A. HCC Algorithm

Algorithm 1 describes hierarchical correlation clustering (HCC) in detail. The algorithm at the beginning assumes n singleton clusters, one for each object. For each cluster, it obtains the nearest cluster and the respective similarity. The algorithm then iteratively performs the following steps. i) Finds the two nearest clusters according to the inter-cluster (dis)similarity function defined in Eq. 5. ii) Merges the respective clusters to build a new cluster at a higher level. iii) Updates the inter-cluster similarity matrix $\mathbf{S}$, the nearest neighbor vector nn_ind and the respective similarities nn_sim.[4]

Algorithm 1 Hierarchical Correlation Clustering.

```
Require: A set of $n$ objects $\mathbf{O} = \{0, ..., n-1\}$ and the
    pairwise similarities $\mathbf{S}$.
 1: for all $i \in \mathbf{O}$ do
 2:     $nn\_ind[i] = \arg\max_j \mathbf{S}[i,j]$
 3:     $nn\_sim[i] = \max_j \mathbf{S}[i,j]$
 4: end for
 5: $n\_c = |\mathbf{O}|$ {shows the number of active clusters}
 6: while $n\_c > 1$ do
 7:     {find the indices of the two nearest clusters, w.l.g.
        we assume $u \leq v$}
 8:     $u = \arg\max_i nn\_sim[i]$
 9:     $v = nn\_ind[u]$
10:     {update the inter-cluster (dis)similarities, active clus-
        ters and other parameters}
11:     for all $i \in \{0, ..., n\_c\}$ do
12:         $new\_sim[i] = \mathbf{S}[i,u] + \mathbf{S}[i,v]$
13:     end for
14:     Remove($new\_sim[v]$)
15:     Remove($new\_sim[u]$)
16:     Remove($\mathbf{S}[v,:]$)
17:     Remove($\mathbf{S}[:,v]$)
18:     Remove($\mathbf{S}[u,:]$)
19:     Remove($\mathbf{S}[:,v]$)
20:     Append($\mathbf{S}, new\_sim$)
21:     Append($\mathbf{S}, [new\_sim, 0]^T$)
22:     $n\_c = n\_c - 1$
23:     Remove($nn\_ind[v]$)
24:     Remove($nn\_sim[u]$)
25:     Update ($nn\_ind$)
26:     Update ($nn\_sim$)
27:     Append($nn\_ind, \arg\max_j \mathbf{S}[n\_c, j]$)
28:     Append($nn\_sim, \max_j \mathbf{S}[n\_c, j]$)
29: end while
    Return the intermediate clusters and the dendrogram.
```

[4]In our implementation, we use a data structure similar to the linkage matrix used by *scipy* package in Python to encode the dendrogram and store the intermediate clusters.

B. Proofs

B.1. Proof of Lemma 1

One can show that the pairwise minimax dissimilarities across any given graph are identical to the pairwise minimax dissimilarities present in any minimum spanning tree obtained from the same graph. The proof is similar to the *maximum capacity* problem [40]. Thereby, the minimax dissimilarities are obtained by

$$\begin{aligned}\mathbf{D}^{MM}_{i,j} &= \min_{p \in \mathcal{P}_{ij}(\mathcal{G})} \left\{ \max_{1 \leq l \leq |p|-1} \mathbf{D}_{p(l)p(l+1)} \right\} \\ &= \max_{1 \leq l \leq |p_{ij}|-1} \mathbf{D}_{p(l)p(l+1)},\end{aligned} \quad (15)$$

where p_{ij} indicates the (only) path between i and j on a minimum spanning tree computed on $\mathcal{G}$. To obtain the minimax dissimilarities $\mathbf{D}^{MM}_{ij}$, we can just select the maximal edge weight on the only path between i and j on the minimum spanning tree.

On the other hand, the single linkage method and the Kruskal's minimum spanning tree algorithm are equivalent [35]. Thus, the dendrogram T obtained via single linkage sufficiently contains the pairwise minimax dissimilarities. Now, we elaborate that the minimax dissimilarities in Eq. 15 equal the dissimilarities defined in Eq. 6, i.e., $\mathbf{X}_{ij}$ is the largest edge weight on the path between i and j in the hierarchy.

Given i, j, let

$$T^* = \arg\min linkage(T') \quad \text{s.t. } i, j \in T' \text{ and } T' \in \mathcal{T}_T\,. \quad (16)$$

Then, T^* represents a minimum spanning subtree, which includes a path between i and j (because the root cluster of T^* contains both i and j) and it is consistent with a complete minimum spanning on all the objects. On the other hand, we know that for each pair of clusters $\mathbf{u}, \mathbf{v} \in T^*$ which have direct or indirect parent-child relation, we have, $linkage(\mathbf{u}) \geq linkage(\mathbf{v})$ iff $level(\mathbf{u}) \geq level(\mathbf{v})$. This implies the linkage of the root cluster of T^* represents the maximal edge weight on the path between i and j represented by the dendrogram T. Thus, $\mathbf{X}_{ij}$ represents $\mathbf{D}^{MM}_{ij}$

Thereby, minimax dissimilarities correspond to building a single linkage dendrogram and using the linkage as the pairwise dissimilarity between the objects in the two respective clusters. We know that according to Proposition 1 single linkage dendrogram is shift-invariant. Therefore, by shifting the pairwise dissimilarities by a sufficient α, there will be no change in the paths between the clusters of single linkage dendrogram, nor in the paths representing the minimax dissimilarities.

B.2. Proof of Theorem 2

Over a graph, we define a path between i and j to be *positive* if all the edge weights on the path are positive. Then, we have the following observations.

1. On a general graph $\mathcal{G}(\mathbf{O}, \mathbf{S})$, one can see that in the optimal solution of correlation clustering, if the two objects i and j are in the same cluster, then there is at least one positive path between them (the proof can be done by contradiction; if there is no such a path, then the two objects should be in separate clusters in order to avoid the increase in the cost function).

2. Whenever there is a positive path between i and j, then their minimax similarity $\mathbf{S}^{MM}_{i,j}$ will be necessarily positive. Therefore, when we apply correlation clustering to graph $\mathcal{G}(\mathbf{O}, \mathbf{S}^{MM})$, all the intra-cluster similarities of the optimal clusters will be positive. This corresponds to having a positive path between every two objects that are in the same optimal cluster, i.e., they are in the same connected component of $\mathcal{G}(\mathbf{O}, \mathbf{S}')$ where $\mathbf{S}'$ is defined as

$$\mathbf{S}'_{ij} = \begin{cases} 1, & \text{if } \mathbf{S}_{ij} > 0. \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

3. We can also conclude that when we apply correlation clustering to graph $\mathcal{G}(\mathbf{O}, \mathbf{S}^{MM})$, then for any optimal cluster $\mathbf{c}$, there is no object $i \notin \mathbf{c}$ such that i has a positive path to an object in $\mathbf{c}$. Otherwise, i and all the other objects outside $\mathbf{c}$ with positive paths to i would have positive paths to all the objects in $\mathbf{c}$ such that all of them should be clustered together.

Now we study the connection of connected components of graph $\mathcal{G}(\mathbf{O}, \mathbf{S}')$ to the optimal correlation clustering on $\mathcal{G}(\mathbf{O}, \mathbf{S}^{MM})$.

- There is a positive path between every two objects in a connected component of $\mathcal{G}(\mathbf{O}, \mathbf{S}')$. Thus, they are in the same optimal cluster of $\mathcal{G}(\mathbf{O}, \mathbf{S}^{MM})$.

- If two nodes i and j are at two different connected components, then there is no positive path between them either on $\mathcal{G}(\mathbf{O}, \mathbf{S}')$ or on $\mathcal{G}(\mathbf{O}, \mathbf{S}^{MM})$. Thus, they cannot be in the same cluster if we apply correlation clustering on $\mathcal{G}(\mathbf{O}, \mathbf{S}^{MM})$.

Thus, we conclude that the connected components of $\mathcal{G}(\mathbf{O}, \mathbf{S}')$ correspond to the optimal correlation clustering on graph $\mathcal{G}(\mathbf{O}, \mathbf{S}^{MM})$.

B.3. Proof of Theorem 3

The approximate algorithm in [5] iteratively picks an unclustered object and its positive neighbors as a new cluster. According to Theorem 1, the optimal solution of correlation clustering applied to $\mathcal{G}(\mathbf{O}, \mathbf{S}^{MM})$ corresponds to extracting the connected components of graph $\mathcal{G}(\mathbf{O}, \mathbf{S}')$, where $\mathbf{S}'$ is defined in Eq. 17.

Thus it is sufficient to show the two followings.

1. If the algorithm in [5] on $\mathcal{G}(\mathbf{O}, \mathbf{S}^{MM})$ picks i and j in the same cluster then $\mathbf{S}^{MM}_{i,j} = +1$. This indicates that i and j have a positive path on $\mathcal{G}(\mathbf{O}, \mathbf{S})$ (a positive path is defined in Theorem 1), i.e., i and j are in the same connected component of $\mathcal{G}(\mathbf{O}, \mathbf{S}')$.

2. If i and j are in different clusters according to algorithm [5] applied to $\mathcal{G}(\mathbf{O}, \mathbf{S}^{MM})$, then $\mathbf{S}^{MM}_{i,j} = -1$. This indicates that there is no positive path between i and j on $\mathcal{G}(\mathbf{O}, \mathbf{S})$ and also on $\mathcal{G}(\mathbf{O}, \mathbf{S}^{MM})$, i.e., i and j are in different connected component of $\mathcal{G}(\mathbf{O}, \mathbf{S}')$.

C. Additional Experimental Results

C.1. More results with UCI datasets

In Figure 5, we demonstrate the performance of HCC on different UCI datasets w.r.t. Rand score, and compare the results with other agglomerative methods (the results w.r.t. MI are shown in the main text in Figure 2). We observe that consistent with the MI measure, HCC yields the best scores, even at high noise levels and when the datasets are difficult to cluster.

C.2. HCC on 20 newsgroup data

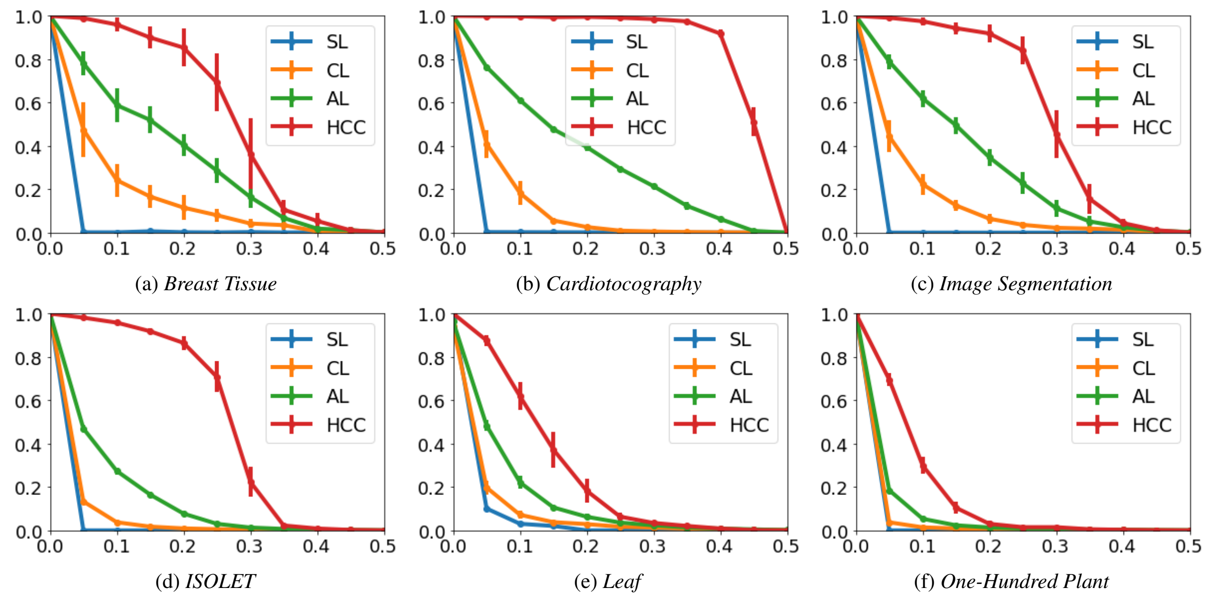
In the following, we study the performance of different methods on several subsets of 20 newsgroup data collection chosen randomly from different categories.

1. *news1*: the 3901 documents of the categories 'misc.forsale', 'rec.motorcycles', 'talk.politics.mideast', 'sci.med' (48596 dimensions).

2. *news2*: the 3743 documents of the categories 'alt.atheism', 'comp.sys.mac.hardware', 'sci.electronics', 'soc.religion.christian' (40735 dimensions).

3. *news3*: the 1984 documents of the categories 'sci.space', 'soc.religion.christian' (30749 dimensions).

4. *news4*: the 2877 documents of the categories 'comp.graphics', 'rec.sport.baseball', 'talk.politics.guns' (38177 dimensions).



(a) *Breast Tissue* (b) *Cardiotocography* (c) *Image Segmentation*

(d) *ISOLET* (e) *Leaf* (f) *One-Hundred Plant*

Figure 5. Rand score of hierarchical clustering methods applied to the UCI datasets (the x-axis shows the flip noise parameter η). Similar to the MI measure, HCC provides the best scores, even when the datasets are difficult to cluster.

Table 3. Performance of different hierarchical clustering methods on 20 newsgroup datasets. HCC yields the best results according to the different evaluation measures.

	news1		news2		news3		news4	
method	MI	Rand	MI	Rand	MI	Rand	MI	Rand
SL	0.034	0.017	0.043	0.039	0.021	0.044	0.020	0.038
CL	0.266	0.277	0.255	0.230	0.501	0.594	0.121	0.116
AL	0.287	0.228	0.342	0.344	0.685	0.750	0.498	**0.548**
HCC	**0.331**	**0.287**	**0.370**	**0.368**	**0.794**	**0.854**	**0.541**	0.499

Table 4. Performance of tree-preserving embedding methods on different 20 newsgroup datasets applied with GMM. The embeddings obtained by HCC yield better results.

	news1		news2		news3		news4	
method	MI	Rand	MI	Rand	MI	Rand	MI	Rand
SL	0.034	0.017	0.043	0.039	0.021	0.044	0.020	0.038
SL+GMM	0.191	0.158	0.108	0.097	0.166	0.210	0.134	0.120
CL	0.266	0.277	0.255	0.230	0.501	0.594	0.121	0.116
CL+GMM	0.271	0.275	0.272	0.239	0.522	0.587	0.118	0.119
AL	0.287	0.228	0.342	0.344	0.685	0.750	0.498	**0.548**
AL+GMM	0.309	0.279	0.358	0.350	0.701	0.773	0.503	0.525
HCC	0.331	0.287	0.370	0.368	0.794	0.854	0.541	0.499
HCC+GMM	**0.344**	**0.297**	**0.439**	**0.443**	**0.831**	**0.892**	**0.560**	0.519

For each dataset, we compute the TF-IDF vectors of the documents and apply PCA with 50 principal components. We obtain the similarity between every two documents via the cosine similarity between their respective PCA vectors. As has been discussed in detail in [37], adding a fixed number to all the pairwise similarities can possibly improve the clustering results.[5] Table 3 shows the results for various hierarchical clustering methods w.r.t. different evaluation criteria. Among the different methods, HCC usually yields

[5]It is suggested in [18] to adaptively shift the pairwise similarities so that the sum of pairwise similarities equals zero for each object in the dataset.

Table 5. Performance of different methods on webpage dataset. The embeddings obtained by HCC yield better results.

method	MI	Rand
SL	0.218	0.192
SL+GMM	0.254	0.211
CL	0.386	0.417
CL+GMM	0.392	0.405
AL	0.459	0.488
AL+GMM	0.472	0.491
HCC	0.583	0.575
HCC+GMM	**0.622**	**0.630**

the best results, and AL is the second best choice.

In the following, we investigate tree-preserving embedding on these datasets. Table 4 presents the results of tree-preserving embedding compared to hierarchical clustering. Specifically, we investigate the benefits of using hierarchical clustering for feature extraction. We observe, i) employing hierarchical clustering to extract features for a method such as GMM usually yields improving the results, and ii) HCC, whether used directly for clustering or for feature extraction, often gives superior results compared to the alternatives.

C.3. Experiments with web data

Finally, we investigate HCC for both hierarchical clustering and feature extraction on a web dataset. The dataset consists of $15,000$ webpages collected about topics such as politics, finance, sport, art, entertainment, health, technology, environment, cars and films. Similar to 20 newsgroup datasets, we compute the TF-IDF vectors, apply PCA and then obtain the pairwise cosine similarities between the web pages. Table 5 demonstrates the performance of different methods on this dataset. We observe that, consistent with the previous experiments, both HCC and HCC+GMM yield improving the results compared to the baselines. In addition, using HCC to compute intermediate features for GMM (i.e., HCC+GMM) results in better scores than using HCC to produce the final clusters.