Missing Details of Section: Algorithm

Proof of Theorem 15. An important primitive in the MPC model is sorting (Goodrich 1999; Goodrich, Sitchinava, and Zhang 2011), and it can be done in $O(1)$ rounds. Now let us look at how to implement Algorithm 3 in the MPC model. For the core point set construction stage, we need to run Algorithm 2. In Algorithm 2, we can use sorting primitive to group points by their rounding value $\hat{h}(\cdot)$. Then each point can be handled separately. A point will be added into core point set, if its group has size larger than k. In the stage of constructing the graph over core points, we can make each core point t copies. This step will increase the space needed by a factor at most t. For the i-th copies of points, we can run sorting primitive again to group points via their hash value $h_i(\cdot)$. The points with the same hash value will be mapped to the same machine, and we are able to create a star in G for each group. In the stage of handling non-core points, we still make t copies of each point, and run sorting to group the points. For each group, a non-core point checks whether there is a core point in the group. Each non-core point may find at most t core points. Then we can do sorting again to make these t core points which are associated to the same non-core point to the same machine. Then, we can choose at most one core point for each non-core point. Since we only need to run sorting primitive in each step, the total number of rounds needed before final clustering step is $O(1)$. $\square$

Missing Details of Section: Theoretical Analysis for Density Level Set Estimation

Proof of Lemma 8. Suppose there is a point $x \in B(L_f(\lambda), r_c)$ with $f(x) < \lambda - \lambda_c$. Due to Assumption 5, $\mathcal{X}$ is convex and f is continuous, we can find a point x' such that $f(x') = \lambda - \lambda_c$ and $d(x', L_f(\lambda)) < r_c$. By Assumption 7, we have $f(x') \geq \lambda - C_2 \cdot d(x', L_f(\lambda))^\beta > \lambda - C_2 \cdot r_c^\beta = \lambda - \lambda_c$ which leads to a contradiction. Thus, $\forall x \in B(L_f(\lambda), r_c) \setminus L_f(\lambda)$, we have $x \in L_f(\lambda - \lambda_c) \setminus L_f(\lambda)$. By applying Assumption 7, we obtained the desired bound. $\square$

Detailed parameters setup. In particular, in our analysis, we will choose k to be in the range:

$$\begin{aligned} &100 \cdot (100\sqrt{D})^{2\beta+D} \left(\frac{\lambda}{\lambda_c}\right)^2 \left(\frac{C_2}{C_1}\right)^2 \cdot C_{\delta,n}^2 \sqrt{D \log n} \\ &\leq k \leq \left(\frac{C_{\delta,n}}{C_2}\right)^{\frac{2D}{2\beta+D}} \left(\frac{1}{4D}\right)^{\frac{\beta D}{2\beta+D}} \lambda^{\frac{2\beta+2D}{2\beta+D}} n^{\frac{2\beta}{2\beta+D}}. \end{aligned}$$

We choose $t = 100 \log(n/\delta)$.

Proof of Theorem 13. Lemma 12 shows that $\sup_{x \in L_f(\lambda)} d(x, C) \leq \frac{1}{2}\left(\frac{\lambda}{C_2} \cdot C_{\delta,n}/\sqrt{k}\right)^{1/\beta}$. Lemma 10 shows that $\sup_{x \in C} d(x, L_f(\lambda)) \leq 2\left(\frac{\lambda}{C_1} \cdot 10 C_{\delta,n}/\sqrt{k}\right)^{1/\beta}$. $\square$

If we choose maximum possible k, then the quantity in Theorem 13 is at most

$$2 \cdot 10^{1/\beta} \cdot \left(\frac{1}{C_1}\right)^{1/\beta} C_2^{\frac{D}{\beta(2\beta+D)}} C_{\delta,n}^{\frac{2}{2\beta+D}} \cdot (4D)^{\frac{D}{4\beta+2D}} \cdot \left(\frac{\lambda}{n}\right)^{\frac{1}{2\beta+D}}.$$

Estimating the Connected Components

Lemma 16. *With probability at least $1 - \delta/3$, for any $x \in X$ and $y \in X$, if $\|x - y\|_1 \leq 1.5\varepsilon$, then $\exists i \in [t], h_i(x) = h_i(y)$.*

Proof. Fix arbitrary $x \in X$ and $y \in X$ such that $\|x - y\|_1 \leq 1.5\varepsilon$. Consider $i \in [t]$. We have:

$$\begin{aligned} \Pr[h_i(x) \neq h_i(y)] &\leq \sum_{j \in [D]} \Pr\left[\lceil (x^{(j)} + \eta_i)/(2\varepsilon) \rceil \neq \lceil (y^{(j)} + \eta_i)/(2\varepsilon) \rceil\right] \\ &= \|x - y\|_1/(2\varepsilon) \leq 3/4. \end{aligned}$$

Thus, $\mathbf{E}[|\{i \in [t] \mid h_i(x) = h_i(y)\}|] \geq 25 \log(n/\delta)$. By Chernoff bound, with probability at least $1 - \delta/(3n^2)$, $\exists i \in [t], h_i(x) = h_i(y)$.

By taking union bound over all pairs $x \in X$ and $y \in X$, with probability at least $1 - \delta/3$, for any $x \in X$ and $y \in X$, if $\|x - y\|_1 \leq 1.5\varepsilon$, $\exists i \in [t]$ such that $h_i(x) = h_i(y)$. $\square$

A direct corollary of above lemma is as the following:

Corollary 17. *With probability at least $1 - \delta/3$,*

1. *for any $x \in C$ and $y \in C$, if $\|x - y\|_1 \leq 1.5\varepsilon$, then x and y are in the same connected component in G;*
2. *for any $x \in X \setminus C$ and $y \in C$, if $\|x - y\|_1 \leq 1.5\varepsilon$, then $\exists y' \in C$, x and y' are in the same connected component in G.*

Theorem 18. *For each connected component $\mathcal{C}$ in $L_f(\lambda)$, there is a unique connected component $\hat{\mathcal{C}}$ in G such that*

$$d_{Haus}(\mathcal{C}, \hat{\mathcal{C}}) \leq 4\left(\frac{\lambda}{C_1} \cdot 10C_{\delta,n}/\sqrt{k}\right)^{1/\beta}.$$

Proof. Let $\mathcal{C}$ be a connected component in $L_f(\lambda)$. Let $Q = X \cap B(\mathcal{C}, \frac{\varepsilon}{2})$. According to Lemma 12, for $x \in \mathcal{C}$, we can always find $x' \in Q$ such that $\|x - x'\|_2 \leq \frac{\varepsilon}{2}$. Next, let us show that Q is connected in G. Consider two points $x', y' \in Q$, we can find a curve $\rho \subset \mathcal{C}$ such that $d(x', \rho), d(y', \rho) \leq \varepsilon/2$. We can find a sequence of points $u_0, u_1, \cdots, u_m$ on ρ such that $d(u_{i-1}, u_i) \leq \varepsilon/2$ and $d(u_0, x'), d(u_m, y') \leq \varepsilon/2$. According to Lemma 12, $\forall i \in \{0, 1, \cdots, m\}$, $\exists u'_i \in C$ such that $\|u_i - u'_i\|_2 \leq \varepsilon/2$. Notice that $\forall i \in [m]$, $\|u'_i - u'_{i-1}\|_2 \leq 1.5\varepsilon$, and $\|u'_0 - x'\|_2, \|u'_m - y'\|_2 \leq 1.5\varepsilon$. By Corollary 17, x', u'_0 are connected in G, $\forall i \in [m], u'_i, u'_{i-1}$ are connected in G, and y', u'_m are connected in G. Thus Q is connected in G.

Let $\hat{\mathcal{C}}$ be the connected component in G containing Q. We have $\sup_{x\in\mathcal{C}} d(x, \hat{\mathcal{C}}) \leq \varepsilon/2 \leq 4\left(\frac{\lambda}{C_1} \cdot 10C_{\delta,n}/\sqrt{k}\right)^{1/\beta}$.

Consider an arbitrary point x' in $\hat{\mathcal{C}}$. There must be a core point $y' \in \hat{\mathcal{C}} \cap C$ such that $\|x' - y'\|_2 \leq 2\varepsilon\sqrt{D}$. We can find a sequence of core points $v'_0, v'_1, \cdots, v'_s$ such that $v'_0 = y', v'_s \in Q$ and $\forall i \in [s], v'_i, v'_{i-1}$ are connected in G. By Lemma 10, we can find a sequence of points $v_0, v_1, \cdots, v_s \in L_f(\lambda)$ such that $\forall i \in \{0, 1, \cdots, s\}, d(v_i, v'_i) \leq 2\left(\frac{\lambda}{C_1} \cdot 10C_{\delta,n}/\sqrt{k}\right)^{1/\beta} \leq r_c/3$ and $v_s \in \mathcal{C}$. By Corollary 9, $v_0, v_1, \cdots, v_s$ must be connected in $L_f(\lambda)$. Thus, $v_0 \in \mathcal{C}$. It implies that $d(y', \mathcal{C}) \leq 2\left(\frac{\lambda}{C_1} \cdot 10C_{\delta,n}/\sqrt{k}\right)^{1/\beta}$. Then, we have:

$$d(x', \mathcal{C}) \leq d(y', \mathcal{C}) + 2\varepsilon\sqrt{D} \leq 2\left(\frac{\lambda}{C_1} \cdot 10C_{\delta,n}/\sqrt{k}\right)^{1/\beta} + 2\varepsilon\sqrt{D} \leq 4\left(\frac{\lambda}{C_1} \cdot 10C_{\delta,n}/\sqrt{k}\right)^{1/\beta}.$$

Thus, $\sup_{x'\in\hat{\mathcal{C}}} d(x', \mathcal{C}) \leq 4\left(\frac{\lambda}{C_1} \cdot 10C_{\delta,n}/\sqrt{k}\right)^{1/\beta}$ □

Removal of False Clusters

In theory, to remove false clusters, we need additional steps. We need to connect connected components with larger ε. This is known as *pruning false clusters* in the literature (see (Kpotufe and Von Luxburg 2011; Jiang 2017; Jang and Jiang 2018)).

1. Run Algorithm 3 with k and $\varepsilon = \left(\frac{k}{n\lambda\cdot(1-2C_{\delta,n}/\sqrt{k})}\right)^{1/D}$. Let $\mathcal{G}$ be returned clusters. Let C be the corresponding core points.
2. Let $\varepsilon' = 8\left(\frac{\lambda}{C_1} \cdot 10C_{\delta,n}/\sqrt{k}\right)^{1/\beta}$.
3. Construct t independent hash functions $h'_1, h'_2, \cdots, h'_t : \mathbb{R}^D \to \mathbb{Z}^D$, where h'_i is constructed as the following: choose $\eta'_i \in [0, 2\varepsilon']$ uniformly at random, and $\forall x \in \mathbb{R}^D$, let
$$h'_i(x) := \left\lceil \frac{x + \eta'_i \cdot \vec{1}_D}{2\varepsilon'} \right\rceil.$$
4. Construct a graph G': for $i \in [t]$ and for each subset S of points with the same value of $h'_i(\cdot)$, create a star connecting every points in this subset. Let $\mathcal{G}'$ be the connected componetns of G'.
5. Let $\widetilde{\mathcal{G}}$ be the clusters obtained by merging clusters from $\mathcal{G}$ which are subsets of the same cluster in $\mathcal{G}'$.

Lemma 19. *For $x, y \in X$, if x, y are in the same connected component in G, x, y are in the same connected component in G'.*

Proof. If x, y are connected in G, there is a path $x = u_0, u_1, \cdots, u_m = y$ in G where $\forall i \in [m], d(u_{i-1}, u_i) \leq 2\sqrt{D}\varepsilon \leq \varepsilon'$. According to Corollary 17, there is a path $u_0, u_1, \cdots, u_m$ in G'. □

Theorem 20. *For each connected component $\tilde{\mathcal{C}}$ in $\widetilde{\mathcal{G}}$ which is not an isolated vertex, there is a unique connected component $\mathcal{C}$ in $L_f(\lambda)$ such that*

$$d_{Haus}(\mathcal{C}, \tilde{\mathcal{C}}) \leq 4\left(\frac{\lambda}{C_1} \cdot 10C_{\delta,n}/\sqrt{k}\right)^{1/\beta}.$$

Proof. Let x' be an arbitrary point in $\widetilde{\mathcal{C}}$. There is a core point $y' \in C$ such that $\|x' - y'\|_2 \leq 2\sqrt{D}\varepsilon$. By Lemma 10, there is a point $y \in L_f(\lambda)$ such that $\|y' - y\|_2 \leq 2\left(\frac{\lambda}{C_1} \cdot 10C_{\delta,n}/\sqrt{k}\right)^{1/\beta}$. Let $\mathcal{C}$ be the connected component in $L_f(\lambda)$

containing y. Thus, $d(x',\mathcal{C}) \leq 4\left(\frac{\lambda}{C_1}\cdot 10C_{\delta,n}/\sqrt{k}\right)^{1/\beta}$. Let a' be an another arbitrary point in $\widetilde{C}$. We can find a sequence of core points $u_0', u_1', \cdots, u_m' \in C$ such that $d(x', u_0') \leq 2\sqrt{D}\varepsilon$, $d(y', u_m') \leq 2\sqrt{D}\varepsilon$, and $\forall i \in [m], d(u_{i-1}', u_i') \leq 2\sqrt{D}\varepsilon'$. According to Lemma 10, we can find a sequence of points $u_0, u_1, \cdots, u_m$ in $L_f(\lambda)$ such that $\forall i \in \{0, 1, \cdots, m\}, d(u_i, u_i') \leq 2\left(\frac{\lambda}{C_1}\cdot 10C_{\delta,n}/\sqrt{k}\right)^{1/\beta}$. By triangle inequality, we have $\forall i \in [m], d(u_{i-1}, u_i) \leq 6\sqrt{D}\varepsilon' \leq \frac{(\lambda_c/C_2)^{1/\beta}}{2}$. By Corollary 9, we know that $u_0, u_1, \cdots, u_m$ are in $\mathcal{C}$. Thus, $d(a', \mathcal{C}) \leq 4\left(\frac{\lambda}{C_1}\cdot 10C_{\delta,n}/\sqrt{k}\right)^{1/\beta}$ which implies that

$$\sup_{a'\in\widetilde{C}} d(a',\mathcal{C}) \leq 4\left(\frac{\lambda}{C_1}\cdot 10C_{\delta,n}/\sqrt{k}\right)^{1/\beta}.$$

According to Lemma 18, there is a unique connected component $\hat{\mathcal{C}}$ in $\mathcal{G}$ such that $d_{\text{Haus}}(\mathcal{C},\hat{\mathcal{C}}) \leq 4\left(\frac{\lambda}{C_1}\cdot 10C_{\delta,n}/\sqrt{k}\right)^{1/\beta}$. Thus, $\forall a' \in \tilde{\mathcal{C}}$, there is a $b' \in \hat{\mathcal{C}}$ such that $d(a', b') \leq 8\left(\frac{\lambda}{C_1}\cdot 10C_{\delta,n}/\sqrt{k}\right)^{1/\beta} \leq \varepsilon'$. According to Corollary 17, a', b' are in the same connected component in G'. By combining with Lemma 19, $\hat{\mathcal{C}}$ is a subset of $\widetilde{\mathcal{C}}$, which implies that

$$\sup_{x\in\mathcal{C}} d(x,\widetilde{\mathcal{C}}) \leq 4\left(\frac{\lambda}{C_1}\cdot 10C_{\delta,n}/\sqrt{k}\right)^{1/\beta}.$$

□

Missing Details of Section: Experiments

Detailed description of the implemented algorithms. The first version of DBSCAN (DSv1) exactly follows the description of Algorithm 1. The second version of DBSCAN (DSv2) has a slight modification and is described in Algorithm 4. The main difference is that instead of constructing the graph over all points, the modified version only constructs the graph over the core points and assigns each non-core point to the closest core point. According to the analysis of (Jiang 2017; Jang and Jiang 2018), it is easy to verify that the graph over core points has the same density level set estimation guarantee.

Algorithm 4 Modified DBSCAN

- **Inputs:** $X \subset \mathbb{R}^D, \varepsilon, k$
- Initialize core $C \leftarrow \emptyset$.
- Check each point $x \in X$: if $|\{y \in X \mid d(x,y) \leq \varepsilon\}| \geq k$, then add x to the core C.
- Construct a graph G where each vertex corresponds to a point in X.
- For each core point $c \in C$, add an edge in G between c and each vertex $x \in C$ which satisfies $d(c,x) \leq \varepsilon$.
- For each $x \in X \setminus C$, add an edge in G between x and $c \in C$ where c is the closest core point to x.
- Return connected components of G.

The detailed implementation of DBSCAN++ (Jang and Jiang 2018) is presented in Algorithm 5.[6] For DBSCAN++ with uniform initialization (DS++ unif), S is a subset of m uniform samples from X. For DBSCAN++ with k-center initialization (DS++ k-ctr), the choice of S is described in Algorithm 6.

Algorithm 5 DBSCAN++

- **Inputs:** $X \subset \mathbb{R}^D, m, \varepsilon, k$
- Select a subset $S \subseteq X$ with $|S| = m$.
- Initialize core $C \leftarrow \emptyset$.
- Check each point $x \in S$: if $|\{y \in X \mid d(x,y) \leq \varepsilon\}| \geq k$, then add x to the core C.
- Construct a graph G where each vertex corresponds to a point in X.
- For each core point $c \in C$, add an edge in G between c and each vertex $x \in C$ which satisfies $d(c,x) \leq \varepsilon$.
- For each $x \in X \setminus C$, add an edge in G between x and $c \in C$ where c is the closest core point to x.
- Return connected components of G.

A detailed implementation of our near linear time DBSCAN (refered to as "Ours" in the charts) is shown in Algorithm 7. Notice that hash functions $h_i(\cdot), h_i'(\cdot)$ have the same guarantees as LSH mentioned in Lemma 2. Thus, the graph G constructed in line 4 has the similar density level set estimation guarantee of the graph constructed by Algorithm 3 over core points. After line 4, Algorithm 7 tries to assign each non-core point to an approximately closest core point. Since we increase ε' exponentially, the total running time of Algorithm 7 will blow up by a factor $\log(R/\varepsilon)$ where R is the diameter of the point set. The running time is still near linear.

[6] The details are confirmed by personal communications with the authors of DBSCAN++ (Jang and Jiang 2018).

Algorithm 6 K-Center Initialization

Inputs: $X \subset \mathbb{R}^D, m$
$S \leftarrow \{x_1\}$.
for $i := 1 \rightarrow m-1$ **do**
    $S \leftarrow S \cup \{\arg\max_{x\in X} d(x, S)\}$
end for

Algorithm 7 Detailed Implementation of Near Linear time DBSCAN

Inputs: $X \subset \mathbb{R}^D, t, \varepsilon, k$
Compute core C by Algorithm 2.
Draw t independent hash functions $h_1, h_2, \cdots, h_t : \mathbb{R}^D \rightarrow \mathbb{Z}^D$, where h_i is constructed as the following: choose $\vec{\eta}_i \in [0, 2\varepsilon]^D$ uniformly at random, and $\forall x \in \mathbb{R}^D$, let $h_i(x) := \left\lfloor \frac{x+\vec{\eta}_i}{2\varepsilon} \right\rfloor$.
Construct G: for $i \in [t]$ and for each part $S \subseteq C$ with the same value of $h_i(\cdot)$, choose an arbitrary point $s \in S$, and add an edge in G between s and every $x \in S$ with $d(s, x) \leq \varepsilon$. .
for $\varepsilon' := \varepsilon, 2\varepsilon, 4\varepsilon, 8\varepsilon, \cdots$ until no isolated non-core point **do**
    Draw t independent hash functions $h'_1, h'_2, \cdots, h'_t : \mathbb{R}^D \rightarrow \mathbb{Z}^D$, where h'_i is constructed as the following: choose $\vec{\eta}'_i \in [0, 2\varepsilon']^D$ uniformly at random, and $\forall x \in \mathbb{R}^D$, let $h'_i(x) := \left\lfloor \frac{x+\vec{\eta}'_i}{2\varepsilon'} \right\rfloor$.
    For each isolated non-core point $x \in X \setminus C$, find one arbitrary core point $c \in C$ such that $\exists i \in [t], h'_i(c) = h'_i(x)$. If such point c exists and $d(c, x) \leq \varepsilon'$, connect x to c in G.
end for
Return connected components of G.

Further discussion of experimental results. For experiments on real datasets (A)-(K), we do linear search to find best ε for each compared algorithm. In particular we search 100 times for each dataset. According to (Jang and Jiang 2018), the search range for each dataset is shown in Table 6. The Adjusted Random Index scores and Adjusted Mutual Information scores for different ε are shown in Figure 2.

Table 6: **The range of ε in the optimal tuning procedure for real datasets.**

(A)	[0,4]	(B)	[0,500]	(C)	[0,100]
(D)	[0,150]	(E)	[0,1500]	(F)	[0,1.6]
(G)	[0,800]	(H)	[0,3.5]	(I)	[0,8]
(J)	[0,10]	(K)	[0,45]		

As shown by Table 2, the quality of the clusters obtained by the second version of DBSCAN (DSv2) is uniformly better than the quality of the clusters obtained by the original version of DBSCAN (DSv1). This implies that the connected components of the graph over core points usually provide better clusters than the connected components of the graph over all points. Similar to the experimental results presented by (Jang and Jiang 2018), the accuracy of DBSCAN++ is uniformly better than the accuracy of the original version of DBSCAN. But we show that the accuracy of the modified version of DBSCAN is comparable to the accuracy of DBSCAN++. Our near linear time DBSCAN has similar accuracy as modified DBSCAN and DBSCAN++.

As shown in all experiments, KDTree cannot improve the running time of DBSCAN and DBSCAN++ on these datasets due to large overhead.

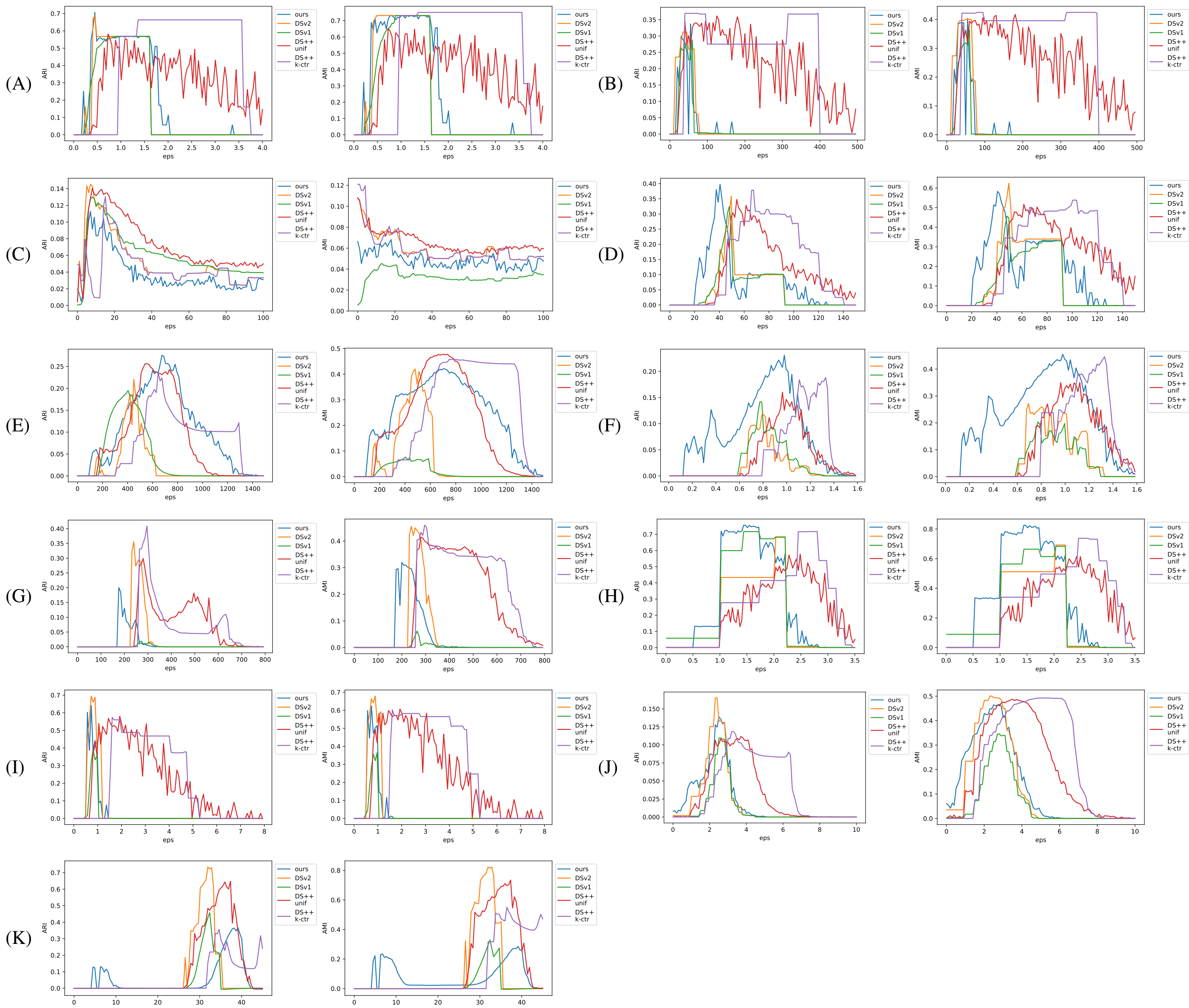

Figure 2: **The qualities of clustering results over different** ε: For each dataset (A)-(K), the left column corresponds to the Adjusted Random Index scores, and the right column corresponds to the Adjusted Mutual Information scores. x-axis correponds to the tunning range of ε and y-axis corresponds to the scores.