

Unimodal Strategies in Density-Based Clustering

Oron Nir^{1,2}, Jay Tenenbaum², and Ariel Shamir¹ (✉)

¹ CANVAS Lab, Reichman University, Herzeliya, Israel, arik@runi.ac.il

² Microsoft, Herzeliya, Israel, {niroron,tenenbaumjay}@microsoft.com

Abstract. Density-based clustering methods often surpass centroid-based counterparts, when addressing data with noise or arbitrary data distributions common in real-world problems. In this study, we reveal a key property intrinsic to density-based clustering methods regarding the relation between the number of clusters and the neighborhood radius of core points — we empirically show that it is nearly unimodal, and support this claim theoretically in a specific setting. We leverage this property to devise new strategies for finding appropriate values for the radius more efficiently based on the Ternary Search algorithm. This is especially important for large scale data that is high-dimensional, where parameter tuning is computationally intensive. We validate our methodology through extensive applications across a range of high-dimensional, large-scale NLP, Audio, and Computer Vision tasks, demonstrating its practical effectiveness and robustness. This work not only offers a significant advancement in parameter control for density-based clustering but also broadens the understanding regarding the relations between their guiding parameters. Our code is available at <https://github.com/oronnir/UnimodalStrategies>.

Keywords: Density-based clustering · Efficient parameter search

1 Introduction

Clustering, a fundamental task in machine learning, is pivotal in uncovering patterns and structures in unlabeled data. Among various clustering algorithms, density-based methods have gained significant attention due to their ability to identify clusters of arbitrary shapes and sizes. Notably, variants of Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [1] are widely used today by researchers and data scientists. Unlike centroid-based methods that assume spherical cluster shapes, density-based approaches are adept at discovering clusters with complex geometries and noisy examples, making them more suitable for real-world data that often exhibit irregular distributions [2].

A significant challenge in leveraging the full potential of density-based clustering lies in parameter tuning. Parameters such as ϵ and *MinPts*. These user-defined parameters affect cluster formation in DBSCAN and its variants, critically influence the clustering outcome, such as the resulting number of clusters denoted by k . Misconfigured parameter invocation can lead to over-segmentation or under-segmentation of data, significantly affecting the quality and interpretability of the results. This is particularly challenging in high-dimensional, large-scale

datasets where the intuitive understanding of parameter effects is less apparent and search for values is costly [3].

Our key insight centers on the relationship between these parameters and the resulting number of clusters. Specifically, we observe that when *MinPts* is held constant, the number of clusters k produced by DBSCAN varies with the neighborhood radius ε in a quasi-unimodal fashion. This means that as ε increases from very small to very large values, the number of clusters first increases, reaches a peak, and then decreases. Using this observation, we address the parameter tuning challenge, by automatically finding the parameter ε^* that maximizes the number of clusters $k(\varepsilon)$. While ε^* may not always yield the “optimal” clustering for all datasets, it provides critical insight into the clustering structure. For values $\varepsilon \gg \varepsilon^*$ there is over-clustering (i.e., one large cluster at the extreme), and for values $\varepsilon \ll \varepsilon^*$ there is under-clustering (i.e., too many samples are treated as noise). Hence, it is clear that values around ε^* are the “interesting” and important ones. Our method approximates ε^* , addressing the challenge of parameter tuning in density-based clustering.

To demonstrate the efficacy of our proposed method, we apply it to high-dimensional, large-scale classification datasets. Our experiments reveal that not only does our method consistently achieve the target number of clusters, but it also enhances the quality of the clustering supervised evaluation metrics over state-of-the-art (SOTA) methods. Empirical evidence also underscores the advantage of our method in achieving less noise points which is important when working with noisy data.

This study contributes to the field of density-based clustering by:

- Discovering the Unimodal property in density-based clustering, demonstrating this both theoretically and practically.
- Proposing the efficient Ternary Search for real-world data parameter tuning.
- Sharing our code at <https://github.com/oronnir/UnimodalStrategies>.

2 Related work

Clustering methods in data mining have been extensively studied, sometimes focusing on handling synthetic, separable, and low-dimension data distributions using benchmarks which could be insufficient for a rigorous evaluation and may lead to overfitting [4]. Density-based methods like DBSCAN [1] excel at handling noise and discovering arbitrary-shaped clusters in high dimensions, surpassing centroid-based approaches.

Several variants of DBSCAN, such as OPTICS [5], VDBSCAN [6], and ADBSCAN [7], have been introduced to address varying-density clusters and improve scalability. These approaches typically trade off runtime efficiency for parameter optimization. HDBSCAN [8], for instance, adopts a multi-resolution framework to self-tune parameters. Methods that extend DBSCAN while retaining its dependence on ε are not considered as a baseline in this paper. Note that unsupervised learning methods inherently depend on specific mathematical properties of data, making it unlikely for any single method to be universally optimal. Notably,

DBSCAN’s assumption of uniform density regions and the existence of an ideal $(\varepsilon, MinPts)$ pair is often unmet in practice.

Parameter Selection in Density-Based Clustering: The performance of DBSCAN and its variants depends heavily on parameter settings: while $MinPts$ is intuitive as an application-dependent integer, $\varepsilon \in \mathbb{R}_{>0}$ is challenging to tune in high-dimensional spaces. Ester et al. [1] suggests using the Elbow Method manually over the k-dist plot which is considered a folklore heuristic for density shift and k selection in algorithms like k-means. However, this method is found sub-optimal by Schubert [2]. Several studies have automated and further optimized the k-dist plot heuristic for parameter selection e.g., [6,9]. These methods aim to reduce the user intervention required in the clustering process, but often face challenges in handling high-dimensional and large-scale datasets. Researchers [10,11] revisit the challenge, noting that $MinPts$ is easier to tune than ε , and suggest setting $MinPts = 2D$ where D is the dimension of the data. Assuming for example, $D > 100$, such a $MinPts$ value could lead to either enhanced noise robustness or into an under-segmented solution. Another common practice is dimensionality reduction. However, in this work we aim at enhancing density-based clustering over the raw data in high-dimension.

Advancements in Parameter Optimization: Recent studies have explored various optimization techniques. SS-DBSCAN [12] and AMD-DBSCAN [13] both suggest an exhaustive grid-search approach for $MinPts$, where the former includes an automated version of the Elbow method using stratified sampling. AEDBSCAN [14] assigns a per-point radius to optimize ε for a fixed $MinPts$. These algorithms essentially apply exhaustive search of the optimal parameters without relying on the underlying algorithm properties and their results are reported over low dimensional synthetic datasets. We consider the following density-based parameter tuning methods as the SOTA baselines [8,6,5,12,13,14,9].

In summary, while density-based clustering methods offer advantages in handling non-linearly separable data, their reliance on parameter settings poses a significant challenge, especially in high-dimensional and large-scale scenarios. Our work builds upon these foundations, proposing an efficient method for parameter tuning that is responsive to these data representation challenges.

3 Preliminaries

We establish standard DBSCAN algorithm definitions and notations,

Definition 1 (Dataset). Let (M, d) be a metric space and a distance metric. A dataset $X = \{x_1, \dots, x_N\} \subseteq M$ is a finite subset of M .

Definition 2 (DBSCAN Parameters). For a dataset $X \in \mathbb{R}^{N \times D}$, DBSCAN receives $\varepsilon \in \mathbb{R}_{>0}$ and $MinPts \in \mathbb{N}_{\geq 2}$ as user-defined parameters.

Definition 3 (ε -ball and Neighborhood). For a point $p \in \mathbb{R}^D$, X , and $\varepsilon > 0$:

Let $B_\varepsilon(p) = \{y \in \mathbb{R}^D : d(p, y) \leq \varepsilon\}$ be the ε -ball centered at p
and let $N_\varepsilon(p) = B_\varepsilon(p) \cap X$ be the ε -neighborhood of p

Definition 4 (Core and Border Points). For ε and $MinPts$:

- A point $p \in X$ is a core point if $|N_\varepsilon(p)| \geq MinPts$
- A point $b \in X$ is a border point if:
 - $b \in B_\varepsilon(p)$ for some core point p
 - $|N_\varepsilon(p)| < MinPts$ (not itself a core point)
- p is a noise point if it is neither a core nor a border point

Definition 5 (Density-Reachability). Two points $p, q \in X$ are density-reachable at radius ε if there exists a sequence of points $\{p_1, \dots, p_t\} \subseteq X$ such that:

1. $p_1 = p$ and $p_t = q$
2. Each p_i is a core point for $i < t$
3. $p_{i+1} \in B_\varepsilon(p_i)$ for all $i < t$

Definition 6 (Cluster). A cluster c is a maximal set of points where:

- At least one point in c is a core point
- Non-core points in c are within ε -radius of a core point in c
- All core points in c are mutually density-reachable

DBSCAN forms clusters by identifying core points and their density-reachable neighbors within ε -radii. Non-core points that are density-reachable from core points become border points, and the remaining points are classified as noise [11].

Let $\mathbf{A} : \mathbb{R}^{N \times D} \times \mathbb{R} \times \mathbb{N} \rightarrow \mathbb{N}^N$ denote DBSCAN or a variant of DBSCAN, let $\mathcal{C} := \mathbf{A}(X, \varepsilon, MinPts) \in \mathbb{N}^N$ be \mathbf{A} 's output clustering assignment, and let $\mathbf{K}(\mathcal{C}) = |\{c_i \in \mathcal{C}\}_{i=1}^N|$ be the function which counts the number of clusters \mathbf{A} returns. We often refer to $\mathbf{K}(\mathcal{C})$ by using the variable k when clear from context and characterize it as a function of ε ,

$$k(\varepsilon) := \mathbf{K}(\mathbf{A}(X, \varepsilon, MinPts)) \quad (1)$$

4 The Unimodality Property

We make a fundamental observation regarding DBSCAN, that for a fixed $MinPts$, $k(\varepsilon)$ (Eq. 1) is *near-unimodal*. This is since **1.** low values of ε label more examples as noise so less clusters are formed (specifically for $\varepsilon < \min_{i \neq j} d(x_i, x_j)$ there are no core points), while **2.** high values of ε combine clusters together and gradually reduce the number of clusters (for $\varepsilon \geq \max_{i,j} d(x_i, x_j)$ there is a single cluster since all points are mutually density-reachable).

We identify in a counter-example in Fig. 1, that $k(\varepsilon)$ is not necessarily strictly unimodal as per the standard definition, i.e., monotonically non-decreasing up to the mode and monotonically non-increasing thereafter [15]. However, we empirically demonstrate the *near-unimodality* of $k(\varepsilon)$ over 24 real-life datasets, and support this by a statistical test (DIP [15]) in Sec. 6, and a theoretical analysis.

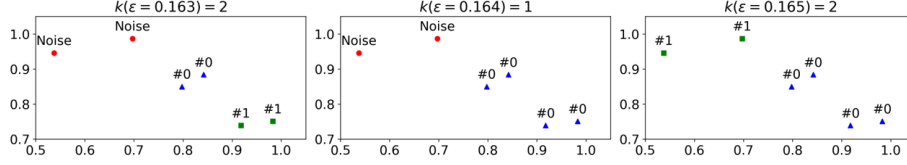


Fig. 1: A counter-example with non-unimodal behavior of $k(\varepsilon)$ in \mathbb{R}^2 and L_2 .

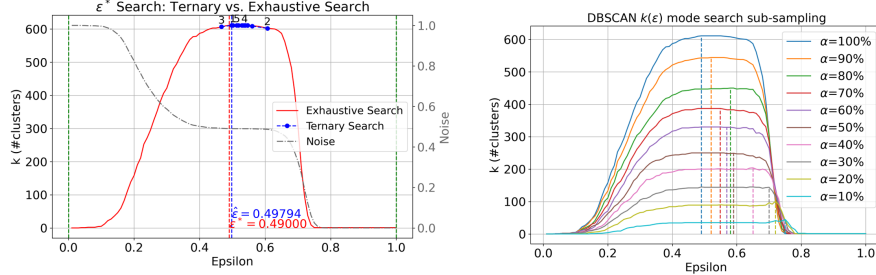


Fig. 2: TS convergence steps in blue vs. Exhaustive Search in red over the FACE dataset ($N=45k$ with noise). The grey line illustrates the percentage of noise. Fig. 3: Exhaustive search of $k(\varepsilon)$ unimodality for different sample sizes (α). When α grows ε_α^* shrinks monotonically, hence used as an UB in our method.

4.1 Theoretical Analysis:

To support our unimodality claim, we give a theoretical analysis of DBSCAN running on a dataset $X = \{x_1, \dots, x_N\}$ of iid uniform samples $\forall i, x_i \sim U[0, 1]^D$. We acknowledge this distribution is non-standard for clustering, but it is a first step towards understanding this property in more general distributions. First, for $MinPts = 2$, a common choice for this parameter, and $D = 1$. Theorem 1 gives and proves the explicit unimodal function describing $E_X[k(\varepsilon)]$, and gives its mode. We note that in experimental evaluations which are omitted, we saw that a law of large numbers appears, and for $N > 1,000$, even for any single random dataset $X \sim U[[0, 1]^N]$, almost always the resulting $k(\varepsilon)$ was surprisingly close to the unimodal function $E_X[k(\varepsilon)]$ over the whole domain $\varepsilon > 0$, hence near-unimodal itself. Then, we consider the setting where N is large, and assume that $MinPts$ grows as a function of N (this is standard in DBSCAN for large datasets). For ease of presentation, we assume a constant ratio $\rho = MinPts/N \in (0, 1)$ between $MinPts$ and N . This setting is natural for DBSCAN since doubling the dataset size means that the expected number of points in each ε -ball should roughly double. To warm up, Theorem 2 considers $D = 1$, and then Theorem 3 considers a general $D \in \mathbb{N}$, and they both prove that $k(\varepsilon)$ is not trivially 0 or 1 only for $\varepsilon \approx \frac{1}{2}\rho^{\frac{1}{D}}$ (much stronger than the bounds to come in Section 5.1). We stress that while we present our results for a constant ratio ρ , they in fact hold for a sufficiently large but reasonable N for any setting where $MinPts$ grows (e.g.,

logarithmically) with N , and linearly with D (a common assumption is that $\text{MinPts} > D$). This suggests that when we search for ε^* , first identifying the often small region in which $k(\cdot)$ is not trivial, can improve runtime and accuracy.

Theorem 1. *Consider DBSCAN running on a uniformly sampled dataset $X \sim U[[0, 1]^N]$ with parameter $\text{MinPts} = 2$. The expected number of clusters as a function of $\varepsilon > 0$ is $E_X[k(\varepsilon)] = (N-1) \cdot (1-\varepsilon)_+^N - (N-2) \cdot (1-2\varepsilon)_+^N$ where $y_+ := \max(y, 0)$. This is a unimodal function maximized at $\varepsilon_0 = \frac{a-1}{2a-1} \approx_{N \rightarrow \infty} \frac{\ln(2)}{N}$ for $a := (\frac{2N-2}{N-2})^{\frac{1}{N-1}}$, with expected value $E_X[k(\varepsilon_0)] \approx_{N \rightarrow \infty} \frac{N}{4}$.*

Proof (Proof Idea). Consider the order statistics $u_1 \leq \dots \leq u_N$ of X , and the differences, often called *spacings*, $s_i := u_i - u_{i-1}$ (by convention $u_0 := 0$, $u_{N+1} := 1$). Let U_i be the event that u_i is the rightmost point of a cluster. Note that for $1 < i \leq N-1$, U_i occurs iff u_i is a core point and the cluster does not extend to the right, i.e., $U_i = (s_i < \varepsilon) \wedge (s_{i+1} > \varepsilon)$. This occurs w.p.

$$P_X[U_i] = P_X[(s_i < \varepsilon) \wedge (s_{i+1} > \varepsilon)] = P_X[s_{i+1} > \varepsilon] - P_X[s_i > \varepsilon \wedge s_{i+1} > \varepsilon].$$

For spacings, it is well known [16] that for any nonnegative values b_1, \dots, b_N it holds that, $P[\forall i, s_i > b_i] = (1 - \sum_i b_i)_+^N$, so we conclude $P_X[U_i] = (1 - \varepsilon)_+^N - (1 - 2\varepsilon)_+^N$. Note that $P[U_1] = 0$ since the leftmost point cannot have a left neighbor, and that $P_X[U_N] = P_X[s_N < \varepsilon] = (1 - \varepsilon)_+^N$ because there is no point to its right. Since $E[A] = P[A]$ for a Bernoulli RV A , and by linearity of the expectation, since the number of clusters is the number of rightmost points of clusters,

$$E_X[k(\varepsilon)] = E_X[\sum_i \mathbb{1}_{U_i}] = \sum_i P_X[U_i] = (N-1) \cdot (1 - \varepsilon)_+^N - (N-2) \cdot (1 - 2\varepsilon)_+^N.$$

To find the optimal ε , note that it is attained at $\varepsilon < 1/2$ and solve $\frac{d}{d\varepsilon} E_X[k(\varepsilon)] = 0$ for ε , i.e., $\frac{d}{d\varepsilon} E_X[k(\varepsilon)] = 2N(N-1)(1-2\varepsilon)^{N-1} - N(N-2)(1-\varepsilon)^{N-1} = 0$. \square

We prove the concentration bound,

Theorem 2. *Consider DBSCAN running on a uniformly sampled dataset $X \sim U[[0, 1]^N]$, with parameter $\text{MinPts} = \rho \cdot N$ for a constant $\rho < \frac{1}{4}$. Then for any $\beta > 1$ and $\delta > 0$, both conditions hold: **1.** for any $\varepsilon > \beta \cdot \frac{\rho}{2}$ and $N > N_\varepsilon$, $P_X[k(\varepsilon) = 1] > 1 - \delta$, and **2.** for any $\varepsilon < \frac{1}{\beta} \cdot \frac{\rho}{2}$ and $N > N_\varepsilon$, $P_X[k(\varepsilon) = 0] > 1 - \delta$, for an appropriately large $N_\varepsilon = \Omega\left(\log(\frac{1}{\delta\varepsilon q})/(\rho q^2)\right)$ for $q := 1 - \frac{1}{\sqrt{\beta}}$.*

Proof (Proof Idea). For the case $\varepsilon > \beta \cdot \frac{\rho}{2}$, divide $[0, 1]$ to the set of segments $S_l = [al - b, al + b]$, for all $l \in \mathbb{N}$ where $b = \frac{\varepsilon}{\sqrt{\beta}}$ is slightly below ε , and $a = \varepsilon - b$. By the Hoeffding inequality and a union bound, w.h.p. all segments S_l centered in $[\varepsilon, 1 - \varepsilon]$ have $\geq \text{MinPts}$ points, so for each $x_i \in [\varepsilon, 1 - \varepsilon]$, the ε -ball around it contains such a segment (for some $l \in \mathbb{N}$, by the definition of a and the triangle inequality), hence x_i is a core point. Combined with a high probability event that for large enough N , all points are ε -near each other, and that the cluster covers also $[0, \varepsilon]$ and $[1 - \varepsilon, 1]$, we get a single cluster.

For the case $\varepsilon < \frac{1}{\beta} \cdot \frac{\rho}{2}$, divide to segments as above with $b = \varepsilon \cdot \sqrt{\beta}$ slightly above ε , and $a = b - \varepsilon$. By Hoeffding inequality and a union bound, w.h.p. all segments S_l centered in $[\varepsilon, 1 - \varepsilon]$ have $< MinPts$ points, so since the ε -neighborhood of each point x_i is contained in such a segment (for some $l \in \mathbb{N}$, by triangle inequality and $a = b - \varepsilon$), there are no core points hence no clusters. \square

We follow similar arguments to those in Theorem 2, and extend it to uniform datasets on the unit cube $[0, 1]^D$ of arbitrary dimension $D \in \mathbb{N}$, incurring only an additional \sqrt{D} factor on the upper bound and no additional factor on the lower bound.

Theorem 3. *Consider DBSCAN running on a dataset $X \sim U[[0, 1]^{N \times D}]$ of N uniformly sampled points in the unit D -dimensional cube, with parameter $MinPts = \rho \cdot N$ for a constant $\rho < \frac{1}{4}$. Then for any $\beta > 1$ and $\delta > 0$, both conditions hold: **1.** for any $\varepsilon > \sqrt{D}\beta \cdot \frac{1}{2}\rho^{\frac{1}{D}}$ and $N > N_\varepsilon$, $P_X[k(\varepsilon) = 1] > 1 - \delta$, and **2.** for any $\varepsilon < \frac{1}{\beta} \cdot \frac{1}{2}\rho^{\frac{1}{D}}$ and $N > N_\varepsilon$, $P_X[k(\varepsilon) = 0] > 1 - \delta$, for an appropriately large*

$$N_\varepsilon = \Omega\left(\frac{1}{\rho} \cdot \left[\frac{\sqrt{\beta}}{(\frac{1}{\sqrt{\beta}} - 1)^2}\right] \cdot \left[D \log\left(\frac{2}{\beta - \sqrt{\beta}}\right) + \log\left(\frac{1}{\rho\alpha}\right)\right]\right).$$

Proof (Proof Idea). For the case $\varepsilon > \sqrt{D}\beta \cdot \frac{1}{2}\rho^{\frac{1}{D}}$, divide $[0, 1]^D$ to the set of cubes $S_{l_1, \dots, l_D} = [al_1 - b, al_1 + b] \times \dots \times [al_D - b, al_D + b]$, for all $l_1, \dots, l_D \in \mathbb{N}$ where $b = \varepsilon/(\sqrt{D}\beta^{\frac{1}{D}})$ is slightly below ε/\sqrt{D} , and $a = \varepsilon/\sqrt{D} - b$. By the Hoeffding inequality and a union bound, w.h.p. all cubes S_{l_1, \dots, l_D} centered in the set $[\varepsilon, 1 - \varepsilon]^D \subset [0, 1]^D$ have $\geq MinPts$ points, so for each $x_i \in [\varepsilon, 1 - \varepsilon]^D$, the ε -ball around it contains such a cube (for some $l \in \mathbb{N}$, by the definition of a and the triangle inequality, and that the diameter of such cubes are $= \sqrt{D}b$ slightly below ε), hence x_i is a core point. Combined with a high probability event that for large enough N , all points are ε -near each other, and that the cluster covers also the exterior $[0, 1]^D \setminus [\varepsilon, 1 - \varepsilon]^D$, we get a single cluster.

For the case $\varepsilon < \frac{1}{\beta} \cdot \frac{1}{2}\rho^{\frac{1}{D}}$, divide to cubes as above with $b = \varepsilon \cdot \beta^{\frac{1}{2D}}$ slightly above ε , and $a = b - \varepsilon$. By Hoeffding inequality and a union bound, w.h.p. all cubes S_{l_1, \dots, l_D} centered in the set $[\varepsilon, 1 - \varepsilon]^D$ have $< MinPts$ points, so since the ε -neighborhood of each point x_i is contained in such a cube (for some $l \in \mathbb{N}$, by triangle inequality and $a = b - \varepsilon$), there are no core points hence no clusters. \square

These theoretical results demonstrate that, under natural assumptions, to find the value of ε that maximize $k(\varepsilon)$, we can first focus on a small range where $k(\varepsilon)$ is neither trivially 1 nor 0, and subsequently exploit its *near-unimodal* behavior to efficiently find its mode, as discussed in the next section.

5 Method

Our task is to efficiently find the mode of $k(\varepsilon)$,

$$\varepsilon^* = \operatorname{argmax}_{\varepsilon} \{\mathbf{K}(\mathbf{A}(X, \varepsilon, \operatorname{MinPts}))\} \quad (2)$$

The Ternary Search (TS) algorithm by Bajwa et al. [17] finds a maximum in a unimodal (discrete) array. In Section 5.1, we adapt it to functions, leveraging the *near unimodality* of $k(\varepsilon)$, to find ε^* using fewer evaluations of $k(\cdot)$ compared to a linear search (see Fig. 2). Then, in Section 5.2 we introduce an even quicker estimator (TSE) for ε^* .

5.1 Ternary Search for ε^* (TS)

Our TS algorithm (Alg. 1) leverages the fact that the function $k(\varepsilon)$ is near-unimodal with mode ε^* . It starts with an initial lower bound (LB) and upper bound (UB) for ε^* , and iteratively (itr times) divides it to 3 equal parts and removes at least one of them (see Alg. 2), to reduce the search space size and still contain the mode. Specifically, for Alg. 2, we let $m_l = \frac{2LB+UB}{3}$ and $m_r = \frac{LB+2UB}{3}$, and $k_l = k(m_l)$ and $k_r = k(m_r)$ be the respective cluster counts. Recall that the cluster count as a function of ε is initially 0, then increases until it reaches the mode, and then decreases back to 1. Hence, the space reduction logic is as follows: 1. If $k_l = k_r = 1$, the mode must be below them, 2. If $k_l = k_r = 0$, the mode must be above them, 3. if $k_l = 0$ and $k_r = 1$, then the mode is between them, 4. if $k_l > k_r$ then the mode is to the left of m_r and otherwise the mode is to the right of m_l . The edge-case in which $k = 1$ for the very first formed cluster i.e., by chance and not by convergence, is easily detected by the Noise ratio thus omitted for simplicity. We provide both algorithms pseudo-code below.

Algorithm	1	Algorithm	2
$TS(X, LB, UB, \operatorname{MinPts}, itr)$		$Cond(LB, UB, m_l, m_r, k_l, k_r)$	
1: for $i=0$ to itr do		1: if $k_l == 1$ and $k_r == 1$ then	
2: $m_l \leftarrow \frac{2LB+UB}{3}$		2: return $\langle LB, m_l \rangle$	
3: $m_r \leftarrow \frac{LB+2UB}{3}$		3: else if $k_l == 0$ and $k_r == 1$ then	
4: $\mathcal{C}_l \leftarrow \mathbf{A}(X, m_l, \operatorname{MinPts})$		4: return $\langle m_l, m_r \rangle$	
5: $\mathcal{C}_r \leftarrow \mathbf{A}(X, m_r, \operatorname{MinPts})$		5: else if $k_l == 0$ and $k_r == 0$ then	
6: $\langle LB, UB \rangle \leftarrow Cond(LB, UB, \dots$		6: return $\langle m_r, UB \rangle$	
7: $m_l, m_r, \mathbf{K}(\mathcal{C}_l), \mathbf{K}(\mathcal{C}_r))$		7: else if $k_l > k_r$ then	
8: end for		8: return $\langle LB, m_r \rangle$	
9: return $\frac{m_l + m_r}{2}$		9: else	
		10: return $\langle m_l, UB \rangle$	
		11: end if	

Upper and Lower Bounds: To efficiently initialize the search space, we use tight bounds UB, LB for ε^* . For bounded metrics, a trivial upper bound UB^0 is simply their bound, and for (unbounded) metrics in general, as in Sec. 4, an upper bound is the Diameter $\max_{i,j \in \{1, \dots, N\}} d(x_i, x_j)$ of X , which we approximate in linear time by doubling a 2-approximation of it. A trivial lower bound LB^0 is 0. We provide an improved heuristic for UB and LB via sampling:

- *Upper Bound (UB):* We observe that empirically, as depicted in Fig. 3, sub-sampling an α fraction of the data requires a larger radius to form a core-point since data is more sparse, so ε^* increases. Hence, we produce the upper bound

$$UB = TS(X_{\mathcal{R}, 1:D}, LB^0, UB^0, MinPts, itr) \quad (3)$$

for $\mathcal{R} \sim \text{Uniform}(G \subset \{1, \dots, N\} : |G| = \lceil \alpha N \rceil)$

- *Lower Bound (LB):* Empirically, projecting the data on a random subset of αD dimensions, reduces ε^* since it brings the data closer. Hence, define

$$LB = TS(X_{1:N, \mathcal{T}}, LB^0, UB, MinPts, itr) \quad (4)$$

for $\mathcal{T} \sim \text{Uniform}(H \subset \{1, \dots, D\} : |H| = \lceil \alpha D \rceil)$

Our resulting clustering algorithm TSclustering (Alg. 3) invokes TS 3 times to find UB, LB , and ε^* .

Algorithm 3 TSclustering($X, MinPts, itr$)

```

1:  $UB^0 \leftarrow 2 \max_{i \in \{2, \dots, N\}} d(X_1, X_i)$ 
2:  $LB^0 \leftarrow 0$ 
3:  $\mathcal{R} \leftarrow$  sample  $\lceil \alpha N \rceil$  points from  $X$ 
4:  $\mathcal{T} \leftarrow$  sample  $\lceil \alpha D \rceil$  dimensions from  $X$ 
5:  $UB \leftarrow TS(X_{\mathcal{R}, 1:D}, LB^0, UB^0, MinPts, itr)$ 
6:  $LB \leftarrow TS(X_{1:N, \mathcal{T}}, LB^0, UB, MinPts, itr)$ 
7:  $\varepsilon^* \leftarrow TS(X, LB, UB, MinPts, itr)$ 
8: return  $A(X, \varepsilon^*, MinPts)$ 
```

Runtime Analysis: TS executes itr iterations, each invoking DBSCAN twice, resulting in $O(itr \cdot DN^2)$. With itr empirically set to 6, the overall complexity remains $O(DN^2)$, improving upon prior works' computational efficiency.

5.2 Ternary Search Estimator (TSE)

To further optimize the runtime of the heavy part of Alg. 3 (line 7) which uses the whole data and full dimension, we propose an estimator (*TSE*) for ε^* obtained by sampling an α fraction of the data and dimensions simultaneously. The intuition is that the opposite influences of sampling the data and dimensions on ε^* should roughly cancel out. Formally, we produce the estimate by replacing line 7 with $TS(X', LB, UB, MinPts, itr)$, where $X' = X_{\mathcal{R}, \mathcal{T}}$ and \mathcal{R}, \mathcal{T} are sampled as in Eq. 3,4. To reduce the variance, we repeat the above m times ($m = 30$ in our experiments) and average the estimates.

6 Evaluation

We evaluated the unimodality property and our methods across various domains, tasks, datasets, and representation techniques, using both qualitative and quantitative analyses over 24 datasets (Tab. 1). To evaluate clustering with noise, we employed the standard Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and Noise [12,11,1] metrics. NMI normalizes by cluster entropy, suitable for varying cluster counts, while ARI provides chance-adjusted accuracy with error-balance penalty. Together, these metrics offer complementary perspectives [18,19].

Unimodality Analysis: Qualitatively, Fig. 4 in the appendix demonstrates the unimodal shape (in red) of $k(\varepsilon)$ across 6 datasets from various fields. We used evenly spaced samples of ε for these curves, and executed DBSCAN exhaustively for each value. See Fig. 6 in the appendix for 15 additional datasets.

Table 1: DIP Test over datasets [20,21,22,23,24,25,26] and embeddings [26,27,28,29,30,31].

Dataset	Labels	N	Embed.	D	Task	p_{val}^{DIP}	Dataset	Labels	N	Embed.	D	Task	p_{val}^{DIP}
LFW	1,680	13,233	DNet	256	Face	>99.9%	AMCDv5	N/A	13,406	CAST	2,048	Anim	14.9%
ImNet1k	1,000	50,000	CLIP	512	OD	>99.9%	AMCDv6	N/A	14,372	CAST	2,048	Anim	6.4%
ImNet1k	1,000	50,000	Hiera	1,000	OD	33.8%	AMCDv7	N/A	14,460	CAST	2,048	Anim	8.4%
CIFAR	100	60,000	CLIP	512	OD	8.9%	AMCDv8	N/A	14,748	CAST	2,048	Anim	14.8%
CIFAR	100	60,000	Hiera	1,000	OD	>99.9%	CASTv1	N/A	2,648	CAST	2,048	Anim	6.4%
Reuters	135	21,578	ADA2	1,536	Doc	99.8%	CASTv2	N/A	4,215	CAST	2,048	Anim	79.1%
ESC-50	50	1,024	CLAP	1,024	Audio	41.3%	CASTv3	N/A	4,633	CAST	2,048	Anim	14.8%
FACE	N/A	45,207	DNet	256	Face	>99.9%	CASTv4	N/A	4,163	CAST	2,048	Anim	99.4%
AMCDv1	N/A	15,395	CAST	2,048	Anim	29.3%	CASTv5	N/A	4,959	CAST	2,048	Anim	14.8%
AMCDv2	N/A	13,102	CAST	2,048	Anim	52.5%	CASTv6	N/A	5,639	CAST	2,048	Anim	99.4%
AMCDv3	N/A	14,676	CAST	2,048	Anim	79.0%	CASTv7	N/A	4,795	CAST	2,048	Anim	52.5%
AMCDv4	N/A	14,676	CAST	2,048	Anim	29.4%	Urban8k	N/A	8,732	CLAP	1,024	Audio	99.6%

Quantitatively, to validate the unimodality of $k(\varepsilon)$ empirically we perform the DIP test [15]. Its null hypothesis is that the data is unimodal, and it is rejected for $p_{val} < 5\%$. The test demonstrated strong insignificance for all 24 NLP, Vision, and Audio datasets in Tab. 1, i.e., $k(\varepsilon)$ is unimodal on these datasets.

Cluster Analysis: Prior studies predominantly focus on synthetic, low-dimensional datasets. In contrast, this work emphasizes applications in high-dimension, comparing our methods (TS and TSE) to KMeans [32], HDBSCAN [8], VDBSCAN [6], OPTICS [5], SS-DBSCAN [12], AMD-DBSCAN [13], AEDB-SCAN [14], and AutoEps [9]. Due to the absence of open-source implementations, we re-implemented the latter four algorithms ourselves. We find that the baseline algorithms usually struggle in high dimensions, seldom producing degenerate outputs. We test against the ground-truth labels of classification datasets from NLP (Reuters), Vision (LFW), and Audio (ESC) using the metrics NMI , ARI , k , Noise, and runtime (Tab. 2).

An ideal clustering approximates the true number of clusters k , which in our case is the number of classification labels. TS/TSE provided the closest k approximations (roughly 5% error), reinforcing our hypothesis that ε^* reveals the natural clustering. Methods like SS-DBSCAN, which subsample data, overestimate ε , resulting in a single cluster. Note that KMeans requires the parameter k , which we set via the Elbow Method for maximal Inertia curvature [2].

For NMI and ARI , since all the algorithms assign points as noise,³ we excluded these noise-labeled points from the NMI and ARI computation to isolate the clustering quality from the noise prediction. TS/TSE consistently achieved the best scores across datasets with $P_{val} < 10^{-5}$ in Friedman non-parametric test.

Regarding noise detection, the ideal outcome identifies true noise. OPTICS underestimated ε , labeling nearly all points as noise, which is clearly incorrect, whereas VDBSCAN and AEDBSCAN significantly overestimated k . For the LFW dataset, where we defined noisy points as people with only one image (4,069/13,233 \approx 30.7%), TS/TSE provided noise estimates closely matching this true value (\approx 30.5%). For runtime, TSE improved on TS which was competitive with the baselines.

Table 2: Evaluation over Reuters, LFW, and ESC. We compare our methods (TS, TSE) with: KMeans (KM), HDBSCAN (HD), VDBSCAN (VD), OPTICS (OP), SS-DBSCAN (SS), AMD-DBSCAN (AM), AEDBSCAN (AE) and AutoEps (Ep). In gray, results with $Noise > 90\%$.

Dataset	Reuters (k= 135)					LFW (k= 1,680)					ESC (k= 50)				
Method	$NMI \uparrow$	$ARI \uparrow$	k	$Noise \downarrow$	T[s]↓	$NMI \uparrow$	$ARI \uparrow$	k	$Noise \downarrow$	T[s]↓	$NMI \uparrow$	$ARI \uparrow$	k	$Noise \downarrow$	T[s]↓
KM [32]	58.5%	19.9%	41	0.0%	1,917	78.0%	78.1%	773	0.2%	17,315	95.1%	83.3%	43	0.0%	306
HD [8]	62.0%	2.4%	1,247	61.4%	240	72.1%	36.3%	393	56.8%	105	86.2%	44.6%	52	17.3%	8
VD [6]	55.4%	0.3%	2,296	27.5%	246	92.3%	12.0%	2,661	38.0%	84	78.7%	20.9%	447	23.3%	10
OP [5]	61.3%	20.5%	37	97.1%	505	64.1%	24.5%	390	65.8%	202	56.3%	3.8%	53	59.7%	16
SS [12]	0.0%	0.0%	1	22.9%	230	13.5%	4.6%	2	52.7%	252	85.9%	46.6%	43	16.0%	9
AM [13]	41.1%	28.3%	134	6.2%	69	71.7%	20.4%	281	34.0%	29	83.4%	31.5%	93	18.3%	13
AE [14]	49.8%	4.5%	974	24.6%	144	91.8%	23.6%	1,944	48.7%	53	83.9%	46.7%	230	20.8%	7
Ep [9]	66.4%	67.1%	646	67.5%	2,377	11.9%	1.7%	56	57.1%	82	91.8%	77.3%	144	36.0%	24
TS (ours)	77.5%	93.9%	138	55.2%	152	99.0%	96.8%	1,697	30.5%	60	97.4%	90.3%	57	32.3%	5
TSE (ours)	77.8%	92.9%	150	38.0%	24	99.0%	96.7%	1,694	30.4%	41	96.7%	85.2%	48	14.7%	2

Hyper-Parameter Tuning: TS/Clustering relies on two hyper-parameters: α for sub-sampling N and D to give initial upper and lower bounds for ε^* , and itr to bound the number of ternary search iterations within TS. Both parameters trade off precision and runtime. Fig. 5(a) illustrates the gap between LB and UB as a decreasing function of α . We selected $\alpha = 0.2$ to balance the gap size reduction with runtime. Recall that we try to maximize k and find its mode k^* . In Fig. 5(b), we illustrate the resulting approximation ratio ($\frac{k}{k^*}$) as a function of TS iterations over the Reuters, LFW, ESC, and CAST datasets. We selected $itr = 6$ since the approximation ratio converges just before this value.

³ For KMeans clustering, we define the noisy points to be the singleton clusters.

7 Conclusion

This paper addresses the problem of parameter tuning in DBSCAN. We identify a unimodality property of $k(\varepsilon)$, and support it empirically and theoretically. We find that maximizing $k(\varepsilon)$ provides a good clustering, and give a novel method to automatically find this ε with an adapted version of the Ternary Search algorithm. Our empirical results on diverse datasets demonstrate improved precision and reduced noise, highlighting its potential for various data mining applications.

Future works may include: **1.** creating a sub-linear estimator for ε^* , **2.** improving the runtime with numerical optimization algorithms, which perhaps incorporate priors or gradients, **3.** adapting the approach to multi-density distributions, and **4.** scaling out for distributed big-data clustering.

8 Acknowledgments

This work was supported by the Joint NSFC-ISF Research Grant no. 3077/23.

References

1. Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, Portland, Oregon, 1996. AAAI.
2. Erich Schubert. Stop using the elbow criterion for k-means and how to choose the number of clusters instead. *SIGKDD*, 25(1):36–42, 2023.
3. Jiawei Han, Micheline Kamber, and Data Mining. Concepts and techniques. *Morgan kaufmann*, 340:94104–3205, 2006.
4. Theresa Ullmann, Anna Beer, Maximilian Hünemörder, Thomas Seidl, and Anne-Laure Boulesteix. Over-optimistic evaluation and reporting of novel cluster algorithms: An illustrative study. *Advances in Data Analysis and Classification*, 17(1):211–238, 2023.
5. Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. *SIGMOD*, 28(2):49–60, 1999.
6. Peng Liu, Dong Zhou, and Naijun Wu. VDBSCAN: varied density based spatial clustering of applications with noise. In *ICSSSM*, pages 1–4, Chengdu, China, 2007. IEEE, IEEE.
7. Mohammad Mahmudur Rahman Khan, Md Abu Bakr Siddique, Rezoana Bente Arif, and Mahjabin Rahman Oishe. Adbscan: Adaptive density-based spatial clustering of applications with noise for identifying clusters with varying densities. In *iCEEiCT*, pages 107–111, Dhaka, Bangladesh, 2018. IEEE.
8. Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *PA-KDD*, pages 160–172, Berlin, 2013. Springer.
9. Manisha Naik Gaonkar and Kedar Sawant. Autoepsdbscan: Dbscan with eps automatic for large dataset. *IJACTE*, 2(2):11–16, 2013.
10. Junhao Gan and Yufei Tao. Dbscan revisited: Mis-claim, un-fixability, and approximation. In *SIGMOD*, pages 519–530, Melbourne, Australia, 2015. SIGMOD.

11. Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.
12. Gloriana Monko and Masaomi Kimura. Optimized dbscan parameter selection: Stratified sampling for epsilon and gridsearch for minimum samples. *Software Engineering and Automation*, 1:43–61, 11 2023.
13. Ziqing Wang, Zhi qin Ye, Yuyang Du, Yi Mao, Yanying Liu, Ziling Wu, and Jun Wang. AMD-DBSCAN: An adaptive multi-density dbscan for datasets of extremely variable density. *DSAA*, pages 1–10, 2022.
14. Vidhi Mistry, Urja Pandya, Anjana Rathwa, Himani Kachroo, and Anjali Jivani. AEDBSCAN—adaptive epsilon density-based spatial clustering of applications with noise. In *Progress in Advanced Computing and Intelligent Engineering*, pages 213–226, Singapore, 2021. Springer.
15. J. A. Hartigan and P. M. Hartigan. The dip test of unimodality. *The Annals of Statistics*, 13(1):70–84, 1985.
16. Luc Devroye and Luc Devroye. Uniform and exponential spacings. *Non-Uniform Random Variate Generation*, pages 206–245, 1986.
17. Manpreet Singh Bajwa, Arun Prakash Agarwal, and Sumati Manchanda. Ternary search algorithm: Improvement of binary search. In *Indiacom*, pages 1723–1725, New Delhi, India, 2015. IEEE, IEEE.
18. Christopher D Manning. *Introduction to information retrieval*. Syngress, 2008.
19. Douglas Steinley. Properties of the hubert-arable adjusted rand index. *Psychological methods*, 9(3):386, 2004.
20. Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Workshop on faces in ‘Real-Life’ Images: detection, alignment, and recognition*, Rio de Janeiro, Brazil, 2008. IEEE.
21. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, Miami, Florida, 2009. Vision Sciences Society, IEEE.
22. Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *University of Toronto*, 1:1–54, 2009.
23. David Lewis. Reuters-21578 text categorization test collection. *Distribution 1.0, AT&T Labs-Research*, 1, 1997.
24. Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *ACM MM*, pages 1015–1018, Brisbane, Australia, 2015. ACM Press.
25. Krishna Somandepalli, Naveen Kumar, Tanaya Guha, and Shrikanth S Narayanan. Unsupervised discovery of character dictionaries in animation movies. *IEEE Transactions on Multimedia*, 20(3):539–551, 2017.
26. Oron Nir, Gal Rapoport, and Ariel Shamir. Cast: Character labeling in animation using self-supervision by tracking. In *Computer Graphics Forum*, volume 41, pages 135–145, Reims, France, 2022. Wiley Online Library, EuroGraphics.
27. Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.
28. Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, Jitendra Malik, Yanghao Li, and Christoph Feichtenhofer. Hiera: a hierarchical vision transformer without the bells-and-whistles. In *ICML, ICML’23*, Honolulu, Hawaii, USA, 2023. JMLR.org.

29. Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, Virtual, 2021. PMLR.
30. Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1, 2019.
31. Benjamin Elizalde, Soham Deshmukh, and Huaming Wang. Natural language supervision for general-purpose audio representations. In *ICASSP*, pages 336–340, Seoul, Korea, 2024. IEEE.
32. J MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*, 1967.

9 Appendix

9.1 Additional Qualitative Evaluations of Near-Unimodality

To support our qualitative claims of Near-Unimodality, in Fig. 6 we illustrate $k(\varepsilon)$ over 15 additional datasets from SAIL AMCD [25] and CAST [26], and plot the run of TSclustering over them. SAIL-AMCD and CAST are collections of 15 animated videos of different styles where each video has its own set of detected characters embedded using the CAST embeddings of dimension 2,048.

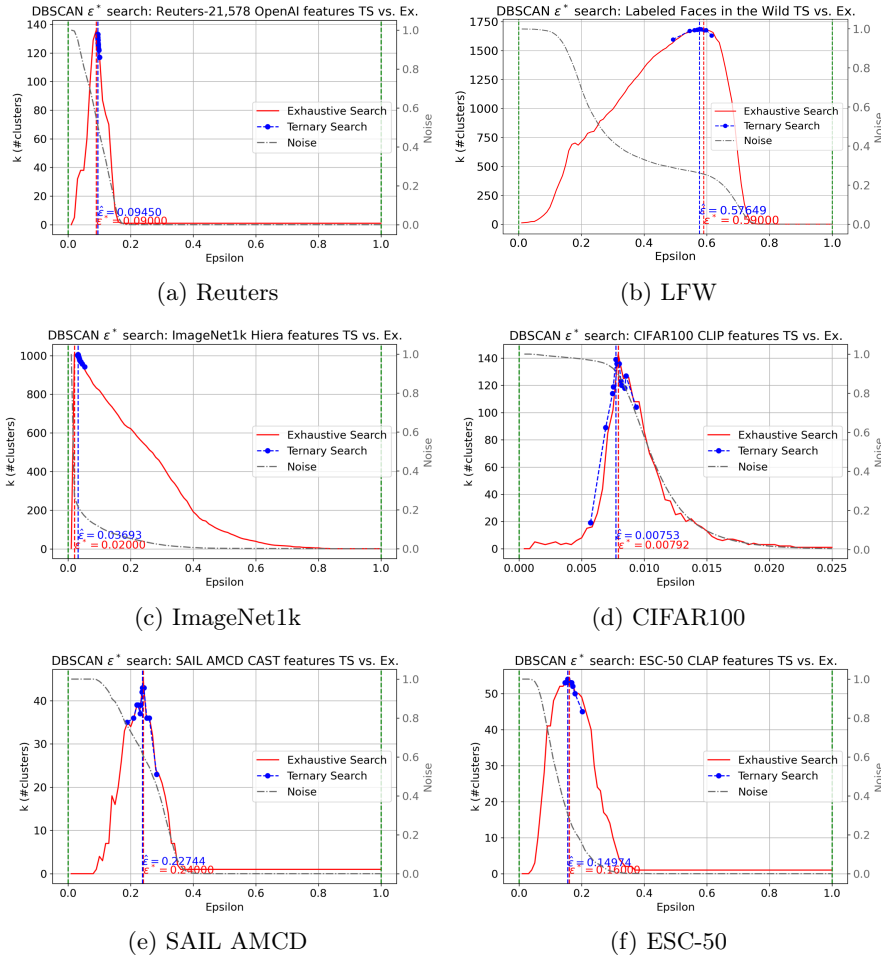
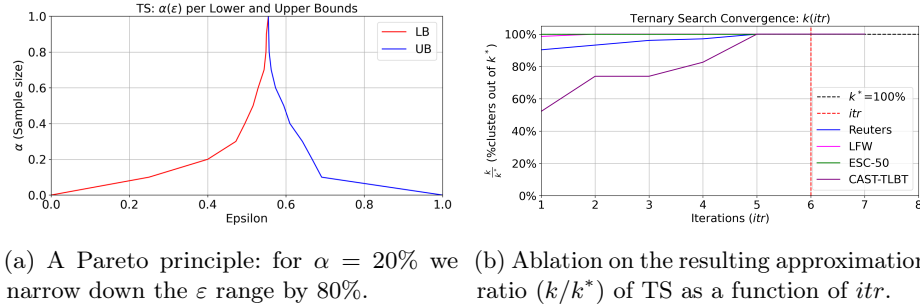
Fig. 4: Datasets across Vision, Audio, and NLP demonstrate unimodality of $k(\epsilon)$.

Fig. 5: Hyper-parameter tuning.

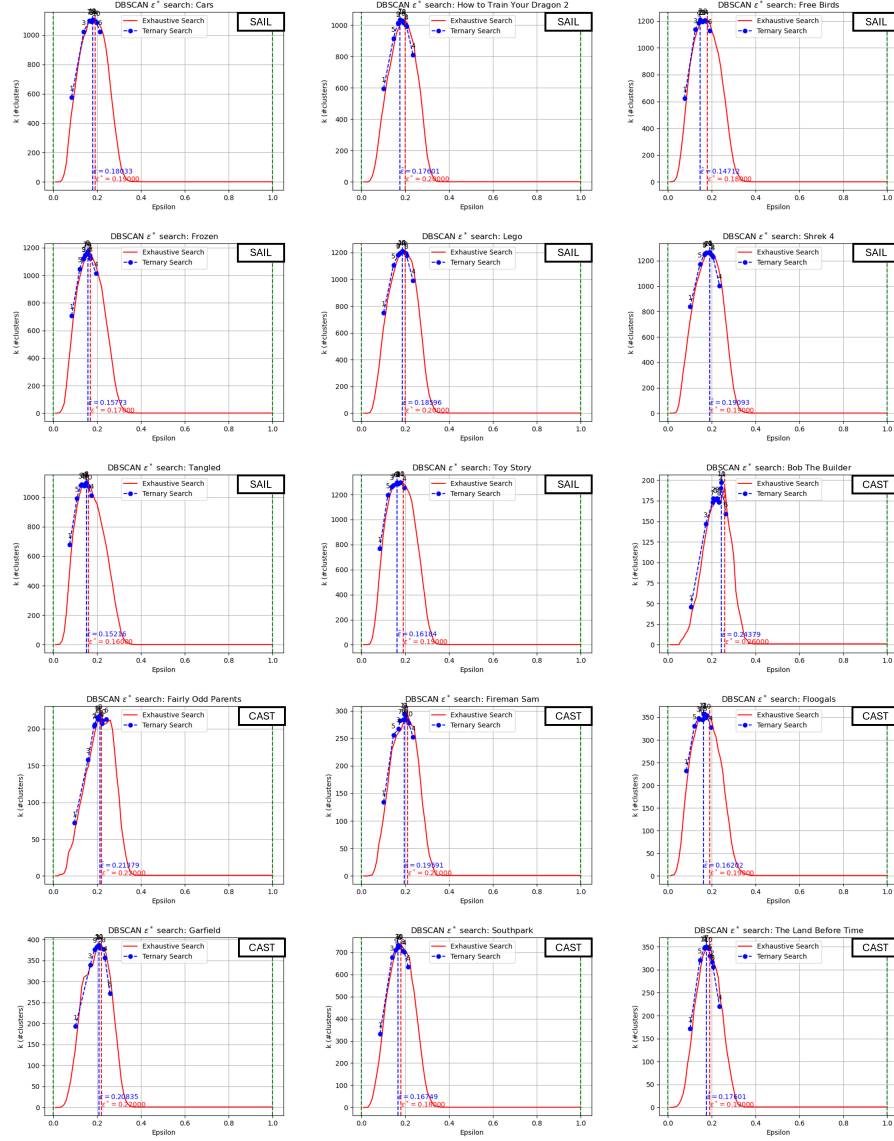


Fig. 6: The function $k(\epsilon)$ plotted over 15 datasets from SAIL-AMCD and CAST, each containing embeddings of multiple characters in a video. We observe a clear unimodal shape on all datasets.