Improved Algorithms for Overlapping and Robust Clustering of Edge-Colored Hypergraphs: An LP-Based Combinatorial Approach

Changyeol Lee[*]
Department of Computer Science and Engineering
Yonsei University
Seoul, South Korea
777john@yonsei.ac.kr

Yongho Shin[*]
Institute of Computer Science
University of Wrocław
Wrocław, Poland
yongho@cs.uni.wroc.pl

Hyung-Chan An[†]
Department of Computer Science and Engineering
Yonsei University
Seoul, South Korea
hyung-chan.an@yonsei.ac.kr

Abstract

Clustering is a fundamental task in both machine learning and data mining. Among various methods, edge-colored clustering (ECC) has emerged as a useful approach for handling categorical data. Given a hypergraph with (hyper)edges labeled by colors, ECC aims to assign vertex colors to minimize the number of edges where the vertex color differs from the edge's color. However, traditional ECC has inherent limitations, as it enforces a nonoverlapping and exhaustive clustering. To tackle these limitations, three versions of ECC have been studied: LOCAL ECC and GLOBAL ECC, which allow overlapping clusters, and ROBUST ECC, which accounts for vertex outliers. For these problems, both linear programming (LP) rounding algorithms and greedy combinatorial algorithms have been proposed. While these LP-rounding algorithms provide high-quality solutions, they demand substantial computation time; the greedy algorithms, on the other hand, run very fast but often compromise solution quality. In this paper, we present a family of algorithms that combines the strengths of LP with the computational efficiency of combinatorial algorithms. Both experimental and theoretical analyses show that our algorithms efficiently produce high-quality solutions for all three problems: LOCAL, GLOBAL, and ROBUST ECC. We complement our algorithmic contributions with complexity-theoretic inapproximability results and integrality gap bounds, which suggest that significant theoretical improvements are unlikely. Our results also answer two open questions previously raised in the literature.

1 Introduction

Clustering is a fundamental task in both machine learning and data mining [24, 35, 25]. *Edge-colored clustering* (ECC), in particular, is a useful model when interactions between the items to be clustered are represented as categorical data [8, 4]. To provide intuition, let us consider the following simple, illustrative example from prior work [34, 4, 37, 51, 48, 19, 20]: given a set of food ingredients,

[*]Equal contributions.
[†]Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

recipes that use them, and a (noisy) labeling of these recipes indicating their cuisine (e.g., Italian or Indian), can we group the food ingredients by their cuisine? To address this question, we can begin by considering a hypergraph whose vertices correspond to ingredients, (hyper)edges represent recipes, and edge colors correspond to cuisines. We can then find a labeling of the ingredients such that, in most recipes, all ingredient labels match the recipe's label. This is precisely what ECC does: given an edge-colored hypergraph, the goal is to assign colors to its vertices so that the number of edges where vertex colors differ from the edge color is minimized. Intuitively, this problem offers an approach for clustering vertices when edge labels are noisy.

However, ECC has an inherent limitation in that it insists on assigning exactly one color to every vertex, enforcing a nonoverlapping and exhaustive clustering. In the above illustrative example, food ingredients are often shared across geographically neighboring cuisines, indicating that overlapping clustering may be preferable. Moreover, some ingredients, such as salt, commonly appear in nearly all cuisines and may be considered outliers that should ideally be excluded from the clustering process. To address these limitations, three generalizations of ECC, namely, LOCAL ECC, GLOBAL ECC, and ROBUST ECC, have been proposed [19]. Among them, LOCAL ECC and GLOBAL ECC allows overlapping clustering: in LOCAL ECC, a *local budget* b_{local} that specifies the maximum number of colors each vertex can receive is given as an input parameter, thereby allowing clusters to overlap. In GLOBAL ECC, vertices may be assigned multiple colors, but with the total number of extra assignments constrained by a *global budget* b_{global} given as input. On the other hand, ROBUST ECC enhances robustness against vertex outliers by allowing up to b_{robust} vertices to be deleted from the hypergraph. This budget b_{robust} is also specified as part of the input. (Alternatively, this can be viewed as designating those vertices as "wildcards" that can be treated as any color.)

While LOCAL ECC, GLOBAL ECC, and ROBUST ECC are useful extensions of ECC that effectively address its limitations, these problems are unfortunately NP-hard, making exact solutions computationally intractable. This directly follows from the NP-hardness of ECC [8], a common special case of all three problems. This computational intractability naturally motivates the study of approximation algorithms for these problems. Recall that an algorithm is called a *ρ-approximation algorithm* if it runs in polynomial time and guarantees a solution within a factor of ρ relative to the optimum.

In this paper, we present a new family of algorithms for overlapping and robust clustering of edge-colored hypergraphs that is linear programming-based (LP-based) yet also combinatorial. Previously, combinatorial algorithms and (non-combinatorial) LP-based algorithms have been proposed for these problems. For LOCAL ECC, Crane et al. [19] gave a greedy combinatorial r-approximation algorithm, where r is the rank of the hypergraph. Their computational evaluation demonstrated that this algorithm runs remarkably faster than their own LP-rounding algorithm, at the expense of a trade-off in solution quality. The theoretical analysis [19] of the LP-rounding algorithm successfully obtains an approximation ratio that does not depend on r: they showed that their algorithm is a $(b_{\mathsf{local}} + 1)$-approximation algorithm. They state it as an open question whether there exists an $O(1)$-approximation algorithm for LOCAL ECC. For ROBUST ECC as well, Crane et al. gave a greedy r-approximation algorithm; however, their LP-rounding algorithm in this case does not guarantee solution feasibility. According to their computational evaluation, solutions produced by the LP-rounding algorithm were of very high quality but violated the budget constraint, which is reflected in the theoretical result: their algorithm is a *bicriteria* $(2+\epsilon, 2+\frac{4}{\epsilon})$-approximation algorithm for any positive ϵ, i.e., an algorithm that produces an $(2+\epsilon)$-approximation solution but violates the budget constraint by a multiplicative factor of at most $2+\frac{4}{\epsilon}$. Finally for GLOBAL ECC, Crane et al. gave similar results: a greedy r-approximation algorithm and a bicriteria $(b_{\mathsf{global}} + 3 + \epsilon, 1 + \frac{b_{\mathsf{global}}+2}{\epsilon})$-approximation algorithm for any positive ϵ, where the latter, empirically, was slow but produced solutions of high quality. Since their bicriteria approximation ratio is not $(O(1), O(1))$ for GLOBAL ECC, Crane et al. left it another open question whether bicriteria $(O(1), O(1))$-approximation is possible for GLOBAL ECC.

The primal-dual method is an algorithmic approach that constructs combinatorial algorithms based on LP, allowing one to combine the strengths of both worlds [29, 30]. Our algorithms are designed using the primal-dual method. We analyze its performance both experimentally and theoretically. For LOCAL ECC, our approach yields a combinatorial $(b_{\mathsf{local}} + 1)$-approximation algorithm, which is the same approximation ratio as Crane et al.'s LP-rounding algorithm; however, our algorithm is combinatorial and runs in linear time. The experiments confirmed that, compared to the previous combinatorial algorithm, our algorithm brings improvement in both computation time and solution

quality. We complement this algorithmic result by showing inapproximability results that match our approximation ratio; this answers one of Crane et al.'s open questions. For ROBUST ECC and GLOBAL ECC, our results give a true (non-bicriteria) approximation algorithm, avoiding the need for bicriteria approximation.[3] Our true approximation algorithm for ROBUST ECC, with the ratio of $2(b_{\mathsf{robust}}+1)$, was enabled by our new LP relaxation: the integrality gap of the relaxation used by previous results is $+\infty$ [19], whereas our LP has an integrality gap of $O(b_{\mathsf{robust}})$. In fact, we show that our gap is $\Theta(b_{\mathsf{robust}})$, suggesting that our ratio may be asymptotically the best one can achieve based on this relaxation. For GLOBAL ECC, our true approximation algorithm has the ratio of $2(b_{\mathsf{global}}+1)$, and our bicriteria approximation algorithm has the ratio of $(2+\epsilon, 1+\frac{2}{\epsilon})$. This affirmatively answers another open question of Crane et al.: bicriteria $(O(1), O(1))$-approximation for GLOBAL ECC is indeed possible. We also show that our relaxation has the integrality gap of $\Theta(b_{\mathsf{global}})$.

Below, we summarize which contributions of our work are presented in which sections of the paper.

- In Section 3.1, we present our algorithm for LOCAL ECC; its performance is analyzed both experimentally (Section 4.2) and theoretically (Section 3.1 and Appendix A.1). We also present the inapproximability result (Theorems 3.3 and 3.4) that answers Crane et al.'s open question [19], whose technical proof is deferred to Appendix A.3.
- In Section 3.2, we present our true approximation algorithm for ROBUST ECC based on a new stronger LP formulation. Our algorithm's performance is analyzed both experimentally (Section 4.3) and theoretically (Section 3.2 and Appendix B.2), including an integrality gap lower bound (Section 3.2; note that an upper bound is implied by the proof of Theorem 3.5).
- In Section 3.2 and Appendix C.2, we present our true approximation algorithm for GLOBAL ECC, whose performance is analyzed both experimentally (Section 4.3) and theoretically (Appendix C.3). This algorithm extends to the bicriteria setting (Section 3.2 and Appendix C.5), answering another open question of Crane et al. [19].

We note that LP-rounding algorithms based on our relaxations can match the ratios of our combinatorial true approximation algorithms. However, we omit them from this paper, as they offer no improvement in performance guarantees while requiring significantly more computation time to solve LPs.

Related work. ECC has been used for a variety of tasks including categorical community detection, temporal community detection [4], and diverse and experienced group discovery [5]; recently, it has also been applied to fair and balanced clustering [20]. Angel et al. [8] initiated the study of clustering edge-colored graphs (not hypergraphs). After showing its NP-hardness, they gave the first approximation algorithm for the (maximization) problem, with the approximation ratio of e^{-2}. Subsequent studies [1, 3, 2] improved this ratio, and recently, Crane et al. [20] achieved $\frac{154}{405}$-approximation. Veldt [48] showed its APX-hardness.

Given the emerging importance of clustering data with higher-order interactions [12, 40], Amburg et al. [4] addressed clustering on edge-colored *hypergraphs* for the first time, and gave 2-approximation algorithms. Veldt [48] presented a combinatorial 2-approximation algorithm along with a UGC-hardness ruling out any constant smaller than 2.

As was highlighted by previous studies [6, 4, 48], ECC is closely related to correlation clustering problems [9], which has been extensively studied in machine learning and data mining [52, 11, 43, 50]. They share the common feature of taking (hyper)edges representing similarity between vertices as input, and thus both have been applied to similar sets of taks such as community detection [49, 4]. However, correlation clustering differs from ECC in that it treats the absence of an edge as an indication of dissimilarity, whereas ECC interprets it merely as a lack of information. Chromatic correlation clustering, which introduces categorical edges to correlation clustering, is another closely related problem to ECC [15, 6, 37, 51]. Interestingly, unlike correlation clustering which was studied on hypergraphs and received significant interest [36, 26, 28], it appears that the chromatic hypergraph correlation clustering has never been studied to the best of our knowledge. We note that this may be an

[3] If, in some contexts, a bicriteria approximation algorithm is acceptable for use, we could instead use a true approximation algorithm with a relaxed budget. Thus, once a true approximation algorithm becomes available, the need for bicriteria approximation algorithms is reduced. However, our algorithms can also be analyzed in the bicriteria setting for both GLOBAL ECC and ROBUST ECC. See Appendices C.5 and B.3.

interesting future direction of research. Other variants of correlation clustering, including overlapping variants [16, 7, 39, 17], and robust variants [22, 32] have been studied. We refer interested readers to the book by Bonchi, García-Soriano, and Gullo [14] and references therein.

2 Problem definitions

In this section, we formally define the problems considered in this paper. First, we describe the part of the input that is common to all three problems. We are given a hypergraph $H = (V, E)$ and a set C of colors as input. Since H is a hypergraph, we have $E \subseteq 2^V$. Each edge $e \in E$ is associated with a color $c_e \in C$.

Given a *node coloring* $\sigma : V \to C$, we say an edge $e \in E$ is a *mistake* if there exists a node $v \in e$ whose assigned color $\sigma(v)$ differs from c_e, i.e., $c_e \neq \sigma(v)$. Otherwise, we say that e is *satisfied*. In LOCAL ECC and GLOBAL ECC, a node coloring $\sigma : V \to 2^C$ assigns (possibly) a multiple number of colors to each node. In these problems, we say $e \in E$ is a *mistake* if there exists a node $v \in e$ whose assigned color does not include c_e, i.e., $c_e \notin \sigma(v)$.

Definition 2.1. In LOCAL ECC, in addition to H, C, and $\{c_e\}_{e\in E}$, a *local budget* $b_{\mathsf{local}} \in \mathbb{Z}_{\geq 1}$ is given as input. The goal is to find a node coloring $\sigma : V \to 2^C$ such that $|\sigma(v)| \leq b_{\mathsf{local}}$ for all v to minimize the number of mistakes.

Definition 2.2. In GLOBAL ECC, in addition to H, C, and $\{c_e\}_{e\in E}$, a *global budget* $b_{\mathsf{global}} \in \mathbb{Z}_{\geq 0}$ is given as input. The goal is to find a node coloring $\sigma : V \to 2^C$ such that $|\sigma(v)| \geq 1$ for all v and $\sum_{v\in V} |\sigma(v)| \leq |V| + b_{\mathsf{global}}$, to minimize the number of mistakes.

Definition 2.3. In ROBUST ECC, in addition to H, C, and $\{c_e\}_{e\in E}$, a *node-removal budget* $b_{\mathsf{robust}} \in \mathbb{Z}_{\geq 0}$ is given as input. The goal is to remove at most b_{robust} nodes from the hypergraph and find a node coloring $\sigma : (V \setminus V_R) \to C$ to minimize the number of mistakes, where V_R denotes the set of removed nodes.

Recall that removing a node from H makes the node disappear from all the incident edges.

In Section 3, our algorithms will be presented for slightly generalized versions of the problems. We introduce edge weights $w_e \in \mathbb{Q}_{\geq 0}$ so that we minimize the total weight, not number, of mistakes. In LOCAL ECC, instead of b_{local} that uniformly applies to all nodes, we will let each node v specify its own budget b_v. Note that it suffices to solve these generalizations.

We conclude this section by introducing notation to be used throughout this paper. For $F \subseteq E$, let $\chi(F) := \{c_e \mid e \in F\}$ be the set of colors of the edges in F. For $v \in V$, let $\delta(v)$ be the set of edges that are incident with v; $d_v := |\delta(v)|$ is the degree of v. Let $\delta_c(v)$ be the set of edges in $\delta(v)$ whose color is c, i.e., $\delta_c(v) := \{e \in \delta(v) \mid c_e = c\}$.

3 Proposed algorithms

3.1 Local ECC

In this section, we informally present our approximation algorithm for LOCAL ECC. Although we will discuss all the necessary technical details here, we will still present a formal analysis in Appendix A for the completeness' sake.

Following are an LP relaxation (left) and its dual (right). Intuitively, $x_{v,c} = 1$ indicates that node v is colored with c and $x_{v,c} = 0$ otherwise; $y_e = 1$ if e is a mistake and $y_e = 0$ otherwise.

$$\begin{aligned}
\min\ & \textstyle\sum_{e\in E} w_e y_e && && \max\ & \textstyle\sum_{e\in E, v\in e} \beta_{e,v} - \sum_{v\in V} b_v \alpha_v && \\
\text{s.t.}\ & \textstyle\sum_{c\in C} x_{v,c} \leq b_v, && \forall v \in V, && \text{s.t.}\ & \textstyle\sum_{e\in\delta_c(v)} \beta_{e,v} \leq \alpha_v, && \forall v \in V, c \in C, \\
& x_{v,c_e} + y_e \geq 1, && \forall e \in E, v \in e, && & \textstyle\sum_{v\in e} \beta_{e,v} \leq w_e, && \forall e \in E, \\
& x_{v,c} \geq 0, && \forall v \in V, c \in C, && & \alpha_v \geq 0, && \forall v \in V, \\
& y_e \geq 0, && \forall e \in E. && & \beta_{e,v} \geq 0, && \forall e \in E, v \in e.
\end{aligned}$$

As a primal-dual algorithm, our algorithm maintains a dual solution (α, β), which changes throughout the execution of the algorithm but remains feasible at all times. The algorithm constructs the "primal"

solution partially guided by the complementary slackness: namely, it allows an edge e to be a mistake only if the corresponding dual constraint $\sum_{v \in e} \beta_{e,v} \leq w_e$ is *tight*, i.e., $\sum_{v \in e} \beta_{e,v} = w_e$. This is useful since the cost of the algorithm's output can then be written as $\sum_{e \in E_m} w_e = \sum_{e \in E_m} \sum_{v \in e} \beta_{e,v} \leq \sum_{e \in E} \sum_{v \in e} \beta_{e,v}$, where E_m is the set of mistakes in the output. Let $B_v := \sum_{e \in \delta(v)} \beta_{e,v}$, and the algorithm's output cost is no greater than $\sum_{v \in V} B_v$ at termination.

In order to maintain dual feasibility, the algorithm begins with a trivial dual feasible solution $(\alpha, \beta) = (\mathbf{0}, \mathbf{0})$ and only increases dual variables, never decreasing them. The first set of constraints will never be violated because whenever we increase $\sum_{e \in \delta_c(v)} \beta_{e,v}$, we will increase α_v by the same amount. The second set of constraints will never be violated simply because we will stop increasing all $\beta_{e,v}$ for $v \in e$ once edge e becomes tight.

We are now ready to present the algorithm. To better convey the intuition, we will describe the algorithm as if it is a "continuous" process that continuously increases a set of variables as time progresses. In this *process over time* perspective (see, e.g., [31]), a primal-dual algorithm starts with an initial (usually all-zero) dual solution at time 0, and the algorithm specifies the increase rate at which each dual variable increases. The dual variables continue to increase at the specified rates until an event of interest—typically, a dual constraint becomes tight—occurs. At that point, the algorithm pauses the progression of time to handle the event and recompute the increase rates. Once updated, time proceeds again. In Appendix A.5, we also provide a discretized version of the algorithm, making all implementation details explicit.

Consider the following algorithm. It maintains a set L of all those edges that are not tight. We call these edges *loose*. One point that requires additional explanation in this pseudocode is that it increases *a sum of variables* $\sum_{e \in \delta_c(v) \cap L} \beta_{e,v}$ at unit rate, rather than a single variable. This should be interpreted as increasing the variables in the summation in an arbitrary way, provided that their total increase rate is 1 and that no variable is ever decreased. The algorithm's analysis holds for any such choice of the increase rates of individual variables as long as their total is 1.

Algorithm 1 Proposed algorithm for LOCAL ECC

$\alpha \leftarrow \mathbf{0}$; $\beta \leftarrow \mathbf{0}$
$L \leftarrow \{e \in E \mid w_e > 0\}$
for $v \in V$ **do**
  while $|\chi(\delta(v) \cap L)| > b_v$ **do**
    increase α_v and $\sum_{e \in \delta_c(v) \cap L} \beta_{e,v}$ for each $c \in \chi(\delta(v) \cap L)$ at unit rate, until there exists e such that $\sum_{u \in e} \beta_{e,u} = w_e$
    if $\exists e \ \sum_{u \in e} \beta_{e,u} = w_e$ **then** remove all such edges from L
  $\sigma(v) \leftarrow \chi(\delta(v) \cap L)$
return σ

This algorithm can be implemented as a usual discrete algorithm using the standard technique for emulating "continuous" algorithms by discretizing them. Once the increase rates are determined, the discretized algorithm computes, for each edge, after how much time the edge would become tight if we continuously and indefinitely increased the dual variables, and selects the minimum among them. That is the amount of time the emulated algorithm runs before getting paused. The discretized algorithm then handles the event, recomputes the increase rates, and repeat. See Appendix A.5.

It is easy to see that Algorithm 1 returns a feasible solution: we assign $\chi(\delta(v) \cap L)$ to v only after ensuring $|\chi(\delta(v) \cap L)| \leq b_v$. The analysis can focus on bounding the final value of $\sum_{v \in V} B_v$: recall that it was an upper bound on the algorithm's output cost. We will compare $\sum_{v \in V} B_v$ against the dual objective value, which is a lower bound on the true optimum from the LP duality.

Both $\sum_{v \in V} B_v$ and the dual objective value change throughout the algorithm's execution. At the beginning, both are zeroes because $(\alpha, \beta) = (\mathbf{0}, \mathbf{0})$. How do they change over the execution? In each iteration of the **while** loop, the algorithm increases α_v at unit rate and B_v at rate $|\chi(\delta(v) \cap L)|$, where v is the vertex being considered at the moment. (Note that $B_v = \sum_{c \in \chi(\delta(v) \cap L)} \sum_{e \in \delta_c(v) \cap L} \beta_{e,v} + \sum_{e \in \delta(v) \setminus L} \beta_{e,v}$.) That is, at any given moment of the algorithm's execution, the rate by which $\sum_{u \in V} B_u$ gets increased is $|\chi(\delta(v) \cap L)| > b_v$, and the increase rate of the dual objective is

$|\chi(\delta(v) \cap L)| - b_v$. Note that the ratio between these two rates is $\frac{|\chi(\delta(v)\cap L)|}{|\chi(\delta(v)\cap L)|-b_v} \leq b_v + 1$ since $|\chi(\delta(v) \cap L)| > b_v$. Since the upper bound on the algorithm's output and the lower bound on the true optimum were initially both zeroes and the ratio between their increase rate is no greater than $b_v + 1$ at all times, the overall approximation ratio is $b_{\mathsf{max}} + 1$ where $b_{\mathsf{max}} := \max_{v \in V} b_v$. Note that $b_{\mathsf{max}} = b_{\mathsf{local}}$ under the original definition of LOCAL ECC.

Theorem 3.1. *Algorithm 1 is a* $(b_{\mathsf{local}} + 1)$*-approximation algorithm for* LOCAL ECC.

Algorithm 1 can be implemented to run in linear time (see Lemma A.3 in Appendix A.1).

Our algorithm harnesses the full "power" of the LP relaxation, in that its approximation ratio matches the integrality gap of the relaxation. We defer the proof of Theorem 3.2 to Appendix A.2.

Theorem 3.2. *There is a sequence of instances of* LOCAL ECC *such that the ratio between a fractional solution and an optimal integral solution converges to* $b_{\mathsf{local}} + 1$.

In fact, our inapproximability results further show that our approximation ratio is essentially the best possible. We note that these results answer one of the open questions raised by Crane et al. [19], namely, whether an $O(1)$-approximation algorithm is possible for LOCAL ECC.

Theorem 3.3. *For any constant* $\epsilon > 0$*, it is* UGC*-hard to approximate* LOCAL ECC *within a factor of* $b_{\mathsf{local}} + 1 - \epsilon$.

If one prefers a milder complexity-theoretic assumption, we show the following theorem as well.

Theorem 3.4. *For any* $b_{\mathsf{local}} \geq 2$ *and any constant* $\epsilon > 0$*, there does not exist a* $(b_{\mathsf{local}} - \epsilon)$*-approximation algorithm for* LOCAL ECC *unless* $\mathrm{P} = \mathrm{NP}$.

The proofs of Theorems 3.3 and 3.4 are deferred to Appendix A.3.

Final remarks. Since our algorithm considers the nodes one by one and operates locally, Algorithm 1 immediately works as an online algorithm, in which vertices are revealed to the algorithm in an online manner.[4] In Appendix A.4, we also show that the algorithm can be analyzed in the bicriteria setting, yielding a $(1 + \epsilon, 1 + \frac{1}{b_{\mathsf{local}}}\lceil \frac{b_{\mathsf{local}}}{\epsilon} \rceil - \frac{1}{b_{\mathsf{local}}})$-approximation for $\epsilon \in (0, b_{\mathsf{local}}]$. Finally, we note that Algorithm 1 does not specify the order in which the **for** loop processes the vertices, which may leave room for optimization in practice. However, our empirical evaluation indicated that such room was rather limited.

3.2 Robust ECC and Global ECC

In this section, we summarize our algorithmic results for ROBUST ECC and GLOBAL ECC. Both problems involve global constraints, and as a result, their LP formulations (and the proposed algorithms) become quite similar. As such, in the interest of space, we will sketch our algorithm only for ROBUST ECC in this section. The only real difference between the two algorithms is in the constraints of the dual LPs.

Following are an LP relaxation (left) and its dual (right) used by the algorithm for ROBUST ECC, where $z_v = 1$ indicates that the node v is removed from the hypergraph.

$$
\begin{aligned}
\min\ & \textstyle\sum_{e \in E} w_e y_e & & \\
\text{s.t. } & z_v + \textstyle\sum_{c \in C} x_{v,c} \leq 1, & & \forall v \in V, \\
& z_v + x_{v,c_e} + y_e \geq 1, & & \forall e \in E, v \in e, \\
& \textstyle\sum_{v \in V} z_v \leq b_{\mathsf{robust}}, & & \\
& x_{v,c} \geq 0, & & \forall v \in V, c \in C, \\
& y_e \geq 0, & & \forall e \in E, \\
& z_v \geq 0, & & \forall v \in V.
\end{aligned}
\qquad
\begin{aligned}
\max\ & \textstyle\sum_{e \in E, v \in e} \beta_{e,v} - \sum_{v \in V} \alpha_v - \lambda b_{\mathsf{robust}} & & \\
\text{s.t. } & \textstyle\sum_{e \in \delta_c(v)} \beta_{e,v} \leq \alpha_v, & & \forall v \in V, c \in C, \\
& \textstyle\sum_{v \in e} \beta_{e,v} \leq w_e, & & \forall e \in E, \\
& \textstyle\sum_{e \in \delta(v)} \beta_{e,v} - \alpha_v \leq \lambda, & & \forall v \in V, \\
& \beta_{e,v} \geq 0, & & \forall e \in E, v \in e, \\
& \lambda \geq 0. & &
\end{aligned}
$$

[4] There are several ways to describe the online setting, but a simple (albeit slightly weak) formulation is as follows: initially, the algorithm is given only the number of hyperedges. Then, at each timestep, when a new vertex v arrives, the algorithm is informed of which hyperedges are incident to v. At that point of time, the algorithm is required to irrevocably color v.

We note that the only difference between our LP and Crane et al.'s [19] lies in the constraint $z_v + \sum_{e \in C} x_{v,c} \leq 1$. (The two LPs use opposite senses for the binary variable x, but this is not an inherent difference.) This difference, which does not change the value of optimal integral solutions, turns out to be enough to reduce the integrality gap of our relaxation. See Theorem 3.5.

Let us now sketch the algorithm for ROBUST ECC we propose. The algorithm maintains a dual feasible solution (α, β, λ), initially set as $(\mathbf{0}, \mathbf{0}, 0)$. The set L will be kept as the set of loose edges; $R \subseteq V$ is the set of nodes with at least two incident loose edges of distinct colors. Intuitively, R is the set of nodes we will remove from the hypergraph. The algorithm therefore continues its execution until $|R| \leq b_{\mathsf{robust}}$ holds. When increasing the dual variables, the algorithm increases variables associated with all vertices in R at the same time, unlike Algorithm 1 which handles one node at a time. The following two properties will hold:

(i) The algorithm increases λ and $\sum_{e \in \delta(v) \cap L} \beta_{e,v} - \alpha_v$ for each $v \in R$ at the same rate.

(ii) For each $v \in R$, the algorithm increases α_v and $\sum_{e \in \delta_c(v) \cap L} \beta_{e,v}$ for each $c \in \chi(\delta(v) \cap L)$ at the same rate. In general, the increase rate of α_{v_1} may be different from that of α_{v_2} for $v_1 \neq v_2$.

These properties can be ensured as follows: the increase rate of λ is set as 1. For each $v \in R$, we increase α_v and $\sum_{e \in \delta_c(v) \cap L} \beta_{e,v}$ for each $c \in \chi(\delta(v) \cap L)$ at rate $\frac{1}{|\chi(\delta(v) \cap L)| - 1}$.

Once $|R|$ becomes less than or equal to b_{robust}, the algorithm removes R from the hypergraph and assigns every node $v \in V \setminus R$ the (only) color in $\chi(\delta(v) \cap L)$. If $\chi(\delta(v) \cap L) = \emptyset$, an arbitrary color can be assigned without affecting the theoretical guarantee on solution quality; in practical implementation, we could employ heuristics for marginal improvement. In the interest of space, the full pseudocodes are deferred to Appendices B.1 and B.4.

We prove the following theorems in Appendices B.2 and C.3.

Theorem 3.5. *Algorithm 3 is a* $2(b_{\mathsf{robust}} + 1)$*-approximation algorithm for* ROBUST ECC.

Theorem 3.6. *Algorithm 5 is a* $2(b_{\mathsf{global}} + 1)$*-approximation algorithm for* GLOBAL ECC.

Both algorithms can be implemented to run in $O(|E| \sum_{v \in V} d_v)$ time (see Lemma C.3).

The LP relaxation of Crane et al. [19] for ROBUST ECC has infinite integrality gap, whereas the integrality gap of our LP is $O(b_{\mathsf{robust}})$, following from the proof of Theorem 3.5. This makes it possible to obtain a true (non-bicriteria) approximation algorithm based on our LP. In fact, the following theorems show that our LP for ROBUST ECC (and GLOBAL ECC) has an integrality gap of $\Theta(b_{\mathsf{robust}})$ (and $\Theta(b_{\mathsf{global}})$), respectively. The proof of Theorem 3.8 is deferred to Appendix C.4.

Theorem 3.7. *The integrality gap of our LP for* ROBUST ECC *is at least* $b_{\mathsf{robust}} + 1$.

Proof. Consider a hypergraph $H = (V = \{v_1, \cdots, v_{b_{\mathsf{robust}}+1}\}, E = \{e_1, e_2\})$ where $e_1 = e_2 = V$, $w_{e_1} = w_{e_2} = 1$, and $c_{e_1} \neq c_{e_2}$. Any integral solution incurs at least 1 since at least one node should remain in the hypergraph and at least one edge cannot be satisfied. However, consider the solution given by $z_v = \frac{b_{\mathsf{robust}}}{b_{\mathsf{robust}}+1}$, $x_{v,c_{e_1}} = x_{v,c_{e_2}} = \frac{1}{2(b_{\mathsf{robust}}+1)}$ for all $v \in V$ and $y_{e_1} = y_{e_2} = \frac{1}{2(b_{\mathsf{robust}}+1)}$. This solution is feasible and the cost is $\frac{1}{b_{\mathsf{robust}}+1}$. □

Theorem 3.8. *The integrality gap of the LP for* GLOBAL ECC *is at least* $b_{\mathsf{global}} + 1$.

Final remarks. Our algorithms can be analyzed in the bicriteria setting as well, yielding a bicriteria $(2 + \epsilon, 1 + \frac{1}{b}\lceil \frac{2b}{\epsilon} \rceil - \frac{1}{b})$-approximation algorithm for all $\epsilon \in (0, 2b]$, where $b = b_{\mathsf{robust}}$ for ROBUST ECC and $b = b_{\mathsf{global}}$ for GLOBAL ECC. This improves the best bicriteria approximation ratios previously known; furthermore, it affirmatively answers one of the open questions of Crane et al. [19], namely, whether there exists a bicriteria $(O(1), O(1))$-approximation algorithm for GLOBAL ECC. See Appendices B.3 and C.5.

4 Experiments

In this section, we analyze the performance of the proposed family of algorithms through experiments. We describe the setup in Section 4.1. We evaluate and discuss the performance of our algorithm for LOCAL ECC in Section 4.2. In Section 4.3, we address ROBUST and GLOBAL ECC.

Table 1: Statistics of the benchmark datasets.

Datasets	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert C\rvert$	r	$\bar{d}$	Δ_χ	$\bar{d}_\chi$	ρ
Brain	638	21,180	2	2	66.4	2	1.92	0.91
MAG-10	80,198	51,889	10	25	2.3	9	1.26	0.18
Cooking	6,714	39,774	20	65	63.8	20	4.35	0.61
DAWN	2,109	87,104	10	22	162.7	10	3.72	0.74
Walmart	88,837	65,898	44	25	5.1	40	2.65	0.52
Trivago	207,974	247,362	55	85	3.6	32	1.55	0.23

4.1 Setup

Our experiments used the same benchmark as Crane et al. [19], which contains six datasets. See Appendix D for further description of the individual datasets. We remark that these datasets have been used as a benchmark to experimentally evaluate ECC also in other prior work [4, 48]. Table 1 summarizes some statistics of the datasets: the number of nodes $|V|$, number of edges $|E|$, number of colors $|C|$, rank $r := \max_{e\in E} |e|$, average degree $\bar{d} := \sum_{v\in V} d_v/|V|$, maximum color-degree $\Delta_\chi := \max_{v\in V} |\chi(\delta(v))|$, average color-degree $\bar{d}_\chi := \sum_{v\in V} |\chi(\delta(v))|/|V|$, and the ratio ρ of vertices whose color degree is at least 2, i.e., $\rho := |\{v \in V \mid |\chi(\delta(v))| \geq 2\}|/|V|$.

All experiments were performed on a machine with Intel Core i9-9900K CPU and 64GB of RAM. In our experiments, we used the original code of Crane et al. [47, 19] as the implementation of the previous algorithms. Since their code was written in Julia, we implemented our algorithms also in Julia to ensure a fair comparison. When running the original codes for the LP-rounding algorithms, we used Gurobi-12.0 as the LP solver. Gurobi was the solver of choice in previous work [47, 19, 48, 4], and it is widely recognized for its excellent speed [41, 42].

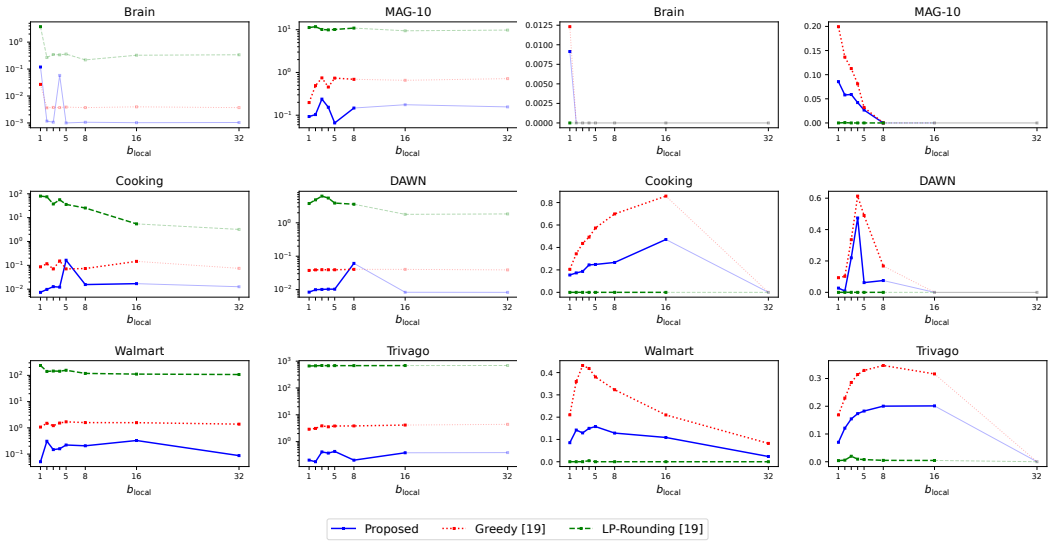
Our experiments focus on two aspects of the algorithms' performance: solution quality and running time. To compare solution quality, we will use *relative error estimate*, a normalized, estimated error of the algorithm's output cost (or quality) compared to the optimum. Since the problems are NP-hard, it is hard to compute the exact error compared to the optimum; as such, Crane et al. [19] used the optimal solution to their LP relaxation in lieu of the true optimum, giving an overestimate of the error. We followed this approach, but we used our LP relaxation instead since we can prove that our relaxation always yields a better estimate of the true optimum. To normalize the estimated error, we divide it by the estimated optimum: that is, the relative error estimate is defined as $(A - L)/L$, where A denotes the algorithm's output cost and L is the LP optimum.[5]

Crane et al.'s experiment [19] used $b_{\mathsf{local}} \in \{1, 2, 3, 4, 5, 8, 16, 32\}$ for LOCAL ECC, $b_{\mathsf{robust}}/|V| \in \{0, .01, .05, .1, .15, .2, .25\}$ for ROBUST ECC, and $b_{\mathsf{global}}/|V| \in \{0, .5, 1, 1.5, 2, 2.5, 3, 3.5, 4\}$ for GLOBAL ECC. While these choices were carefully made to help avoid *trivial* instances, we decided to extend their choice for GLOBAL ECC. To explain what trivial instances are, suppose that b_{local} is greater than the maximum color-degree Δ_χ in an instance of LOCAL ECC. The problem then becomes trivial, since the local budget allows assigning each vertex *all* the colors of its incident edges. We call an instance of LOCAL ECC *trivial* if $b_{\mathsf{local}} \geq \Delta_\chi$; similarly, ROBUST ECC instances are trivial if $b_{\mathsf{robust}} \geq \rho|V|$, and GLOBAL ECC instances are trivial if $b_{\mathsf{global}} \geq |V|(\bar{d}_\chi - 1)$. For LOCAL ECC and ROBUST ECC, Crane et al.'s choice of budgets ensure that most instances are nontrivial: each data set has 0, 1, or at most 2 trivial instances, possibly with the exception of at most one dataset. However, for GLOBAL ECC, only 44 instances out of 78 in the original benchmark are nontrivial, so we decided to additionally test $b_{\mathsf{global}}/|V| \in \{.1, .2, .3, .4\}$. As a result, we tested thirteen different budgets in total for each dataset for GLOBAL ECC.

4.2 Local ECC

We measured the solution quality and running time of the proposed algorithm in comparison with the greedy combinatorial algorithm and the LP-rounding algorithm of Crane et al. [19].

[5]When $L = 0$, we define the relative error estimate as 0. Note that $L = 0$ implies $A = 0$ since our LP has a bounded integrality gap.



(a) Running times ($\log_{10}$ scaled, in seconds) (b) Relative error estimates

Figure 1: (a) Running times (in seconds, log scale) and (b) relative error estimates of the LOCAL ECC algorithms. Empty square markers denote trivial instances.

Figure 1(a) depicts the running times, and Table 2 (in Appendix E) lists their average for each dataset. Figure 1(a) shows that our proposed algorithm was the fastest in most instances. It is not surprising that our algorithm, with the overall average running time of 0.121sec, was much faster than the LP-rounding algorithm whose overall average running time was 146.470sec, since our algorithm is combinatorial. This gap was no smaller even when we consider only nontrivial instances: the overall average running times were 0.142sec (proposed) and 180.367sec (LP-rounding). Remarkable was that the proposed algorithm was faster than the greedy algorithm, too. In fact, on average, it was more than twice as fast as the greedy algorithm in most datasets except for Brain. Such gap in the running times became more outstanding in larger datasets: for Trivago, our proposed algorithm was 11 times faster than the greedy algorithm and 2,100 times faster than the LP-rounding algorithm.

Figure 1(b) shows the relative error estimates of the algorithms' outputs. We note that, except for Brain and MAG-10, the relative error estimate of our algorithm (and of the greedy algorithm) tends to increase as b_{local} increases, and then at some point starts decreasing. This appears to be the result of the fact that the problem becomes more complex as b_{local} initially increases, but when b_{local} becomes too large, the problem becomes easy again. It can be seen from Figure 1(b) that our proposed algorithm outperformed the greedy algorithm in all cases. The overall average relative error estimate of our proposed algorithm was 0.141, which is less than half of the greedy algorithm's average of 0.297. The LP-rounding algorithm output near-optimal solutions in every case.

Overall, these experimental results demonstrate that the proposed family of algorithms is scalable, and produces solutions of good quality. As was noted by Veldt [48] and observed in this section, LP-rounding approach does not scale well due to its time consumption, even though it produces near-optimal solutions when it is given sufficient amount of time. Compared to the greedy combinatorial algorithm, our proposed algorithm output better solutions in smaller amount of time in most cases. This suggests that the proposed algorithm can provide improvement upon the greedy algorithm.

4.3 Robust ECC and Global ECC

We present the experimental results of both problems together in this section, starting with ROBUST ECC. We measured the performance of our proposed algorithm in addition to the greedy combinatorial algorithm and the LP-rounding algorithm of Crane et al. [19]. However, as their LP-rounding algorithm is a bicriteria approximation algorithm that possibly violates the budget b_{robust}, we cannot directly compare their solution quality with the proposed algorithm. In fact, the LP-rounding algorithm turned out to output "superoptimal" solutions violating b_{robust} in most cases of the experiment.

The bicriteria approximation ratio was chosen as $(6, 3)$, which is the same choice as in Crane et al.'s experiment [19].[6]

Comparing the average running times of each dataset reveals that the proposed algorithm ran much faster than the LP-rounding algorithm for most datasets, except for DAWN. The proposed algorithm was slower than the greedy algorithm for all datasets; however, it tended to produce solutions of much better quality than the greedy algorithm. The relative error estimate of the proposed algorithm was strictly better than that of the greedy algorithm in all nontrivial instances; the overall average relative error estimate of the proposed algorithm was 0.042, six times better than the greedy algorithm's average of 0.272. We also note that the relative error estimate of our algorithm stayed relatively even regardless of the budget, while that of the greedy algorithm fluctuated as b_{robust} changed in some datasets, such as MAG-10 and Trivago. Due to space constraints, a detailed table and a figure presenting the experimental results have been deferred to Appendix E.

For GLOBAL ECC, the bicriteria approximation ratio of the LP-rounding algorithm was chosen as $(2b_{\mathsf{global}} + 5, 2)$, which again is the same choice as in Crane et al.'s experiment. For GLOBAL ECC, the bicriteria approximation algorithm did not violate the budget for any instances of the benchmark. This may be due to the fact that their LP relaxation for GLOBAL ECC has a bounded integrality gap, unlike their LP for ROBUST ECC.[7]

The experimental results for Global ECC exhibited similar trends to those for Robust ECC. The relative error estimate of the proposed algorithm was strictly better than that of the greedy algorithm in all nontrivial instances. The average relative error estimate on nontrivial instances was 0.039 for the proposed algorithm, while that of the greedy algorithm was 0.912—more than 23 times higher. We also note that the relative error estimate of the greedy algorithm rapidly increased as b_{global} increased. While the proposed algorithm was on average slower than the greedy algorithm for all datasets, it was much faster than the LP-rounding algorithm in all datasets except for DAWN. A detailed table and a figure presenting the experimental results have been again deferred to Appendix E due to the space constraints.

The above results together indicate that our proposed algorithms for ROBUST ECC and GLOBAL ECC are likely to be preferable when a high-quality solution is desired possibly at the expense of a small increase in computation time.

5 Conclusion and discussion

In this paper, we presented a new family of algorithms for overlapping and robust clustering of edge-colored hypergraphs. Experimental results demonstrated that our algorithm improves upon the previous combinatorial algorithm for LOCAL ECC in both computation time and solution quality; compared to LP-rounding, it achieves significantly faster computation, with a slight trade-off in solution quality. For ROBUST ECC and GLOBAL ECC, our approach delivers improved solution quality with a slight increase in computation time compared to the previous combinatorial algorithms, while strictly satisfying the budget constraint. On the theoretical side, our analyses show that we achieve true $(b_{\mathsf{local}} + 1)$-, $2(b_{\mathsf{robust}} + 1)$-, $2(b_{\mathsf{global}} + 1)$-approximation for LOCAL, ROBUST, and GLOBAL ECC, respectively. We also provide inapproximability results for LOCAL ECC and integrality gap results for all three problems, suggesting that significant theoretical improvements are unlikely. These results lead to answers to two open questions posed in the literature [19].

There remain a few promising directions for future research. Although our combinatorial algorithm runs significantly faster than LP-rounding algorithms, its running time is still superlinear for ROBUST ECC and GLOBAL ECC. Can we optimize the dual update steps of our algorithms to obtain a linear-time algorithm for these two problems? Also, while our work focused on giving a better algorithm for ECC, it would be also interesting to explore additional applications of ECC, e.g., to the clustering tasks solved via correlation clustering. Given that k-PARTIAL VERTEX COVER admits a 2-approximation algorithm [27], another interesting question is if we can obtain an $O(1)$-approximation algorithm for ROBUST ECC as well.

[6] As a side remark, when we reran the proposed algorithm with the budget tripled to enable a comparison with the LP-rounding $(6, 3)$-approximation algorithm, the number of mistakes made by the proposed algorithm was, on average, as small as 57.2% of that made by the bicriteria algorithm.

[7] When we reran the proposed algorithm with the budget doubled, the number of mistakes made by the proposed algorithm was, on average, as small as 68.9% of that made by the bicriteria $(2b_{\mathsf{global}} + 5, 2)$-approximation algorithm.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers for their helpful comments. Supported by NCN grant number 2020/39/B/ST6/01641. This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2021-II212068, Artificial Intelligence Innovation Hub). This work was partly supported by an IITP grant funded by the Korean Government (MSIT) (No. RS-2020-II201361, Artificial Intelligence Graduate School Program (Yonsei University)). This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2025-00563707). Part of this research was conducted while Y. Shin was at Yonsei University.

References

[1] Alexander Ageev and Alexander Kononov. Improved approximations for the max k-colored clustering problem. In *Proceedings of the International Workshop on Approximation and Online Algorithms (WAOA)*, pages 1–10. Springer, 2014.

[2] Alexander Ageev and Alexander Kononov. A 0.3622-Approximation Algorithm for the Maximum k-Edge-Colored Clustering Problem. In *International Conference on Mathematical Optimization Theory and Operations Research (MOTOR)*, pages 3–15. Springer, 2020.

[3] Yousef M Alhamdan and Alexander Kononov. Approximability and inapproximability for maximum k-edge-colored clustering problem. In *Computer Science–Theory and Applications: 14th International Computer Science Symposium in Russia (CSR)*, pages 1–12. Springer, 2019.

[4] Ilya Amburg, Nate Veldt, and Austin Benson. Clustering in graphs and hypergraphs with categorical edge labels. In *Proceedings of The Web Conference (WWW)*, pages 706–717, 2020.

[5] Ilya Amburg, Nate Veldt, and Austin R Benson. Diverse and experienced group discovery via hypergraph clustering. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pages 145–153. SIAM, 2022.

[6] Yael Anava, Noa Avigdor-Elgrabli, and Iftah Gamzu. Improved theoretical and practical guarantees for chromatic correlation clustering. In *Proceedings of the 24th International Conference on World Wide Web (WWW)*, pages 55–65, 2015.

[7] Carlos E Andrade, Mauricio GC Resende, Howard J Karloff, and Flávio K Miyazawa. Evolutionary algorithms for overlapping correlation clustering. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 405–412, 2014.

[8] Eric Angel, Evripidis Bampis, A Kononov, Dimitris Paparas, Emmanouil Pountourakis, and Vassilis Zissimopoulos. Clustering on k-edge-colored graphs. *Discrete Applied Mathematics*, 211:15–22, 2016.

[9] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56:89–113, 2004.

[10] Nikhil Bansal and Subhash Khot. Inapproximability of hypergraph vertex cover and applications to scheduling problems. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 250–261. Springer, 2010.

[11] Thorsten Beier, Thorben Kroeger, Jorg H Kappes, Ullrich Kothe, and Fred A Hamprecht. Cut, glue & cut: A fast, approximate solver for multicut partitioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 73–80, 2014.

[12] Austin R Benson, David F Gleich, and Jure Leskovec. Higher-order organization of complex networks. *Science*, 353(6295):163–166, 2016.

[13] Manuel Blum, Robert W Floyd, Vaughan Pratt, Ronald L Rivest, and Robert E Tarjan. Linear time bounds for median computations. In *Proceedings of the 4th Annual ACM Symposium on Theory of Computing (STOC)*, pages 119–124, 1972.

[14] Francesco Bonchi, David García-Soriano, and Francesco Gullo. *Correlation Clustering*. Morgan & Claypool Publishers, 2022.

[15] Francesco Bonchi, Aristides Gionis, Francesco Gullo, Charalampos E Tsourakakis, and Antti Ukkonen. Chromatic correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(4):1–24, 2015.

[16] Francesco Bonchi, Aristides Gionis, and Antti Ukkonen. Overlapping correlation clustering. In *2011 IEEE 11th International Conference on Data Mining (ICDM)*, pages 51–60, 2011.

[17] Guilherme Oliveira Chagas, Luiz Antonio Nogueira Lorena, and Rafael Duarte Coelho dos Santos. A hybrid heuristic for the overlapping cluster editing problem. *Applied Soft Computing*, 81:105482, 2019.

[18] Philip S Chodrow, Nate Veldt, and Austin R Benson. Hypergraph clustering: from blockmodels to modularity. *Science Advances*, 2021.

[19] Alex Crane, Brian Lavallee, Blair D Sullivan, and Nate Veldt. Overlapping and robust edge-colored clustering in hypergraphs. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 143–151, 2024.

[20] Alex Crane, Thomas Stanley, Blair D. Sullivan, and Nate Veldt. Edge-colored clustering in hypergraphs: Beyond minimizing unsatisfied edges. In *Forty-second International Conference on Machine Learning (ICML)*, 2025.

[21] Nicolas A Crossley, Andrea Mechelli, Petra E Vértes, Toby T Winton-Brown, Ameera X Patel, Cedric E Ginestet, Philip McGuire, and Edward T Bullmore. Cognitive relevance of the community structure of the human brain functional coactivation network. *Proceedings of the National Academy of Sciences*, 110(28):11583–11588, 2013.

[22] Devvrit, Ravishankar Krishnaswamy, and Nived Rajaraman. Robust Correlation Clustering. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*, volume 145, pages 33:1–33:18, 2019.

[23] Irit Dinur, Venkatesan Guruswami, Subhash Khot, and Oded Regev. A new multilayered PCP and the hardness of hypergraph vertex cover. *SIAM Journal on Computing*, 34(5):1129–1146, 2005.

[24] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*, page 226–231, 1996.

[25] Absalom E Ezugwu, Abiodun M Ikotun, Olaide O Oyelade, Laith Abualigah, Jeffery O Agushaka, Christopher I Eke, and Andronicus A Akinyelu. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, 110:104743, 2022.

[26] Takuro Fukunaga. Lp-based pivoting algorithm for higher-order correlation clustering. *Journal of Combinatorial Optimization*, 37:1312–1326, 2019.

[27] Rajiv Gandhi, Samir Khuller, and Aravind Srinivasan. Approximation algorithms for partial covering problems. *Journal of Algorithms*, 53(1):55–84, 2004.

[28] David F Gleich, Nate Veldt, and Anthony Wirth. Correlation clustering generalized. In *29th International Symposium on Algorithms and Computation (ISAAC)*, 2018.

[29] Michel X Goemans and David P Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24(2):296–317, 1995.

[30] Michel X. Goemans and David P. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. In Dorit S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*. 1996.

[31] Fabrizio Grandoni, Jochen Könemann, Alessandro Panconesi, and Mauro Sozio. A primal-dual bicriteria distributed algorithm for capacitated vertex cover. *SIAM Journal on Computing*, 38(3):825–840, 2008.

[32] Sai Ji, Gaidi Li, Dongmei Zhang, and Xianzhao Zhang. Approximation algorithms for the capacitated correlation clustering problem with penalties. *Journal of Combinatorial Optimization*, 45(1), January 2023.

[33] jprenci, Walmart Competition Admin, and Will Cukierski. Walmart Recruiting: Trip Type Classification. `https://kaggle.com/competitions/walmart-recruiting-trip-type-classification`, 2015.

[34] Wendy Kan. What's Cooking? `https://kaggle.com/competitions/whats-cooking`, 2015. Kaggle.

[35] Alboukadel Kassambara. *Practical guide to cluster analysis in R: Unsupervised machine learning*, volume 1. Sthda, 2017.

[36] Sungwoong Kim, Sebastian Nowozin, Pushmeet Kohli, and Chang Yoo. Higher-order correlation clustering for image segmentation. *Advances in Neural Information Processing Systems (NIPS)*, 24, 2011.

[37] Nicolas Klodt, Lars Seifert, Arthur Zahn, Katrin Casel, Davis Issac, and Tobias Friedrich. A color-blind 3-approximation for chromatic correlation clustering and improved heuristics. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*, pages 882–891, 2021.

[38] Peter Knees, Yashar Deldjoo, Farshad Bakhshandegan Moghaddam, Jens Adamczak, Gerard-Paul Leyson, and Philipp Monreal. Recsys challenge 2019: Session-based hotel recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 570–571, 2019.

[39] Pan Li, Hoang Dau, Gregory Puleo, and Olgica Milenkovic. Motif clustering and overlapping clustering for social network analysis. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pages 1–9. IEEE, 2017.

[40] Pan Li and Olgica Milenkovic. Inhomogeneous hypergraph clustering with applications. *Advances in neural information processing systems (NeurIPS)*, 30, 2017.

[41] Hans D. Mittelmann. Latest benchmark results. In *INFORMS Annual Conference*, Phoenix, AZ, USA, 2018.

[42] Hans D. Mittelmann. Latest progress in optimization software. In *INFORMS Annual Meeting*, Phoenix, AZ, USA, 2023.

[43] Divya Pandove, Shivani Goel, and Rinkle Rani. Correlation clustering methodologies and their fundamental results. *Expert Systems*, 35(1):e12229, 2018.

[44] Mikail Rubinov and Olaf Sporns. Complex network measures of brain connectivity: uses and interpretations. *Neuroimage*, 52(3):1059–1069, 2010.

[45] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Hsu, and Kuansan Wang. An overview of Microsoft Academic Service (MAS) and Applications. In *Proceedings of the 24th International Conference on World Wide Web (WWW) Companion*, pages 243–246, 2015.

[46] Substance Abuse and Mental Health Services Administration. Drug Abuse Warning Network (DAWN). `https://www.samhsa.gov/data/data-we-collect/dawn-drug-abuse-warning-network`, 2011.

[47] TheoryInPractice. Github repository for "Overlapping and Robust Edge-Colored Clustering in Hypergraphs" (Crane et al. [19]). `https://github.com/TheoryInPractice/overlapping-ecc`, 2024.

[48] Nate Veldt. Optimal LP rounding and linear-time approximation algorithms for clustering edge-colored hypergraphs. In *International Conference on Machine Learning (ICML)*, pages 34924–34951. PMLR, 2023.

[49] Nate Veldt, David F Gleich, and Anthony Wirth. A correlation clustering framework for community detection. In *Proceedings of the 2018 World Wide Web Conference (WWW)*, pages 439–448, 2018.

[50] Dewan F Wahid and Elkafi Hassini. A literature review on correlation clustering: cross-disciplinary taxonomy with bibliometric analysis. In *Operations Research Forum*, volume 3, page 47. Springer, 2022.

[51] Qing Xiu, Kai Han, Jing Tang, Shuang Cui, and He Huang. Chromatic correlation clustering, revisited. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:26147–26159, 2022.

[52] Julian Yarkony, Alexander Ihler, and Charless C Fowlkes. Fast planar correlation clustering for image segmentation. In *Computer Vision: 12th European Conference on Computer Vision (ECCV)*, pages 568–581. Springer, 2012.

A Technical details and proofs for Local ECC deferred from Section 3.1

A.1 Formal proof of Theorem 3.1

Lemma A.1. *Algorithm 1 satisfies the following:*

(a) At any moment, (α, β) is feasible to the dual LP.

(b) At any moment, for all $v \in V$, $\alpha_v \leq \frac{1}{b_v+1} \sum_{e \in \delta(v)} \beta_{e,v}$.

(c) At termination, every mistake e under σ is tight, i.e., not loose.

Proof. Properties (a) and (b) were shown in Section 3.1; let us show Property (c). Let e be an arbitrary loose edge, and suppose towards contradiction that $c_e \notin \sigma(v)$ for some $v \in e$. Observe that, once an edge becomes tight, the algorithm never makes it loose again. Therefore, e was loose at the end of the iteration for v of the **for** loop. Then $\delta(v) \cap L$ contained e and therefore $c_e \in \chi(\delta(v) \cap L)$, leading to contradiction. □

Let ALG be the total weight of mistakes in the output of Algorithm 1 and OPT be the weight of an optimal solution.

Lemma A.2. *We have $\mathsf{ALG} \leq (b_{\mathsf{max}} + 1) \cdot \mathsf{OPT}$.*

Proof. By Properties (a) and (b) of Lemma A.1, we have

$$\mathsf{OPT} \geq \sum_{e \in E} \sum_{v \in e} \beta_{e,v} - \sum_{v \in V} b_v \alpha_v = \sum_{v \in V} \Big(\sum_{e \in \delta(v)} \beta_{e,v} - b_v \alpha_v \Big) \geq \sum_{v \in V} \Big(\frac{1}{b_v + 1} \sum_{e \in \delta(v)} \beta_{e,v} \Big),$$

where the first inequality is due to the (weak) LP duality. On the other hand, we have

$$\mathsf{ALG} \leq \sum_{e \in E \setminus L} w_e = \sum_{e \in E \setminus L} \sum_{v \in e} \beta_{e,v} \leq \sum_{e \in E} \sum_{v \in e} \beta_{e,v} = \sum_{v \in V} \sum_{e \in \delta(v)} \beta_{e,v},$$

where the first inequality is due to Property (c). The two inequalities together completes the proof. □

Now we need to show that the algorithm runs in polynomial time. In fact, the algorithm can be implemented to run in linear time. Recall that the size of H is $\sum_{v \in V} d_v$.

Lemma A.3. *Algorithm 1 can be implemented to run in $O(\sum_{v \in V} d_v)$ time.*

Proof. For each edge, let us maintain the "level" $\ell_e := \sum_{u \in e} \beta_{e,u}$. Consider an iteration for node $v \in V$. By enumerating $\delta(v)$, we can compute, for every color $c \in \chi(\delta(v))$, the "slack" $\mathsf{slack}(c) := \sum_{e \in \delta_c(v)} (w_e - \ell_e)$. Let $c^\star$ be the color $c \in \chi(\delta(v))$ that has the $(b_v + 1)$-st largest slack. Let $s^\star := \mathsf{slack}(c^\star)$. Note that we can identify $c^\star$ in $O(d_v)$ time using the algorithm of Blum et al. [13]. Once we found $c^\star$, for every color $c \in \chi(\delta(v))$, we increase $\sum_{e \in \delta_c(v)} \beta_{e,v}$ by $\min\{\mathsf{slack}(c), s^\star\}$ while maintaining the dual feasibility, i.e., for each edge $e \in E$, $\sum_{v \in e} \beta_{e,v} \leq w_e$ must be satisfied at the end. We then update $\{\ell_e\}_{e \in \delta_c(v)}$ accordingly. Note that a single iteration can be implemented to run in $O(d_v)$, completing the proof. □

Theorem 3.1. *Algorithm 1 is a* $(b_{\mathsf{local}} + 1)$*-approximation algorithm for* LOCAL ECC.

Proof. Immediate from Lemmas A.2 and A.3. □

A.2 Proof of Theorem 3.2

Theorem 3.2. *There is a sequence of instances of* LOCAL ECC *such that the ratio between a fractional solution and an optimal integral solution converges to* $b_{\mathsf{local}} + 1$.

Proof. Consider a hypergraph $H = (V, E)$ where $|E|$ is sufficiently large and $|V| = \binom{|E|}{b_{\mathsf{local}}+1}$. All edge weights are 1. In the hypergraph, each node is uniquely labeled by a subset S of E such that $|S| = b_{\mathsf{local}} + 1$. Let v_S denote the node whose label is S. For each $v_S \in V$, the set of edges that are incident to v_S is S. The colors of edges are distinct, i.e., $c_e \neq c_{e'}$ for all $e \neq e' \in E$.

We claim that, in any integral solution, the number of *satisfied* edges (i.e., edges that are not mistakes) does not exceed b_{local}. Suppose toward contradiction that there is a color assignment where at least $b_{\mathsf{local}} + 1$ edges are satisfied. Let S be any subset of the satisfied edges of size exactly $b_{\mathsf{local}} + 1$. Since the colors are all distinct, node v_S must be colored with (at least) $b_{\mathsf{local}} + 1$ colors, contradicting the budget constraint. Therefore, the total number of mistakes of any integral solution is at least $|E| - b_{\mathsf{local}}$.

Now consider the following fractional solution. Let x_{v_S, c_e} has value $\frac{b_{\mathsf{local}}}{b_{\mathsf{local}}+1}$ if $e \in S$, otherwise 0. Let $y_e = \frac{1}{b_{\mathsf{local}}+1}$ for all $e \in E$. Observe that the constructed solution is feasible to the LP and its cost is $\frac{|E|}{b_{\mathsf{local}}+1}$. The integrality gap is at least

$$\frac{|E| - b_{\mathsf{local}}}{\frac{|E|}{b_{\mathsf{local}}+1}} = \frac{(|E| - b_{\mathsf{local}})(b_{\mathsf{local}} + 1)}{|E|} = b_{\mathsf{local}} + 1 - \frac{b_{\mathsf{local}}(b_{\mathsf{local}} + 1)}{|E|},$$

which converges to $b_{\mathsf{local}} + 1$ as $|E|$ tends to infinity. □

A.3 Proof of Theorems 3.3 and 3.4

Theorem 3.3. *For any constant* $\epsilon > 0$*, it is* UGC*-hard to approximate* LOCAL ECC *within a factor of* $b_{\mathsf{local}} + 1 - \epsilon$.

Theorem 3.4. *For any* $b_{\mathsf{local}} \geq 2$ *and any constant* $\epsilon > 0$*, there does not exist a* $(b_{\mathsf{local}} - \epsilon)$*-approximation algorithm for* LOCAL ECC *unless* P = NP.

We say a hypergraph $H = (V, E)$ is k-uniform if, for all $e \in E$, $|e| = k$. Given a k-uniform hypergraph $H = (V, E)$, Ek-VERTEX-COVER asks to find a minimum-size subset $S \subseteq V$ of vertices, called a *vertex cover*, such that every hyperedge $e \in E$ intersects S, i.e., $e \cap S \neq \emptyset$ for each $e \in E$. Bansal and Khot [10] showed the following theorem.

Theorem A.4 (Bansal and Khot [10])**.** *For any* $k \geq 2$ *and any constant* $\epsilon > 0$*, there does not exists a* $(k - \epsilon)$*-approximation algorithm for* Ek-VERTEX-COVER *assuming the Unique Game Conjecture.*

Dinur, Guruswami, Khot, and Regev [23] showed the following theorem.

Theorem A.5 (Dinur et al. [23])**.** *For any* $k \geq 3$ *and any constant* $\epsilon > 0$*, there does not exists a* $(k - 1 - \epsilon)$*-approximation algorithm for* Ek-VERTEX-COVER *unless* P = NP.

Due to Theorems A.4 and A.5, it suffices to present an approximation-preserving reduction from Ek-VERTEX-COVER to LOCAL ECC with $b_{\mathsf{local}} := k - 1$.

Proof of Theorems 3.3 and 3.4. Given a k-uniform hypergraph $H = (W, F)$ as an input to Ek-VERTEX-COVER, let $H' := (V, E)$ be a hypergraph defined as follows:

- $V := \{v_f \mid f \in F\}$ and
- $E := \{e_w \mid w \in W\}$ where $e_w := \{v_f \mid f \ni w\}$.

Let $C := \{c_w \mid w \in W\}$ be a set of $|W|$ number of distinct colors. Let us then consider the input to LOCAL ECC where H' is given as the hypergraph, the color of e_w is c_w for every $e_w \in E$, and the budget b_{local} is set to $k - 1$.

For any vertex cover $S \subseteq W$ in H, let σ_S be the node coloring defined as follows: for every $v_f \in V$, $\sigma_S(v_f) := \{c_w \mid w \in f \setminus S\}$. Note that, for every $w \in W \setminus S$, e_w is satisfied by σ_S. Moreover, since $|f| = k$ and $f \cap S \neq \emptyset$, we can see that $|\sigma_S(v_f)| \leq k - 1 = b_{\mathsf{local}}$. This shows σ_S is indeed a feasible node coloring whose number of mistakes is at most $|S|$. We can therefore deduce that the minimum size of a vertex cover in the original input is at least the minimum number of mistakes in the reduced input.

For the other direction, let us now consider a feasible node coloring σ. Observe that, for any $v_f \in V$, at least one edge in $\delta(v_f)$ must be a mistake since $|\delta(v_f)| = |f| = k = b_{\mathsf{local}} + 1$ and the colors of E are distinct. This implies that, for every $f \in F$, there exists a vertex $w \in W$ such that $e_w \in E$ is a mistake due to σ in the reduced input. This shows that, given a feasible node coloring σ to the reduced input, we can construct in polynomial time a feasible vertex cover in the original input whose size is the same as the number of mistakes due to σ. Together with the above argument that the minimum number of mistakes in the reduced input is at most the minimum size of a vertex cover in the original input, this implies an approximation-preserving reduction from Ek-VERTEX-COVER to LOCAL ECC. □

A.4 Bicriteria algorithm for Local ECC

Theorem A.6. *For any $\epsilon \in (0, b_{\mathsf{local}}]$, there exists a $(1 + \epsilon, 1 + \frac{1}{b_{\mathsf{local}}}\lceil \frac{b_{\mathsf{local}}}{\epsilon} \rceil - \frac{1}{b_{\mathsf{local}}})$-approximation algorithm for* LOCAL ECC.

Proof. Let $\tau := \lceil \frac{b_{\mathsf{local}}}{\epsilon} \rceil - 1$. Consider the algorithm where the condition of **while** loop of Algorithm 1 is replaced by $|\chi(\delta(v) \cap L)| > b_{\mathsf{local}} + \tau$. Let σ be the assignment output by the modified algorithm. Observe first that the number of colors assigned to each v is at most $b_{\mathsf{local}} + \tau = b_{\mathsf{local}} \cdot (1 + \frac{1}{b_{\mathsf{local}}}\lceil \frac{b_{\mathsf{local}}}{\epsilon} \rceil - \frac{1}{b_{\mathsf{local}}})$. Observe that Properties (a) and (c) of Lemma A.1 still hold. Moreover, instead of Property (b), it is easy to show a stronger property that, at any moment, for all $v \in V$,

$$\alpha_v \leq \frac{1}{b_{\mathsf{local}} + \tau + 1} \sum_{e \in \delta(v)} \beta_{e,v}. \tag{1}$$

We therefore have

$$\begin{aligned}
\mathsf{ALG} &\leq \sum_{v \in V} \sum_{e \in \delta(v)} \beta_{e,v} \leq \sum_{v \in V} \frac{b_{\mathsf{local}} + \tau + 1}{\tau + 1} \left(\sum_{e \in \delta(v)} \beta_{e,v} - b_{\mathsf{local}} \alpha_v \right) \\
&= \sum_{v \in V} \left(1 + \frac{b_{\mathsf{local}}}{\tau + 1} \right) \left(\sum_{e \in \delta(v)} \beta_{e,v} - b_{\mathsf{local}} \alpha_v \right) \\
&\leq (1 + \epsilon) \sum_{v \in V} \left(\sum_{e \in \delta(v)} \beta_{e,v} - b_{\mathsf{local}} \alpha_v \right) \leq (1 + \epsilon)\mathsf{OPT},
\end{aligned}$$

where the first inequality follows from Property (c), the second from Equation (1), and the last from Property (a). □

Note that $1 + \frac{1}{b_{\mathsf{local}}}\lceil \frac{b_{\mathsf{local}}}{\epsilon} \rceil - \frac{1}{b_{\mathsf{local}}} < 1 + \frac{1}{\epsilon}$.

A.5 Discretized version of Algorithm 1

Algorithm 2 is a discretized version of Algorithm 1. Note that the proof Lemma A.3 is based on this discretized version.

Algorithm 2 Discretized primal-dual algorithm for LOCAL ECC

$\ell_e \leftarrow 0$ for all $e \in E$
$L \leftarrow \{e \in E \mid w_e > 0\}$
for $v \in V$ **do**
  if $|\chi(\delta(v) \cap L)| > b_v$ **then**
    $\mathsf{slack}(c) \leftarrow 0$ for all $c \in \chi(\delta(v) \cap L)$
    for $e \in \delta(v) \cap L$ **do**
      $\mathsf{slack}(c_e) \leftarrow \mathsf{slack}(c_e) + (w_e - \ell_e)$
    let $s^\star$ be the $(b_v + 1)$-st largest value in the (multi)set $\{\mathsf{slack}(c)\}_{c \in \chi(\delta(v) \cap L)}$
    for $c \in \chi(\delta(v) \cap L)$ **do**
      $\ell_e \leftarrow \ell_e + \frac{\min\{\mathsf{slack}(c), s^\star\}}{\mathsf{slack}(c)}(w_e - \ell_e)$ for all $e \in \delta_c(v) \cap L$
      if $\mathsf{slack}(c) \leq s^\star$ **then**
        $L \leftarrow L \setminus \delta_c(v)$
  $\sigma(v) \leftarrow \chi(\delta(v) \cap L)$
return σ

B Technical details and proofs for Robust ECC deferred from Section 3.2

B.1 Proposed algorithm for Robust ECC

In Section 3.2, we sketched our algorithm for ROBUST ECC. We present its pseudocode below.

Algorithm 3 Proposed algorithm for ROBUST ECC

$\alpha \leftarrow \mathbf{0}$; $\beta \leftarrow \mathbf{0}$; $\lambda \leftarrow 0$
$L \leftarrow \{e \in E \mid w_e > 0\}$
$R \leftarrow \{v \in V \mid |\chi(\delta(v) \cap L)| \geq 2\}$
while $|R| > b_{\mathsf{robust}}$ **do**
  increase λ and α_v and $\beta_{e,v}$ for $v \in R$ and $e \in \delta(v) \cap L$ in a way that the increase rate of λ and that of $\sum_{e \in \delta(v) \cap L} \beta_{e,v} - \alpha_v$ for each $v \in R$ are uniform and, for each $v \in R$, the increase rate of α_v and that of $\sum_{e \in \delta_c(v) \cap L} \beta_{e,v}$ for each $c \in \chi(\delta(v) \cap L)$ are uniform, until there exists e such that $\sum_{u \in e} \beta_{e,u} = w_e$
  if $\exists e$ $\sum_{u \in e} \beta_{e,u} = w_e$ **then** remove all such edges from L
  if $\exists v$ $|\chi(\delta(v) \cap L)| \leq 1$ **then** remove all such nodes from R
remove R from the hypergraph
for $v \notin R$ **do**
  if $|\chi(\delta(v) \cap L)| = 1$ **then**
    $\sigma(v) \leftarrow c$ where $c \in \chi(\delta(v) \cap L)$
  else
    $\sigma(v) \leftarrow c$ where c is an arbitrary color
return σ

B.2 Proof of Theorem 3.5

We have the following key lemma. Let us prove only Property (a), since Properties (b) and (c) can be seen from the same argument as the one for Lemma A.1.

Lemma B.1. *Algorithm 3 satisfies the following:*

(a) At any moment, (α, β, λ) is feasible to the dual LP.

(b) At any moment, for all $v \in V$, $\alpha_v \leq \frac{1}{2}\sum_{e\in\delta(v)} \beta_{e,v}$.

(c) At termination, every mistake e under σ is tight.

Proof of Property (a). Recall the two properties of the algorithm. Observe that the first set of dual constraints remain feasible due to Property (ii); the second set of constraints are satisfied since the algorithm stops increasing dual variables as soon as it discovers a tight edge; the third set of dual constraints are kept feasible due to Property (i). □

Let ALG be the total weight of mistakes in the output of Algorithm 3 and OPT be the weight of an optimal solution.

Lemma B.2. *We have* $\mathsf{ALG} \leq (2b_{\mathsf{robust}} + 2) \cdot \mathsf{OPT}$.

Proof. Observe that, if $|R| \leq b_{\mathsf{robust}}$ from the very beginning of the algorithm, the algorithm immediately terminates and incurs no weight. Let us thus assume that $|R| > b_{\mathsf{robust}}$ at the beginning.

Consider the timepoint when the algorithm terminates. Let R_0 denote the value of R at termination. Let $R' := \{v \in V \mid \sum_{e\in\delta(v)} \beta_{e,v} - \alpha_v = \lambda\}$. We claim that $R_0 \subsetneq R'$ and $|R'| > b_{\mathsf{robust}}$. (*Proof.* Right before the algorithm terminates, it removes some nodes from R. Consider the moment right before this removal. At this moment, every node v in R satisfies $\sum_{e\in\delta(v)} \beta_{e,v} - \alpha_v = \lambda$ and therefore is in R'. Note that R contains more than b_{robust} vertices at this moment, since otherwise the algorithm would have terminated earlier. Note that R_0 is the set resulting from the removal.) Let R'' be any set such that $R_0 \subseteq R'' \subsetneq R'$ with $|R''| = b_{\mathsf{robust}}$, and let w denote an arbitrary node in $R' \setminus R''$. We can then bound OPT from below as follows:

$$\begin{aligned}
\mathsf{OPT} &\geq \sum_{e\in E}\sum_{v\in e} \beta_{e,v} - \sum_{v\in V} \alpha_v - \lambda b_{\mathsf{robust}} = \sum_{e\in E}\sum_{v\in e} \beta_{e,v} - \sum_{v\in V} \alpha_v - \sum_{v\in R''} \Big(\sum_{e\in\delta(v)} \beta_{e,v} - \alpha_v \Big) \\
&= \sum_{v\in V\setminus R''} \Big(\sum_{e\in\delta(v)} \beta_{e,v} - \alpha_v \Big) \qquad (2) \\
&\geq \frac{1}{2} \sum_{v\in V\setminus R''} \sum_{e\in\delta(v)} \beta_{e,v}, \qquad (3)
\end{aligned}$$

where the first inequality is due to Property (a) and the second inequality is due to Property (b). Moreover, since $w \in R' \setminus R''$ and $|R''| = b_{\mathsf{robust}}$, we can find another lower bound on OPT from (2):

$$\begin{aligned}
\mathsf{OPT} &\geq \sum_{v\in V\setminus R''} \Big(\sum_{e\in\delta(v)} \beta_{e,v} - \alpha_v \Big) \geq \sum_{e\in\delta(w)} \beta_{e,w} - \alpha_w = \frac{1}{b_{\mathsf{robust}}} \sum_{v\in R''} \Big(\sum_{e\in\delta(v)} \beta_{e,v} - \alpha_v \Big) \\
&\geq \frac{1}{2b_{\mathsf{robust}}} \sum_{v\in R''} \sum_{e\in\delta(v)} \beta_{e,v}, \qquad (4)
\end{aligned}$$

where the equality follows from the fact that $w \in R'$ and the last inequality is again due to Property (b). Therefore, by Property (c), we have

$$\mathsf{ALG} \leq \sum_{v\in V}\sum_{e\in\delta(v)} \beta_{e,v} = \sum_{v\in R''}\sum_{e\in\delta(v)} \beta_{e,v} + \sum_{v\in V\setminus R''}\sum_{e\in\delta(v)} \beta_{e,v} \leq (2b_{\mathsf{robust}} + 2) \cdot \mathsf{OPT}, \qquad (5)$$

where the last inequality follows from (3) and (4). Note that, if $b_{\mathsf{robust}} = 0$, (5) immediately follows from (3) without (4). □

Lemma C.3 in Appendix C.3 shows that Algorithm 3 can be implemented to run in $O(|E| \sum_{v\in V} d_v)$ time.

Theorem 3.5. *Algorithm 3 is a $2(b_{\mathsf{robust}} + 1)$-approximation algorithm for* ROBUST ECC.

Proof. Immediate from Lemmas B.2 and C.3. □

B.3 Bicriteria algorithm for Robust ECC

Theorem B.3. *Suppose that $b \geq 1$. For any $\epsilon \in (0, 2b_{\mathsf{robust}}]$, there exists a $(2+\epsilon, 1+\frac{1}{b_{\mathsf{robust}}}\lceil\frac{2b_{\mathsf{robust}}}{\epsilon}\rceil - \frac{1}{b_{\mathsf{robust}}})$-approximation algorithm for* ROBUST ECC.

Proof. Let $\tau := \lceil\frac{2b_{\mathsf{robust}}}{\epsilon}\rceil - 1$. Consider the algorithm where the condition of the **while** loop in Algorithm 3 is replaced by $|R| > b_{\mathsf{robust}} + \tau$. It is clear that the number of removals is at most $b_{\mathsf{robust}} + \tau = b_{\mathsf{robust}} \cdot (1 + \frac{1}{b_{\mathsf{robust}}}\lceil\frac{2b_{\mathsf{robust}}}{\epsilon}\rceil - \frac{1}{b_{\mathsf{robust}}})$. Note also that the modified algorithm satisfies all the properties of Lemma B.1.

We basically follow the proof of Lemma B.2. Let R_0 be the set of nodes that are removed from the hypergraph H. Observe that $|R_0| \leq b_{\mathsf{robust}} + \tau$ from the construction. Let σ be the color assignment of $V \setminus R_0$ output by the algorithm and let E_m be the set of mistakes under σ. We have

$$\sum_{e\in E_m} w_e \leq \sum_{v\in V}\sum_{e\in\delta(v)} \beta_{e,v} \leq 2\sum_{v\in V}\left(\sum_{e\in\delta(v)}\beta_{e,v} - \alpha_v\right), \tag{6}$$

where the first inequality follows from Property (c) and the second from Property (b). Let $R' := \{v \in V \mid \sum_{e\in\delta(v)}\beta_{e,v} - \alpha_v = \lambda\}$. Observe that $R_0 \subsetneq R'$ and $|R'| > b_{\mathsf{robust}} + \tau$. Let R'' be a subset of R' such that $R \subseteq R''$ with $|R''| = b_{\mathsf{robust}} + \tau$. We then have

$$\begin{aligned}
\mathsf{OPT} &\geq \sum_{e\in E}\sum_{v\in e}\beta_{e,v} - \sum_{v\in V}\alpha_v - \lambda b_{\mathsf{robust}} \\
&= \sum_{v\in V}\left(\sum_{e\in\delta(v)}\beta_{e,v} - \alpha_v\right) - \frac{b_{\mathsf{robust}}}{b_{\mathsf{robust}}+\tau}\sum_{v\in R''}\left(\sum_{e\in\delta(v)}\beta_{e,v} - \alpha_v\right) &(7)\\
&= \sum_{v\in V\setminus R''}\left(\sum_{e\in\delta(v)}\beta_{e,v} - \alpha_v\right) + \frac{\tau}{b_{\mathsf{robust}}+\tau}\sum_{v\in R''}\left(\sum_{e\in\delta(v)}\beta_{e,v} - \alpha_v\right). &(8)
\end{aligned}$$

Let w denote any node in $R' \setminus R''$. We give a lower bound on the first term of (8) as follows:

$$\sum_{v\in V\setminus R''}\left(\sum_{e\in\delta(v)}\beta_{e,v} - \alpha_v\right) \geq \sum_{e\in\delta(w)}\beta_{e,w} - \alpha_w = \frac{1}{b_{\mathsf{robust}}+\tau}\sum_{v\in R''}\left(\sum_{e\in\delta(v)}\beta_{e,v} - \alpha_v\right). \tag{9}$$

Therefore by plugging (9) into (8), we have

$$\mathsf{OPT} \geq \frac{\tau+1}{b_{\mathsf{robust}}+\tau}\sum_{v\in R''}\left(\sum_{e\in\delta(v)}\beta_{e,v} - \alpha_v\right). \tag{10}$$

We then have

$$\begin{aligned}
&\sum_{v\in V}\left(\sum_{e\in\delta(v)}\beta_{e,v} - \alpha_v\right) \\
&= \left[\sum_{v\in V}\left(\sum_{e\in\delta(v)}\beta_{e,v} - \alpha_v\right) - \frac{b_{\mathsf{robust}}}{b_{\mathsf{robust}}+\tau}\sum_{v\in R''}\left(\sum_{e\in\delta(v)}\beta_{e,v} - \alpha_v\right)\right] \\
&\qquad + \frac{b_{\mathsf{robust}}}{b_{\mathsf{robust}}+\tau}\sum_{v\in R''}\left(\sum_{e\in\delta(v)}\beta_{e,v} - \alpha_v\right) \\
&\leq \left(1 + \frac{b_{\mathsf{robust}}}{\tau+1}\right)\mathsf{OPT} \leq \left(1+\frac{\epsilon}{2}\right)\mathsf{OPT},
\end{aligned}$$

where the first inequality follows from (7) and (10). Combining this inequality with (6) proves the theorem. □

Recall that Crane et al. [19] gave an LP-rounding bicriteria $(2+\epsilon, 2+\frac{4}{\epsilon})$-approximation algorithm for ROBUST ECC for $\epsilon > 0$. Note that this algorithm provides an better performance guarantee.

B.4 Discretized version of Algorithm 3

Algorithm 4 is a discretized version of Algorithm 3. Note that the proof Lemma C.3 is based on this discretized version.

Algorithm 4 Discretized primal-dual algorithm for ROBUST ECC

$\ell_e \leftarrow 0$ for all $e \in E$
$L \leftarrow \{e \in E \mid w_e > 0\}$
$R \leftarrow \{v \in V \mid |\chi(\delta(v) \cap L)| \geq 2\}$
while $|R| > b_{\mathsf{robust}}$ **do**
  $\mathsf{rate}(e) \leftarrow 0$ for all $e \in L$
  for $v \in R$ **do**
    for $c \in \chi(\delta(v) \cap L)$ **do**
      $\mathsf{rate}(e) \leftarrow \mathsf{rate}(e) + \frac{1}{|\chi(\delta(v)\cap L)|-1} \cdot \frac{1}{|\delta_c(v)\cap L|}$ for all $e \in \delta_c(v) \cap L$
  $t_{\mathsf{tighten}}(e) \leftarrow \frac{w_e - \ell_e}{\mathsf{rate}(e)}$ for all $e \in L$
  $t^\star \leftarrow \min_{e \in L} t_{\mathsf{tighten}}(e)$
  for $e \in L$ **do**
    $\ell_e \leftarrow \ell_e + t^\star \cdot \mathsf{rate}(e)$
    if $\ell_e = w_e$ **then**
      $L \leftarrow L \setminus \{e\}$
      remove all $v \in e$ from R such that $|\chi(\delta(v) \cap L)| \leq 1$
remove R from the hypergraph
for $v \notin R$ **do**
  if $|\chi(\delta(v) \cap L)| = 1$ **then**
    $\sigma(v) \leftarrow c$ where $c \in \chi(\delta(v) \cap L)$
  else
    $\sigma(v) \leftarrow c$ where c is an arbitrary color
return σ

C Technical details and proofs for Global ECC deferred from Section 3.2

C.1 Primal and dual LP

Following is the LP relaxation, where $z_v \in \mathbb{Z}_{\geq 0}$ indicates the number of additional colors budget assigned to v, i.e., $z_v + 1$ number of colors is assigned to v.

$$\begin{aligned}
\min\ & \textstyle\sum_{e \in E} w_e y_e \\
\text{s.t. } & \textstyle\sum_{c \in C} x_{v,c} \leq z_v + 1, && \forall v \in V, \\
& x_{v,c_e} + y_e \geq 1, && \forall e \in E, v \in e, \\
& \textstyle\sum_{v \in V} z_v \leq b_{\mathsf{global}}, \\
& x_{v,c} \geq 0, && \forall v \in V, c \in C, \\
& y_e \geq 0, && \forall e \in E, \\
& z_v \geq 0, && \forall v \in V.
\end{aligned}$$

Following is the dual of this LP.

$$\begin{aligned}
\max\ & \textstyle\sum_{e \in E, v \in e} \beta_{e,v} - \sum_{v \in V} \alpha_v - \lambda b_{\mathsf{global}} \\
\text{s.t. } & \textstyle\sum_{e \in \delta_c(v)} \beta_{e,v} \leq \alpha_v, && \forall v \in V, c \in C, \\
& \textstyle\sum_{v \in e} \beta_{e,v} \leq w_e, && \forall e \in E, \\
& \alpha_v \leq \lambda, && \forall v \in V, \\
& \alpha_v \geq 0, && \forall v \in V, \\
& \beta_{e,v} \geq 0, && \forall e \in E, v \in e, \\
& \lambda \geq 0.
\end{aligned}$$

C.2 Proposed algorithm for Global ECC

Let us now present our algorithm. Similarly, we consider the problem where each edge has an associated weight w_e.

The algorithm shares many similarity with the algorithm for ROBUST ECC. It maintains a dual feasible solution (α, β, λ), starting from $(\mathbf{0}, \mathbf{0}, 0)$, the set L of loose edges, and the set R of nodes with at least two incident loose edges of distinct colors. It also simultaneously increases λ and (α, β) associated with the nodes in R. Intuitively, R is the set of nodes that we will assign (possibly) more than one color.

As the dual formulation differs, the way the algorithm increases the dual solution slightly varies. The following properties will hold:

(i) λ and α_v for each $v \in R$ increas at the same rate.

(ii) For all $v \in R$, α_v and $\sum_{e \in \delta_c(v) \cap L} \beta_{e,v}$ for each $c \in \chi(\delta(v) \cap L)$ increase at the same rate.

Observe that these properties can be easily ensured.

The algorithm increases the dual solution until $\sum_{v \in R}(|\chi(\delta(v) \cap L)| - 1)$ becomes at most b_{global}, and once the algorithm reaches this point, it assigns every node $v \in V$ the color in $\chi(\delta(v) \cap L)$. It is clear that the returned node coloring is feasible. As before, if $\chi(\delta(v) \cap L) = \emptyset$, an arbitrary color can be assigned without affecting the theoretical guarantee; in practical implementation, we could employ heuristics for marginal improvement. See Algorithm 5 for a detailed pseudocode. The full discretized version of the algorithm is presented in Appendix C.6.

Algorithm 5 Proposed algorithm for GLOBAL ECC

$\alpha \leftarrow \mathbf{0}$; $\beta \leftarrow \mathbf{0}$; $\lambda \leftarrow 0$
$L \leftarrow \{e \in E \mid w_e > 0\}$
$R \leftarrow \{v \in V \mid |\chi(\delta(v) \cap L)| \geq 2\}$
while $\sum_{v \in R}(|\chi(\delta(v) \cap L)| - 1) > b_{\mathsf{global}}$ **do**
  increase λ and α_v and $\beta_{e,v}$ for $v \in R$ and $e \in \delta(v) \cap L$ in a way that the increase rate of λ and that of α_v for each $v \in R$ are uniform and, for each $v \in R$, the increase rate of α_v and that of $\sum_{e \in \delta_c(v) \cap L} \beta_{e,v}$ for each $c \in \chi(\delta(v) \cap L)$ are uniform, until there exists e such that $\sum_{u \in e} \beta_{e,u} = w_e$
  if $\exists e$ $\sum_{u \in e} \beta_{e,u} = w_e$ **then** remove all such edges from L
  if $\exists v$ $|\chi(\delta(v) \cap L)| \leq 1$ **then** remove all such nodes from R
for $v \in V$ **do**
  if $|\chi(\delta(v) \cap L)| \geq 1$ **then**
    $\sigma(v) \leftarrow \chi(\delta(v) \cap L)$
  else
    $\sigma(v) \leftarrow \{c\}$ where c is an arbitrary color
return σ

C.3 Proof of Theorem 3.6

We have the following key lemma. Let us prove only Property (a), since Properties (b) through (d) can be seen from the same argument as the one for Lemma A.1.

Lemma C.1. *Algorithm 5 satisfies the following:*

(a) At any moment, (α, β, λ) is feasible to the dual LP.

(b) At any moment, for all $v \in V$, $\alpha_v \leq \frac{1}{2} \sum_{e \in \delta(v)} \beta_{e,v}$.

(c) At any moment, for all $v \in R$, $\alpha_v \leq \frac{1}{|\chi(\delta(v) \cap L)|} \sum_{e \in \delta(v)} \beta_{e,v}$.

(d) At termination, every mistake e under σ is tight.

Proof of Property (a). Recall the two properties of the algorithm. Observe that the first set of dual constraints remain feasible due to Property (ii); the second set of constraints are satisfied since the

algorithm stops increasing dual variables as soon as it discovers a tight edge; the third set of dual constraints are kept feasible due to Property (i). □

Given $L \subseteq E$, let $\kappa_L(v) := |\chi(\delta(v) \cap L)|$ for $v \in V$. Let $\mathsf{budget}_L(S) := \sum_{v \in R}(\kappa_L(v) - 1)$ for $S \subseteq V$. Let ALG be the total weight of mistakes in the output of Algorithm 5 and OPT be the weight of an optimal solution.

Lemma C.2. *We have* $\mathsf{ALG} \leq (2b_{\mathsf{global}} + 2) \cdot \mathsf{OPT}$.

Proof. Observe that, if $\mathsf{budget}_L(R) \leq b_{\mathsf{global}}$ from the very beginning of the algorithm, the algorithm immediately terminates and incurs no weight. Let us thus assume that $\mathsf{budget}_L(R) > b_{\mathsf{global}}$ at the beginning.

Consider the last iteration of **while** loop of the algorithm. In this iteration, the algorithm removes some edges from L (and possibly removes some vertices). Consider the moment right before the removal of edges. Let R' and L', respectively, denote the value of R and L at this moment. Let $b' := \mathsf{budget}_{L'}(R')$. Note that $b' > b_{\mathsf{global}}$, since otherwise the algorithm would have terminated earlier. Moreover, at this moment–or at the termination–we have $\alpha_v = \lambda$, for every $v \in R'$.

We then bound OPT from below as follows:

$$\begin{aligned}
\mathsf{OPT} &\geq \sum_{e \in E}\sum_{v \in e}\beta_{e,v} - \sum_{v \in V}\alpha_v - \lambda b_{\mathsf{global}} = \sum_{v \in V}\left(\sum_{e \in \delta(v)}\beta_{e,v} - \alpha_v\right) - \frac{b_{\mathsf{global}}}{b'}\sum_{v \in R'}(\kappa_{L'}(v) - 1)\alpha_v \\
&= \sum_{v \in V \setminus R'}\left(\sum_{e \in \delta(v)}\beta_{e,v} - \alpha_v\right) + \sum_{v \in R'}\left(\sum_{e \in \delta(v)}\beta_{e,v} - \left(1 + \frac{b_{\mathsf{global}}}{b'}(\kappa_{L'}(v) - 1)\right)\alpha_v\right) \\
&\geq \sum_{v \in V \setminus R'}\left(\frac{1}{2}\sum_{e \in \delta(v)}\beta_{e,v}\right) + \sum_{v \in R'}\left(\left(1 - \frac{1}{\kappa_{L'}(v)}\left(1 + \frac{b_{\mathsf{global}}}{b'}(\kappa_{L'}(v) - 1)\right)\right)\sum_{e \in \delta(v)}\beta_{e,v}\right) \\
&= \sum_{v \in V \setminus R'}\left(\frac{1}{2}\sum_{e \in \delta(v)}\beta_{e,v}\right) + \sum_{v \in R'}\left(\left(1 - \frac{1}{\kappa_{L'}(v)}\right)\left(1 - \frac{b_{\mathsf{global}}}{b'}\right)\sum_{e \in \delta(v)}\beta_{e,v}\right) \\
&\geq \sum_{v \in V \setminus R'}\left(\frac{1}{2}\sum_{e \in \delta(v)}\beta_{e,v}\right) + \sum_{v \in R'}\left(\frac{1}{2}\left(1 - \frac{b_{\mathsf{global}}}{b'}\right)\sum_{e \in \delta(v)}\beta_{e,v}\right) \\
&\geq \frac{1}{2}\left(1 - \frac{b_{\mathsf{global}}}{b'}\right)\sum_{v \in V}\sum_{e \in \delta(v)}\beta_{e,v} \qquad (11)
\end{aligned}$$

where the first inequality is due to Property (a), the first equality comes from $\lambda = \alpha_v$ for every $v \in R'$, the second inequality is due to Property (b) and (c), and the second to last inequality comes from $\kappa_{L'}(v) \geq 2$ for every $v \in R'$. Therefore, by Property (d), we have

$$\mathsf{ALG} \leq \sum_{v \in V}\sum_{e \in \delta(v)}\beta_{e,v} \leq (2b_{\mathsf{global}} + 2) \cdot \mathsf{OPT},$$

where the last inequality comes from $b' \geq b_{\mathsf{global}} + 1$, which implies $1 - \frac{b_{\mathsf{global}}}{b'} \geq 1 - \frac{b_{\mathsf{global}}}{b_{\mathsf{global}}+1} = \frac{1}{b_{\mathsf{global}}+1}$. □

Lemma C.3. *Both Algorithm 3 and Algorithm 5 can be implemented to run in* $O(|E|\sum_{v \in V} d_v)$ *time.*

Proof. It suffices to show that we can decide in $O(\sum_{v \in V} d_v)$ time which edge becomes tight, as well as the increment of the dual variables. By iterating each node v and its incident edges $\delta(v)$, we can compute the increase rates of α_v and $\{\beta_{e,v}\}_{e \in \delta(v)}$. From this, we can obtain the increase rate of the "level" $\ell_e := \sum_{u \in e}\beta_{e,u}$ for each $e \in L$. Let $\mathsf{rate}(e)$ denote this increase rate. As $e \in L$ will become tight in $t_{\mathsf{tighten}}(e) := \frac{w_e - \sum_{u \in e}\beta_{e,u}}{\mathsf{rate}(e)}$ time, we can determine the edge that will become tight the earliest. We can also compute the increment of the dual variables accordingly. □

Theorem 3.6. *Algorithm 5 is a* $2(b_{\mathsf{global}}+1)$*-approximation algorithm for* GLOBAL ECC.

Proof. Immediate from Lemmas C.2 and C.3. □

We note that our approach yields a true (non-bicriteria) approximation algorithm. This shows that the LP, which is equivalent to that of Crane et al. [19], has an integrality gap of $O(b_{\mathsf{global}})$.

C.4 Proof of Theorem 3.8

Theorem 3.8. *The integrality gap of the LP for* GLOBAL ECC *is at least* $b_{\mathsf{global}}+1$.

Proof. We construct an instance similar to the one used in the proof of Theorem 3.7. Consider a hypergraph $H=(V=\{v_1,\cdots,v_{b_{\mathsf{global}}+1}\},E=\{e_1,e_2\})$ where $e_1=e_2=V$, $w_{e_1}=w_{e_2}=1$, and $c_{e_1}\neq c_{e_2}$. Any integral solution incurs at least 1 since at least one node is assigned one color and at least one edge cannot be satisfied. However, consider the solution given by $z_v=\frac{b_{\mathsf{global}}}{b_{\mathsf{global}}+1}$, $x_{v,c_{e_1}}=x_{v,c_{e_2}}=\frac{2b_{\mathsf{global}}+1}{2(b_{\mathsf{global}}+1)}$ for all $v\in V$ and $y_{e_1}=y_{e_2}=\frac{1}{2(b_{\mathsf{global}}+1)}$. This solution is feasible and the cost is $\frac{1}{b_{\mathsf{global}}+1}$. □

C.5 Bicriteria algorithm for GLOBAL ECC

Theorem C.4. *Suppose that* $b\geq 1$. *For any* $\epsilon\in(0,2b_{\mathsf{global}}]$*, there exists a* $(2+\epsilon,1+\frac{1}{b_{\mathsf{global}}}\lceil\frac{2b_{\mathsf{global}}}{\epsilon}\rceil-\frac{1}{b_{\mathsf{global}}})$*-approximation algorithm for* GLOBAL ECC.

Proof. Let $\tau:=\lceil\frac{2b_{\mathsf{global}}}{\epsilon}\rceil-1$. Consider the algorithm where the condition of the **while** loop in Algorithm 5 is replaced by $\sum_{v\in R}(|\chi(\delta(v)\cap L)|-1)>b_{\mathsf{global}}+\tau$. It is clear that

$$\sum_{v\in V}\max\{|\sigma(v)|-1,0\}\leq b_{\mathsf{global}}+\tau=b_{\mathsf{global}}\cdot\left(1+\frac{1}{b}\left\lceil\frac{2b_{\mathsf{global}}}{\epsilon}\right\rceil-\frac{1}{b_{\mathsf{global}}}\right).$$

Note also that the modified algorithm satisfies all the properties of Lemma C.1.

We basically follow the proof of Lemma C.2. With the same definition of R' and L', we have $b':=\mathsf{budget}_{L'}(R')>b_{\mathsf{global}}+\tau$. Note that $1-\frac{b_{\mathsf{global}}}{b'}\geq 1-\frac{b_{\mathsf{global}}}{b_{\mathsf{global}}+\tau+1}=\frac{\tau+1}{b_{\mathsf{global}}+\tau+1}$. Together with equation (11),

$$\mathsf{ALG}\leq\sum_{v\in V}\sum_{e\in\delta(v)}\beta_{e,v}\leq(2+\frac{2b_{\mathsf{global}}}{\tau+1})\cdot\mathsf{OPT}\leq(2+\epsilon)\cdot\mathsf{OPT},$$

where the last inequality comes from $\tau\geq\frac{2b_{\mathsf{global}}}{\epsilon}-1$. □

Recall that Crane et al. [19] gave an LP-rounding bicriteria $(b_{\mathsf{global}}+3+\epsilon,1+\frac{b_{\mathsf{global}}+2}{\epsilon})$-approximation algorithm for GLOBAL ECC for $\epsilon>0$. Both approximation factor and violation factor of Crane et al.'s algorithm are linear in b_{global}. They raised an open question whether we can give a bicriteria approximation algorithm for GLOBAL ECC with both factor being constant (or give a hardness result). Observe $1+\frac{1}{b}\lceil\frac{2b_{\mathsf{global}}}{\epsilon}\rceil-\frac{1}{b_{\mathsf{global}}}<1+\frac{2}{\epsilon}$. Our bicriteria algorithm satisfies the condition, answering the open question of Crane et al. [19].

C.6 Discretized version of Algorithm 5

Algorithm 6 is a discretized version of Algorithm 5. Note that the proof Lemma C.3 is based on this discretized version.

Algorithm 6 Discretized primal-dual algorithm for GLOBAL ECC

$\ell_e \leftarrow 0$ for all $e \in E$
$L \leftarrow \{e \in E \mid w_e > 0\}$
$R \leftarrow \{v \in V \mid |\chi(\delta(v) \cap L)| \geq 2\}$
while $\sum_{v \in R}(|\chi(\delta(v) \cap L)| - 1) > b_{\mathsf{global}}$ **do**
  $\mathsf{rate}(e) \leftarrow 0$ for all $e \in L$
  for $v \in R$ **do**
    for $c \in \chi(\delta(v) \cap L)$ **do**
      $\mathsf{rate}(e) \leftarrow \mathsf{rate}(e) + \frac{1}{|\delta_c(v) \cap L|}$ for all $e \in \delta_c(v) \cap L$
  $t_{\mathsf{tighten}}(e) \leftarrow \frac{w_e - \ell_e}{\mathsf{rate}(e)}$ for all $e \in L$
  $t^\star \leftarrow \min_{e \in L} t_{\mathsf{tighten}}(e)$
  for $e \in L$ **do**
    $\ell_e \leftarrow \ell_e + t^\star \cdot \mathsf{rate}(e)$
    if $\ell_e = w_e$ **then**
      $L \leftarrow L \setminus \{e\}$
      remove all $v \in e$ from R such that $|\chi(\delta(v) \cap L)| \leq 1$
$\sigma(v) \leftarrow \chi(\delta(v) \cap L)$ for all $v \in V$
return σ

D Dataset description

The benchmark data of Crane et al. [19] contains six datasets. Cooking [34, 4], described in Section 1, is a hypergraph whose nodes correspond to food ingredients, edges represent recipes, and edge colors indicate cuisines. Brain [44, 21] contains the relation between brain regions: there are two types of relations, coactivation and connectivity, encoded by colors. The nodes corresponds to brain regions, and colored edges represent relations between them. In MAG-10 [45, 4], each node corresponds to a researcher, and an edge indicates the author set of a published paper. Its color represents the publication venue (e.g., NeurIPS). DAWN [46, 4] is a dataset on the relation between drug use and emergency room (ER) visit disposition such as "discharged", "surgery", and "transferred". Each node corresponds to a drug, an edge corresponds to the combination of drugs taken by an ER patient, and colors represent the visit disposition. In Walmart [33, 4], each node represents a product, an edge indicates a set of products purchased together, and colors are "trip type" labels determined by Walmart. Lastly in Trivago [38, 18], nodes are vacation rental properties and an edge represents the set of rental properties clicked during a single browsing session of a single user. Colors correspond to the countries where the browsing sessions happen.

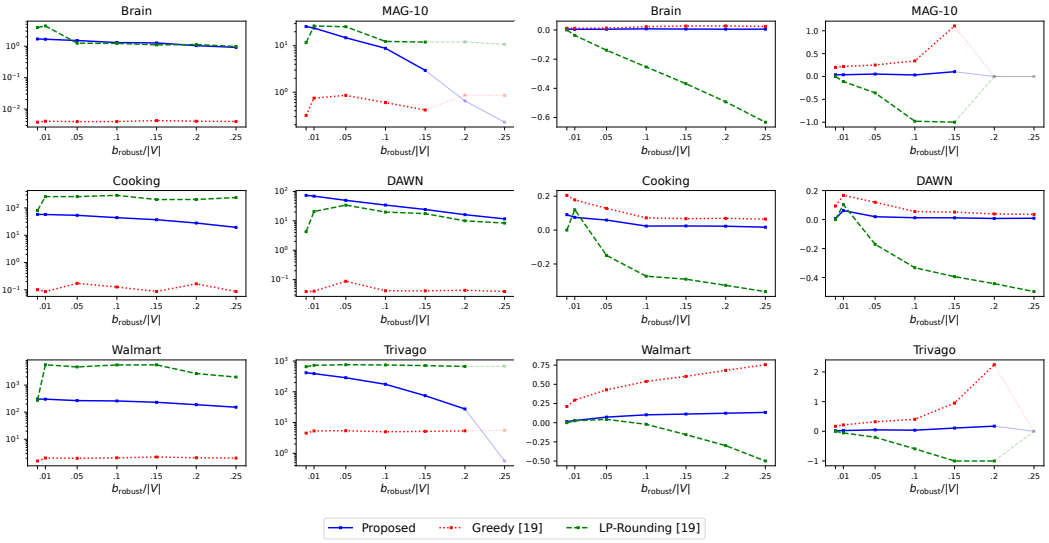
E Tables and figures deferred from Section 4

Table 2: Average running times of each dataset (in seconds): LOCAL ECC. Values in parentheses are averages excluding trivial instances.

	Proposed		Greedy		LP-rounding	
Brain	0.023	(0.120)	0.007	(0.028)	0.743	(3.739)
MAG-10	0.142	(0.134)	0.587	(0.554)	10.413	(10.677)
Cooking	0.032	(0.035)	0.099	(0.103)	39.702	(44.916)
DAWN	0.016	(0.019)	0.040	(0.040)	3.948	(4.658)
Walmart	0.190	(0.190)	1.443	(1.443)	145.427	(145.427)
Trivago	0.323	(0.313)	3.709	(3.608)	678.585	(677.036)

Table 3: Average running times of each dataset (in seconds): ROBUST ECC. Values in parentheses are averages excluding trivial instances.

	Proposed		Greedy		LP-rounding	
Brain	1.345	(1.345)	0.005	(0.005)	2.007	(2.007)
MAG-10	11.056	(15.303)	0.666	(0.588)	15.871	(17.660)
Cooking	42.571	(42.571)	0.118	(0.118)	220.107	(220.107)
DAWN	39.905	(39.905)	0.048	(0.048)	16.464	(16.464)
Walmart	243.881	(243.881)	1.995	(1.995)	3766.539	(3766.539)
Trivago	195.337	(227.799)	5.210	(5.144)	705.323	(709.378)

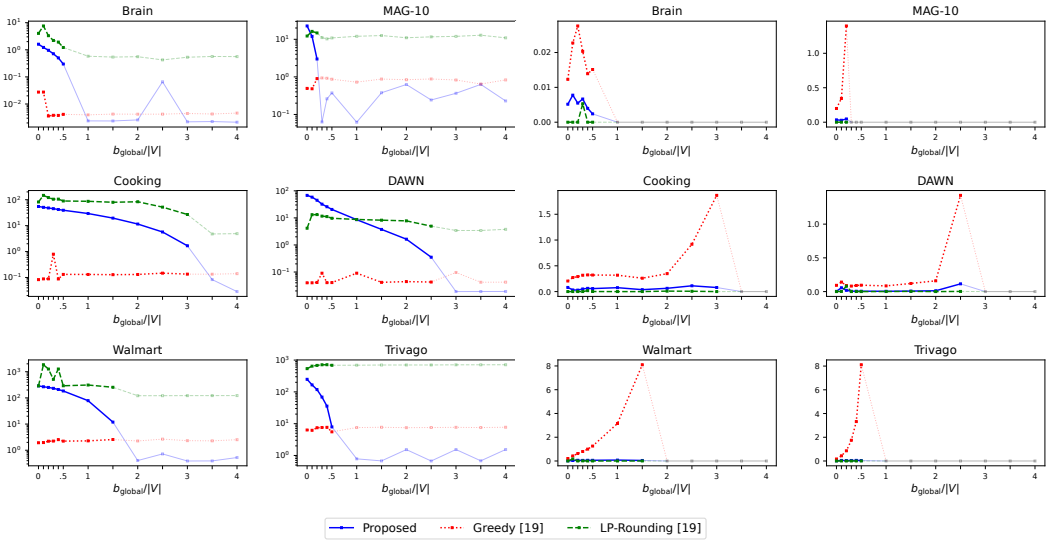


(a) Running times ($\log_{10}$ scaled, in seconds) (b) Relative error estimates

Figure 2: (a) Running times (in seconds, log scale) and (b) relative error estimates of the ROBUST ECC algorithms. Empty square markers denote trivial instances.

Table 4: Average running times of each dataset (in seconds): GLOBAL ECC. Values in parentheses are averages excluding trivial instances.

	Proposed		Greedy		LP-rounding	
Brain	0.415	(0.885)	0.008	(0.012)	1.849	(3.381)
MAG-10	3.143	(12.541)	0.787	(0.625)	12.194	(14.418)
Cooking	26.755	(31.609)	0.171	(0.177)	75.953	(88.883)
DAWN	20.345	(26.443)	0.054	(0.052)	7.944	(9.263)
Walmart	117.046	(189.895)	2.310	(2.247)	511.011	(754.344)
Trivago	50.162	(107.449)	7.234	(6.795)	684.776	(662.595)



(a) Running times ($\log_{10}$ scaled, in seconds) (b) Relative error estimates

Figure 3: (a) Running times (in seconds, log scale) and (b) relative error estimates of the GLOBAL ECC algorithms. Empty square markers denote trivial instances.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract provides a high-level overview of our contributions, and the introduction (Section 1) elaborates them.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Scalability and computational efficiency of the proposed algorithms are heavily discussed throughout the paper, including Sections 4.2 and 4.3 and Appendices A, B, and C, as they are one of the key contributions of this paper. Other limitations and possible future directions of research related to them are also discussed in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Theorems are stated in Section 3; some proofs are sketched in Section 3.1, and the full proofs are presented in Appendices A, B, and C.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Detailed description of the algorithms are presented in Section 3.1 and Appendices A.5, B.1, B.4, C.2, and C.6, and the experimental setting and details can be found in Sections 4.1, 4.2, and 4.3. Moreover, the code is available as a supplemental material, and the data used by the experiments are publicly available [47]. One also needs to obtain a license for Gurobi LP solver; academic licenses are available.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code is available as a supplemental material. The data used by the experiments are publicly available [47].

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All experimental setting and details can be found in Sections 4.1, 4.2, and 4.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Neither the proposed algorithms nor the data used in the experiments involve any randomness in their execution or preparation. Due to this deterministic nature of the experiments, statistical errors do not need to be reported; however, the guideline explicitly states that NA means that the paper does not include experiments at all, which does not

apply to this paper. Although not directly related to statistical errors, we do report the approximation errors of the algorithms as a *relative error estimate*—a normalized, estimated error of the algorithms' output (see Section 4.1 for its definition).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the information on the experiment setup in Section 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: This paper presents foundational research which is generic and does not directly lead to any societal impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly credited the original owners of the code and data used in this paper [47, 4, 44, 21, 45, 34, 46, 33, 38, 18]. We also obtained a valid academic license of Gurobi and used it.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.