A Proofs of Results from Section 3

A.1 Proofs from Section 3.1

Let's look at the projection of a set of centers C to the set of *sites* S. This projection is a multiset defined as $\Pi(C,S) = \{\arg\min_{s\in S}(\|s-\mu\|_2) \mid \mu \in C\}$. Let $C' = \Pi(C,S)$ such that $C_i^{'}$ is the projected point corresponding to C_i. We define the projection infinity distance of C onto S as the maximum distance between these pairs, i.e. $\max_{i\in 1\dots|C|} \|C_i^{'} - C_i\|_2$, and denote it as $\|C_i - S\|_\infty$.

The following lemma comes from the folklore.

Lemma 5. *Let μ be the centroid (mean) of a set of points P, and $\hat{\mu}$ another point in space. Then $\ell(\hat{\mu},P) = |P|\cdot\|\mu-\hat{\mu}\|_2^2 + \ell(\mu,P)$.*

Corollary 3. *Let P be a set of points, $\boldsymbol{\mu}$ the set of k optimal centers, and $\hat{\boldsymbol{\mu}}$ k alternative centers such that $\|\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}\|_\infty \le \epsilon$ then $\ell(\hat{\boldsymbol{\mu}},P) \le |P|\cdot\epsilon^2 + \ell(\boldsymbol{\mu},P)$*

We now turn to the main theorem

Theorem 5. *Let $S = \{i\delta \mid 0 \le i \le \delta^{-1}\}^d \subset \mathbb{R}^d$ be the set of* sites *and let $\mathcal{E} = \{C \subset S \mid |C| = k\}$ be the set of experts (k-centers chosen out of the sites). Then, for any $0 < \alpha \le 1/4$, with $\delta = T^{-\alpha}$, the* MWUA *with the expert set $\mathcal{E}$ achieves regret $\tilde{O}(T^{1-2\alpha})$; the per round running time is $O(T^{\alpha kd})$.*

Proof. Following section (3.9) in Arora et al. (2012). The grid distance δ results with $n = \frac{1}{\delta^d}$ sites hence $N = \binom{n}{k}$ experts, which is $N = O(\delta^{-kd})$. Denote the regret with respect to the grid experts as the *grid-regret*. Running a single step in MWUA requires sampling and weight update, taking $O(kdN)$ time and the algorithm guarantees a grid-regret of at most $2\sqrt{\ln(N)T} = 2\sqrt{kd\ln(\delta^{-1})T}$.

Let $C_g = \Pi(C^\star T+1, S)$ be the closest grid sites to $C^\star T+1$. Because $\|c^*_{T+1} - C_g\|_\infty \le \frac{\sqrt{d}}{2}\delta$, (3) gives $\ell(C_g, X_{1:T}) - \ell(C^\star T+1, X_{1:T}) \le \frac{d\delta^2}{4}T$. Hence the regret of the algorithm is at most $2\sqrt{kd\ln(\delta^{-1})T} + \frac{d\delta^2}{4}T$, so choosing $\delta = T^{-\alpha/2}$ for $\alpha \in (0, \frac{1}{2}]$ yields an algorithm with $\tilde{O}(T^{1-\alpha})$ regret and a per step time complexity of $O(T^{\alpha kd/2})$. □

A.2 Proofs from Section 3.2

The following is a proof for Theorem 2

Proof. We will reduce the offline problem with point set P to the online problem by generating a stream X of T uniformly sampled points from P and running $\mathcal{A}$ on X; $\mathcal{A}$ generates T intermediate cluster centers $\{C_t\}_{t=1}^T$, and we return the best one with respect to the entire set P.

Denote the best offline k-means solution as $C^\star$– the optimal clustering for the stream $C^\star T+1$ may not coincide with the optimal offline clustering $C^\star$, but it must perform at least as good as $C^\star$ on $X_{1:T}$. Denote r the internal randomness of $\mathcal{A}$. The regret guarantee gives

$$\mathbb{E}_r[\mathbb{E}_X[\sum_{t=1}^T \ell(C_t,x_t) - \ell(C^\star T+1, X)]] \le T^\alpha$$

Using the linearity of expectation and noticing $C^\star T+1$ doesn't depend on r.

$$\begin{aligned}\sum_{t=1}^T \mathbb{E}_r[\mathbb{E}_X[\ell(C_t,x_t)]] &\le T^\alpha + \mathbb{E}_X[\ell(C^\star T+1, X)] \\ &\le T^\alpha + \mathbb{E}_X[\ell(C^\star, X)]\end{aligned}$$

Using $\forall C : \mathbb{E}_{x_t}[\ell(C,x_t)] = \frac{\ell(C,P)}{|P|}$

$$\sum_{t=1}^T \mathbb{E}_r[\mathbb{E}_X[\frac{\ell(C,P)}{|P|}]] \le T^\alpha + \frac{T\mathsf{OPT}}{|P|}$$

Define ϵ_t s.t. $\mathbb{E}_r[\mathbb{E}_X[\ell(C,P)]] = (1+\epsilon_t)\mathsf{OPT}$. Because $C^\star$ is optimal, we know $\forall t : \epsilon_t \geq 0$.

$$\sum_{t=1}^{T} \frac{(1+\epsilon_t)\mathsf{OPT}}{|P|} \leq T^\alpha + \frac{T\mathsf{OPT}}{|P|}$$

Rearranging

$$\sum_{t=1}^{T} \epsilon_t \leq \frac{|P|T^\alpha}{\mathsf{OPT}}$$

Denote $\epsilon^* = \min_t(\epsilon_t)$

$$T \cdot \epsilon^* \leq \frac{|P|T^\alpha}{\mathsf{OPT}}$$
$$\epsilon^* \leq \frac{|P|T^{\alpha-1}}{\mathsf{OPT}}$$

So provided $\mathsf{OPT}^{-1} = \text{poly}(|P|)$ one can choose $T = \text{poly}(|P|)$ s.t. ϵ^* is arbitrarily small. ϵ^* is a non-negative random variable, hence this condition suffices to produce an approximation algorithm with arbitrary ϵ for k-means. Awasthi et al. (2015) shows that this is NP-hard, finishing the proof. □

B Proofs from Section 4

The following is a proof for Theorem 3

Proof. We will present an algorithm that generates a stream of points on a line, for $k = 2$ and any value of T, such that the regret FTL obtains for the stream can be bounded from below by $c \cdot T$ where c is some constant. Extending the result to arbitrary k can be done by contracting the bounding box where the algorithm generates points by a factor of $2k$, and adding data points outside of it in $k-2$ equally spaced locations, to force the creation of $k-2$ clusters, one for each location.

The stream will consist of points in 3 locations $-\delta, 0, (1-\delta)$ for $\delta < \frac{1}{4}$. We will call C_t a $(-\delta)$-clustering if it puts all the points at $(-\delta)$ in one cluster and the rest in the other cluster, and a $(1-\delta)$-clustering puts all the point at $(1-\delta)$ in one cluster and the rest in the other cluster. We will define a stream that has a $(1-\delta)$ optimal $C^\star T + 1$ clustering.

We will keep the amount of points in 0 and $(-\delta)$ equal up to a difference of 1 at any step by alternative between the two any time we put points in one of them. There exists $n^* = f(\delta) = O(\delta^{-2})$ such that if there is one point at $(1-\delta)$ and n^* points in each of $0, -\delta$ then the optimal clustering is the $(1-\delta)$-clustering and for n^*+1 points in each of $0, -\delta$ the optimal clustering is the $(-\delta)$-clustering.

Our algorithm will start with a point in $(1-\delta)$ making C_t a $(-\delta)$-clustering. The next points will be added as follows– as long as C_t is a $(1-\delta)$-clustering add a point to $(-\delta)$ or 0, balancing them; if C_t has just become a $(-\delta)$-clustering, the next point will be $(1-\delta)$, inflicting an loss for FTL which depends only on δ hence it is $O(1)$, and making C_{t+1} a $(1-\delta)$-clustering again. Halt when we have T points.

When the algorithm halts a $(1-\delta)$-clustering is either the optimal clustering or is at most $O(1)$ below the optimal clustering, so we will say that $C^\star T + 1$ is a $(1-\delta)$-clustering without loss of generality. FTL will jump from $(-\delta)$-clustering to $(1-\delta)$-clustering $O(T/n^*)$ times, and suffer a loss of at least the loss $C^\star T + 1$ incurs at that step (because we balance $(-\delta)$ and 0, FTL moves the lower cluster away from the middle hence suffer a slightly larger loss than $C^\star T + 1$ at the next stage). This means that the regret is larger than $O(T/n^*)$ which is bounded from below by a linear function in T for a fixed δ. □

C Details and Proofs of Results from Section 5

C.1 Incremental Coreset

C.1.1 Maintaining an $O(1)$-Approximation for k-means

The first part of our algorithm is to maintain an *incremental* bicriteria $(O(k\log^2 T), O(1))$-approximate solution $S^{(t)}$ for the k-means problem. Namely, $S^{(t)}$ contains at most $O(k\log^2 T)$ centers and its loss is at most some constant times the loss of the best k-means solution using at most k centers. Moreover, the sequence $\{S^{(t)}\}_{t=1}^{T}$ is an *incremental clustering*: First, it must be a monotone sequence, i.e. for any time step t, $S^{(t-1)} \subseteq S^{(t)}$. Furthermore, if a data point x of the data stream is assigned to a center point $c \in S^{(t)}$ at time t, it remains assigned to c in any $S^{(\tau)}$ for $t \leq \tau \leq T$, i.e. until the end of the algorithm.

To achieve this, we will use the algorithm of Liberty et al. (2016) (for the k-median problem one can use Charikar et al. (2003)), which we call $\mathcal{A}_{cop}$, and whose performance guarantees are summarized by the following proposition, that is a restating from Liberty et al. (2016).

Theorem 6 (Liberty et al. (2016))**.** *Let $\Delta > 0$ and $\delta > 0$ be two parameters for $\mathcal{A}_{cop}$. Consider a set of T points such that the optimal k-means cost is at most $T\Delta$ and at least δ, then with probability at least 1/2, at any time t, the output of $\mathcal{A}_{cop}$ is a set of $O(k\log(T\Delta/\delta))$ centers which form an $O(1)$-approximation to the k-means problem defined by the t first elements.*

We claim that we may assume without loss of generality that the pairwise distance between input points is at least d/T^3 since making such an additive error of d/T^3 for each of the T input points yields an additive regret of $d/T^2 < d\sqrt(T)$, our total target bound of the regret. Thus, up to losing a constant factor in the regret bound, we may assume that $\delta > 1/T^3$ (since the case where all the input points are at at most k different locations can be easily detected and so the optimum solution pays at least d^2/T^6 and at most dT since the maximum distance is $\sqrt{d}$. We thus have that the Liberty et al. (2016)'s theorem yields a bicriteria $(O(k\log T), O(1))$-approximate solution.

C.1.2 Maintaining an explicit incremental coreset for k-means

Our algorithm maintains a coreset Q based on the solutions $S^{(t)}$ for $1 \leq t \leq T$, maintained by $\mathcal{A}_{cop}$. It makes use of coresets through the coreset construction introduced by Chen (2009) whose properties are summarized in the following theorem.

Theorem 7 (Thm. 5.5/3.6 in Chen (2009))**.** *Given a set P of T points in a metric space and parameters $1 > \varepsilon_{\mathsf{c}} > 0$ and $\lambda > 0$, one can compute a weighted set Q such that $|Q| = O(k\varepsilon_{\mathsf{c}}^{-2}\log T(k\log T + \log(1/\lambda))$ and Q is a $(1+\varepsilon_{\mathsf{c}})$-coreset of P for k-means clustering, with probability $1-\lambda$.*

We now review the coreset construction of Chen. Given a bicriteria (α,β)-approximation $S_0^{(t)}$ to the k-means problem, Chen's algorithm works as follows. For each center $c \in S_0^{(t)}$, consider the *cluster of* c, namely the set of points in P whose closest center in $S_0^{(t)}$ is c and proceed as follows. For each i, we define the i^{th} ring of c to be the set of points of cluster c that are at distance $[2^i, 2^{i+1})$ to c. The coreset construction simply samples $\zeta \geq \beta\varepsilon_{\mathsf{c}}^{-4}k\log T$ points among the points whose distance to c is in the range $[2^i, 2^{i+1})$ (if the number of such points is below ζ simply take the whole set). This ensures that the total number of points in the coreset is $\alpha k\zeta\log\Delta$, where Δ is the maximum to minimum distance ratio (which can be assume to be polynomial in n without loss of generality).

Our algorithm uses the bicriteria approximation algorithm $\mathcal{A}_{cop}$ of Section C.1.1 as follows. For each solution $S^{(t)}$ stored by $\mathcal{A}_{cop}$, the algorithm uses it to compute a coreset via a refinement of Chen's construction. Consider first applying Chen's construction to a solution $S^{(t)}$ maintained by $\mathcal{A}_{cop}$. Since $\mathcal{A}_{cop}$ is incremental, whatever decisions the algorithm has made until time t, center open and point assignment, will remain unchanged until the end. Thus, applying Chen's construction seems possible. The only problem is that for a given set $S^{(t)}$, a given center $c \in S^{(t)}$ and a given ring of c, we don't know in advance how many points are going to end up in the ring and so, what should be sampling rate so as to sample $\Theta(\zeta)$ elements uniformly, and define a mapping that assigns the same number of points to each coreset point of the ring.

To circumvent this issue, our algorithm proceeds as follows. We break the set of points in the given ring of a cluster center into *epochs*. The set of points which are the mth points inserted in the given ring of a cluster

center for $m \in [(1+\varepsilon)^j, (1+\varepsilon)^{j+1})$ forms the jth epoch of this ring. Our algorithm aims at sampling ζ points in each epochs. It immediately follows that the coreset size is at most $O(\varepsilon^{-1} \log T)$ times larger than the size of the coreset constructed by Chen's algorithm and so of total size $\text{poly}(k \log T \varepsilon^{-1})$. The quality of the coreset is at least as good as the one produced by Chen's algorithm. Then, our algorithm works as follows: for each epoch j consisting of less than $\log^2 n\zeta$ points, the algorithm keeps all the points (with weight 1), mapping each point to itself. For the other epochs, our algorithm keeps each point of epoch j with probability $(\zeta \log n)/(\varepsilon(1+\varepsilon)^j)$. Thus, an immediate application of multiplicative Chernoff bound shows that the number of points sampled during epoch j is at most $(1+\varepsilon)\zeta \log n$ with high probability. Taking a union bound over all epochs, all centers and all rings shows that the total size of the coreset computed is of the claimed size.

We now describe how the assignment ϕ is computed. The assignment provided is as follows. For a given epoch j of a ring of a cluster with center c, each point is assigned to the earliest inserted *non-full* sampled point of the epoch (or if no point of the epoch has been sampled so far), or to the center c if all sampled points of the epoch are full. A sampled point is said to be *full* if the total assigned point is more than $(1+\varepsilon)^2(\varepsilon(1+\varepsilon)^j)/(\zeta \log n)$.

Lemma 6. *For any $0 < \varepsilon < 1/2$, for any time t, the coreset at time t is an $O(\varepsilon)$-coreset.*

Proof. We break the analysis into *complete epochs* and *incomplete epoch*. An epoch j is complete if epoch $j+1$ is non-empty.

Consider a complete epoch j of a ring of a cluster with center c. We first argue that the number of points of this epoch assigned to c is at most $\varepsilon^2(1+\varepsilon)^j$.

The ℓth point of the epoch gets assigned to c if and only if the number of sampled points at time ℓ is smaller than

$$\delta_\ell = \frac{\ell(\zeta \log n)}{(1+\varepsilon)^2(\varepsilon(1+\varepsilon)^j)}$$

On the other hand, the expected number of sampled points when the ℓth point gets inserted is

$$\mu_\ell = \ell \cdot \frac{(\zeta \log n)}{(\varepsilon(1+\varepsilon)^j)}.$$

Thus, if $\ell \geq \varepsilon^2(1+\varepsilon)^j$, then an immediate application of multiplicative Chernoff bound yields that the number of sampled points at time ℓ is at least $(1 - 1/\log n)\mu_\ell$ with high probability, and so at least δ_ℓ. Conditioning on this even happening for all $\ell \geq \varepsilon^2(1+\varepsilon)^j$, it follows that non of the ℓth inserted points, for $\ell \geq \varepsilon^2(1+\varepsilon)^j$, is assigned to c.

Finally, we only have to worry about the ℓth inserted points, for $\ell \leq \varepsilon^2(1+\varepsilon)^j$. However, in the worst case all of them gets assigned to c, but this is at most $\varepsilon^2(1+\varepsilon)^j$ as desired. This implies that the total number of points of a given ring assigned to c is at most ε times the size of the ring. For this set of points, the cost increase in any solution is by triangle inequality at most twice the cost paid in the bicriteria approximation, and so at most an ε fraction of the cluster cost. Therefore at most an $O(\varepsilon)$ fraction of the optimum solution. Finally, condition on the event that the number of point sampled $\hat{\zeta}$ is in $[(1-\varepsilon)\zeta \log n; (1+\varepsilon)\zeta \log n]$ (since this event happens with high probability, we will remove the conditioning by taking a union bound over the probability of failure at the end). Hence, the weight of each coreset point in our construction is within a $(1+\epsilon)$ factor of its weight in Chen's construction and so the coreset guarantees induced by Chen's construction is preserved up to a $(1+\epsilon)$ factor.

Finally turn to an incomplete epoch j: the distance between the locations the points of epoch j are mapped to in the coreset and their original locations is at most twice their cost in the constant factor approximation. Since an incomplete epoch is of size at most ε times the total number of points already inserted in the ring, the triangle inequality implies that the total cost difference for the points in the incomplete error is at most 2ϵ times the total cluster cost. Overall clusters of the bicriteria approximation, this error is at most $O(\varepsilon)$ times the cost of the optimum solution. We conclude that C_t is an ε-coreset. □

C.2 Hierarchical Region Decomposition

C.2.1 Detailed Algorithm Description

Given a sequence of points in $\mathbb{R}^d$, we describe an algorithm that maintains a hierarchical region decomposition with d-dimensional hypercube regions as follows. Let $\varepsilon_{\mathsf{hrd}} > 0$ be a parameter s.t. $\frac{\varepsilon_{\mathsf{hrd}}}{2T^3\sqrt{d}}$ is a power of 2, and

denote $\delta_t = \frac{\varepsilon_{\mathsf{hrd}}}{2t^3}$ and $\delta_T = \frac{\varepsilon_{\mathsf{hrd}}}{2T^3}$. One can obtain a *Full Grid* of side length $\frac{\varepsilon_{\mathsf{hrd}}}{2T^3\sqrt{d}}$ (diameter δ_T) by repeated halving of all regions of space; Denote the region tree structure that corresponds to this process as the *Full Grid Tree*. Consider a step t, $R \in \mathcal{R}_t$ and $x \in [0,1]^d$. Denote the diameter of R as ΔR, and $R(x) = \min_{p \in R} \|p - x\|_2$ the distance between R and x. Notice that if $x \in R$ then $R(x) = 0$. We define the *refinement criteria induced by x at time t* as $q(R)$, which takes the value true if and only if the diameter of R is smaller or equal $\max(\varepsilon_{\mathsf{hrd}} \cdot R(x)/2, \delta_t)$. At a given time t, a new point x_t is received and the hierarchical region decomposition obtained at the end of time $t-1$, $\mathcal{H}_{t-1}$, is refined using Algorithm 2, which guarantees that all the new regions satisfy the refinement criteria induced by all the points $X_{1:t}$ at the corresponding insertion times.

C.2.2 Structural Properties

Proposition 2. *Consider the hierarchical region decomposition $\{\mathcal{R}_1, \ldots, \mathcal{R}_t\}$ produced by the algorithm at any time t. Consider a region $R \in \mathcal{R}_{t-1}$ and let ΔR be the diameter of region R, then the following holds. Either region R belongs to $\mathcal{R}_t$ or each child region of R in $\mathcal{R}_t$ has diameter at most $\frac{1}{2}\Delta R$.*

Corollary 4. *Consider $\{R_t \in \mathcal{R}_t\}_{t=1}^T$ such that $\forall t: \quad R_{t+1} \subseteq R_t$, a sequence of nested regions of length T. We say that such a sequence cannot be refined more than Λ times, i.e. $|\{t | R_{t+1} \neq R_t\}| \leq \Lambda$. Proposition 2 along with the fact that the algorithm does not refine regions with diameter smaller than δ_T give us that*

$$\Lambda \leq -\log\left(\delta_T/\sqrt{d}\right) = -\log\left(\frac{\varepsilon_{\mathsf{hrd}}}{2T^3\sqrt{d}}\right).$$

The proof of the following is provided in Appendix C.

Lemma 7. *For any stream, using the above algorithm, we have that the total number of regions that are added at step t is at most $(9\sqrt{d}/\varepsilon_{\mathsf{hrd}})^d \log(T^3)$.*

Proof. At any time step t, each point x_t corresponds to a refinement criteria at time t over regions in the *Full Grid Tree*, namely, $q(\cdot)$, s.t. there exists a frontier (i.e. the leaves of a subtree of the Full Grid Tree) that separates the vertices from above, which have $q(R) =$ False, and the vertices below have $q(R) =$ True. This frontier can be thought of as the minimal refinement requirement along each path in the Full Grid Tree that corresponds to adding x_t. In order to satisfy the requirements of all the points in the stream and have that all the regions in $\mathcal{R}_t$ have $q(R) =$ True (for the corresponding time steps $t' \leq t$) the algorithm iterates the existing frontier in the Full Grid Tree that corresponds to the current region decomposition and extends it to match the requirements due to $X_{1:t-1}$ together with the new minimum requirements introduced by the new point. Hence the algorithm removes a subset of regions from the $\mathcal{R}_{t-1}$ and replaces them with a subset of the frontier that corresponds to x_t. We now turn to prove that the frontier in the Full Grid Tree of any point in space is of size $(9\sqrt{d}/\varepsilon_{\mathsf{hrd}})^d \log(T^3)$, proving the lemma.

Let x denote an arbitrary point in space whose frontier size we are trying to bound. The frontier is made up of an area in space close to x that contains only regions of the minimum diameter δ_T, where we use T instead of t as it lower bounds for the diameter. When the distance from x is larger than $r_{hrd} = 2\delta_T/\varepsilon_{\mathsf{hrd}}$ the dominant term in the refinement criteria is $\varepsilon_{\mathsf{hrd}} \cdot R(x)/2$, hence this is a hypersphere of radius r_{hrd} centered around x.

Outside of the r_{hrd} sphere we will have thick shells with inner radius $r_{hrd} \cdot 2^i$ and outer radius of $r_{hrd} \cdot 2^{i+1}$ where i is some nonnegative integer, where the refinement criteria requires that the maximum diameter will be $\delta_T \cdot 2^i$.

Consider a bounding box for the shell that corresponds to $i \geq 0$. The sides of such a hypercube are of length $2 \cdot r_{hrd} \cdot 2^{i+1}$. In order to partition this hypercube to regions of diameter $\delta_T \cdot 2^i$ we require $(8\sqrt{d}/\varepsilon_{\mathsf{hrd}})^d$ regions. If we iterate the shells from the outermost shell inward, we can assume that a sphere twice as large as shell i was already partitioned to regions with half the resolution of shell i before iterating shell i.

Now we can account for the fact that the spheres, of radius r, are not aligned with the regions of the Full Grid Tree for the corresponding resolution/depth. Let us extend the bounding box of shell i such that it is aligned with the full grid in the resolution of the next shell $(i+1)$. This means that the edge length of the box is enlarged by at most two units of edge length $\delta_T \cdot 2^i/\sqrt{d}$ from each side, resulting in $(4 + 8\sqrt{d}/\varepsilon_{\mathsf{hrd}})^d \leq (9\sqrt{d}/\varepsilon_{\mathsf{hrd}})^d$ regions, for sufficiently small $\varepsilon_{\mathsf{hrd}}$.

There are at most $\log(1/r_{hrd}) = \log(2T^3)$ shells, and noticing that the innermost shell accounts for the core, this gives the bound. □

Corollary 5. *Let $R \in \mathcal{R}_t$ be any region at step t and $S = \{R' \in \mathcal{R}_{t+1} | R' \subseteq R\}$ be the set of regions that refine R in the next time step. Defining $\beta = \log \max_{t,R} |S|$ which we refer to as the log max branch, we have*

$$\beta = \log \max_{t,R} |S| \leq \log \left(\left(\frac{9\sqrt{d}}{\varepsilon_{\mathsf{hrd}}} \right)^d \log(T^3) \right),$$

due to Lemma 7. Furthermore for sufficiently large T we have that $\beta \leq d \cdot \Lambda$.

We will now present a few properties related to the approximation of the k-means problem.

Lemma 8. *Let $\varepsilon_{\mathsf{hrd}} > 0$. Consider an online instance of the weighted k-means problem where a new point and its weight are inserted at each time step. Let x_t be the weighted point that arrives at time t, such that its weight is bounded by t. Consider the hierarchical region decomposition with parameter $\varepsilon_{\mathsf{hrd}}$ produced by the algorithm $\mathcal{H}_t = \{\mathcal{R}_1, \ldots, \mathcal{R}_t\}$, for some time step t.*

Consider two multisets of k centers $S = \{c_1, \ldots, c_k\}, S' = \{c'_1, \ldots, c'_k\}$ such that for all $i \in \{1 \ldots k\}$, c_i and c'_i are contained in the same region of the Region Decomposition of step t. Then, the following holds.

$$\forall 1 \leq \tau < t, \quad \ell(S', x_\tau) \leq (1 + \varepsilon_{\mathsf{hrd}})\ell(S, x_\tau) + \varepsilon_{\mathsf{hrd}}/\tau^5$$

Proof. Fix τ and focus on c_i the cluster center in S closest to x_τ, and its corresponding c'_i.

consider the region R containing c_i. If R also contains x_τ then, since x_τ has been inserted to the stream and so R has diameter of at most δ_t. Because c'_i is also contained in this region the weighted loss is at most $\tau \cdot (\frac{\varepsilon_{\mathsf{hrd}}}{2\tau^3})^2$.

Thus, we turn to the case where R does not contain x_τ, hence $||x_\tau - c_i|| > 0$. Denote $r = ||x_\tau - c_i||$. By definition of the algorithm and again since x_τ has already been inserted to the stream, this region must be a hypercube with a diameter of at most $\Delta R \leq \max(\varepsilon_{\mathsf{hrd}} \cdot r/2, \frac{\varepsilon_{\mathsf{hrd}}}{2\tau^3})$ since otherwise, the region is refined again when x_τ is inserted. Cauchy Schwartz gives us

$$||x_\tau - c'_i||^2 \leq ||x_\tau - c_i||^2 + 2||x_\tau - c_i|| \cdot ||c_i - c'_i|| + ||c_i - c'_i||^2 \leq r^2 + r \cdot \Delta R + (\Delta R)^2$$

If $r \leq 1/\tau^3$ then $\Delta R \leq \varepsilon_{\mathsf{hrd}}/2\tau^3$, hence

$$||x_\tau - c'_i||^2 \leq ||x_\tau - c_i||^2 + \varepsilon_{\mathsf{hrd}}/\tau^6$$

Otherwise, $\varepsilon_{\mathsf{hrd}} \cdot r/2 > \frac{\varepsilon_{\mathsf{hrd}}}{2\tau^3}$ hence $\Delta R \leq \varepsilon_{\mathsf{hrd}} \cdot r/2$

$$||x_\tau - c'_i||^2 \leq ||x_\tau - c_i||^2 (1 + \varepsilon_{\mathsf{hrd}})$$

Hence, accounting for the τ weight one gets

$$\ell(S', x_\tau) \leq \ell(S, x_\tau)(1 + \varepsilon_{\mathsf{hrd}}) + \varepsilon_{\mathsf{hrd}}/\tau^5$$

□

For shorthand, we write $S^* = C^*_{T+1}$, the best solution in hindsight. The next lemma follows directly from applying Lemma 8 to the approximate centers of S^* induced by $\mathcal{H}$, and summing over t.

Lemma 9. *For the optimal set of candidate centers in hindsight S^* and $\tilde{S}_t$ the approximate centers induced by the Hierarchical Region Decomposition at time step t, for a weighted stream $X_{1:t}$*

$$(1 - \varepsilon_{\mathsf{hrd}})\mathsf{OPT} - 2\varepsilon_{\mathsf{hrd}} \leq \sum_{t=1}^{T} \ell(\tilde{S}_{t+1}, x_t) \leq (1 + \varepsilon_{\mathsf{hrd}})\mathsf{OPT} + 2\varepsilon_{\mathsf{hrd}}.$$

As Lemma 4 gives that that $\tilde{S}_{t+1} \neq \tilde{S}_t$ at most $k \cdot \Lambda$ times (each of the k regions may be refined Λ times), and the loss is bounded by d, then along with Lemma 9 we get the following Corollary 1.

C.3 MTMW

The following is a proof for Lemma 2

Proof. We will show a proof by induction on the subtree height h. For $h = 1$ we have $\tilde{V} = \{v\}$ hence the property holds. For $h + 1$ we can denote the children of v as U and use the induction hypothesis for each child's subtree and get that

$$\sum_{v' \in \tilde{V}} \mathcal{M}(v') = \sum_{v' \in U} \mathcal{M}(v') = \sum_{v' \in U} \mathcal{M}(v)/|U| = \mathcal{M}(v)$$

Where the last equality used the recursive mass definition. □

The following is a proof for Theorem 1

Proof. For a rooted path $p = (v_1 \dots v_T)$ define the *uniform weight of p at time t* with respect to the stream $X_{1:t}$, as $u_p^{(t)} = \prod_{\tau=1}^{t-1}(1 - \eta \ell_1(v_\tau, x_\tau))$, which corresponds to the weight MWUA with uniform initial weights would associate that expert p before witnessing x_t. We extend this notation for any rooted path p, as long as it has length at least t. In our modified algorithm, the weight associated to expert p at step t is $w_p^{(t)} = \mathcal{M}(p) u_p^{(t)}$. Denote $p(v)$ the path from the root to vertex v. Using the modified initial weight, the probability MWUA will output the prediction that corresponds to any node v_t at depth t at step t, due to choosing some path that contains it, is given by (before normalizing)

$$w_{v_t}^{(t)} = \sum_{p \in \mathcal{P}(\mathcal{T}_T): v_t \in p} w_p^{(t)} = \sum_{p \in \mathcal{P}(\mathcal{T}_T): v_t \in p} u_p^{(t)} \cdot \mathcal{M}(p) = u_{p(v_t)}^{(t)} \cdot \mathcal{M}(v_t)$$

The weight and mass are functions of known quantities at step t, where the equality is from the definition of $u_p^{(t)}$ and due to Lemma 2. Following the proof of Theorem (2.1) of Arora et al. (2012), we define

$$\Phi^{(t)} = \sum_{p \in \mathcal{P}(\mathcal{T}_T)} w_p^{(t)} \qquad (\mathbf{m}^{(t)})_p = \ell_1(p, x_t) \qquad (\mathbf{p}^{(t)})_p \propto w_p^{(t)}$$

The potential, the loss vector, and the normalized probability vector, respectively. For any path p, due to the fact $\Phi^{(1)} = 1$ we can modify Equation (2.2) in (Arora et al., 2012) to

$$\Phi^{(T+1)}/\mathcal{M}(p) \leq (1/\mathcal{M}(p)) \exp(-\eta \sum_{t=1}^{T} \mathbf{m}^{(t)} \cdot \mathbf{p}^{(t)})$$

Using the weight of p after the last step as lower bound for $\Phi^{(T+1)}$, we change Equation (2.4) to

$$\Phi^{(T+1)}/\mathcal{M}(p) \geq w_p^{(T+1)}/\mathcal{M}(p) = u_p^{(T+1)}$$

Hence the rest of the proof is left intact, where n (the amount of experts) is replaced with $(1/\mathcal{M}(p))$, so the regret changes to

$$\sqrt{-T \ln(\mathcal{M}(p))}$$

The running time is composed of sampling a path which is done by iterating the vertices of depth t and calculating their weights, which finishes the proof. □

C.4 Combining the Components

The following is a proof for Lemma 3

Proof. Denote the times where $\tilde{S}_t \neq \tilde{S}_{t-1}$ as $t_1, t_2 \dots t_{T_0}$, where we consider $t_1 = 1$ for this purpose, and T_0 is the amount of different time steps where this occurs. For ease of notation we denote $t_{T_0+1} = T + 1$. Furthermore, we use $X_{[a:b)}$ to denote $X_{a:b-1}$. As $\tilde{S}_t = \tilde{S}_{t-1}$ for any other time step we have that $\tilde{S}_{t_{i+1}-1} = \tilde{S}_{t_i}$ hence

$$\sum_{t=1}^{T} \ell(\tilde{S}_t, x_t) = \sum_{i=1}^{T_0} \ell(\tilde{S}_{t_i}, X_{[t_i:t_{i+1})}) \leq \sum_{i=1}^{T_0} \ell(\tilde{S}_{t_{i+1}-1}, X_{[t_i:t_{i+1})})$$

Excluding the T_0 points that resulted with a region refinement from the sum, and using the fact that the loss is bounded by d

$$\begin{aligned}\sum_{t=1}^{T}\ell(\tilde{S}_t,x_t) &\leq \sum_{i=1}^{T_0}\ell(\tilde{S}_{t_{i+1}-1},X_{[t_i:t_{i+1}-1)}) + dT_0\\ &\leq \sum_{i=1}^{T_0}\Big(\ell(\tilde{S}_{t_{i+1}-1},X_{[1,t_{i+1}-1)}) - \ell(\tilde{S}_{t_{i+1}-1},X_{[1,t_i)})\Big) + dT_0\end{aligned}$$

As the loss is nonnegative

$$\sum_{t=1}^{T}\ell(\tilde{S}_t,x_t) \leq \sum_{i=1}^{T_0}\Big(\ell(\tilde{S}_{t_{i+1}-1},X_{[1,t_{i+1}-1)}) - \ell(\tilde{S}_{t_{i+1}-1},X_{[1,t_i-1)})\Big) + dT_0$$

Using the coreset property

$$\begin{aligned}\sum_{t=1}^{T}\ell(\tilde{S}_t,x_t) \leq &\sum_{i=1}^{T_0}(1+\varepsilon_{\mathsf{c}})\ell(\tilde{S}_{t_{i+1}-1},\chi(X_{[1,t_{i+1}-1)}))-\\ &\sum_{i=1}^{T_0}(1-\varepsilon_{\mathsf{c}})\ell(\tilde{S}_{t_{i+1}-1},\chi(X_{[1,t_i-1)})) + dT_0\end{aligned}$$

As the Hierarchical Region Decomposition $\mathcal{H}_{t_{i+1}-1}$ was constructed according to $\chi(X_{1:t_{i+1}-2})$ (which contains $\chi(X_{1:t_i-2})$), one can use Corollary 9 and get

$$\begin{aligned}\sum_{t=1}^{T}\ell(\tilde{S}_t,x_t) \leq &\sum_{i=1}^{T_0}(1+\varepsilon_{\mathsf{c}})(1+\varepsilon_{\mathsf{hrd}})\ell(S^*,\chi(X_{[1,t_{i+1}-1)}))-\\ &\sum_{i=1}^{T_0}(1-\varepsilon_{\mathsf{c}})(1-\varepsilon_{\mathsf{hrd}})\ell(S^*,\chi(X_{[1:t_i-1)})) + dT_0 + 2\varepsilon_{\mathsf{hrd}}\end{aligned}$$

Now the series telescope, and along with $\varepsilon_{\mathsf{c}},\varepsilon_{\mathsf{hrd}} < 1$ rearranging gives

$$\sum_{t=1}^{T}\ell(\tilde{S}_t,x_t) \leq \sum_{i=1}^{T_0}((\varepsilon_{\mathsf{c}}+\varepsilon_{\mathsf{hrd}})\ell(S^*,\chi(X_{[1,t_{i+1}-1)}))) + \ell(S^*,\chi(X_{[1,T)})) + 2dT_0$$

As the loss is nonnegative we can extend the substreams until T, and using the coreset property

$$\sum_{t=1}^{T}\ell(\tilde{S}_t,x_t) \leq ((1+\varepsilon_{\mathsf{c}}) + 8T_0(\varepsilon_{\mathsf{c}}+\varepsilon_{\mathsf{hrd}}))\ell(S^*,X_{1:T}) + 2dT_0$$

finally, with Lemma 4 we have that $T_0 \leq k\cdot\Lambda$, which finishes the proof. □

The following is a proof for Lemma 4

Proof. Using Lemma 4 we can see that $\log(\deg(v_t))$ can take a non zero value at most $k\Lambda$ times, each is bounded by $k\cdot\beta$ as each node can be replaced by $((9\sqrt{d}/\varepsilon_{\mathsf{hrd}})^d\log(T^3))^k$ children nodes in the k-tree. □

The following is a proof for Theorem 4

Proof. Using the bounds for $\varepsilon_{\mathsf{hrd}},\varepsilon_{\mathsf{c}}$ we have that

$$\Lambda = \log(\frac{2T^3\sqrt{d}}{\varepsilon_{\mathsf{hrd}}}) \leq \log(\frac{4adT^6k^2}{\varepsilon^2}) \leq 2\log(\frac{akT^3\sqrt{d}}{\varepsilon^2})$$

Next we will show that $\varepsilon_{\mathsf{hrd}}$ is of the same order as $\varepsilon/k\Lambda$

$$\begin{aligned}\varepsilon_{\mathsf{hrd}} \cdot k\Lambda &\leq \frac{\varepsilon^2}{ak^2\log(T^3\sqrt{d})} \cdot 2k\log(\frac{ak\sqrt{d}T^3}{\varepsilon^2}) \leq \\ &\leq \frac{2\varepsilon^2}{ak\log(T^3\sqrt{d})}\log(T^3\sqrt{d})k\log(\frac{a}{\varepsilon^2}) \leq 4(\frac{\varepsilon}{\sqrt{a}})^2\log(\frac{\sqrt{a}}{\varepsilon}) \leq \frac{2\varepsilon}{\sqrt{a}}\end{aligned}$$

As stated in Corollary 5, for sufficiently large T we have $\beta < d \cdot \Lambda$, hence, along with Lemma 4, we have that MWUA has a regret w.r.t. the best path of $k\Lambda\sqrt{dT} \leq 2k\log(\frac{akT^3\sqrt{d}}{\varepsilon^2})\sqrt{dT}$. For $a \geq 34^2$ we get $\frac{2\varepsilon}{\sqrt{a}} \leq \frac{\varepsilon}{17}$, which makes Lemma 3 bound the ε-Approximate Regret of the best path. hence the final regret of

$$O\left(k\log\left(\frac{akT^3\sqrt{d}}{\varepsilon^2}\right)\sqrt{d^3T}\right) + \varepsilon \cdot \mathsf{OPT}$$

Furthermore, as $|\mathcal{Q}_T| \leq k^2\log^4(T)\varepsilon_{\mathsf{c}}^{-4} = O(k^{10}\log^8(T)\log^4(d)\varepsilon^{-8})$, using Lemma 7 with $N = |\mathcal{Q}_T|$ we can bound the k-tree vertices by the amount of leaves times the depth T, and have that the runtime is

$$O(T(|\mathcal{Q}_T|(32\sqrt{d}/\varepsilon_{\mathsf{hrd}})^d\log(T^3))^k) = T \cdot O(k^{10}\log^9(T)\log^4(d)\varepsilon^{-8})^k O(\sqrt{d}k^2\log(T)\varepsilon^{-2})^{dk}$$

□