DAG-Structured Clustering by Nearest Neighbors: Supplement

A Algorithmic Details

A.1 Linkage Functions & k-NN Graph Sparsification

LLAMA, like agglomerative hierarchical clustering methods, uses of a linkage function. The performance (both accuracy and speed) of the clustering algorithm will depend on the linkage function. Frequently used linkages are a function of a *similarity* function between points, $\texttt{sim} : X \times X \rightarrow \mathbb{R}$. We will focus on linkages $f(\cdot, \cdot)$ which are symmetric, ie. $f(C_i, C_j) = f(C_j, C_i)$. For instance, given two sets C_i and C_j, *single linkage* is the maximum similarity between an element in C_i and C_j, $\max_{x_i, x_j \in C_i \times C_k} \texttt{sim}(x_i, x_j)$; and *average linkage* is the average similarity between pairs of elements in the two clusters, $\frac{1}{|C_i||C_j|} \sum_{x_i, x_j \in C_i \times C_k} \texttt{sim}(x_i, x_j)$.

To make the construction of $\mathcal{N}^{(i)}$ more efficient, we build k-nearest neighbor graphs with respect to the *similarity* function ($\texttt{sim}$) for a dataset. We weight the edges of the graph with the similarity between the points. Edges that are missing from the graph are assumed to have 0 similarity. We can use this k-nearest neighbor graph with vertices X and edges E to define an analogous average linkage: $\frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} w_{ab} \mathbb{I}[(a, b) \in E]$ where $\mathbb{I}[(a, b) \in E]$ determines if the edge is in the graph. When computing $f(\cdot, \cdot)$ we can then restrict our consideration for candidate nearest neighbors in $\mathcal{N}^{(i)}$ to connected nodes. When considering candidate nearest neighbors of a cluster, we consider other clusters such that there is at least one edge in the nearest neighbor graph between the points in the clusters.

Previous work (Murtagh, 1983; Müllner, 2011) has shown that we can efficiently update the linkage function values between newly merged clusters using the linkage function values of existing ones. For tree structures, these are well known (Müllner, 2011) and provide for massive speedups in the methods as the average linkage can be computed as the sum of values for cluster pairs (instead of having to consider all of their descendant points). However, if the clusters are overlapping (as in our setting) the standard update rules for certain linkages will no longer be technically correct such as for average linkage. Empirically, however we find that approximating the average linkage by using the standard update rules from Müllner (2011) achieves good performance. Interestingly, for average linkage, this approximation looks very much like doing a bag-based average linkage where the number of times edges are double counted is a function of number of times the nodes has appeared in the overlap of two nodes. Table 2 provides a comparison between using this approximate computation using the update rule and the exact version.

B Theoretical Analysis

In our theoretical analysis, we follow Monath et al. (2019a) and make the assumption that the linkage function f is symmetric and without ties.

B.1 Model-Based Separation Theorem

Lemma 1 Given a dataset X and a linkage function f such that X is model-based separated with respect to f, let $\mathcal{H}^\star$ be the target partition corresponding to the separated data. In each round of LLAMA, each pair of nearest neighbors $(C, C') \in \mathcal{N}^{(i)}$, will satisfy either:

1. $\exists\ C^\star \in \mathcal{H}^\star$ such that $C \subseteq C^\star$ and $C' \subseteq C^\star$, or
2. $\exists\ C^\star \in \mathcal{H}^\star$ such that $C^\star \subseteq C$ or $C^\star \subseteq C'$.

Proof. We will prove this by induction. The first round of the algorithm, in which each point sits in its own cluster, satisfies the above property. Now let us assume that $\mathcal{N}^{(i-1)}$ has the above property. We want to show that $\mathcal{N}^{(i)}$ has the property as well. Each $C \in \mathcal{H}^{(i-1)}$ finds its nearest neighbor in $\mathcal{H}^{(i-1)}$ according to the linkage function f, we denote this as $C' = \text{argmin}_{C'' \in \mathcal{H}^{(i-1)} \setminus C} f(C, C'')$. There are three cases.

Case A: $\exists\ C^\star \in \mathcal{H}^\star$, s.t., $C = C^\star$. In this case, the node C corresponds exactly to the ground truth cluster. For any node that it pairs with it will satisfy Condition (2) above.

Case B: $\exists\ C^\star \in \mathcal{H}^\star$**, s.t.,** $C^\star \subseteq C$. In this case, as a ground truth cluster is already consumed by the cluster C, Condition (2) above is already satisfied.

Case C: $\exists\ C^\star \in \mathcal{H}^\star$**, s.t.,** $C \subset C^\star$. This final case is the most interesting one. We will show that the node it chooses to pair with must be from the same ground truth cluster as C. By condition 1, we know that there must be a C' such that $C' \subseteq C^\star \setminus C$ because the $C \neq C^\star$. There also must exist a C' such that C is connected to C' according to $g(\cdot, \cdot)$. Therefore, by the definition of model-based separation C's nearest neighbor will be some cluster that is is connected to and that is in its cluster. Thus we will maintain property 1 in $\mathcal{N}^{(i)}$. □

Theorem 1 Given a dataset X and a linkage function f such that X is model-based separated with respect to f, let $\mathcal{H}^\star$ be the target partition corresponding to the separated data. Let $\mathcal{D}$ be the DAG-structured clustering produced by LLAMA (Alg. 1), then $\mathcal{H}^\star$ is a $\mathcal{D}$ consistent partition, $\mathcal{H}^\star \subset \mathcal{D}$.

Proof. We will prove this by contradiction and by Lemma 1. Suppose not, let $C^\star \in \mathcal{H}^\star$ be any of the ground truth clusters and $C^\star \notin \mathcal{D}$. In the first round, each member point of $C^\star$ appears as a singleton cluster. Define the pairs of clusters that are nearest neighbors and are both subsets of $C^\star$ in round i and subsequent nodes:

$$\mathcal{N}^{(i)}(C^\star) = \{(C, C') \mid (C, C') \in \mathcal{N}^{(i)},\ C \subset C^\star \wedge C' \subset C^\star\} \tag{13}$$

$$\mathcal{H}^{(i)}(C^\star) = \{C \cup C' \mid (C, C') \in \mathcal{N}^{(i)}(C^\star)\} \tag{14}$$

Let's consider the earliest round that the above is empty, $\mathcal{N}^{(i)}(C^\star) = \emptyset$, call this round e. We will now show that $C^\star \in \mathcal{H}_{e-1}$. Lemma 1 tells us that each member of the pairs in $\mathcal{N}^{(e-1)}(C^\star)$ must be both subsets $C^\star$ and so we know that: $\forall C'' \in \mathcal{H}^{(e-1)}(C^\star),\ C'' \subseteq C^\star$.

If $\mathcal{N}^{(e)}(C^\star)$ is empty, then we know that for each member $C \in \mathcal{H}^{(e-1)}(C^\star)$ it is the case that C found a nearest neighbor C' such that $C' \not\subset C^\star$. By model based separation, if $C \subset C^\star$ and $C \in \mathcal{H}^{(e-1)}(C^\star)$, then its nearest neighbor must be some other subset that is also a subset of $C^\star$. Some such subset must exist because there exists at least one point that connects the points in C to all other points $C^\star$ in the underlying model-based separation latent graph. And so C must not be a subset of $C^\star$. If this C is not a subset of $C^\star$, then by lemma 1 it must be a superset and by our supposition of $C^\star$ not being in $\mathcal{D}$, a strict superset. But this reaches a contradiction as each member of $\mathcal{H}^{(e-1)}(C^\star)$ was made by merging two pure subsets of $C^\star$. □

B.2 Noisy Model-Based Separation Analysis

Proposition 1 Given a dataset X and a symmetric linkage function f such that X is noisy model-based separated with respect to f, let $\mathcal{H}^\star$ be the target partition corresponding to the noisy model-based separated data. Let $\mathcal{D}$ be the DAG-structured clustering produced by LLAMA (Alg. 1), then $\mathcal{H}^\star$ is a $\mathcal{D}$ consistent partition, $\mathcal{H}^\star \subset \mathcal{D}$.

Proof. The first round of the algorithm creates $\mathcal{N}^{(1)}$ and the clusters that are input to the next round $\mathcal{H}^{(1)}$. We will show that $\mathcal{H}^{(1)}$ is model-based separated with respect to f and the original graph G (not noisy model-based separated) and then apply the results from Theorem 1.

To achieve this, we will show that for each $C^\star \in \mathcal{H}^\star$, $\exists\ C_1, C_2, \ldots, C_K \in \mathcal{H}_1$ such that $\cup_{i=1:k} C_i = C^\star$, there by showing that the original partition $C^\star$ is a model-based separated partition of $\mathcal{H}^{(1)}$. We can partition $\mathcal{H}^{(1)}$ into the connected and disconnected clusters:

$$\mathcal{H}^{(1)}_{\mathsf{conn}} = \{C \mid C \text{ is connected in } G\} \tag{15}$$

$$\mathcal{H}^{(1)}_{\mathsf{sep}} = \{C \mid C \text{ is not connected in } G\} \tag{16}$$

For each ground truth cluster $C^\star$ the noisy model-based separation property tells us that at most $1/2$ of the points of any ground truth cluster can have nearest neighbors that are not connected (and outside the cluster). And so, we have that for each ground truth cluster $C^\star$, each point must participate in at least one member of $\mathcal{H}^{(1)}_{\mathsf{conn}}$, i.e.,

$$\forall C^\star \in \mathcal{H}^\star\ \forall x \in C^\star\ \exists C \in \mathcal{H}^{(1)}_{\mathsf{conn}},\ x \in C \tag{17}$$

Now observe that by definition of G, only those ground truth clusters described in the last equation will be connected. For the remaining rounds of the algorithm, all cluster sizes will be greater than 1 and so all nearest neighbors will be from the same cluster, i.e., model-based separation holds. We can use the result from Theorem 1. □

Proposition 2 There exists a datasets X and symmetric linkage function f such that X is noisy model-based separated wrt f, let $\mathcal{H}^\star$ be the target partition corresponding to the noisy model-based separated data. HAC and GRINCH produces a structure $\mathcal{T}$ such that $\mathcal{H}^\star$ is not a tree consistent partition, $\mathcal{H}^\star \not\subset \mathcal{T}$.

Proof. Consider a very simple dataset with three points $X = \{a, b, c\}$, let the target partition be $\mathcal{H}^\star = \{\{a, b\}, \{c\}\}$. Now let $f(b, c) = 2$ and $f(a, b) = 1$ and $f(a, c) = 0$. We observe that HAC and Grinch will put b and c in the same cluster and so could not represent $\{a, b\}$. However, we also have that a's nearest neighbor is b and so the DAG-structured method would be able to represent the cluster $\{a, b\}$. □

B.3 Complexity

Proposition *(Space Complexity)*. Given a dataset of N points, LLAMA produces DAG-structured clusterings with at most $O(N^2)$ nodes.

Proof. Assuming we have a symmetric linkage function, in each round, each cluster is merged with one other cluster. By the pigeon-hole principle, a round starting with N clusters will produce at most $N - 1$ clusters. Therefore, the total number of nodes that can be produced is $O(\sum_{i=N}^{1} i) = O(N^2)$. □

Proposition *(Time Complexity)*. Given a dataset of N points, R rounds of LLAMA produces requires at most $O(R * N^2)$ linkage function computations.

Proof. In each round, we need to find the nearest neighbor of each of the clusters produced by the previous rounds. Without a nearest neighbor index, each round would require $O(N^2)$ time to compute the nearest neighbor of each cluster. If nearest neighbor index structures are used, this time can of course be reduced. □

Proposition *(Number of Rounds)*. Let $\mathcal{H}^\star$ be the target partition of a dataset that is (noisy) model-based separated, let K be the size of the largest cluster in $\mathcal{H}^\star$. $K = \max_{C \in \mathcal{H}^\star} |C|$. After K rounds, LLAMA produces a structure that contains $\mathcal{H}^\star$.

Proof. We observe that that all points from the same ground truth cluster will be merged before points from different ground truth clusters. In the worst case, this means that a cluster with K points will take K rounds (by the same logic as the space complexity above) to form. □

What DAG structures can be formed by Llama? We note that the LLAMA algorithm cannot produce any DAG-structured clustering. Instead, it is limited to a subset with polynomial size. In future work, we hope to better understand the properties of the kind of structure LLAMA can produce.

C Empirical Analysis

C.1 Analysis of Jaccard-based Clustering Metrics

Dendrogram Purity (Heller and Ghahramani, 2005b) is a metric that is often used to evaluate the quality of a hierarchical clustering of a dataset which has a ground truth flat partition. Rather than demanding a particular flat clustering be extracted from the tree structure, dendrogram purity evaluates the quality of the tree consistent partitions encoded in the hierarchical clustering. It is defined as:

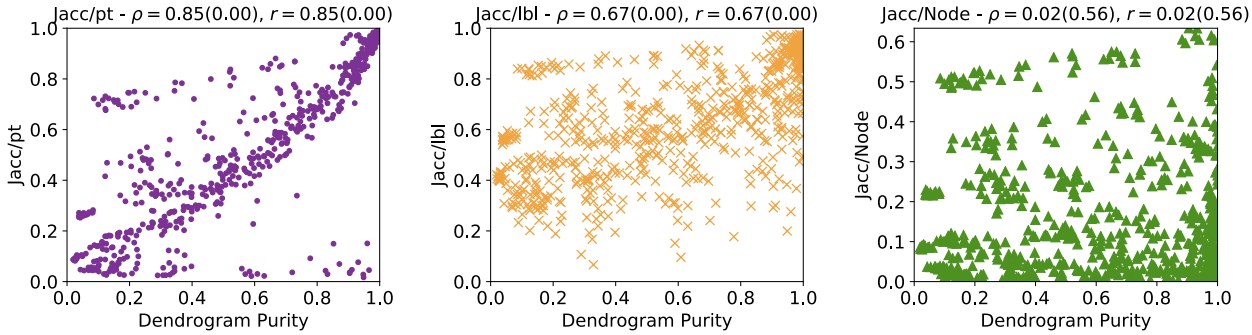

Figure 4: **Dendrogram Purity and Jaccard Metrics on Synthethic Data**. We report the Spearman (ρ) and Pearson r correlation for each and p value in parenthesis. We observe that Jacc/pt is well correlated with dendrogram purity. While the other metrics are not correlated, this does not diminish our interest in them as metrics. Jacc/node captures how precise or compact the structures are, unlike dendrogram purity. Jacc/lbl measures at the label level how well represented the ground truth clusters are. Unlike Jacc/pt and dendrogram purity, Jacc/lbl weights each ground truth cluster equally independent of the size of the cluster. As the data here has CRP distributed cluster sizes, it is no surprise that Jacc/lbl looks quite different than Jacc/pt.

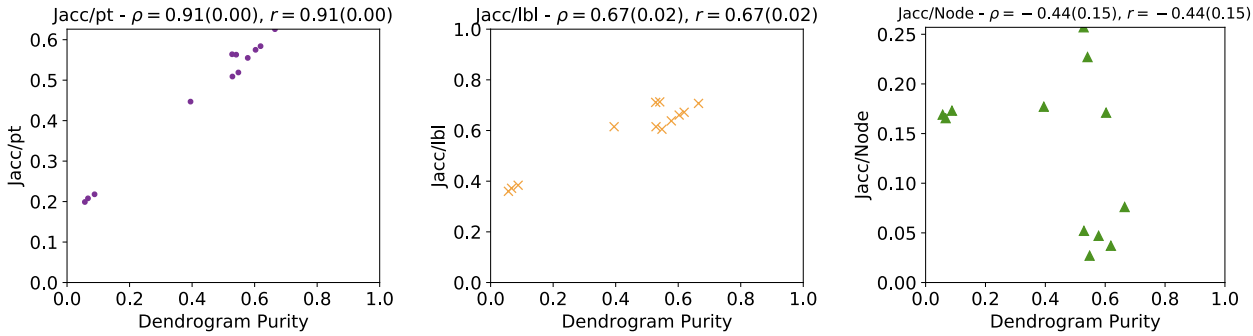

Figure 5: **Dendrogram Purity and Jaccard Metrics on Real Data**. As in Figure 4, we report the values of the metrics in this case on the hierarchical clustering benchmark datasets.

Let $\mathcal{T}$ be a hierarchical clustering of dataset X. Let $\mathcal{H}^\star = \{C_1^\star, \ldots, C_K^\star\}$ be a ground truth flat partition of X. The dendrogram purity (DP) of $\mathcal{T}$ with respect to $C^\star$ is:

$$\frac{1}{Z} \sum_{C^\star \in \mathcal{H}^\star} \sum_{\substack{x, x' \in C^\star \times C^\star, \\ x \neq x'}} \texttt{purity}(\texttt{lca}(x, x', \mathcal{T}), C^\star) \tag{18}$$

$$Z = \sum_{C^\star \in \mathcal{H}^\star} \frac{1}{2} |C^\star|(|C^\star| - 1) \tag{19}$$

where $\texttt{lca}(x, x', \mathcal{T})$ gives the least common ancestor of x and x' in $\mathcal{T}$ and $\texttt{purity}(n, C^\star)$ is defined as the fraction of descent leaves of n that belong to $C^\star$, that is: $\texttt{purity}(n, C^\star) = |\texttt{lvs}(n) \cap C^\star| / |\texttt{lvs}(n)|$, where $\texttt{lvs}(n)$ gives the leaves of the node n.

We note that there are trivial DAG structures which would achieve perfect dendrogram purity. In particular, the DAG structure which contains the cluster for each pair of points in the dataset.

We are interested to understand which of the Jacc/pt, Jacc/lbl, Jacc/node is most correlated to dendrogram purity. To analyze this, we sample synthetic data from Dirichlet Process Mixture Models with spherical variance. We sample 10 datasets from 75 different DPMM hyperparameter settings in $\mathbb{R}^{10}$ for a total of 750 datasets. The 75 settings come from the cartersian product of (number of points ({100, 1000, 5000}), variances ({0.25, 0.4, 0.5, 0.75, 1.0}), and alpha parameters of CRP ({1, 5, 10, 25, 100}). For each dataset we run the best tree-based

		Num. Points	Num. Labels	Dim
Partition-based	**ALOI**	108K	1000	128
	ILSVRC (Sm.)	50K	1000	2048
	Speaker	36.5K	4958	6388
	ImageNet	100K	17K	2048
	ILSVRC (Lg.)	1.2M	1000	2048
Cover-based	**EURLex-4k**	19K	3993	5000
	Bibtex	7K	159	1836
	Delicious	16K	983	500
	MediaMill	43.9K	101	120
	Wiki10-31K	20K	31K	101K

Table 6: **Dataset Statistics**.

method, reciprocal nearest neighbors and report all metrics. We plot each metric against dendrogram purity in Figure 4 and observe that Jacc/pt is most correlated to dendrogram purity. We note that dendrogram purity is by no means the only metric that we are interested in and so the lack of correlation for the other two metrics is not a negative result, it simply implies that they capture something different about the structures. Furthermore, for this particular choice of generative models, which encourages rich-get-richer cluster sizes, it is no surprise that Jacc/lbl looks quite different than Jacc/pt. Similarly, Jacc/node measures the compactness of the structure and so captures properties that dendrogram purity does not.

We also show these same plots for the results of all three tree-based methods compared (reciprocal nearest neighbor, Affinity, Grinch) on the clustering benchmarks. The results follow a similar trend (see Fig. 5).

C.2 Dataset Sizes

Table 6 provides the statistics for each dataset used in the clustering and cover-based evaluations. For evaluation on ILSVRC (Lg.) we use a randomly selected subset of 50K points following Kobren et al. (2017).

C.3 Hyperparameter Analysis

We analyze two hyperpameters of LLAMA, RecipNN, and Affinity, the number of neighbors of the nearest neighbor graph (described above) and the number of rounds of the algorithm used. Figure 6 shows the results. We observe that around 20-40 rounds are required for competitive performance. We observe that the number of nearest neighbors between 3 and 1000 does not lead to major variation in performance. We observe that while LLAMA can become much more expensive than tree structures when the number of rounds or graph density becomes very large, the algorithm does not much more time than the tree-based methods to achieve better-than-tree structure performance.

C.4 Running Time Analysis

In Table 7, we report a timing comparison of running 100 rounds of LLAMA and Reciprocal NN algorithm on the clustering benchmarks. We use 10 threads in parallelizing each algorithm's computation of the neighbors and linkage function values. For LLAMA, we use the comparably efficient approximate average linkage. We report the time of clustering the pre-computed sparse graph (which is done using ScaNN (Guo et al., 2020)).

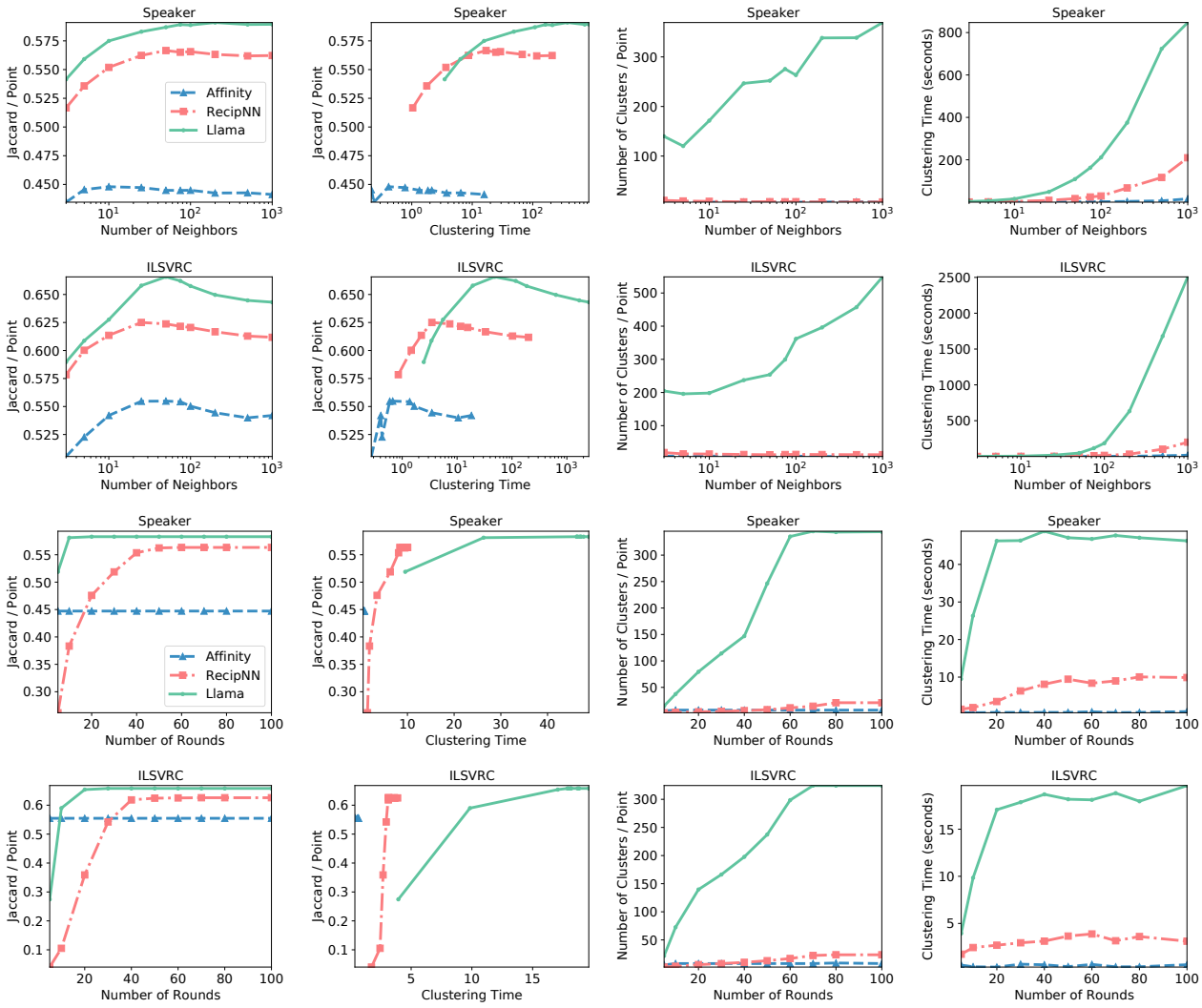

Figure 6: **Hyperparameter Analysis**. We compare performance on the Speaker and ILSVRC (Sm.) datasets using various numbers of rounds and various settings of the number of nearest neighbors in the nearest neighbor graph. We observe comparable performance across various kinds of nearest neighbors. We observe that around 20-40 rounds is required for competitive performance of the metrics. Importantly, while the complexity of Llama does grow faster than the other methods in terms of time and number of nodes, we observe good performance can be achieved in the parts of the time/space curves that are much closer to tree-based methods.

	Running Time (s)		**Avg. Clusters / Point**	
	RcNN	**Llama**	**RcNN**	**Llama**
ALOI	12.32	7.58	18.474	127.621
Speaker	10.43	53.63	19.819	238.21
ILSVRC (Sm.)	3.955	10.66	24.18	162.35
ImageNet	16.754	224.533	21.513	600.67
ILSVRC (Lg.)	86.32	495.19	44.387	356.311

Table 7: **Running Times & Structure Size**. The running time of the two algorithms on each of the clustering benchmarks. Interestingly, Llama takes less time on the ILSVRC (Sm.) dataset than the Speaker dataset, despite it being larger. We hypothesize that the time taken by Llama is directly impacted by the underlying structure of the dataset's similarity graph and with more separation in the data (as seems to be the case here), Llama can be more efficient. For the same runs as the timing numbers, we report the number of average number of clusters each point has been assigned to in the structures, which share a similar trend.