any $\rho > 1$ and any $\epsilon > 0$, there is an algorithm which receives as input $\{x_1, \dots, x_n\}$ and $\{y_1, \dots, y_n\}$ and in time at most $n \cdot \mathrm{poly}^*(n^{(\rho-1)/\rho} \cdot 2^{\rho/(\rho-1)}/\epsilon)$, outputs an $\pm\epsilon r$ approx to $\mathcal{R}_\rho(\mu)$.

Main Theorem: An algorithm for $\mathcal{R}_\rho(\mu)$

New Theorem:

For any $\rho > 1$ and any $\epsilon > 0$, there is an algorithm which receives as input $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_n\}$ and in time at most $n \cdot \mathrm{poly}^*(n^{(\rho-1)/\rho} \cdot 2^{\rho/(\rho-1)}/\epsilon)$, outputs an $\pm\epsilon r$ approx to $\mathcal{R}_\rho(\mu)$.

$$\rho = 1 + \frac{1}{\sqrt{\log n}}$$

Time: $n^{1+o(1)}/\mathrm{poly}(\epsilon)$

Main Theorem: An algorithm for $\mathcal{R}_\rho(\mu)$

New Theorem:

For any $\rho > 1$ and any $\epsilon > 0$, there is an algorithm which receives as input $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_n\}$ and in time at most $n \cdot \mathrm{poly}^*(n^{(\rho-1)/\rho} \cdot 2^{\rho/(\rho-1)}/\epsilon)$, outputs an $\pm\epsilon r$ approx to $\mathcal{R}_\rho(\mu)$.

$$\rho = 1 + \frac{1}{\sqrt{\log n}}$$

Time: $n^{1+o(1)}/\mathrm{poly}(\epsilon)$

EMD ⟷ Sinkhorn

Fast, crude approx — Slow, accurate approx

Talk Outline:

- “Spanner approach”
- Main Result
- **Kernel Density Estimation**
- Techniques

Kernel Density Estimation (KDE)

[Charikar, Siminelakis '17]

Preprocessing:

- A dataset $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$
- A kernel function $\mathsf{K}(x, y)$

Query:

- A point $y \in \mathbb{R}^d$
- Output approximation to $\tau = \frac{1}{n} \sum_{i=1}^{n} \mathsf{K}(x_i, y)$

e.g. $\mathsf{K}(x, y) = \frac{1}{\|x - y\|_2^2}$

Smoothness is very helpful in sketching.

[Backurs, Charikar, Indyk, Siminelakis '18]:

- Message: complexity governed by decay of $\mathsf{K}(x, y)$

Smooth Kernel:

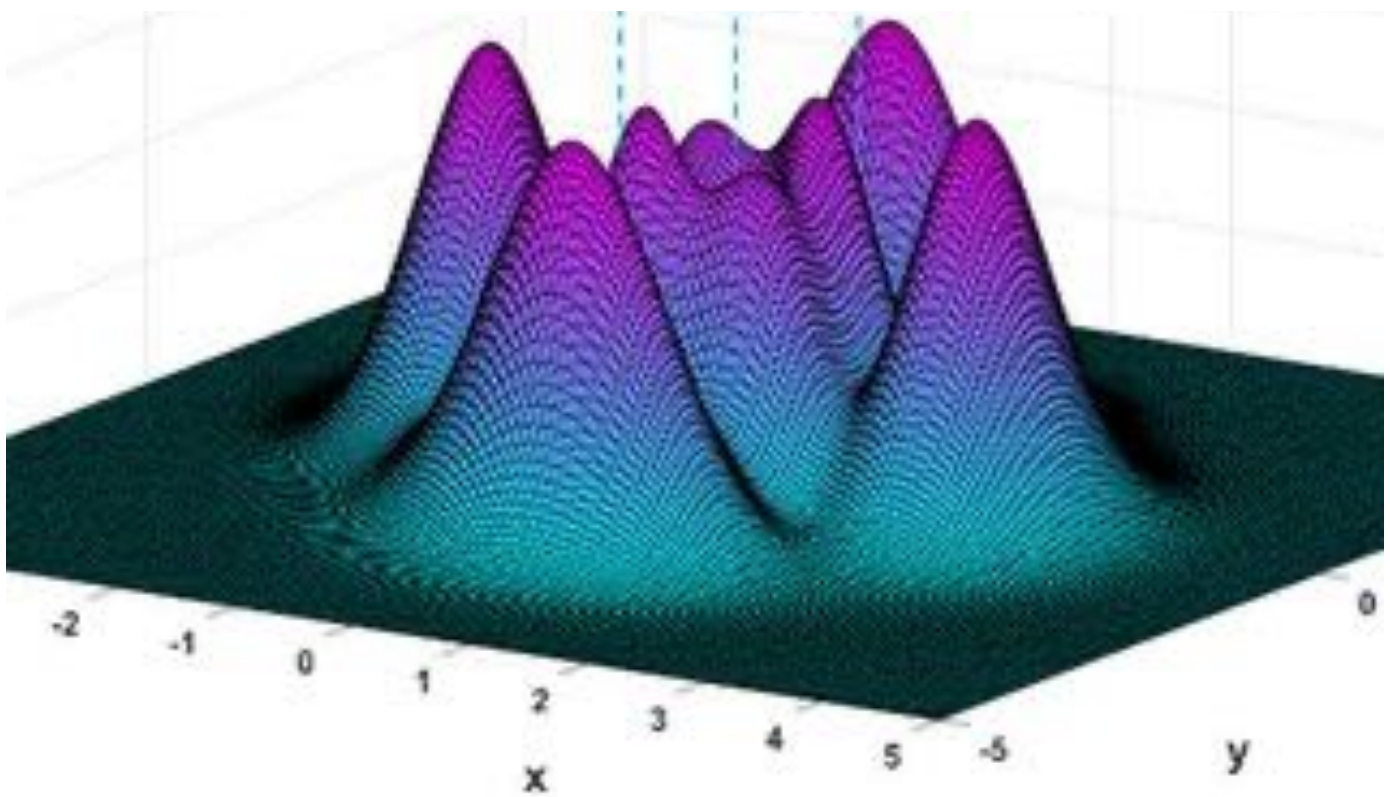
$$\mathsf{K}(x, y) \text{ decays polynomially with } \|x - y\|_2$$

$$\text{e.g.: } \mathsf{K}(x, y) = \frac{1}{\|x - y\|_2^s}$$

Theorem: there exists a data structure for approximating KDE of smooth kernels running in time $\mathrm{poly}(d \log n \cdot 2^s / \epsilon)$.

Smoothness is very helpful in sketching.

Smooth Kernel: $\mathsf{K}(x, y) = \dfrac{1}{\|x - y\|_2^s}$

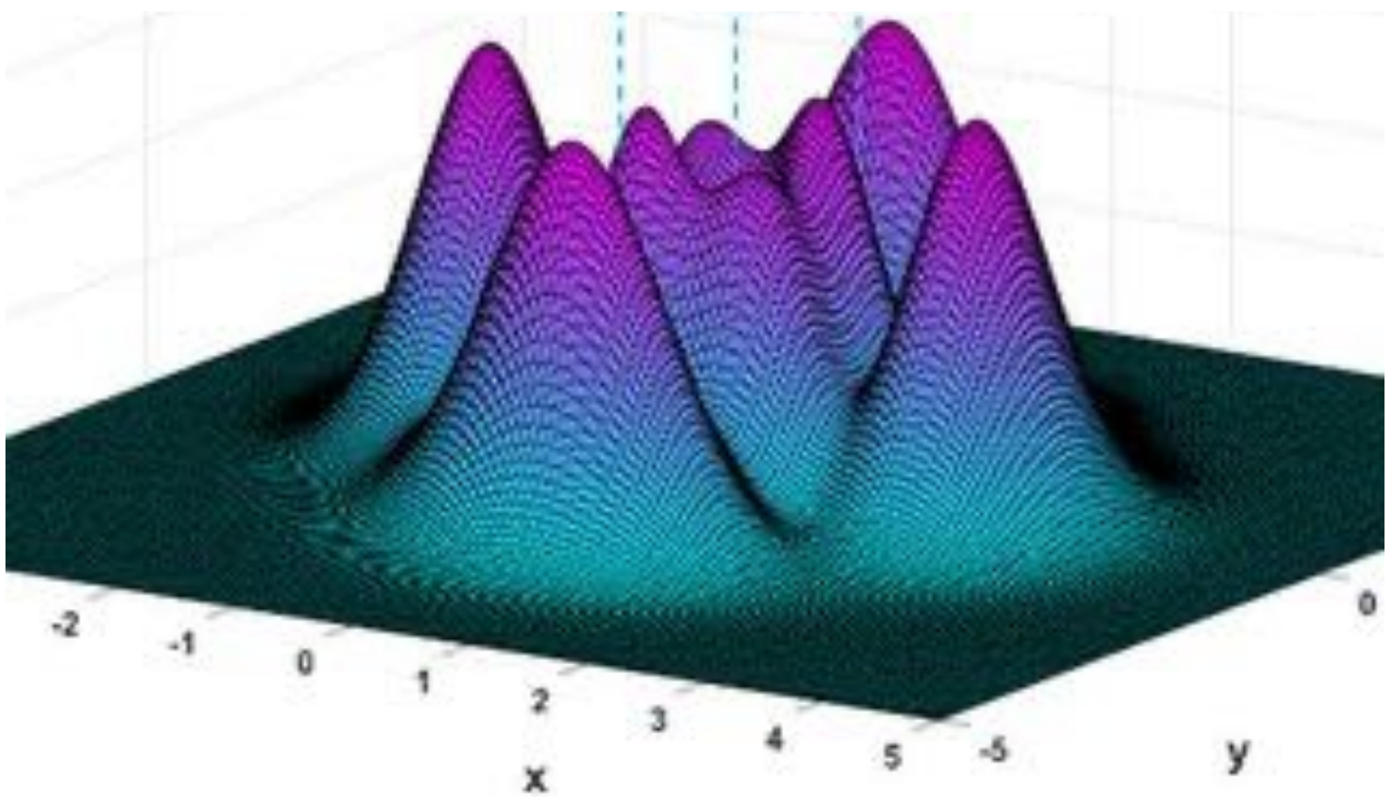


$\mathrm{poly}(d \log n \cdot 2^s / \epsilon)$

time per query.

Smoothness is very helpful in sketching.

Smooth Kernel: $\mathsf{K}(x,y) = \dfrac{1}{\|x-y\|_2^s}$



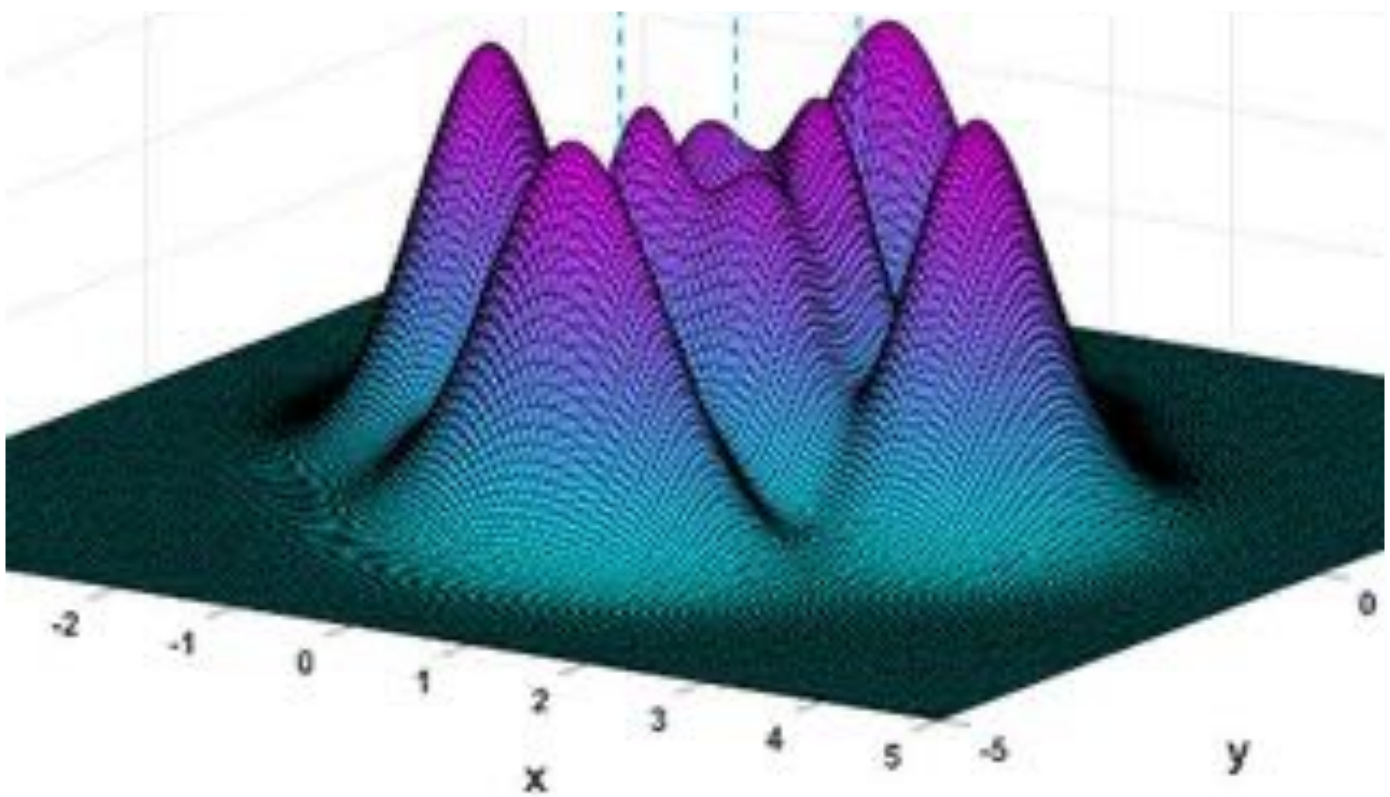
$\text{poly}(d \log n \cdot 2^s/\epsilon)$

time per query.

Looking forward to… $\mathcal{R}_\rho(\mu) \longleftrightarrow s = \dfrac{\rho}{\rho-1}$

Smoothness is very helpful in sketching.

Smooth Kernel: $\mathsf{K}(x,y) = \dfrac{1}{\|x-y\|_2^s}$



$\mathrm{poly}(d \log n \cdot 2^s/\epsilon)$

time per query.

Traditional sketching vectors ℓ_p : $\tilde{\Theta}(d^{1-2/p}/\epsilon^2)$

Techniques:
Using KDE for $\mathcal{R}_\rho(\mu)$.

The Linear Program for the Earth Mover's Distance

$$
\begin{aligned}
\texttt{EMD}(\mu) \quad &= \quad \min_{\gamma} \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{ij} \|x_i - y_j\|_2 \\
\text{s.t} \quad & \sum_{i=1}^{n} \gamma_{ij} = \frac{1}{n} \text{ for all } i \in [n] \\
& \sum_{j=1}^{n} \gamma_{ij} = \frac{1}{n} \text{ for all } j \in [n]
\end{aligned}
$$

The Linear Program for the Earth Mover's Distance

$$\texttt{EMD}(\mu) \quad = \quad \min_{\gamma} \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{ij} \|x_i - y_j\|_2$$

$$\text{s.t} \qquad \sum_{i=1}^{n} \gamma_{ij} = \frac{1}{n} \text{ for all } i \in [n]$$

$$\sum_{j=1}^{n} \gamma_{ij} = \frac{1}{n} \text{ for all } j \in [n]$$

Even representing γ is complicated!

The Linear Program for the Earth Mover's Distance

$$\texttt{EMD}(\mu) \quad = \quad \min_{\gamma} \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{ij} \|x_i - y_j\|_2$$

$$\text{s.t} \qquad \sum_{i=1}^{n} \gamma_{ij} = \frac{1}{n} \text{ for all } i \in [n]$$

$$\sum_{j=1}^{n} \gamma_{ij} = \frac{1}{n} \text{ for all } j \in [n]$$

$$= \quad \max_{\alpha, \beta \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} \alpha_i - \frac{1}{n} \sum_{j=1}^{n} \beta_j$$

$$\text{s.t} \qquad \alpha_i - \beta_j \leq \|x_i - y_j\|_2 \text{ for all } i, j \in [n]$$

The Perturbation $\mathcal{R}_\rho(\mu)$

$$\mathcal{R}_\rho(\mu)^\rho = \min_\gamma \sum_{i=1}^n \sum_{j=1}^n \gamma_{ij}^\rho \|x_i - y_j\|_2^\rho$$

$$\text{s.t} \qquad \sum_{j=1}^n \gamma_{ij} = \frac{1}{n} \text{ for all } i \in [n]$$

$$\sum_{i=1}^n \gamma_{ij} = \frac{1}{n} \text{ for all } j \in [n]$$

The Perturbation $\mathcal{R}_\rho(\mu)$

$$\mathcal{R}_\rho(\mu)^\rho = \min_\gamma \sum_{i=1}^n \sum_{j=1}^n \gamma_{ij}^\rho \|x_i - y_j\|_2^\rho$$

$$\text{s.t} \qquad \sum_{j=1}^n \gamma_{ij} = \frac{1}{n} \text{ for all } i \in [n]$$

$$\sum_{i=1}^n \gamma_{ij} = \frac{1}{n} \text{ for all } j \in [n]$$

$$= \max_{\alpha,\beta \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \alpha_i - \frac{1}{n} \sum_{j=1}^n \beta_j - C_\rho \sum_{i=1}^n \sum_{j=1}^n \left(\frac{(\alpha_i - \beta_j)^+}{\|x_i - y_j\|_2} \right)^s$$

The Perturbation $\mathcal{R}_\rho(\mu)$

$$\mathcal{R}_\rho(\mu)^\rho = \min_\gamma \sum_{i=1}^n \sum_{j=1}^n \gamma_{ij}^\rho \|x_i - y_j\|_2^\rho$$

$$\text{s.t} \qquad \sum_{j=1}^n \gamma_{ij} = \frac{1}{n} \text{ for all } i \in [n]$$

$$\sum_{i=1}^n \gamma_{ij} = \frac{1}{n} \text{ for all } j \in [n]$$

$$= \max_{\alpha,\beta \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \alpha_i - \frac{1}{n} \sum_{j=1}^n \beta_j$$

$$- C_\rho \sum_{i=1}^n \sum_{j=1}^n \left(\frac{(\alpha_i - \beta_j)^+}{\|x_i - y_j\|_2} \right)^s$$

$$s = \rho/(\rho - 1)$$

The Perturbation $\mathcal{R}_\rho(\mu)$

$$\mathcal{R}_\rho(\mu)^\rho = \min_{\gamma} \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{ij}^{\rho} \|x_i - y_j\|_2^{\rho}$$

$$\text{s.t} \qquad \sum_{j=1}^{n} \gamma_{ij} = \frac{1}{n} \text{ for all } i \in [n]$$

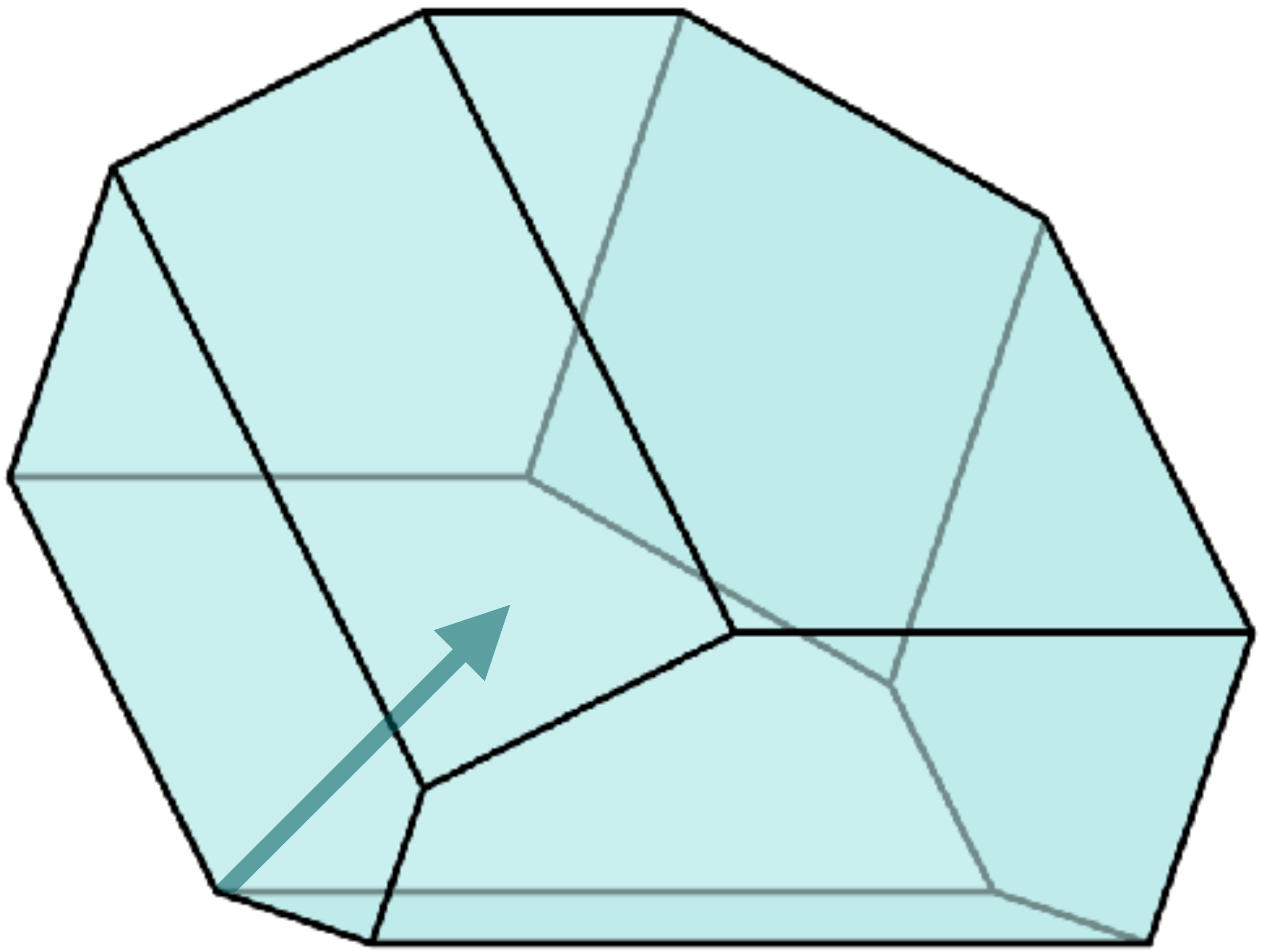
$$\sum_{i=1}^{n} \gamma_{ij} = \frac{1}{n} \text{ for all } j \in [n]$$

$$= \max_{\alpha,\beta \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} \alpha_i - \frac{1}{n} \sum_{j=1}^{n} \beta_j$$

$$- C_\rho \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\frac{(\alpha_i - \beta_j)^+}{\|x_i - y_j\|_2} \right)^s$$

$s = \rho/(\rho - 1)$

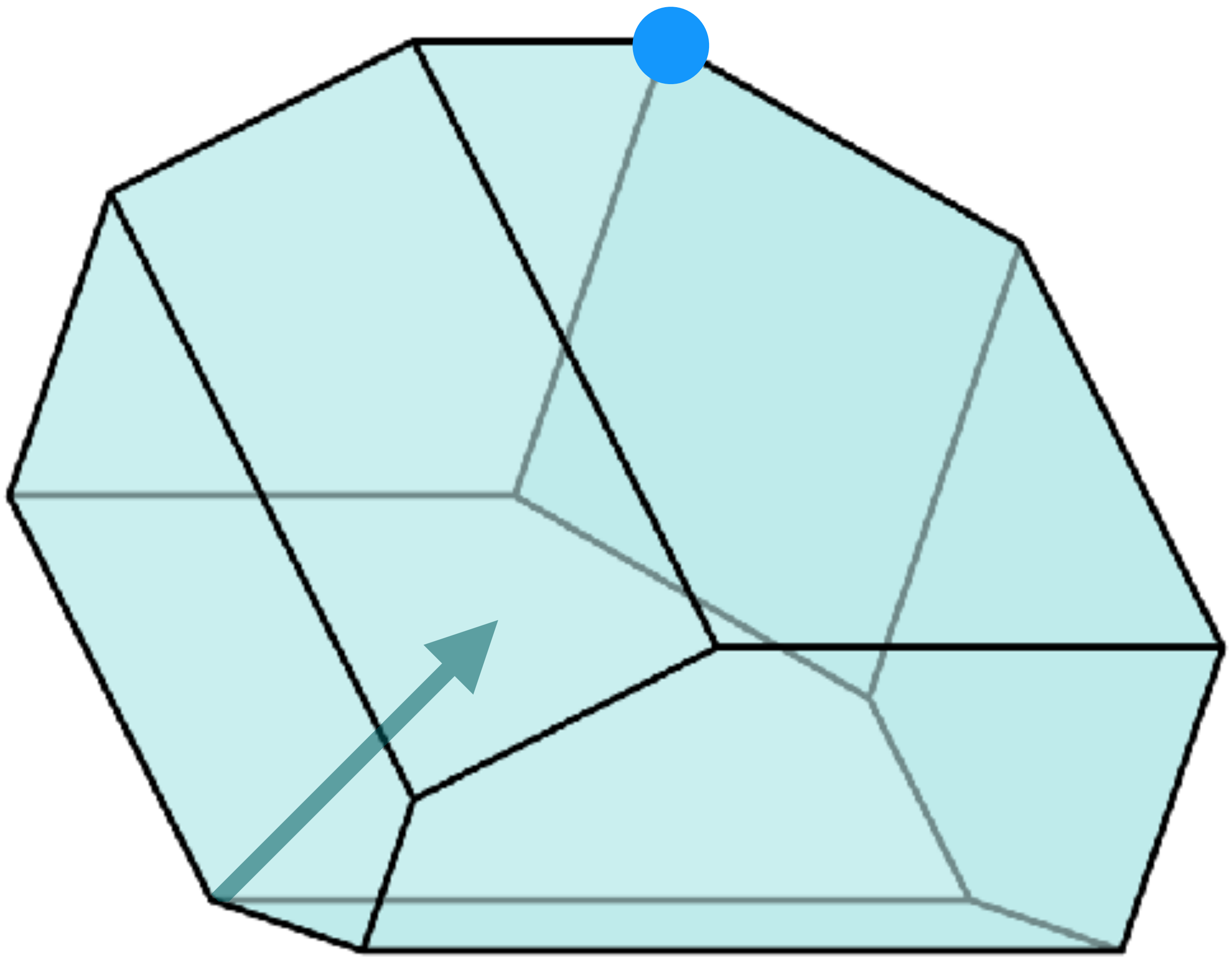
Picture: EMD vs $\mathcal{R}_\rho(\mu)$



$$\max_{\alpha,\beta\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n}\alpha_i - \frac{1}{n}\sum_{j=1}^{n}\beta_j$$

$$\text{s.t} \qquad \alpha_i - \beta_j \leq \|x_i - y_j\|_2$$

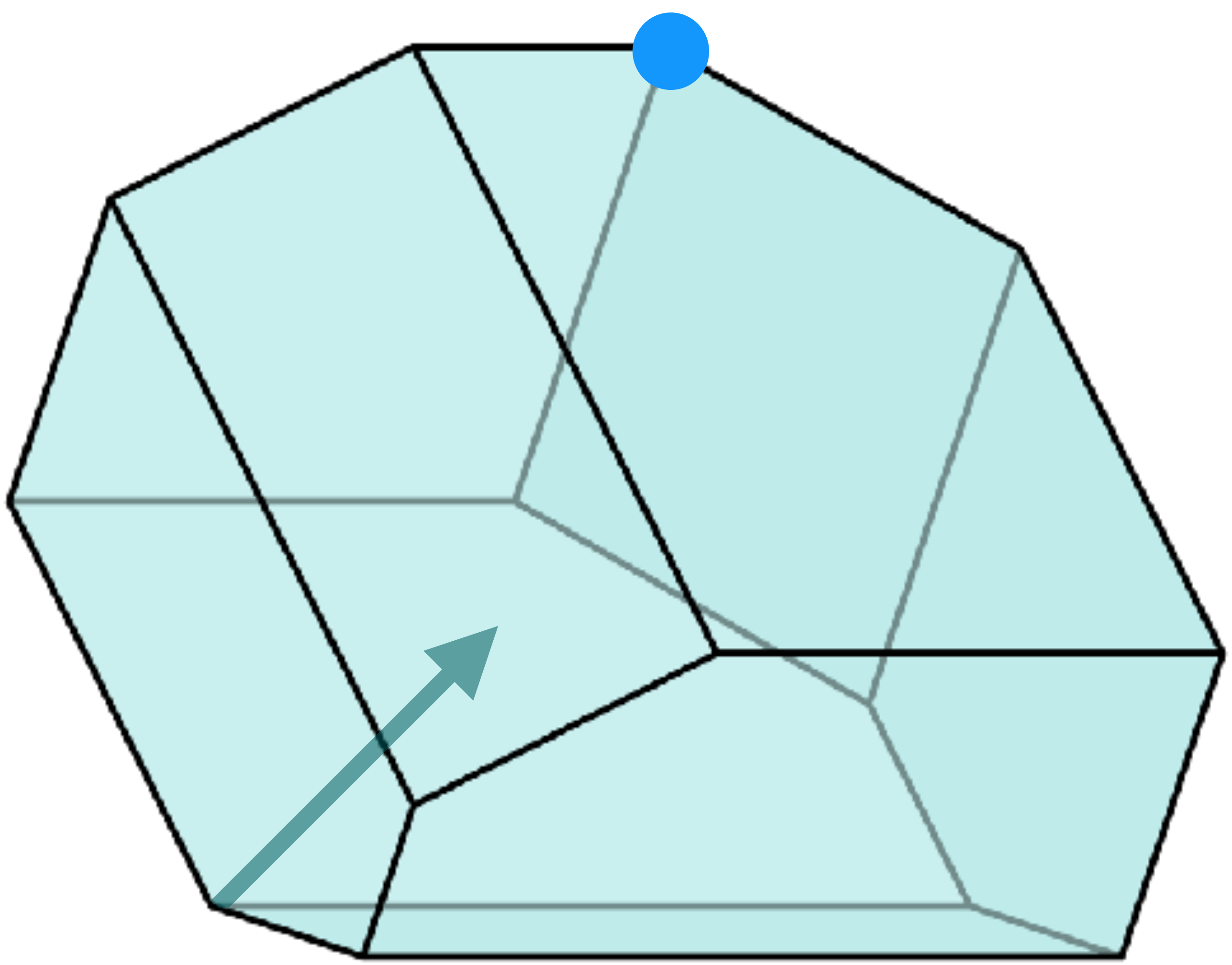
Picture: EMD vs $\mathcal{R}_\rho(\mu)$



$$\max_{\alpha,\beta\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n}\alpha_i - \frac{1}{n}\sum_{j=1}^{n}\beta_j$$

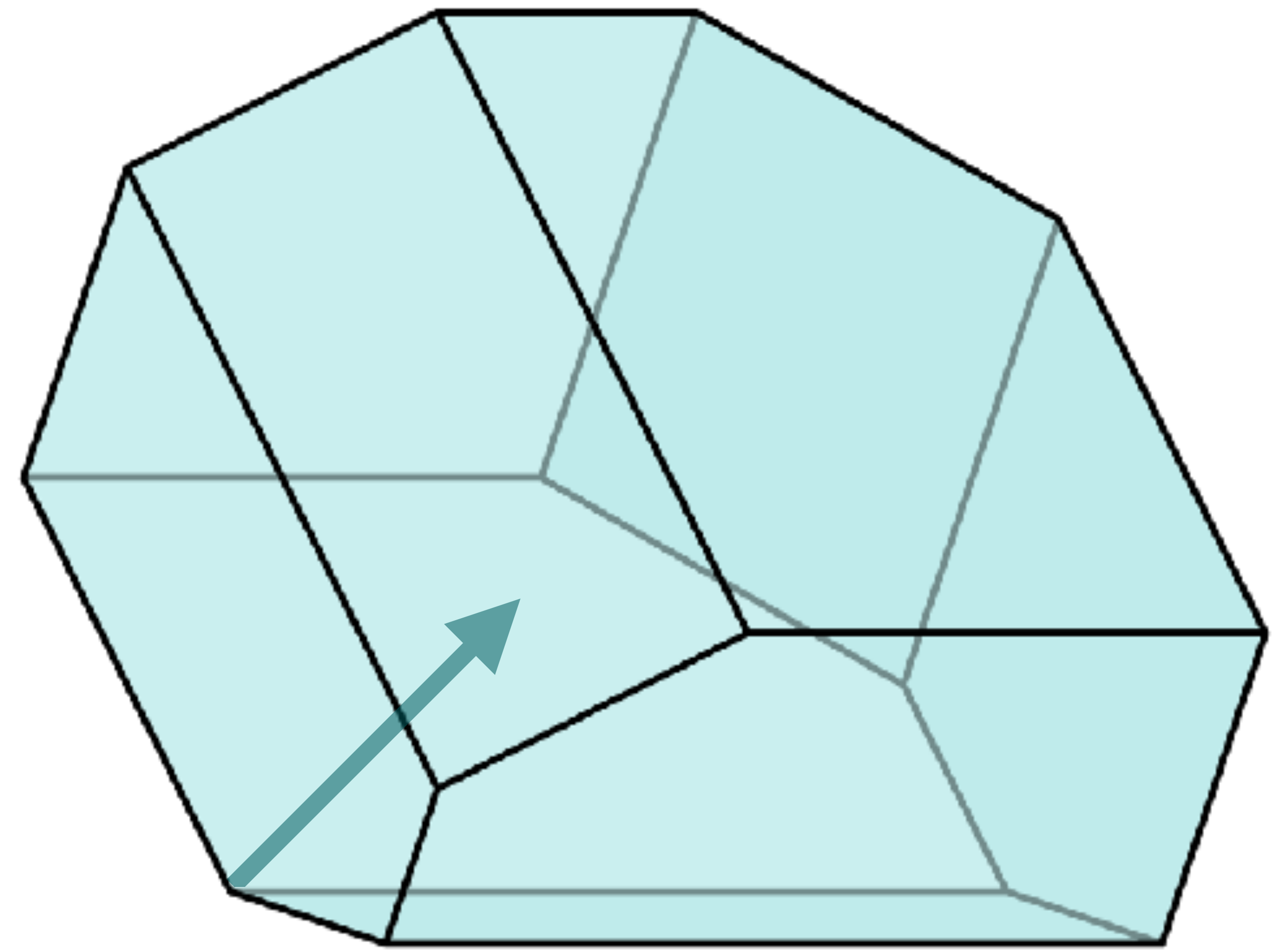
$$\text{s.t} \qquad \alpha_i - \beta_j \leq \|x_i - y_j\|_2$$

Picture: EMD vs $\mathcal{R}_\rho(\mu)$



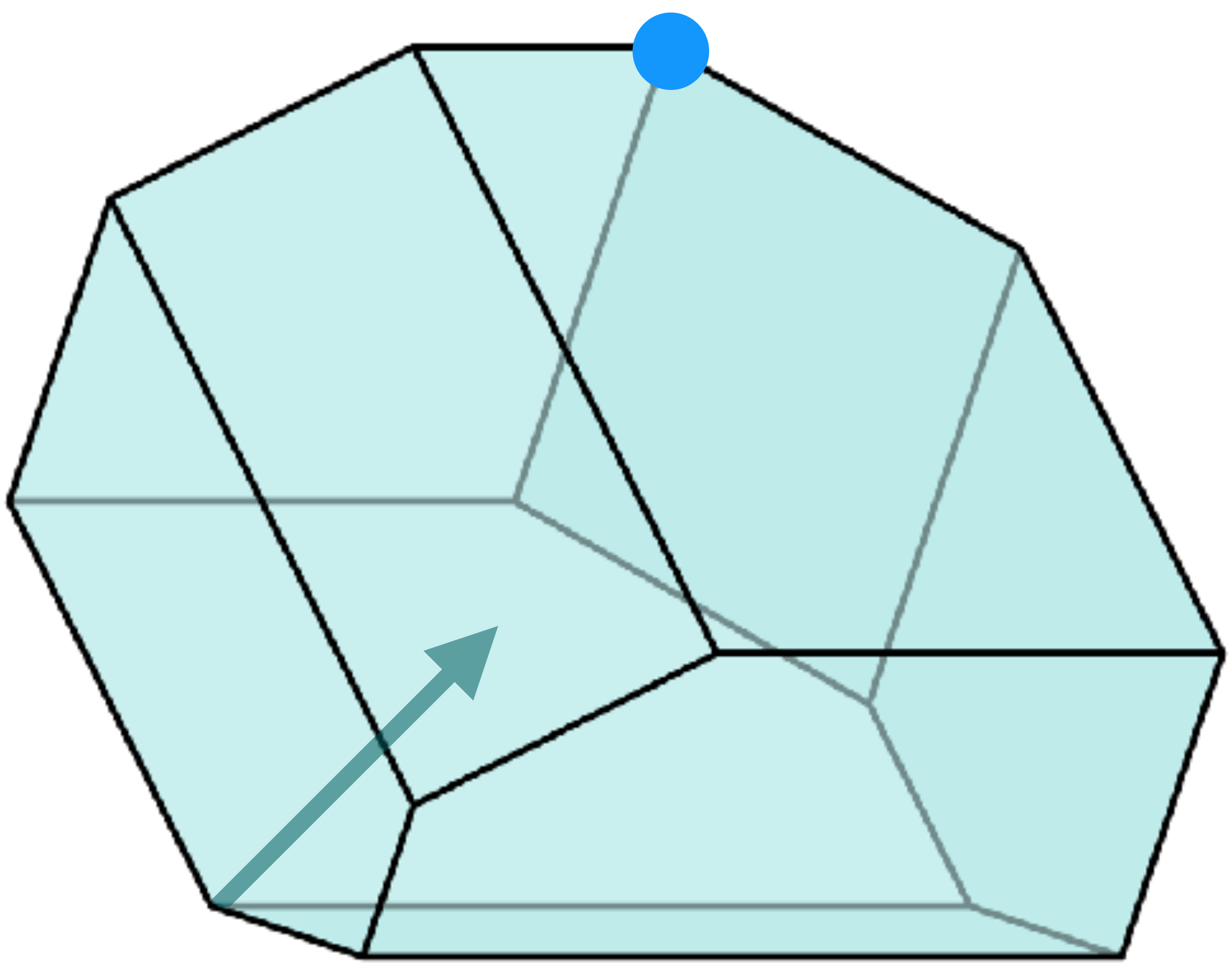
$$\max_{\alpha,\beta\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n}\alpha_i - \frac{1}{n}\sum_{j=1}^{n}\beta_j$$

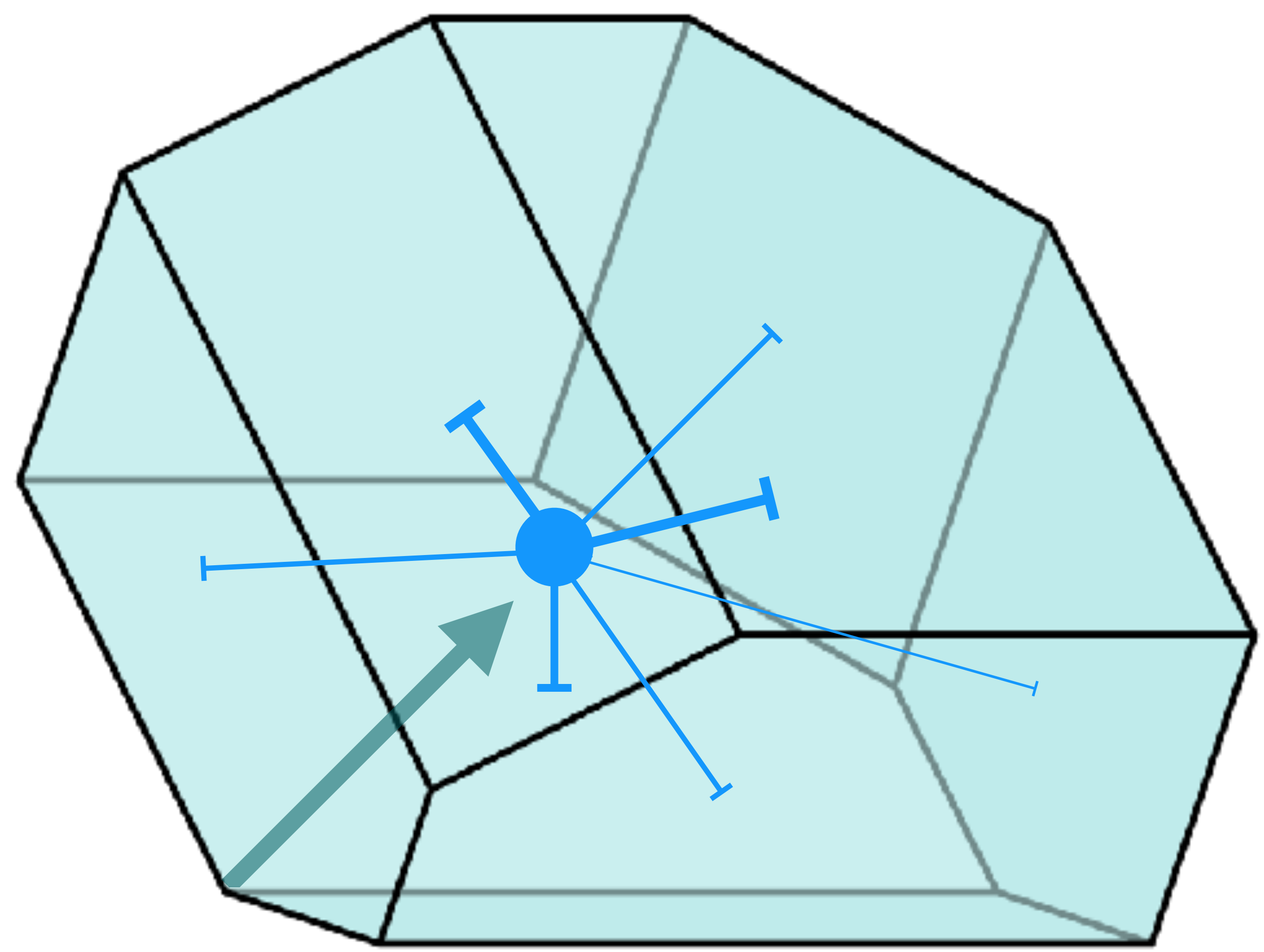
$$\text{s.t} \quad \alpha_i - \beta_j \leq \|x_i - y_j\|_2$$



$$\max_{\alpha,\beta\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n}\alpha_i - \frac{1}{n}\sum_{j=1}^{n}\beta_j - C_\rho \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{(\alpha_i - \beta_j)^+}{\|x_i - y_j\|_2}\right)^s$$

Picture: EMD vs $\mathcal{R}_\rho(\mu)$

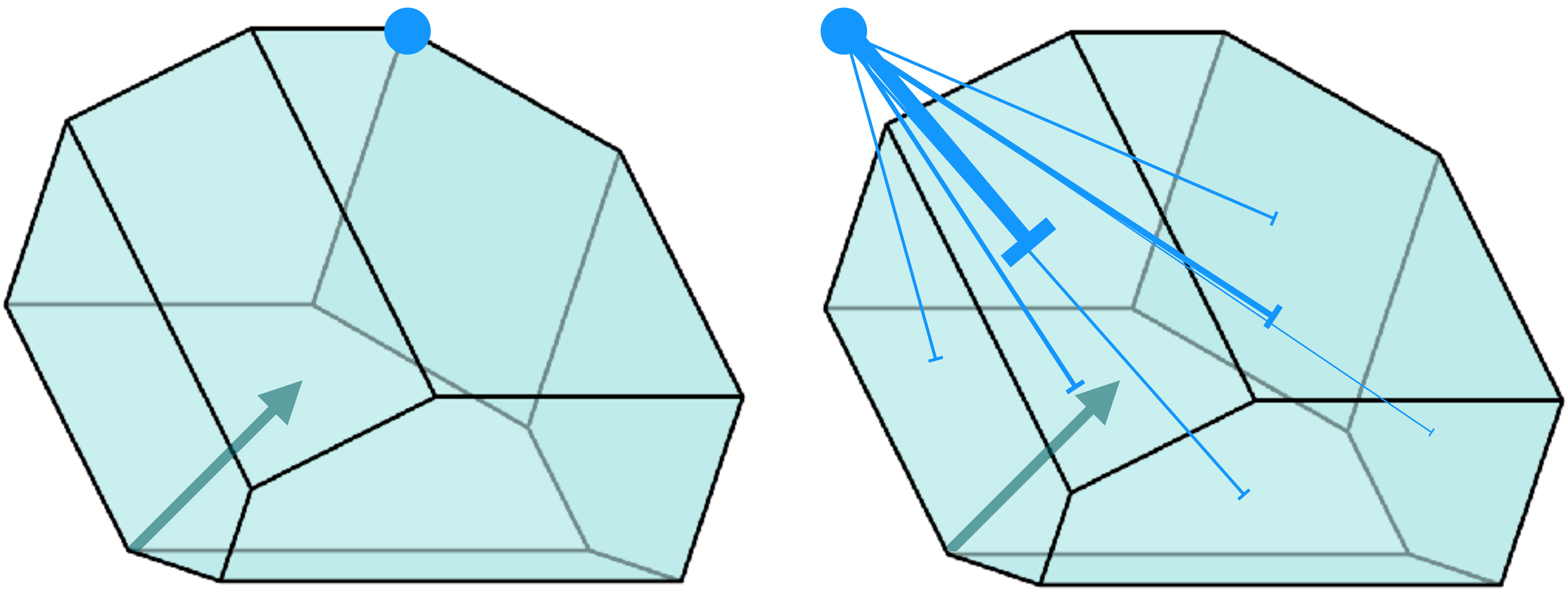




$$\max_{\alpha,\beta\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^n \alpha_i - \frac{1}{n}\sum_{j=1}^n \beta_j$$
$$\text{s.t} \qquad \alpha_i - \beta_j \le \|x_i - y_j\|_2$$

$$\max_{\alpha,\beta\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^n \alpha_i - \frac{1}{n}\sum_{j=1}^n \beta_j$$
$$- C_\rho \sum_{i=1}^n \sum_{j=1}^n \left(\frac{(\alpha_i - \beta_j)^+}{\|x_i - y_j\|_2} \right)^s$$

Picture: EMD vs $\mathcal{R}_\rho(\mu)$

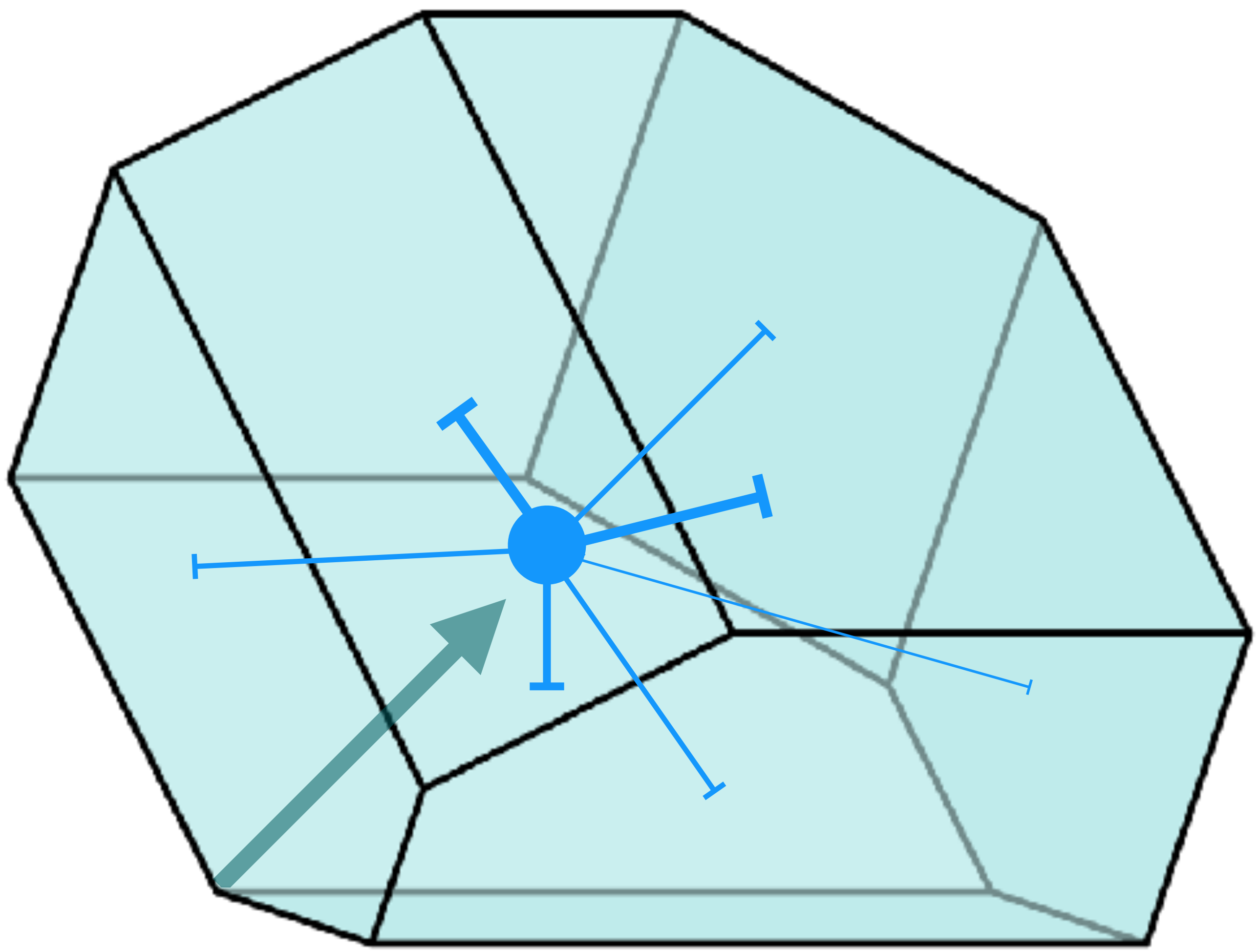


$$\max_{\alpha,\beta\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n}\alpha_i - \frac{1}{n}\sum_{j=1}^{n}\beta_j$$

$$\text{s.t} \qquad \alpha_i - \beta_j \leq \|x_i - y_j\|_2$$

$$\max_{\alpha,\beta\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n}\alpha_i - \frac{1}{n}\sum_{j=1}^{n}\beta_j - C_\rho\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{(\alpha_i-\beta_j)^+}{\|x_i-y_j\|_2}\right)^s$$

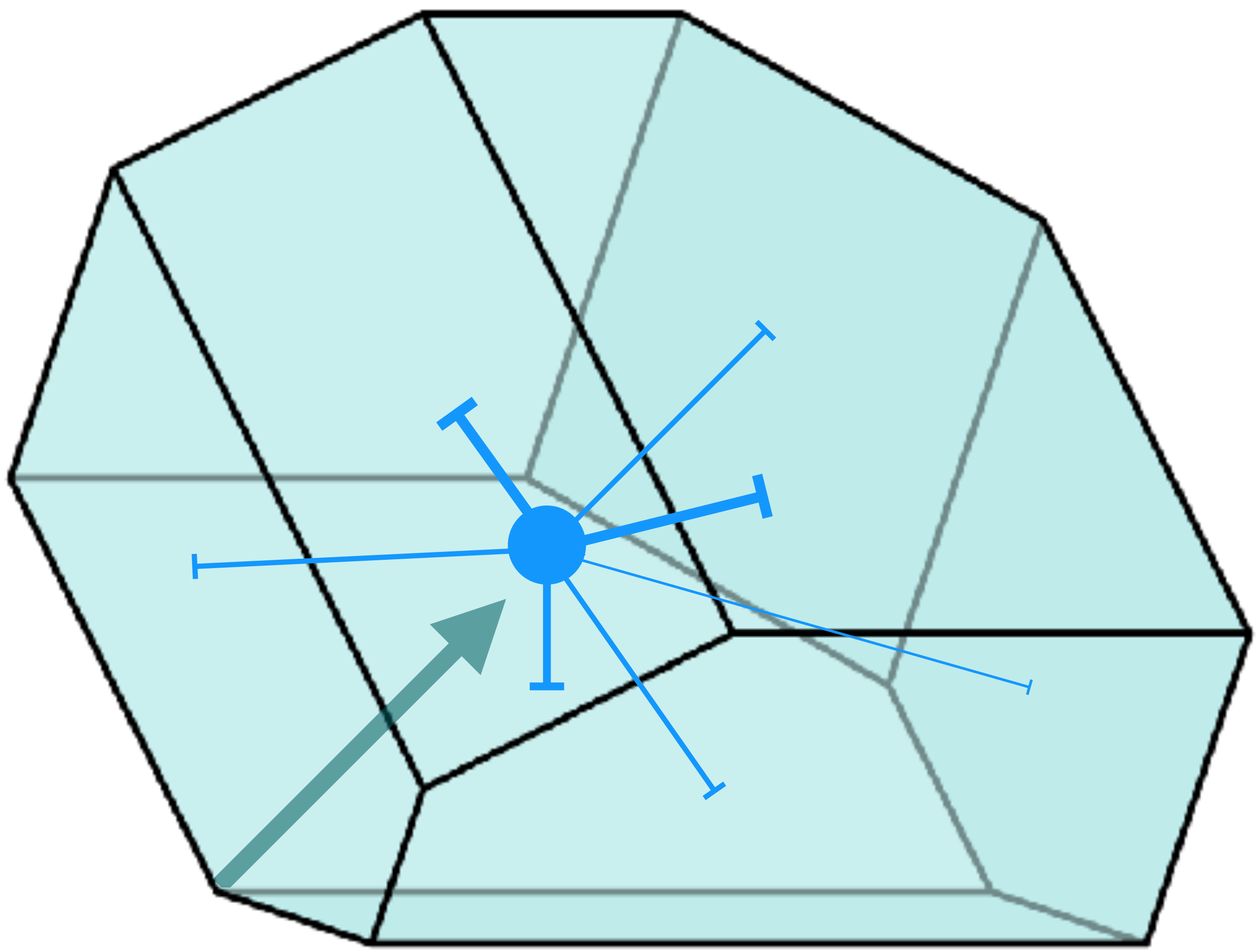
We can efficiently run first-order methods.



$$\max_{\alpha,\beta\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n}\alpha_i - \frac{1}{n}\sum_{j=1}^{n}\beta_j$$

$$- C_\rho \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{(\alpha_i-\beta_j)^+}{\|x_i-y_j\|_2}\right)^s$$

We can efficiently run first-order methods.

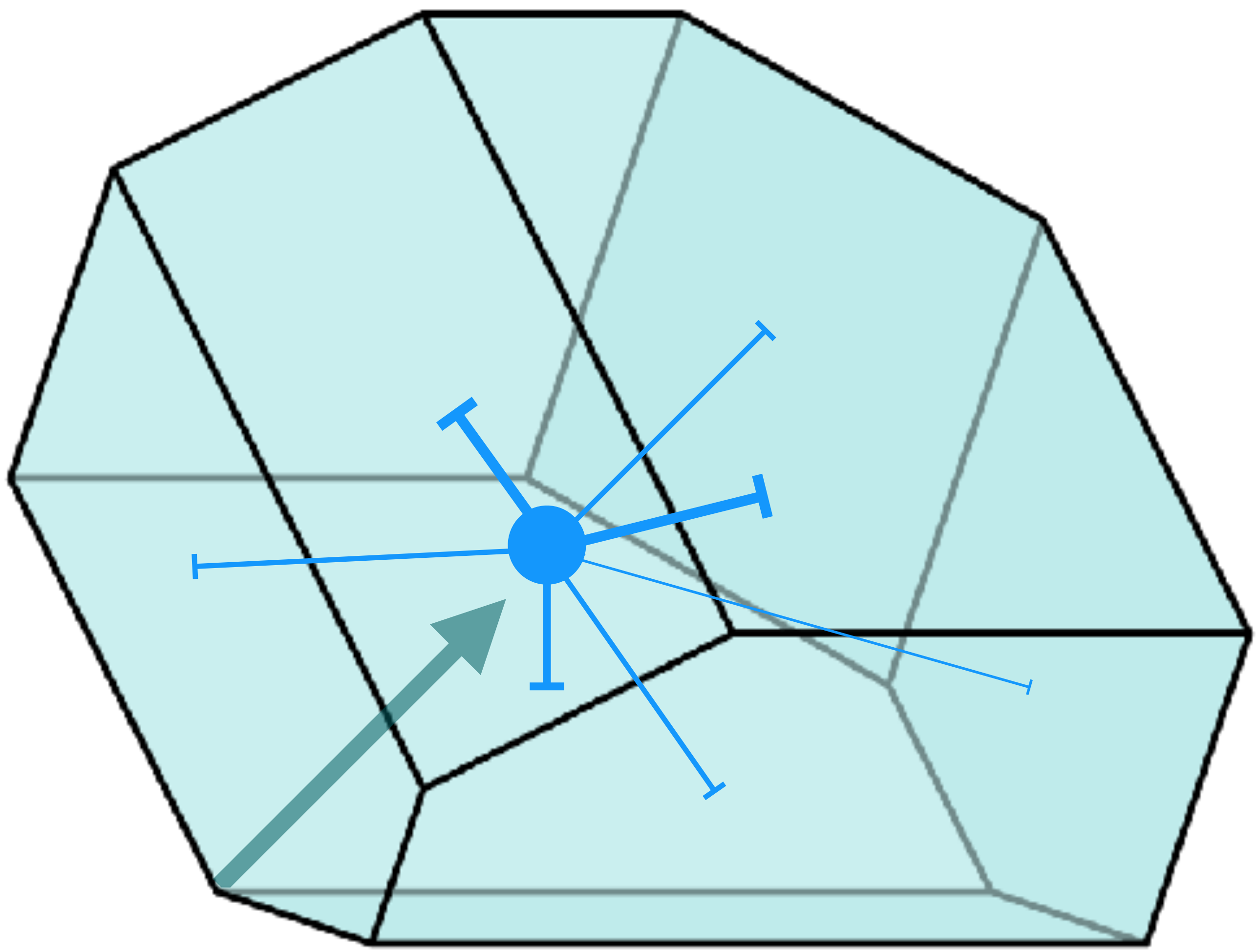


$$\frac{\partial}{\partial \alpha_i} = \frac{1}{n} - sC_\rho \sum_{j=1}^{n} \frac{((\alpha_i - \beta_j)^+)^{s-1}}{\|x_i - y_j\|_2^s}$$

$$\frac{\partial}{\partial \beta_j} = -\frac{1}{n} + sC_\rho \sum_{i=1}^{n} \frac{((\alpha_i - \beta_j)^+)^{s-1}}{\|x_i - y_j\|_2^s}$$

$$\max_{\alpha,\beta \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} \alpha_i - \frac{1}{n} \sum_{j=1}^{n} \beta_j - C_\rho \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\frac{(\alpha_i - \beta_j)^+}{\|x_i - y_j\|_2} \right)^s$$

We can efficiently run first-order methods.

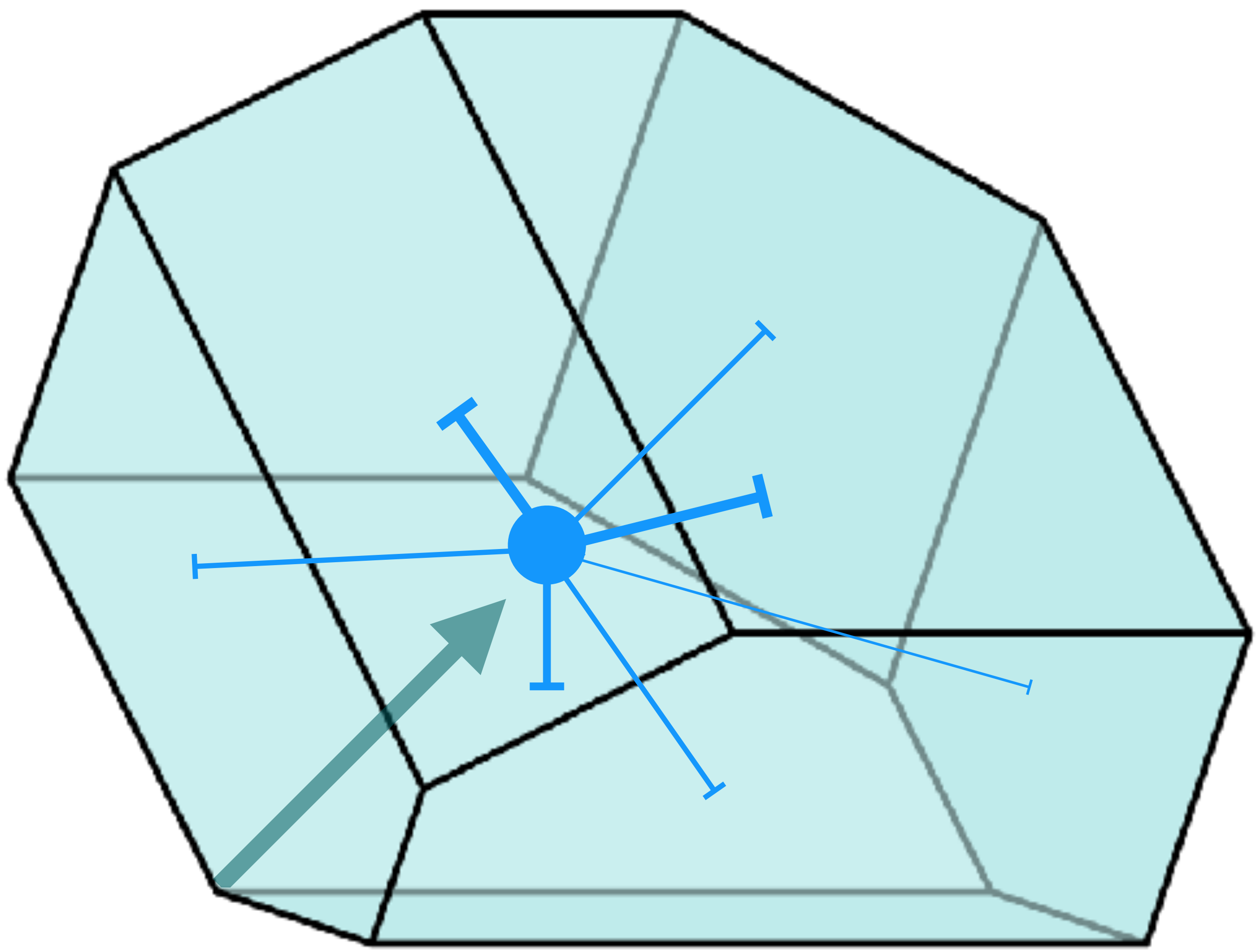


$$\frac{\partial}{\partial \alpha_i} = \frac{1}{n} - sC_\rho \sum_{j=1}^{n} \frac{((\alpha_i - \beta_j)^+)^{s-1}}{\|x_i - y_j\|_2^s}$$

$$\frac{\partial}{\partial \beta_j} = -\frac{1}{n} + sC_\rho \sum_{i=1}^{n} \frac{((\alpha_i - \beta_j)^+)^{s-1}}{\|x_i - y_j\|_2^s}$$

$$\max_{\alpha,\beta\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n}\alpha_i - \frac{1}{n}\sum_{j=1}^{n}\beta_j - C_\rho \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{(\alpha_i - \beta_j)^+}{\|x_i - y_j\|_2}\right)^s$$

We can efficiently run first-order methods.



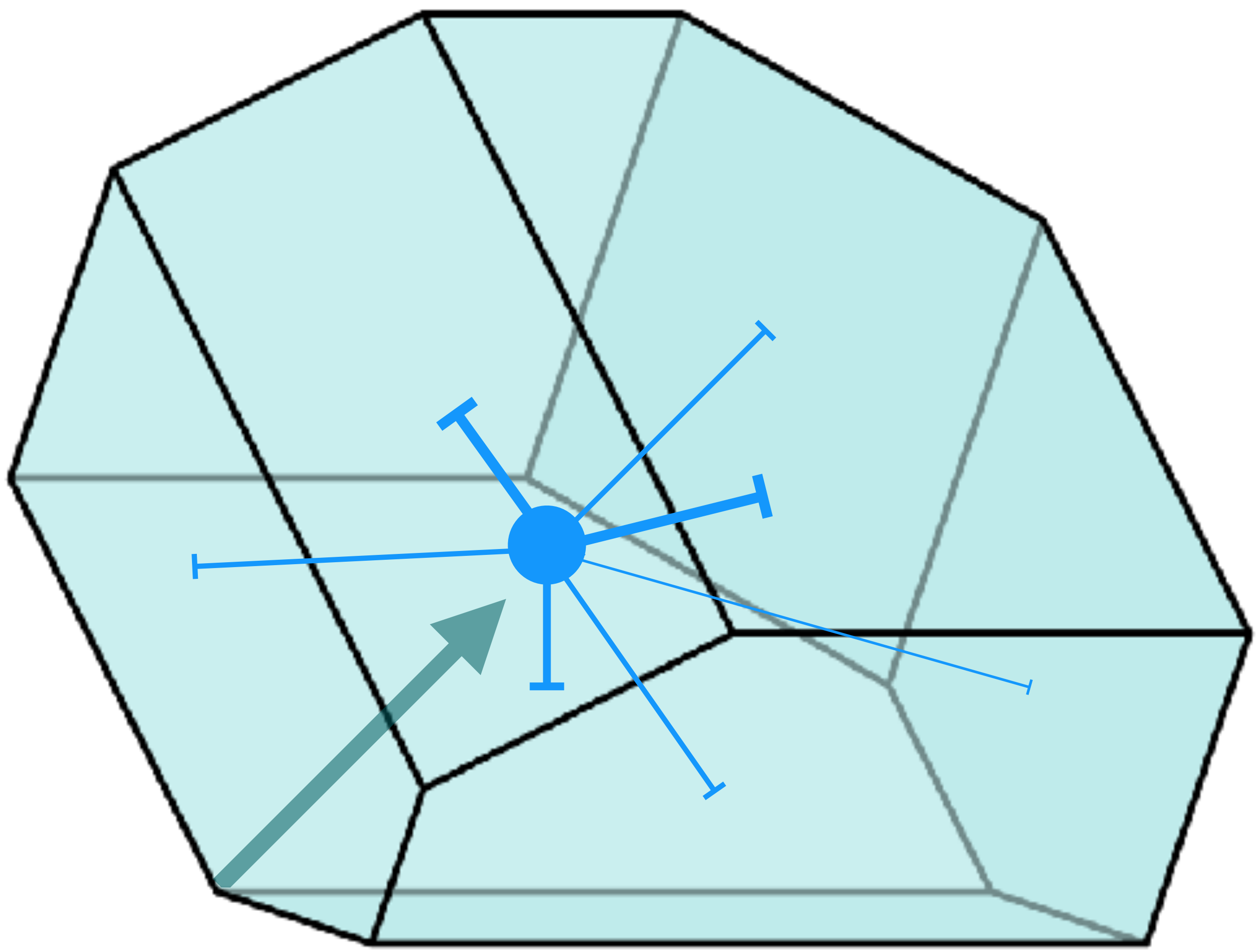
$$\frac{\partial}{\partial \alpha_i} = \frac{1}{n} - sC_\rho \sum_{j=1}^{n} \frac{((\alpha_i - \beta_j)^+)^{s-1}}{\|x_i - y_j\|_2^s}$$

$$\frac{\partial}{\partial \beta_j} = -\frac{1}{n} + sC_\rho \sum_{i=1}^{n} \frac{((\alpha_i - \beta_j)^+)^{s-1}}{\|x_i - y_j\|_2^s}$$

If can handle $((\alpha_i - \beta_j)^+)^{s-1}$,
rest is a smooth kernel.

$$\max_{\alpha,\beta \in \mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n} \alpha_i - \frac{1}{n}\sum_{j=1}^{n} \beta_j - C_\rho \sum_{i=1}^{n}\sum_{j=1}^{n} \left(\frac{(\alpha_i - \beta_j)^+}{\|x_i - y_j\|_2}\right)^s$$

We can efficiently run first-order methods.



$$\frac{\partial}{\partial \alpha_i} = \frac{1}{n} - sC_\rho \sum_{j=1}^{n} \frac{((\alpha_i - \beta_j)^+)^{s-1}}{\|x_i - y_j\|_2^s}$$

$$\frac{\partial}{\partial \beta_j} = -\frac{1}{n} + sC_\rho \sum_{i=1}^{n} \frac{((\alpha_i - \beta_j)^+)^{s-1}}{\|x_i - y_j\|_2^s}$$

If can handle $((\alpha_i - \beta_j)^+)^{s-1}$,
rest is a smooth kernel.

$$\max_{\alpha,\beta \in \mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n} \alpha_i - \frac{1}{n}\sum_{j=1}^{n} \beta_j - C_\rho \sum_{i=1}^{n}\sum_{j=1}^{n} \left(\frac{(\alpha_i - \beta_j)^+}{\|x_i - y_j\|_2}\right)^s$$

In time $O(n) \cdot \text{poly}(d \log n \cdot 2^s/\epsilon)$
good approximations to gradient

Algorithm

1. Maintain α, β.
2. Estimate $\frac{\partial}{\partial\alpha}, \frac{\partial}{\partial\beta}$.
3. If $\partial/(\partial\alpha)$ far from 0, update α
4. If $\partial/(\partial\beta)$ far from 0, update β

$$\max_{\alpha,\beta\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n}\alpha_i - \frac{1}{n}\sum_{j=1}^{n}\beta_j - C_\rho \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{(\alpha_i-\beta_j)^+}{\|x_i-y_j\|_2}\right)^s$$

$$\frac{\partial}{\partial\alpha_i} = \frac{1}{n} - sC_\rho \sum_{j=1}^{n} \frac{((\alpha_i-\beta_j)^+)^{s-1}}{\|x_i-y_j\|_2^s}$$

$$\frac{\partial}{\partial\beta_j} = -\frac{1}{n} + sC_\rho \sum_{i=1}^{n} \frac{((\alpha_i-\beta_j)^+)^{s-1}}{\|x_i-y_j\|_2^s}$$

Open Problems

Open Problems

1. Can we beat the spanner approach?

Open Problems

1. Can we beat the spanner approach?
2. Which geometric problems can we approx accurately?

Open Problems

1. Can we beat the spanner approach?
2. Which geometric problems can we approx accurately?
3. Better dependence on ϵ.

Open Problems

1. Can we beat the spanner approach?
2. Which geometric problems can we approx accurately?
3. Better dependence on ϵ.

Give an algorithm for approximating EMD cost in time $n \cdot \mathrm{poly}(d \log(n)/\epsilon)$,
or
show that it would break complexity theory assumption.

Open Problems

1. Can we beat the spanner approach?
2. Which geometric problems can we approx accurately?
3. Better dependence on ϵ.

Give an algorithm for approximating EMD cost in time $n \cdot \mathrm{poly}(d \log(n)/\epsilon)$,
or
show that it would break complexity theory assumption.

Thanks