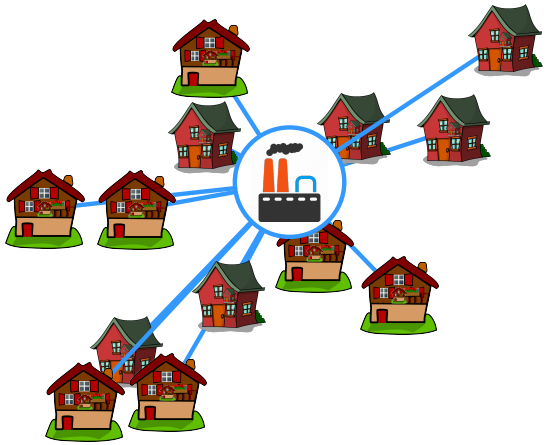
Clustering High Dimensional Dynamic Data Streams

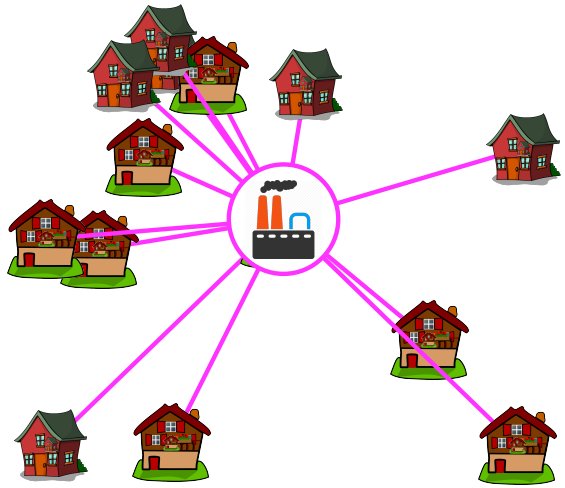
Lin F. Yang

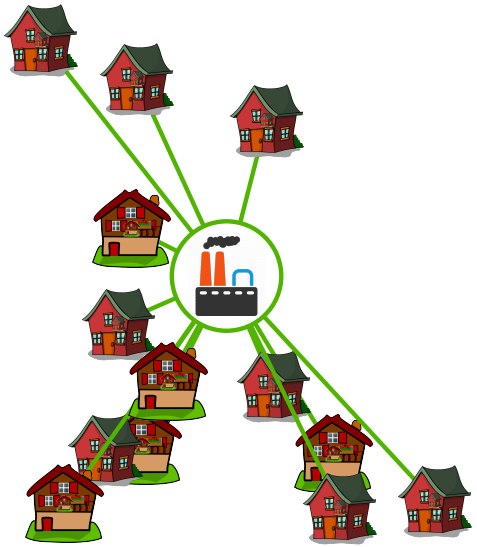
Joint Work With:

Vladimir Braverman (JHU) Gereon Frahling (Linguee GmbH)
Harry Lang (JHU) Christian Sohler (TU Dortmund)

k-Clustering





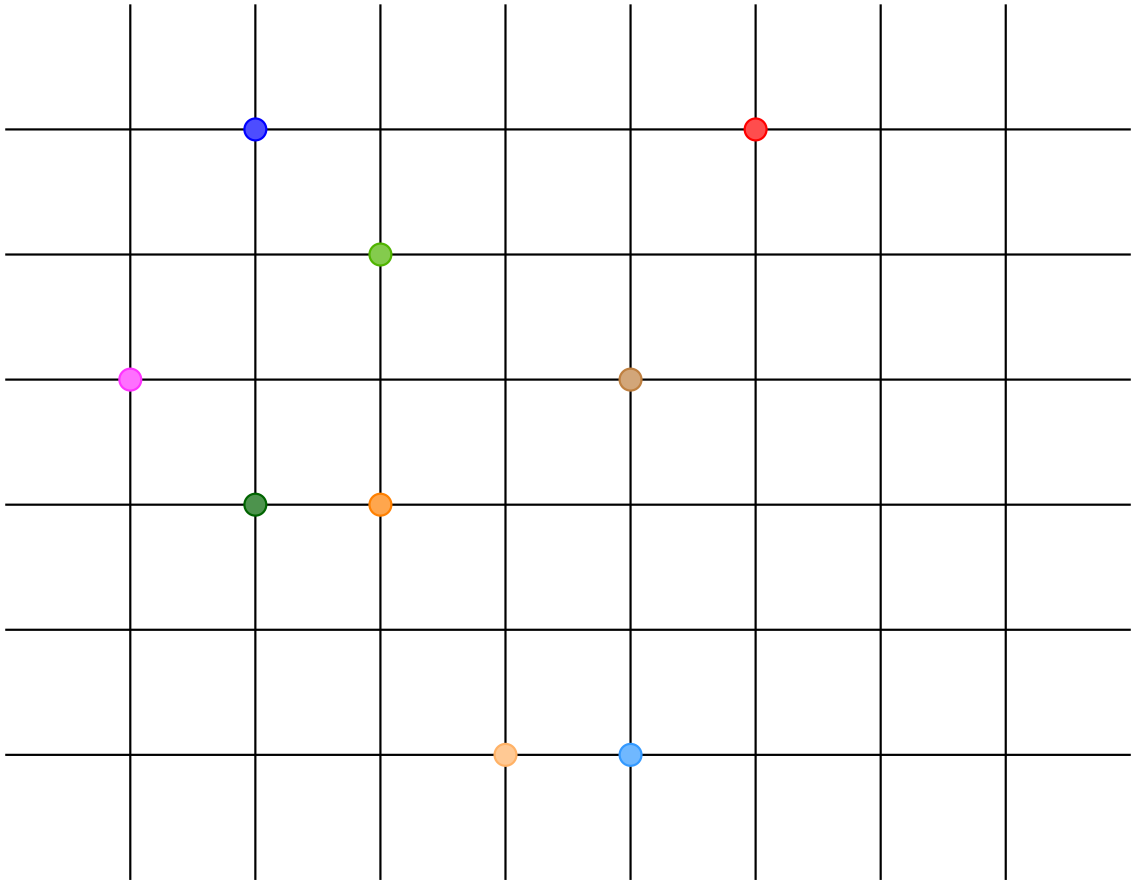


$$\min_{C:|C|=k} \sum_{p \in P} \min_{c \in C} \text{dist}(p, c)$$

Dynamic Points Stream

A sequence of points from discrete Euclidean space $[\Delta]^d$

$P = p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, \cancel{p_1}, \cancel{p_3}, \cancel{p_4}$



Example $\Omega = [\Delta]^2, \Delta = 10$

k-Median Clustering Points Streams

Approximate

$$\sum_{p\in P} \min_{z\in Z^*} \operatorname{dist}(p,z) \leqslant (1+\epsilon)\cdot \min_{C:|C|=k} \sum_{p\in P} \min_{c\in C} \operatorname{dist}(p,c)$$

- Dynamic Stream: allow insertion and deletions
- Points are from $[\Delta]^d$:
  - [Indyk '04]: introduction of dynamic streaming model, $O(k\epsilon^{-O(d)} \log \Delta)$ space, and $\widetilde{O}(n^{kd})$-time algorithm
  - [Frahling, Sohler '05]: $O(k^2\epsilon^{-O(d)} \operatorname{poly}\log \Delta)$ space, efficient algorithm
  - [Braverman, Frahling, Lang, Sohler, Y '17]: $O[k\epsilon^{-2} \operatorname{poly}(d \log \Delta \log(k\epsilon^{-1}))]$ space

First algorithm in space $\operatorname{poly}(d)$**.** Our algorithm is also time efficient!

k-Median Clustering

$$\sum_{p\in P} \min_{z\in Z^*} \operatorname{dist}(p,z) \leqslant (1+\epsilon)\cdot \min_{C:|C|=k} \sum_{p\in P} \min_{c\in C} \operatorname{dist}(p,c)$$

Insertion-Only Stream:

- [Guha, Meyerson, Mishra, Motwani, O'Callaghan '03]: $O(n^c/c)$ space for $O(2^{O(1/c)})$ approximation
- [Charikar, O'Callaghan, Panigrahy '03]: $O(k \log^2 n)$ space for $O(1)$ approximation
- [Braverman, Meyerson, Ostrovsky, Roytman, Shindler, Tagiku '11]: $O(k \log n)$ space for $O(1)$-approximation
- [Har-Peled, Mazumdar '04]: $O(k\epsilon^{-d} \log^{2d+2} n)$ space
- [Har-Peled, Kushal '05]: $O(k^2\epsilon^{-d})$ space
- [Chen '09]: $O(dk^2\epsilon^{-2} \log^8 n)$ space
- [Feldman, Langberg '11]: $O(dk\epsilon^{-2} \log^3 n)$ space

How? Coreset!

- Input point set P
- Coreset S, a multiset, each point s has a weight $w(s)$
- for all $Z \subset \mathbb{R}^d$ with $|Z| = k$

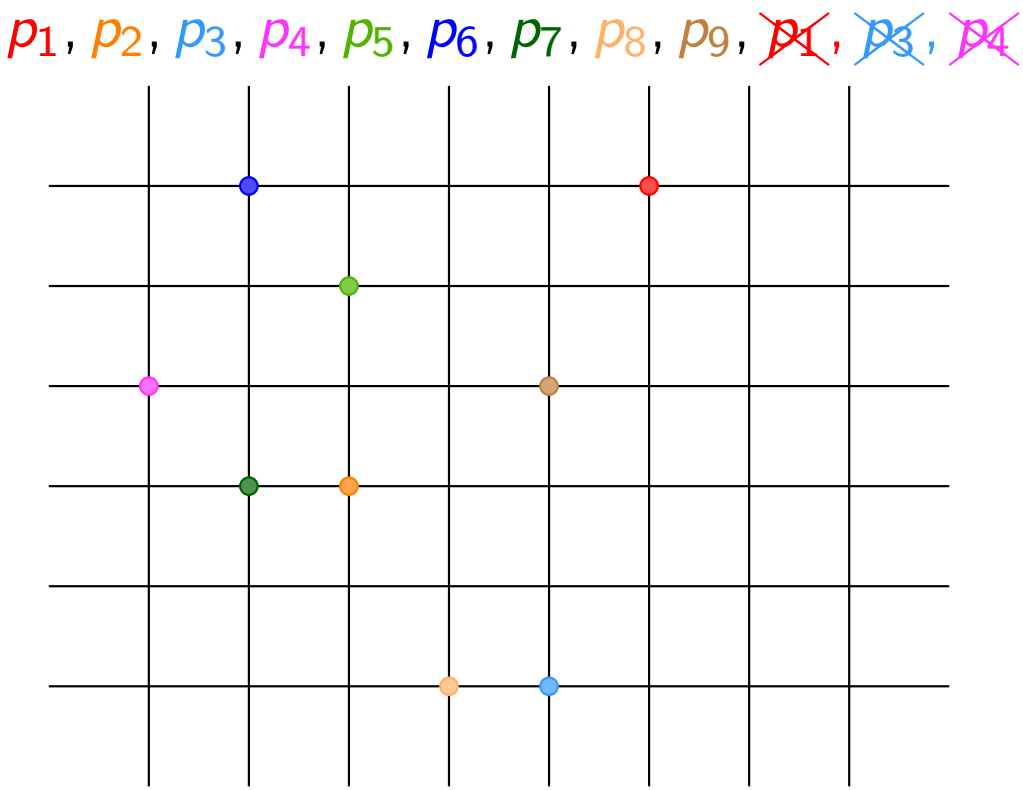
$$\text{cost}(S, Z) = (1 \pm \epsilon)\,\text{cost}(P, Z)$$

- $\text{cost}(S, Z) = \sum_{s \in S} w(s) \cdot \min_{z \in Z} \text{dist}(s, z)$
- Usually $|S| \ll |P|$
- Solve k-median using the small coreset (e.g. [Chen '06])

Our Result

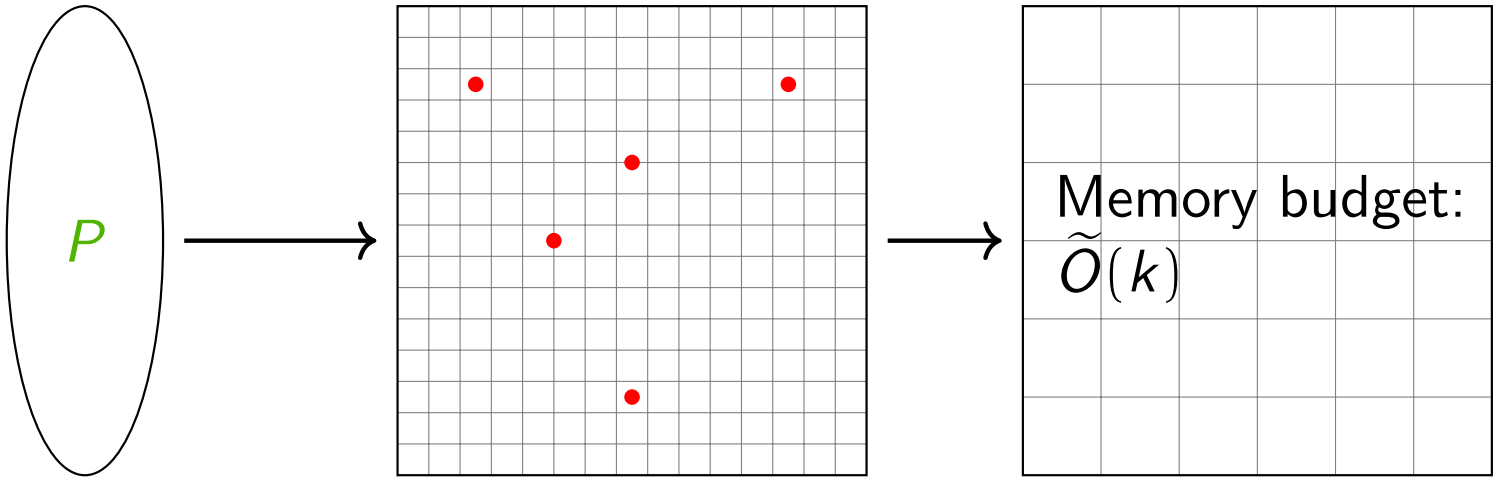
Theorem

There is a data-structure, supports insert and delete operations of points P from $[\Delta]^d$. It maintains a weighted set S, such that with probability $\geqslant 9/10$, S is a k-median ϵ-coreset.
$O(k\epsilon^{-2} \cdot \mathrm{poly}(d, \log \Delta, \log(k\epsilon^{-1})))$ spaces for both S and data-structure.



A New Coreset Framework

- Inspired by Frahling, Sohler' 05 and Chen' 09;
- An *offline oblivious construction* + *K-Set*;



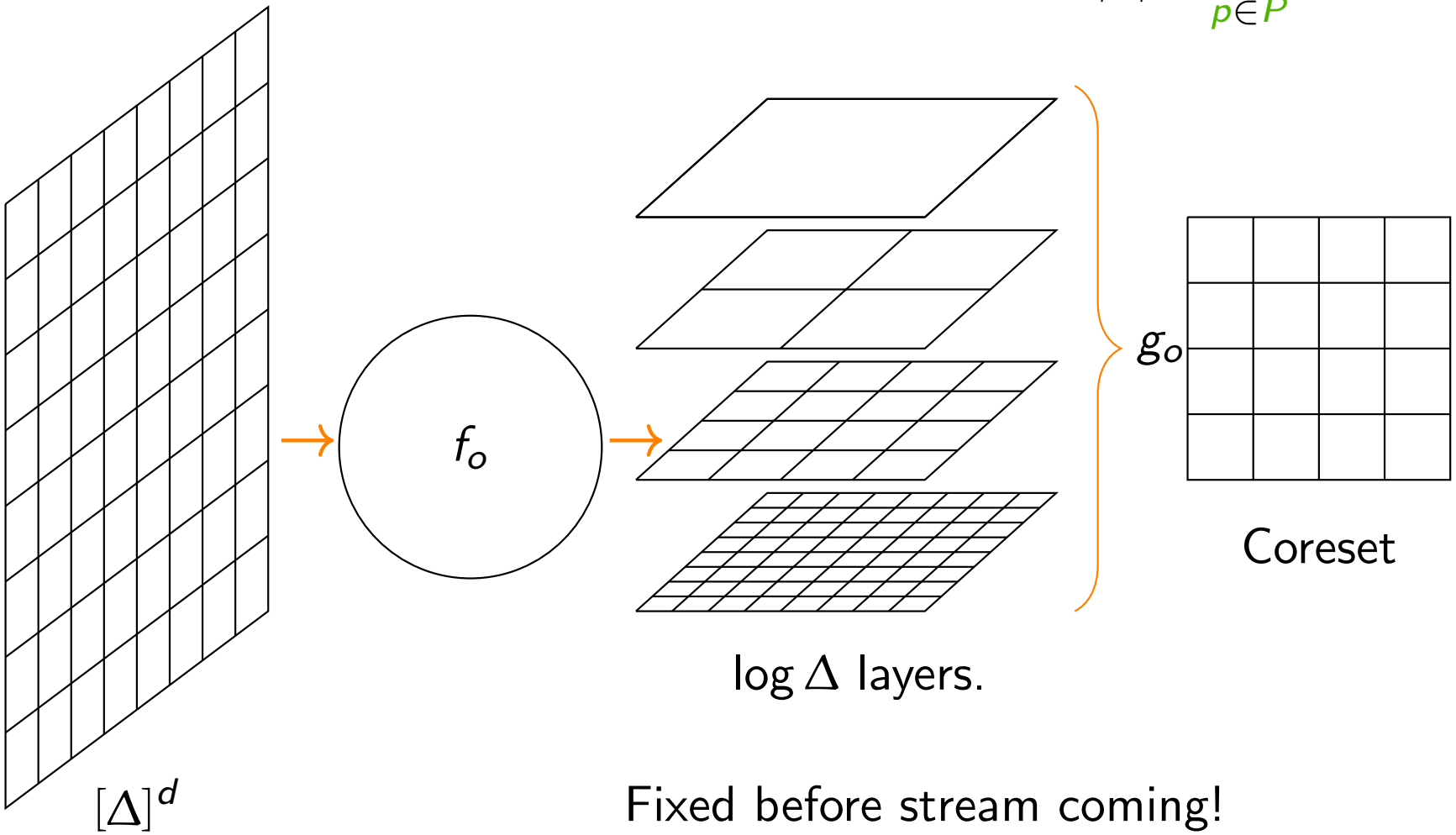
oblivous offline coreset construction K-Set

$P \Rightarrow Q$, $|Q|$ depends on a parameter o Preserve Q, if $|Q|$ is small.

Psuedo-randomized

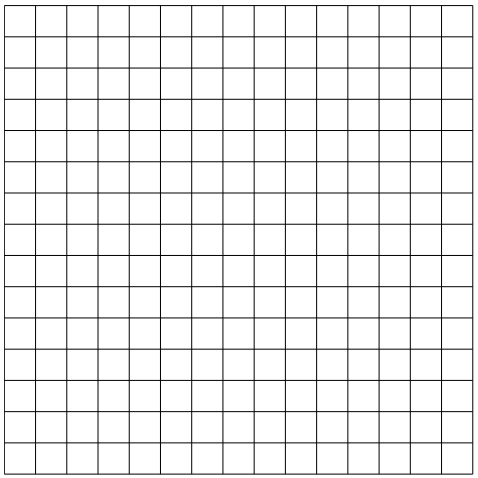
Coreset Framework: Oblivious Offline Coreset Construction

- A guess of the optimal cost function o.
- Two oblivious functions f_o, g_o. $\mathrm{OPT}(P) = \min_{Z:|Z|=k} \sum_{p \in P} \mathrm{dist}(p, Z)$

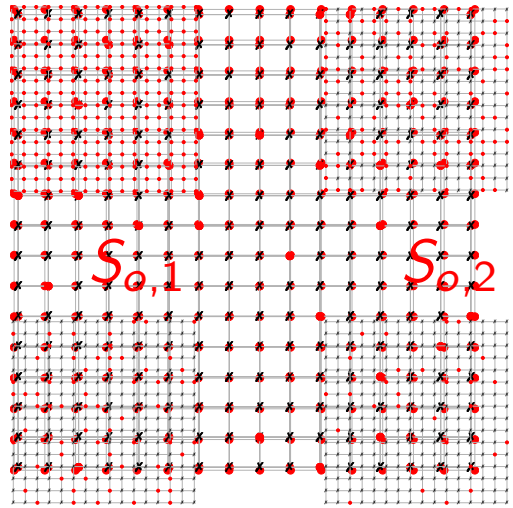


Coreset Framework: Offline Construction f_o

- A guess of the optimal cost function o.
- Two oblivious functions f_o, g_o. $\mathrm{OPT}(P) = \min_{Z:|Z|=k} \sum_{p\in P} \mathrm{dist}(p, Z)$



$f_{o,2}$



$[\Delta]^d$

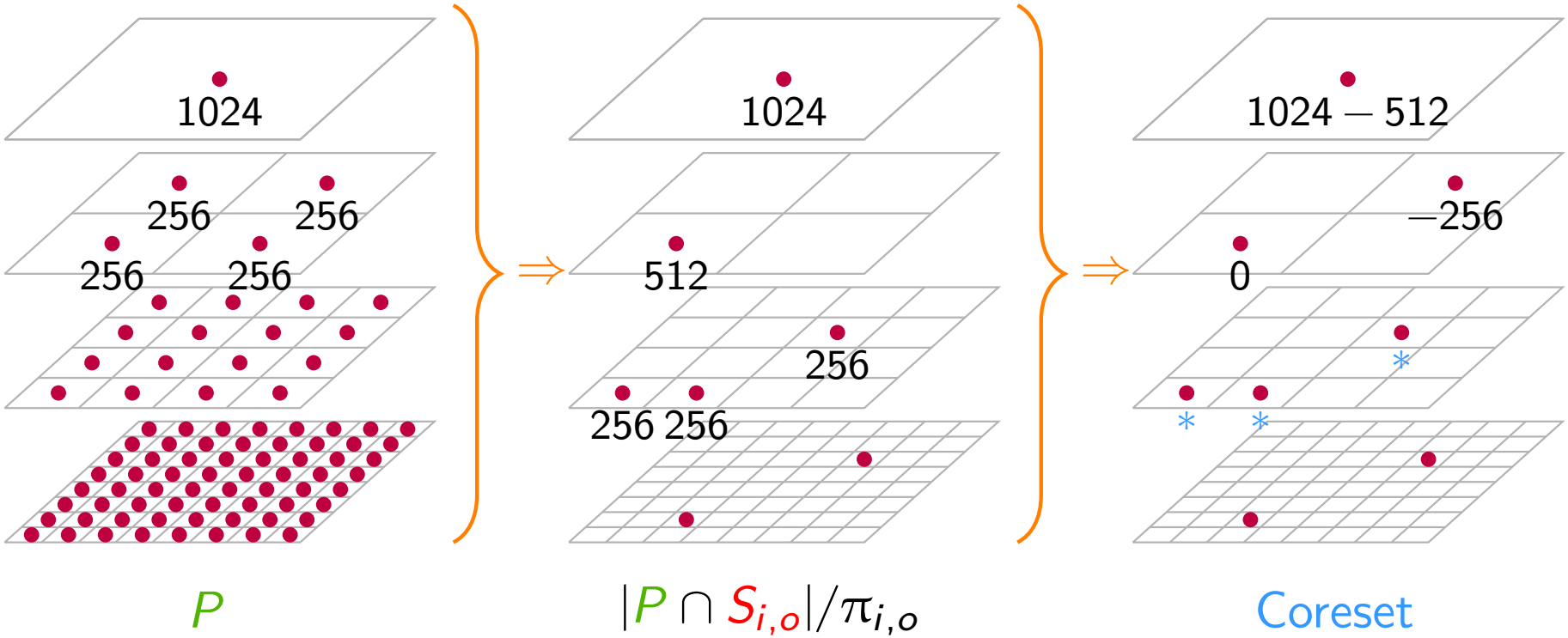
$S_{o,3}$ $S_{o,4}$ $\subset [\Delta]^d$

Each $f_{o,i}$ samples a set $S_{o,i} \subset [\Delta]^d$ with prob. $\sim \widetilde{O}\left(\frac{\Delta}{2^i \cdot o} \cdot \log \Delta^{dk}\right)$

Oblivious sampling: $S_{o,i} \cap P$

Coreset Framework: Offline Construction g_o

- A guess of the optimal cost function o.
- Two oblivious functions f_o, g_o.



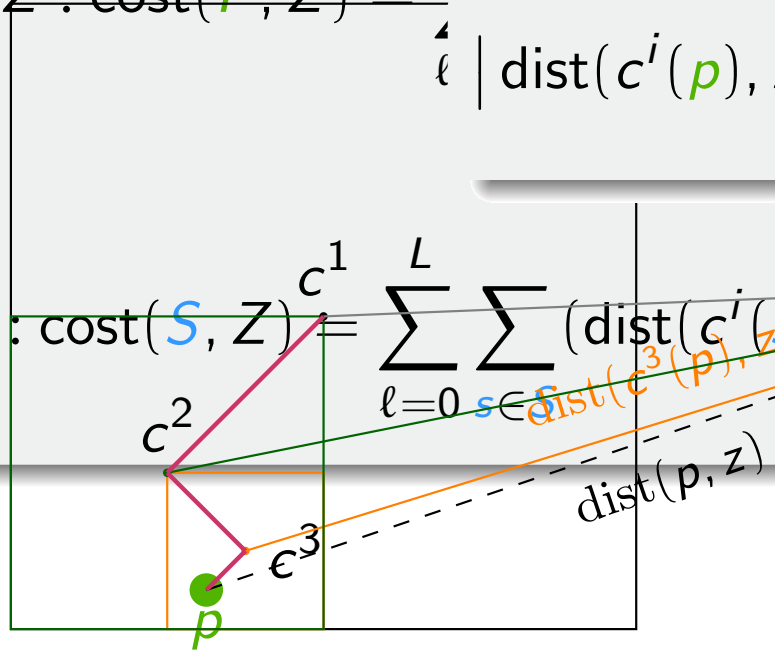
Coreset Framework: Why it works? Telescope sum.

Why the offline construction is a coreset? Telescope sum.

$$\forall Z : \mathrm{dist}(p, Z) = \sum_{\ell=0}^{L} \mathrm{dist}(c^i(p), Z) - \mathrm{dist}(c^{i-1}(p), Z)$$

$$\forall Z : \mathrm{cost}(P, Z) = \sum_{\ell=0}^{L} \sum_{p \in P} \left| \mathrm{dist}(c^i(p), Z) - \mathrm{dist}(c^{i-1}(p), Z) \right| \leqslant \mathrm{diam}(C^i)$$

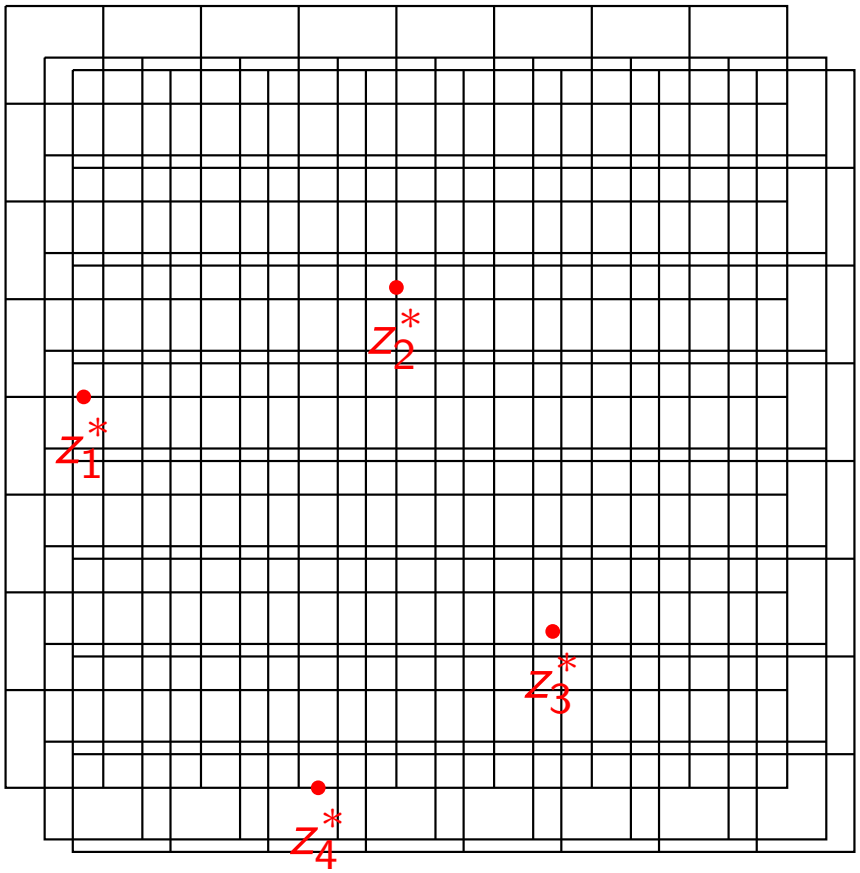
$$\forall Z : \mathrm{cost}(S, Z) = \sum_{\ell=0}^{L} \sum_{s \in S} (\mathrm{dist}(c^i(s), Z) - \mathrm{dist}(c^{i-1}(s)), Z)/\pi_{o,i}$$



Using Bernstein's inequality, a sample $S_{o,\ell} \cap P$ per level is sufficient

Coreset Framework: Why the Number of Non-empty Cells is Small?

- Randomly shift the grid!



- Let P be an arbitray point set (not known)
- $Z^* = (z_1^*, z_2^*, \ldots, z_k^*)$ be an optimal k-median solution
- After random shift, all centers in Z^* are *far* away from boundary! E.g., at least distance d_c
- #cells without a center but with more than OPT/d_c points is at most $O(1)$

Sampling probability: $\pi_{i,o} \approx d_c \cdot (\log \Delta^{dk})/(o\epsilon^2)$

Expected non-empty cells: $\widetilde{O}(k \cdot \mathrm{OPT}/(o\epsilon^2))$

Additive Approximation Error: $\widetilde{O}(\epsilon\,\mathrm{OPT})$

Compare with Chen'09 Framework

- Chen's framework is of size $\widetilde{O}(k^2)$: sampling points from $\widetilde{O}(k)$ partitions, each of $\widetilde{O}(k)$ points
  - Ours is of size $\widetilde{O}(k)$
- Chen's framework works for k-means and M-estimator
  - Ours only works for k-median
  - A recent result by [Song, Y, Zhong' 17]: nearly linear in k for k-means as well, simulates sensitivity-based sampling of [Feldman, Langberg' 11] and [Braverman, Feldman and Lang' 16]
- Positive weights? A Rectify Weights procedure in the paper – much more complicated

Summary

- Our Results:
  - A $\widetilde{O}(k\,\mathrm{poly}(d))$-space k-median coreset algorithm for high dimensional dynamic stream
  - The first algorithm achieving space $\mathrm{poly}(d)$ in dynamic setting
  - Does **not** extends to k-means ($k^2\,\mathrm{poly}(d)$ space is possible)
- Result for k-means and more:
  - [Song, Y, Zhong' 17] : $\widetilde{O}(k\,\mathrm{poly}(d))$ space
  - Sensitivity-based sampling
  - generalizes to other clustering objectives with $\widetilde{O}(k\,\mathrm{poly}(d))$ space