A 4-Approximation Algorithm for Min Max Correlation Clustering

Holger Heidrich
TU Dresden

Jannik Irmai
TU Dresden

Bjoern Andres
TU Dresden
ScaDS.AI Dresden/Leipzig

Abstract

We introduce a lower bounding technique for the min max correlation clustering problem and, based on this technique, a combinatorial 4-approximation algorithm for complete graphs. This improves upon the previous best known approximation guarantees of 5, using a linear program formulation (Kalhan et al., 2019), and 40, for a combinatorial algorithm (Davies et al., 2023a). We extend this algorithm by a greedy joining heuristic and show empirically that it improves the state of the art in solution quality and runtime on several benchmark datasets.

1 INTRODUCTION

Correlation clustering refers to the task of clustering elements based on pairwise similarity. The objective is to find a partition of the set of elements such that, preferably, similar elements are in the same cluster and dissimilar elements are in distinct clusters. The number and size of clusters is not predefined but determined by the pairwise similarities. Originally, the correlation clustering problem is defined by Bansal et al. (2004) for a graph where each edge $\{u, v\}$ is labeled either $+$ or $-$, depending on whether u and v are deemed similar or dissimilar. For any partition of the node set, an edge $\{u, v\}$ is said to be in disagreement with the partition if the edge is labeled $-$ and the nodes u and v are in the same cluster of the partition, or if the edge is labeled $+$ and the nodes u and v are in distinct clusters of the partition. Bansal et al. (2004) study the problem of finding a partition that minimizes the number of disagreeing edges. More recently, Puleo and Milenkovic (2016) have introduced the correlation clustering problem with locally bounded errors. They define for each node in the graph the disagreement of that node as the number of edges that are incident to that node and in disagreement with the partition. They study the problem of finding a partition that minimizes a function of the disagreements of the nodes. Among others, they consider ℓ^p norms for $p \geq 1$. In the special case of $p = \infty$, this leads to the problem of finding a partition such that the maximum disagreement over all nodes is minimal. This special case, called *min max correlation clustering*, has attracted attention as a model of a sense of fairness (Davies et al., 2023a). Here, we concentrate on the special case of complete graphs, i.e., every pair of nodes is either a $+$ or a $-$ edge.

As a first contribution, we present a lower bounding technique for the min max correlation clustering problem for complete graphs. This lower bound can be computed efficiently, by a combinatorial algorithm, and does not require solving a linear program (LP). Moreover, this bound is different from that obtained by solving the canonical LP relaxation, i.e. instances exist where either of these bounds is strictly stronger. As a second contribution, we use this lower bounding technique to derive a 4-approximation algorithm for the min max correlation clustering problem for complete graphs. This improves upon the previous best known approximation guarantees of 5, using an LP formulation (Kalhan et al., 2019), and 40, for a combinatorial algorithm (Davies et al., 2023a). We briefly discuss generalizations to non-complete and weighted graphs. However, we do not establish any approximation guarantees for these cases. As a third contribution, we extend the 4-approximation by a local search algorithm that is designed in due consideration of the insights provided by the lower bound. Empirically, we show: The lower bound and the 4-approximation with the local search extension outperform the current state of the art in both solution quality and runtime on a variety of benchmark datasets. In particular, we present lower bounds and approximate solutions for large graphs out of reach of previous methods.

The remainder of the article is organized as follows. In Section 2, we discuss related work. In Section 3,

Proceedings of the 27$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

we state the min max correlation clustering problem formally. In Section 4, we present the lower bounding technique and derive a 4-approximation algorithm. In Section 5, we extend the 4-approximation by an efficient local search heuristic. In Section 6, we examine the bound and the approximation algorithm empirically, on several benchmark datasets, and compare these to the state of the art. In Section 7, we draw conclusions and discuss perspectives for future work. Implementations of all our algorithms and the complete code for reproducing the experiments are available at `https://github.com/JannikIrmai/min-max-correlation-clustering`.

2 RELATED WORK

The correlation clustering problem is introduced originally by Bansal et al. (2004). It is closely related to the *clique partitioning* problem (Grötschel and Wakabayashi, 1989, 1990) and the *graph partition* problem (Chopra and Rao, 1993) which ask for partitions of a graph with positive and negative edge weights that minimize the costs of the edges within clusters and between clusters, respectively. The graph partition problem is also known as the *multicut* problem (Deza et al., 1992), not to be confused with the *multiterminal cut* problem (Dahlhaus et al., 1994) or *multicommodity cut* problem (Leighton and Rao, 1999). The complexity and hardness of approximation of the correlation clustering problem are studied, among others, by Bansal et al. (2004); Demaine et al. (2006); Ailon et al. (2008); Voice et al. (2012); Bachrach et al. (2013); Veldt (2022); Klein et al. (2023). The best known approximation guarantee for the correlation clustering problem in unweighted complete graphs is $1.996 + \epsilon$, due to Cohen-Addad et al. (2022).

Puleo and Milenkovic (2016) propose a generalization of the correlation clustering problem in which the objective is a function of the disagreements of the nodes. This is motivated by the idea of bounding disagreements locally in order to model a sense of fairness and, e.g., penalize partitions in which individual nodes have a disproportionally large disagreement. As a special case, they introduce the min max correlation clustering problem, show that it is NP-hard for complete graphs, and provide a 48-approximation algorithm. In a subsequent study, Charikar et al. (2017) present approximation algorithms for several variants of the correlation clustering problem, including a 7-approximation for the min max objective. This is further improved by Kalhan et al. (2019) who, among other results, present a 5-approximation algorithm for the correlation clustering problem for complete graphs with the ℓ^p objective for all $p \geq 1$, which includes the min max objective ($p = \infty$). Their algorithm is based on rounding the solution of the canonical LP relaxation of the correlation clustering problem. As solving an LP can be computationally expensive, Davies et al. (2023a) propose a method for computing a feasible solution to this specific LP more efficiently. While this solution is not guaranteed to be optimal for the LP, its objective is at most 8 times greater than that of the optimal integral solution for the min max objective. This results in a 40-approximation for the min max correlation clustering problem for complete graphs, which is the first approximation that does not require solving an LP. They show empirically that their algorithm performs well in practical applications and scales to instances previously out of reach. The arguments we use in order to prove the existence and efficiency of a combinatorial 4-approximation algorithm (Theorem 3) are closely related to the non-local charging argument of Davies et al. (2023a). Our arguments are combinatorial while Davies et al. (2023a) use a correlation metric (i.e. a fractional solution to the LP relaxation) to prove their result. To the best of our knowledge, nothing is known about the hardness of approximation for the min max correlation clustering problem.

The more general ℓ_p objective is further studied by Jafarov et al. (2020) who present approximation algorithms for complete weighted graphs with a bounded weight range. Davies et al. (2023b) establish an algorithm for computing a partition that simultaneously approximates all ℓ_p-norm objectives within a constant factor.

Another closely related variant of the correlation problem is studied by Ahmadi et al. (2019). They consider the objective of minimizing the maximum over the disagreements within all clusters and develop a $\mathcal{O}(\log(n))$-approximation algorithm. An improved approximation guarantee of $2+\epsilon$ is given by Kalhan et al. (2019) who also establish inapproximability within a factor better than 2 assuming the unique games conjecture (UGC). Hence, their approximation guarantee is the best possible if UGC holds.

3 MIN MAX CORRELATION CLUSTERING FOR COMPLETE GRAPHS

To begin with, we state the min max correlation clustering problem for complete graphs formally, using elementary notation. For any set V, let P_V denote the set of all partitions of V. For any $\Pi \in P_V$ and any $u \in V$, let $[u]_\Pi$ denote the unique $U \in \Pi$ such that $u \in U$. For any graph $G = (V, E)$ and any $v \in V$, let us refer to $N_v := \{v\} \cup \{w \in V \mid \{v, w\} \in E\}$ as the neighborhood of v in G, including v itself.

Definition 1. For any graph $G = (V, E)$, the instance

of the *min max correlation clustering problem* with respect to G has the form

$$\min_{\Pi \in P_V} \underbrace{\max_{v \in V} |[v]_\Pi \triangle N_v|}_{=:\ \varphi(\Pi)} \tag{MMCC}$$

where $A \triangle B = (A \setminus B) \cup (B \setminus A)$ is the symmetric difference of sets. Here, $|[v]_\Pi \triangle N_v|$ is the disagreement of node v with partition Π and $\varphi(\Pi)$ is the maximum disagreement of Π.

MMCC can be understood as a problem with respect to an edge signed complete graph. More specifically, one can identify the graph (V, E) in Definition 1 with the edge signed complete graph $(V, E^+ \cup E^-)$ such that $E^+ = E$.

4 COMBINATORIAL LOWER BOUND AND 4-APPROXIMATION

In this section, we present a lower bounding technique for the min max correlation clustering problem for complete graphs. From this technique, we derive a 4-approximation algorithm.

We begin by establishing properties of partitions and their maximal disagreement. The following lemma states that two nodes whose neighborhoods are particularly similar (respectively dissimilar) must be in the same cluster (respectively different clusters) in all partitions whose maximal disagreement is small enough.

Lemma 1. *Let $G = (V, E)$ be a graph. For every partition $\Pi \in P_V$ and every $u, v \in V$:*

(a) If $|N_u \cap N_v| > 2\varphi(\Pi)$ then $[u]_\Pi = [v]_\Pi$.
(b) If $|N_u \triangle N_v| > 2\varphi(\Pi)$ then $[u]_\Pi \neq [v]_\Pi$.

Proof. To prove (a), we show that $[u]_\Pi \neq [v]_\Pi$ implies $|N_u \cap N_v| \leq 2\varphi(\Pi)$. By definition, $\varphi(\Pi) \geq |N_u \triangle [u]_\Pi| \geq |N_u \setminus [u]_\Pi| \geq |(N_u \cap N_v) \setminus [u]_\Pi|$. Analogously, $\varphi(\Pi) \geq |(N_u \cap N_v) \setminus [v]_\Pi|$. The assumption $[u]_\Pi \neq [v]_\Pi$ and the fact that clusters of Π are disjoint implies the desired $|N_u \cap N_v| \leq |(N_u \cap N_w) \setminus [u]_\Pi| + |(N_u \cap N_w) \setminus [v]_\Pi| \leq 2\varphi(\Pi)$.

The second statement can be derived from Proposition 4.1 of Davies et al. (2023a). For completeness, we provide a self-contained prove below. We show that $[u]_\Pi = [v]_\Pi$ implies $|N_u \triangle N_v| \leq 2\varphi(\Pi)$. Let $C = [u]_\Pi = [v]_\Pi$. By definition:

$$\begin{aligned}
|N_u \triangle N_v| &= |N_u \setminus N_v| + |N_v \setminus N_u| \\
&= |(N_u \setminus N_v) \cap C| + |(N_u \setminus N_v) \setminus C| \\
&\quad + |(N_v \setminus N_u) \cap C| + |(N_v \setminus N_u) \setminus C| \\
&\leq |C \setminus N_v| + |N_u \setminus C| + |C \setminus N_u| + |N_v \setminus C| \\
&= |N_u \triangle C| + |N_v \triangle C| \\
&= |N_u \triangle [u]_\Pi| + |N_v \triangle [v]_\Pi| \leq 2\varphi(\Pi) \ . \qquad \square
\end{aligned}$$

If there exists a partition Π with a given maximal disagreement, Lemma 1 can imply that certain nodes are in the same cluster and certain nodes are in distinct clusters. These constraints are captured in the following definition.

Definition 2. Let $G = (V, E)$ be a graph and let $d \in \mathbb{N}$. Let $G_d = (V, E_d)$ with $E_d = \{\{u, v\} \in \binom{V}{2} \mid |N_u \cap N_v| > 2d\}$ be the graph with those pairs of nodes of G as edges whose neighborhoods intersect in more than $2d$ nodes. Let $\Pi_d \in P_V$ be the partition of V into the maximal connected components of G_d. For a cluster $C \in \Pi_d$, let $U_C^d = \{v \in V \mid |N_w \triangle N_u| \leq 2d \ \forall w \in [v]_{\Pi_d} \ \forall u \in C\}$.

Note that the clusters of Π_d are precisely the sets of nodes that, by Lemma 1 (a), must be in the same cluster of any partition Π with maximum disagreement of $\varphi(\Pi) = d$. For all clusters $C \in \Pi_d$, the set $V \setminus U_C^d$ is precisely the set of nodes that, by Lemma 1 (b), cannot be in the cluster containing C of any partition Π with $\varphi(\Pi) = d$.

Theorem 1. *Let $G = (V, E)$ be a graph. The smallest $d \in \mathbb{N}$ with $C \subseteq U_C^d$ for all $C \in \Pi_d$ and*

$$\max_{v \in V} \left| N_v \setminus U^d_{[v]_{\Pi_d}} \right| + \left| [v]_{\Pi_d} \setminus N_v \right| \quad \leq \quad d \ , \tag{1}$$

is a lower bound for the min max correlation clustering problem with respect to graph G. We call this value the combinatorial lower bound *and denote it by* $\mathrm{CLB}(G)$.

Proof. Let $\Pi \in P_V$ be a partition with maximum disagreement $d = \varphi(\Pi)$. By Lemma 1 and Definition 2, the clusters of Π are unions of clusters of Π_d that are contained in the sets U_C^d for $C \in \Pi_d$, i.e. $[v]_{\Pi_d} \subseteq [v]_\Pi \subseteq U^d_{[v]_{\Pi_d}}$ for all $v \in V$. With this, the disagreement of any node $v \in V$ with Π can be bounded by

$$\begin{aligned}
|N_v \triangle [v]_\Pi| &= |N_v \setminus [v]_\Pi| + |[v]_\Pi \setminus N_v| \\
&\geq |N_v \setminus U^d_{[v]_{\Pi_d}}| + |[v]_{\Pi_d} \setminus N_v| \ . \qquad (2)
\end{aligned}$$

Now, let $d = \mathrm{CLB}(G)$ be the smallest $d \in \mathbb{N}$ that satisfies the conditions in the statement of the theorem. Suppose there exists a partition $\Pi \in P_V$ with maximum disagreement $\varphi(\Pi) = d' < d$. By definition of d,

there exists $v \in V$ with $|N_v \setminus U^d_{[v]_{\Pi_{d'}}}| + |[v]_{\Pi_{d'}} \setminus N_v| > d'$. By (2), this implies $\varphi(\Pi) \geq |N_v \triangle [v]_\Pi| > d'$ in contradiction to the assumption. □

Theorem 2. *Let $G = (V, E)$ be a graph. The combinatorial lower bound $\mathrm{CLB}(G)$ can be computed in time $\mathcal{O}(n^2 \log_2(\delta) + n\delta^2)$ where n is the number of nodes in G, and δ is the maximum degree of all nodes in G.*

Proof. To begin with, we show that for all pairs of nodes the size of the intersections of their neighborhoods can be computed in time $\mathcal{O}(n^2 + n\delta^2)$. This can be done by the following algorithm: For all $u, v \in V$, let $I_{\{u,v\}} = 0$. For every $w \in V$ and every $\{u, v\} \in N_w$, increase $I_{\{u,v\}}$ by one to account for the fact that w is in $N_u \cap N_v$. Then, $I_{\{u,v\}} = |N_u \cap N_v|$, and clearly, this can be done in time $\mathcal{O}(n^2 + n\delta^2)$.

Next, we show that for a given $d \in \mathbb{N}$ it can be decided in time $\mathcal{O}(n^2)$ whether (1) holds. The partition Π_d consists of the connected components of the graph whose edges are all pairs of nodes $\{u, v\} \in \binom{V}{2}$ with $I_{\{u,v\}} > 2d$. These connected components can be computed, for example, by breadth first search in time $\mathcal{O}(n^2)$. For every node $v \in V$, the value $|[v]_{\Pi_d} \setminus N_v|$ can clearly be computed in time $\mathcal{O}(n)$. It remains to compute the value $|N_v \setminus U^d_{[v]_{\Pi_d}}|$ for every $v \in V$. To this end, we construct an auxiliary graph $G' = (V', E')$ whose nodes are the clusters in Π_d, and two clusters $C, C' \in \Pi_d$ are connected by an edge if and only if $|N_u \triangle N_{u'}| \leq 2d$ for all $u \in C$ and $u' \in C'$. Clearly, this graph can be computed in time $\mathcal{O}(n^2)$. By definition of $U^d_{[v]_{\Pi_d}}$:

$$|N_v \setminus U^d_{[v]_{\Pi_d}}| = |\{u \in N_v \mid \{[u]_{\Pi_d}, [v]_{\Pi_d}\} \notin E'\}| \ , \quad (3)$$

which is the number of neighbors u of v whose cluster is not connected to the cluster of v in G'. By representing G' with an adjacency matrix, (3) can be evaluated in time $\mathcal{O}(n)$ for each node.

Lastly, the minimal d that satisfies (1) can be computed by a bisection algorithm. Clearly, $0 \leq \mathrm{CLB}(G) \leq \delta$. Thus, $\mathcal{O}(\log_2(\delta))$ bisection steps are sufficient. □

This worst case time complexity of $\mathcal{O}(n^2 \log_2(\delta) + n\delta^2) \subseteq \mathcal{O}(n^3)$ for computing the combinatorial lower bound is smaller than that of solving the LP relaxation with $\mathcal{O}(n^2)$ variables and $\mathcal{O}(n^3)$ constraints.

Below, Lemma 2 and Corollary 1 state that for every partition whose maximal disagreement is less than a quarter of the size of the neighborhood of a given node, the cluster of that node in the partition is uniquely determined. Afterward, Theorem 3 states that this yields a 4-approximation algorithm for min max correlation clustering.

Lemma 2. *Let $G = (V, E)$ be a graph and let $d = \mathrm{CLB}(G)$ be the combinatorial lower bound according to Theorem 1. For any node $v \in V$ with $|N_v| > 4d$:*

$$[v]_{\Pi_d} = U^d_{[v]_{\Pi_d}} = \{u \in V \mid |N_u \cap N_v| > |N_v|/2\} \ .$$

Proof. By definition of the combinatorial lower bound in Theorem 1: $[v]_{\Pi_d} \subseteq U^d_{[v]_{\Pi_d}}$. To prove the claim, it remains to show $U^d_{[v]_{\Pi_d}} \subseteq \{u \in V \mid |N_u \cap N_v| > |N_v|/2\} \subseteq [v]_{\Pi_d}$.

By the assumption that $|N_v| > 4d$, we have $|N_v|/2 > 2d$, which implies that every $u \in V$ that satisfies $|N_u \cap N_v| > |N_v|/2$ is in $[v]_{\Pi_d}$, by definition of Π_d. This yields the second inclusion.

For $u \in V$ with $|N_u \cap N_v| \leq |N_v|/2$, we have

$$\begin{aligned} |N_u \triangle N_v| &= |N_u \cup N_v| - |N_u \cap N_v| \\ &\geq |N_v| - |N_v|/2 = |N_v|/2 > 2d \ , \end{aligned}$$

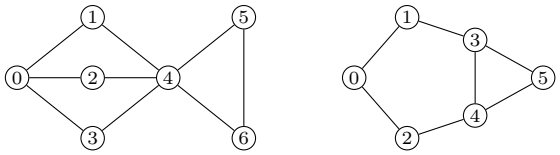
and Lemma 1 (b) implies $u \notin U^d_{[v]_{\Pi_d}}$. This yields the first inclusion. □

Corollary 1. *Let $G = (V, E)$ be a graph. For every partition $\Pi \in P_V$ and every node $v \in V$ with $|N_v| > 4\varphi(\Pi)$: $[v]_\Pi = \{u \in V \mid |N_u \cap N_v| > |N_v|/2\}$.*

Theorem 3. *There exists a 4-approximation algorithm for the min max correlation clustering problem. The 4-approximation can be computed in time $\mathcal{O}(n^2 + n\delta^2)$ where n is the number of nodes and δ is the largest degree.*

Proof. The algorithm starts with the partition into singleton clusters. Iteratively, a node v with largest disagreement with respect to the current partition is selected. If there exists a partition with maximum disagreement strictly less than $|N_v|/4$, then the cluster of v is uniquely determined by $C = \{u \in V \mid |N_u \cap N_v| > |N_v|/2\}$, according to Corollary 1. If the cluster C is in conflict with any previously computed cluster (i.e. it contains nodes that have been assigned to a different cluster in an earlier iteration) or it contains a node whose disagreement with respect to that cluster is greater than $|N_v|/4$, then there cannot exists a partition with disagreement less than $|N_v|/4$. This implies that the current partition is a 4-approximation, and the algorithm terminates. Otherwise, C is added to the current partition and the algorithm continues.

As in Theorem 2, the size of the intersections of neighborhoods of all pairs of nodes can be computed in $\mathcal{O}(n^2 + n\delta^2)$. The algorithm described above terminates after at most $\mathcal{O}(n)$ iterations. In each iteration,



(a) (b)

Figure 1: For the graph depicted in a, the combinatorial bound (3) is stronger than the LP bound ($\frac{7}{4}$). For the graph depicted in b, the LP bound ($\frac{5}{4}$) is stronger than the combinatorial bound (1). For details, see Examples 1 and 2.

the cluster C can be computed in $\mathcal{O}(n)$. The disagreement of a node with a given cluster can be computed in $\mathcal{O}(\delta)$. As each node is assigned to a cluster at most once, the disagreement needs to be computed for at most $\mathcal{O}(n)$ nodes. Together, this implies the claimed runtime. □

4.1 Relation to LP bound

The combinatorial lower bound from Theorem 1 differs from the bound obtained by solving the canonical LP relaxation of the min max correlation clustering problem. In this section, we show by Examples 1 and 2 that neither bound is stronger. For an in-depth discussion of the LP bound, we refer to Kalhan et al. (2019). An efficient algorithm for computing the combinatorial lower bound is detailed in the proof of Theorem 2.

Example 1. Consider the graph $G = (V, E)$ depicted in Figure 1a. Suppose there exists a partition Π with maximum disagreement $\varphi(\Pi) \leq 2$. By Lemma 1 (a), all pairs of nodes whose neighborhoods share at least five nodes must be in the same cluster in Π. In the given graph, the neighborhoods of all pairs of nodes share at most 3 nodes, i.e. $\Pi_2 = \{\{v\} \mid v \in V\}$ consists of singleton clusters. By Lemma 1 (b), all pairs of nodes whose neighborhoods have a symmetric difference containing five or more nodes must be in distinct clusters of Π. For the given graph, this implies that 4 must be in a cluster different from that of 1, 2, and 3, i.e $U^2_{\{4\}} = \{0, 4, 5, 6\}$. We can bound the disagreement of 4 with respect to partition Π by

$$\begin{aligned} |N_4 \triangle [4]_\Pi| &= |N_4 \setminus [4]_\Pi| + |[4]_\Pi \setminus N_4| \\ &\geq |N_4 \setminus \{0, 4, 5, 6\}| + |\{4\} \setminus N_4| = 3 \end{aligned}$$

which is strictly greater than 2, in contradiction to the assumption. Therefore, every partition has maximum disagreement at least 3. In fact, this is the optimal value assumed e.g. by the partition $\Pi = \{\{0, 1\}, \{2\}, \{3\}, \{4, 5, 6\}\}$.

In contrast, the LP bound is $\frac{7}{4}$, which is strictly less than the combinatorial bound of 3. The LP bound is assumed by the solution $x_{\{4,5\}} = 0$, $x_{\{0,1\}} = x_{\{0,2\}} = x_{\{0,3\}} = x_{\{1,4\}} = x_{\{1,5\}} = x_{\{2,4\}} = x_{\{2,5\}} = x_{\{3,4\}} = x_{\{3,5\}} = \frac{1}{2}$, $x_{\{1,6\}} = x_{\{2,6\}} = x_{\{3,6\}} = \frac{3}{4}$, and $x_e = 1$ for all other edges e. The maximum disagreement of $\frac{7}{4}$ is obtained at node 4.

Example 2. Consider the graph $G = (V, E)$ depicted in Figure 1b. Suppose there exists a partition Π with disagreement $\varphi(\Pi) = 0$. Then, by Lemma 1 (a), all pairs of nodes whose neighborhoods intersect in at least one node must be in the same cluster in Π. For the given graph, this implies that all nodes are in the same cluster, i.e. $\Pi_0 = \{V\}$. However, this partition has disagreement 3, in contrast to the assumption. Thus, there cannot exist a partition of with disagreement 0.

Next, suppose there exists a partition Π with disagreement $\varphi(\Pi) = 1$. By Lemma 1 (a), all pairs of nodes whose neighborhoods share at least three nodes must be in the same cluster in Π. This implies that the nodes 3, 4, and 5 are in the same cluster, i.e. $\Pi_1 = \{\{0\}, \{1\}, \{2\}, \{3, 4, 5\}\}$. Lemma 1 (b) implies that all pairs of nodes whose neighborhoods have a symmetric difference containing more than three nodes must be in distinct clusters of Π. For the given graph, we get $U^1_{\{0\}} = \{0, 1, 2\}$, $U^1_{\{1\}} = \{0, 1\}$, $U^1_{\{2\}} = \{0, 2\}$ and $U^1_{\{3,4,5\}} = \{3, 4, 5\}$. For all nodes, the bound (1) is less than or equal to 1. Therefore, the combinatorial lower bound of the given graph is 1.

In contrast, the LP bound is $\frac{5}{4}$ which is strictly greater than the combinatorial lower bound of 1. The LP bound is assumed by the solution $x_{\{0,1\}} = x_{\{0,2\}} = \frac{1}{2}$, $x_{\{1,3\}} = x_{\{2,4\}} = \frac{3}{4}$, $x_{\{3,4\}} = x_{\{3,5\}} = x_{\{4,5\}} = \frac{1}{4}$, and $x_e = 1$ for all other edges. The maximum disagreement of $\frac{5}{4}$ is obtained at nodes 3 and 4.

4.2 Non-complete and weighted graphs

The combinatorial lower bound can be adapted to the case of non-complete graphs. However, Lemma 1 (b) needs to be adjusted: If the number of nodes that are in the negative neighborhood of u and in the positive neighborhood of v, or vice versa, is greater than $2\varphi(\Pi)$, then u and v must be in distinct clusters. This leads to a different definition of U^d in Definition 2 which in turn leads to a different definition of the combinatorial lower bound in Theorem 1. For this adapted version, Lemma 2, Corollary 1, and Theorem 3 no longer hold. Whether this bounding technique for non-complete graphs can be used to derive an approximation algorithm is an open problem.

The bounding technique can even be adapted to the weighted variant of the min max correlation clustering problem. Again, the main difference lies in Lemma 1. For two nodes u, v in distinct clusters of a partition,

each node w in the positive neighborhood of both u and v contributes to either the disagreement of u or v. Therefore, w contributes a disagreement of at least $\min(\theta_{uw}, \theta_{vw})$ where θ_e is the weight of edge e. As a consequence, the maximum disagreement of u and v is at least $\frac{1}{2}\sum_{w \in N_u \cap N_v} \min(\theta_{uw}, \theta_{vw})$. This yields an analogue to Lemma 1 (a) for the weighted case: If $\sum_{w \in N_u \cap N_v} \min(\theta_{uw}, \theta_{vw}) > 2\varphi(\Pi)$, then $[u]_\Pi = [v]_\Pi$. Similarly, an analogue to Lemma 1 (b) can be derived.

5 GREEDY JOINING ALGORITHM

The 4-approximation due to Theorem 3 only adds a cluster if it decreases the maximum disagreement of the nodes contained in it by at least a factor of four. Nodes with smaller degree may remain in singleton clusters. For example, all graphs from practical applications we consider in Section 6 are such that the largest node degree is less than four times the combinatorial lower bound. For these graphs, the partition into singleton clusters is a 4-approximation. In practical applications like these, it is desirable not only to find a feasible solution that satisfies the approximation guarantee but also to further improve this feasible solution by local search.

In this section, we introduce an algorithm that seeks to iteratively improve a given partition by greedily joining clusters. While we do not establish any improved approximation guarantee, the feasible solutions found by this algorithm improve upon the empirical state of the art for the applications we consider in Section 6.

The algorithm is remarkable simple: In each iteration, a node w with largest disagreement with respect to the current partition Π is chosen. Every neighbor v of w that is not in the same cluster as w in Π contributes to the disagreement for w. The disagreement of w can potentially be decreased by joining the cluster of w with the cluster of v. If no such neighbor v exists, the disagreement of w cannot be improved by joining two clusters, and the algorithm terminates. Otherwise, we choose one such v that fits well to w according to Lemma 1, i.e. such that the intersection of neighborhoods $|N_w \cap N_v|$ is large and the symmetric difference of $|N_w \triangle N_v|$ neighborhoods is small. This algorithm is detailed in Algorithm 1.

In several places of Algorithm 1, ties can occur: In Line 3, multiple nodes with largest agreement can exist. In Line 5, multiple neighbors v of w can have the same value $|N_w \cap N_v| - |N_w \triangle N_v|$. We break these ties by considering node degree, as the secondary ordering criterion, and node index, as the tertiary ordering criterion. In Line 3, we select among all nodes with largest disagreement a node with largest/smallest degree and refer to these two options as Design Choice 1 (DC1). In Line 5, we sort neighbors v of w with the same value $|N_w \cap N_v| - |N_w \triangle N_v|$ by increasing/decreasing degree (DC2). We investigate two more choices in the design of Algorithm 1: Instead of sorting the neighbors of w by $|N_w \cap N_v| - |N_w \triangle N_v|$, we consider sorting these only by $|N_w \cap N_v|$ or only by $-|N_w \triangle N_v|$ (DC3). Finally, we consider making Line 10 stricter by discarding a join if there exists $u \in C$ such that the disagreement of u with Π is strictly less than that of w, but the disagreement of u with C is equal to the disagreement of w with Π (DC4).

Algorithm 1: Greedy Joining

```
Data: Graph G = (V, E)
Result: Partition Π of V
1  Let Π be the partition computed by the
     4-approximation algorithm (Theorem 3)
2  while True do
3      w ∈ argmax_{v∈V} |N_v △ [v]_Π|
4      made_join = False
5      for v ∈ N_w sorted by |N_w ∩ N_v| − |N_w △ N_v| in
         descending order do
6          if [v]_Π = [w]_Π then
7              continue // v is already in the
                   cluster of w.
8          C = [v]_Π ∪ [w]_Π // join the clusters of v
               and w
9          d = max_{u∈C} |N_u △ C|  // maximal
               disagreement of all nodes in C
10         if d > |N_v △ [v]_Π| then
11             continue // the maximal disagreement
                   in C is greater than the current
                   maximal disagreement
12         Π := Π \ {[v]_Π, [w]_Π} ∪ {C} // replace the
               clusters of v and w by C.
13         made_join = True
14         break
15     if not made_join then
16         return Π
```

We denote by $\mathcal{A}$ the variant of Algorithm 1 where ties are resolved in favor of largest degree, where neighborhoods are sorted by $|N_w \cap N_v| - |N_w \triangle N_v|$ and with the stricter version of Line 10. We denote by $\mathcal{A}^*$ the algorithm that executes Algorithm 1 for all 24 combinations of DC1-4 and outputs the best solution.

Lemma 3. *Algorithm 1 has worst case time complexity of $\mathcal{O}(n^2\delta^2)$.*

Proof. The algorithm terminates after $\mathcal{O}(n)$ joins. Each join can be computed in time $\mathcal{O}(n\delta^2)$: There are $|N_w| \leq \delta$ candidate clusters that can be joined with the cluster of w (Line 5). For each candidate cluster, the maximum disagreement with respect to the joint cluster (Line 9) is computed in time $\mathcal{O}(n\delta)$. □

Despite its high worst case time complexity, Algorithm 1 is efficient in practice (see Section 6) for the

following reason: In real world instances there are typically only a few nodes with large degree and thus large disagreement. Therefore, Algorithm 1 only operates on a few nodes and their neighborhoods. In particular, it is not necessary to compute the size of the intersections of the neighborhoods of all pairs of nodes. Note, however, that these computations are required in order to compute the combinatorial lower bound as well as in the algorithm by Davies et al. (2023a).

6 EXPERIMENTS

We follow Davies et al. (2023a) and evaluate the combinatorial lower bound and the greedy joining algorithm on social network graphs as well as synthetic graphs. We compare the results to those of Davies et al. (2023a). All experiments were performed on a Lenovo X1 Carbon laptop equipped with an Intel Core i7-10510U CPU @ 1.80GHz and 16 GB LPDDR3 RAM.

6.1 Datasets

Social network graphs The *ego-Facebook* dataset (McAuley and Leskovec, 2012) contains ten graphs that represent circles of friends from the social network Facebook. The graphs contain 52 to 1,034 nodes and 146 to 30,025 edges. The *feather-lastfm-social* dataset (Rozemberczki and Sarkar, 2020) is a social network of 7,624 LastFM users with 27,806 edges. The *ca-HepPh* and *ca-HepTh* datasets (Leskovec et al., 2007) are collaboration networks of authors in the field of high energy physics. They contain 12,008 and 9,977 nodes and 118,489 and 25,998 edges, respectively. Lastly, the *com-Youtube* dataset is a social network of 1,134,890 Youtube users with 2,987,624 edges. These datasets are available online[1].

Synthetic graphs In order to compare algorithms for the min max correlation clustering problem also in a more controlled setting, Davies et al. (2023a) synthesize instances of the problem as follows: Beginning with 10 distinct cliques containing 10 nodes each, a given number of pairs of nodes are selected at random. A selected pair is added to the edge set in case the nodes are not in a clique, and removed from the edge set, otherwise. I.e., edge are flipped from attractive to repulsive, or vice versa. The number of disagreements increases with the number of flips. For each number $f \in \{0, 50, 100, \dots, 1000\}$ of flips, we construct 10 instances randomly, in this way.

Results In Tables 1 and 2, we report empirical results for the social network graphs. In Columns 1-4, we report the name of the dataset, the number of nodes, number of edges and largest degree of the graph. In the columns CLB and LP, we report the combinatorial lower bound and the LP bound. In the columns $\mathcal{A}$ and $\mathcal{A}^*$, we report the maximum disagreements of the partitions found by algorithms $\mathcal{A}$ and $\mathcal{A}^*$, as described in Section 5. In the columns DMN and KMZ, we report the maximum disagreements of the partitions found by the algorithm of Davies et al. (2023a) and Kalhan et al. (2019). Note that the rounding algorithm utilized by the DMN and KMZ algorithm is parameterized by two parameters. We denote by DMN and KMZ the results that are obtained by using the same parameters for all problem instances and denote by DMN* and KMZ* the results that are obtained by searching for the best parameters for each problem instances individually (for details, see Davies et al. (2023a)). In the last columns, we report the times for computing the combinatorial lower bound, for computing the LP bound, and for running the algorithms $\mathcal{A}$, $\mathcal{A}^*$, and DMN. As our algorithms are implemented in c++ and the DMN algorithm of Davies et al. (2023a) is originally implemented in python, we contribute, in addition, a performance optimized c++ implementation of DMN that exploits sparsity of the input graph. We report the runtime of the original python implementation (t_{DMN}), as well as that of our c++ implementation (t^{++}_{DMN}). We do not report the runtime of the KMZ algorithm explicitly as it is dominated by solving the LP (t_{LP}). The LP is solved with Gurobi Optimization, LLC (2023), using the barrier method.

In Figure 2, we report bounds, maximum disagreements and runtimes for the synthetic instances of the min max correlation clustering problem. In this figure, thick lines indicate the median across the 10 random instances, shaded areas indicate the second and third quartile, and dashed lines indicate the 0.1 and 0.9-quantile.

Analysis As can be seen from Tables 1 and 2, the partitions computed by $\mathcal{A}$ have a lower maximum disagreement than those computed by DMN and KMZ, across all instances. For the Facebook ego-network with ID 3437, the disagreement of the partition computed by $\mathcal{A}$ is 58, compared to 107 computed by DMN. The runtime of $\mathcal{A}$ is approximately one order of magnitude shorter than that of DMN. This is due to the fact that in DMN, the intersections of neighborhoods of all pairs of nodes need to be computed, while in $\mathcal{A}$, these intersections only for the node with the largest disagreement and its neighbors need to be computed (Line 5). The disagreement of partitions computed by $\mathcal{A}^*$ is often strictly less than that computed by $\mathcal{A}$. The greatest relative improvement of $\mathcal{A}^*$ over $\mathcal{A}$ can

[1] https://snap.stanford.edu/data/

ID	$\lvert V\rvert$	$\lvert E\rvert$	δ	CLB	LP	$\mathcal{A}$	$\mathcal{A}^*$	DMN	DMN*	KMZ	KMZ*	t_{CLB}	t_{LP}	$t_{\mathcal{A}}$	$t_{\mathcal{A}^*}$	t_{DMN}	t^{++}_{DMN}
0	333	2519	77	32	-	46	44	49	49	-	-	5.32	-	0.36	10.4	159	6.58
107	1034	26749	253	95	-	123	122	152	134	-	-	112.38	-	3.30	84.1	1,294	230.05
348	224	3192	99	39	39.13	61	50	72	71	89	69	2.12	$1.5 \cdot 10^6$	0.53	12.4	117	13.04
414	150	1693	57	18	19.66	27	27	34	31	38	28	2.06	$2.0 \cdot 10^5$	0.54	14.5	72	3.27
686	168	1656	77	31	30.48	45	43	47	43	69	47	1.06	$4.1 \cdot 10^5$	0.21	6.7	84	4.17
698	61	270	29	11	10.64	16	16	20	18	18	17	0.20	$1.6 \cdot 10^3$	0.06	1.5	13	0.27
1684	786	14024	136	52	-	80	78	93	93	-	-	65.76	-	2.11	50.0	814	67.03
1912	747	30025	293	118	-	166	163	220	187	-	-	41.66	-	4.63	116.4	877	272.84
3437	534	4813	107	49	-	58	57	107	77	-	-	7.79	-	0.89	14.9	405	16.24
3980	52	146	18	8	7.34	11	11	12	12	13	13	0.10	$6.7 \cdot 10^2$	0.07	1.2	7	0.20

Table 1: This table summarizes the results for the *ego-Facebook* graphs. The runtime is reported in milliseconds. The LP bound and KMZ objective is only reported for the five smallest instances as solving the LP for the graph 348 already takes approximately 25 minutes.

Name	$\lvert V\rvert$	$\lvert E\rvert$	δ	CLB	$\mathcal{A}$	$\mathcal{A}^*$	DMN	DMN*	t_{CLB}	$t_{\mathcal{A}}$	t_{DMN}	t^{++}_{DMN}
lastfm	7,624	27,806	216	106	116	116	216	160	1,724	8.74	$6.4 \cdot 10^4$	281
HepPh	12,006	118,489	491	208	251	250	333	267	8,023	27.91	$1.2 \cdot 10^5$	2,410
HepTh	9,875	25,973	65	34	42	42	65	58	3,611	8.52	$1.7 \cdot 10^5$	212
Youtube	1,134,890	2,987,624	28,754	-	15,486	14,796	-	-	-	$4.1 \cdot 10^4$	-	-

Table 2: This table summarizes the results for the *feather-lastfm-social*, *ca-HepPh*, *ca-HepTh*, and *com-Youtube* graphs. The runtime is reported in milliseconds. For the largest graph, the combinatorial lower bound and DMN are not computed as this would exceed time and memory constraints.

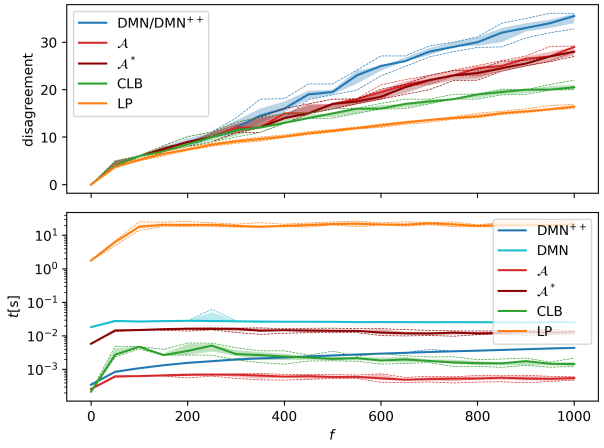


Figure 2: Depicted above are the maximum disagreements of the partitions computed by the DMN algorithm and our algorithms $\mathcal{A}$, $\mathcal{A}^*$ as well as the lower bounds according to the combinatorial lower bound (CLB) and the LP bound. Depicted below are the runtimes in seconds of all algorithms and both bounding techniques. f is the number of random flips in a graph with 100 nodes (10 cliques of size 10).

be observed on the Facebook ego-network with ID 348. Here, the disagreement of 50 is achieved by the greedy joining algorithm in which neighborhoods are sorted by $-|N_v \triangle N_w|$. Similarly, the disagreement of partitions computed by DMN* and KMZ* are often strictly less than that computed by DMN and KMZ. The disagreements achieved by $\mathcal{A}^*$ are less than that of DMN* and KMZ* on all instances except one where there is a tie.

The combinatorial lower bound (CLB) and the LP bound are similar. However, the combinatorial lower bound can be computed many orders of magnitude faster than the LP bound. For the Facebook ego-network with ID 348, solving the LP requires approximately 25 minutes while computing the combinatorial lower bound takes approximately 2 milliseconds. The fact that the combinatorial lower bound can be computed so much more efficiently allows us to compute the first non-trivial lower bounds for larger instances. This includes the five large instances of the *ego-Facebook* dataset for which we cannot report an LP bound in Table 1, and it includes the even larger graphs in Table 2. Only for the *com-Youtube* graph in Table 2 with more than one million nodes have we found the computation the combinatorial lower bound to be impractical. Across all other instances, the greatest relative gap between the maximum disagreement of the partition computed by $\mathcal{A}$ and the lower bound is 1.56 (Facebook ego-network with ID 348).

For synthetic graphs (Figure 2), the results are similar: For small numbers of flips, both DMN and $\mathcal{A}$ compute the optimal partition, which can be seen from the fact that there is no gap between the optimal solution and the lower bound. For larger numbers of flips, the gap between the lower bounds and the maximum disagreement of the computed partitions increases. Notably, the gap of the partitions computed by $\mathcal{A}$ is approximately half of the gap of the partitions computed by DMN. The combinatorial lower bound (CLB) is slightly stronger than the LP bound. The runtime of the greedy joining algorithm $\mathcal{A}$ is about one order of magnitude shorter than that of DMN. The difference in runtime is even greater between CLB and the LP bound. Similar to the social network graphs, the greatest relative gap we observe between CLB and the maximum disagreement of the partition computed by $\mathcal{A}$ is approximately 1.5. In fact, we have not found instances where the relative gap between CLB and the maximum disagreement of the partition computed by $\mathcal{A}$ is greater than 2.

7 CONCLUSION

We have introduced a combinatorial lower bounding technique for the min max correlation clustering problem for complete graphs. There are instances where this bound is stronger than the canonical LP bound, and vice versa. This motivates future work to combine these bounds. To this end, the constraints from Lemma 1 can be expressed in the form of quadratic inequalities. However, we have not observed improvements over the LP bound when adding linear relaxations of these quadratic inequalities to the LP (results not shown). From the combinatorial lower bound, we have derived a 4-approximation that we have extended by a greedy local search heuristic. On all instances we have considered in the experiments for this article, the greedy joining algorithm yields a 2-approximation. Whether the greedy joining algorithm is indeed a 2-approximation algorithm is an open problem. We have discussed briefly generalizations of the combinatorial lower bound to non-complete and weighted graphs. Whether approximation guarantees can be derived for these cases, or other objectives than the min max objective discussed in this article, remains open.

References

Saba Ahmadi, Samir Khuller, and Barna Saha. Min-max correlation clustering via multicut. In *International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, 2019. doi: 10.1007/978-3-030-17953-3_2.

Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM*, 55(5):1–27, 2008.

Yoram Bachrach, Pushmeet Kohli, Vladimir Kolmogorov, and Morteza Zadimoghaddam. Optimal coalition structure generation in cooperative graph games. In *AAAI*, 2013. doi: 10.1609/aaai.v27i1.8653.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1–3): 89–113, 2004. doi: 10.1023/B:MACH.0000033116.57574.95.

Moses Charikar, Neha Gupta, and Roy Schwartz. Local guarantees in graph cuts and clustering. In *International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, 2017. doi: 10.1007/978-3-319-59250-3_12.

Sunil Chopra and Mendu R. Rao. The partition problem. *Mathematical Programming*, 59(1–3):87–115, 1993. doi: 10.1007/BF01581239.

Vincent Cohen-Addad, Euiwoong Lee, and Alantha Newman. Correlation clustering with Sherali-Adams. In *Annual Symposium on Foundations of Computer Science (FOCS)*, 2022. doi: 10.1109/FOCS54457.2022.00068.

Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, Paul D. Seymour, and Mihalis Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23(4):864–894, 1994. doi: 10.1137/S0097539792225297.

Sami Davies, Benjamin Moseley, and Heather Newman. Fast combinatorial algorithms for min max correlation clustering. In *ICML*, 2023a. URL https://proceedings.mlr.press/v202/davies23a.html.

Sami Davies, Benjamin Moseley, and Heather Newman. One partition approximating all ℓ_p-norm objectives in correlation clustering. *arXiv preprint arXiv:2308.01534*, 2023b.

Erik D. Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361 (2–3):172–187, 2006. doi: 10.1016/j.tcs.2006.05.008.

Michel Deza, Martin Grötschel, and Monique Laurent. Clique-web facets for multicut polytopes. *Mathematics of Operations Research*, 17(4):981–1000, 1992. doi: 10.1287/moor.17.4.981.

Martin Grötschel and Yoshiko Wakabayashi. A cutting plane algorithm for a clustering problem. *Mathematical Programming*, 45(1):59–96, 1989. doi: 10.1007/BF01589097.

Martin Grötschel and Yoshiko Wakabayashi. Facets of the clique partitioning polytope. *Mathemati-*

cal Programming, 47:367–387, 1990. doi: 10.1007/BF01580870.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023. URL https://www.gurobi.com.

Jafar Jafarov, Sanchit Kalhan, Konstantin Makarychev, and Yury Makarychev. Correlation clustering with asymmetric classification errors. In *ICML*. PMLR, 2020.

Sanchit Kalhan, Konstantin Makarychev, and Timothy Zhou. Correlation clustering with local objectives. *Neural Information Processing Systems*, 2019. doi: 10.5555/3454287.3455125.

Philip N. Klein, Claire Mathieu, and Hang Zhou. Correlation clustering and two-edge-connected augmentation for planar graphs. *Algorithmica*, pages 1–34, 2023. doi: 10.1007/s00453-023-01128-w.

Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46 (6):787–832, 1999. doi: 10.1145/331524.331526.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data*, 1(1):2–es, 2007. doi: 10.1145/1217299.1217301.

Julian McAuley and Jure Leskovec. Learning to discover social circles in ego networks. *Neural Information Processing Systems*, 2012. doi: 10.5555/2999134.2999195.

Gregory Puleo and Olgica Milenkovic. Correlation clustering and biclustering with locally bounded errors. In *ICML*, 2016. doi: 10.1109/TIT.2018.2819696.

Benedek Rozemberczki and Rik Sarkar. Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models. In *International Conference on Information and Knowledge Management*, 2020. doi: 10.1145/3340531.3411866.

Nate Veldt. Correlation clustering via strong triadic closure labeling: Fast approximation algorithms and practical lower bounds. In *ICML*, 2022. URL https://proceedings.mlr.press/v162/veldt22a.html.

Thomas Voice, Maria Polukarov, and Nicholas R. Jennings. Coalition structure generation over graphs. *Journal of Artificial Intelligence Research*, 45:165–196, 2012. doi: 10.1613/jair.3715.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. Yes (Definition 1, Theorems 2 and 3, Algorithm 1)
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Yes (Theorems 2 and 3, Lemma 3). The space complexity of all algorithm is clearly $\mathcal{O}(n^2)$ for storing adjacency matrices.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. Yes (supplemental material)

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. Yes (Section 4)
   (b) Complete proofs of all theoretical results. Yes (Section 4)
   (c) Clear explanations of any assumptions. Yes (Section 4)

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). Yes (supplemental material)
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Not Applicable
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). Yes (Section 6)
   (d) A description of the computing infrastructure used. Yes (Section 6)

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. Yes
   (b) The license information of the assets, if applicable. Not Applicable
   (c) New assets either in the supplemental material or as a URL, if applicable. Not Applicable
   (d) Information about consent from data providers/curators. Not Applicable
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. Not Applicable

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. Not Applicable
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. Not Applicable
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. Not Applicable