Online Correlation Clustering: Simultaneously Optimizing All ℓ_p-Norms

Sami Davies
Department of EECS, UC Berkeley, CA, USA

Benjamin Moseley
Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA, USA

Heather Newman
Department of Computer Science, Vassar College, Poughkeepsie, NY, USA

Abstract

The ℓ_p-norm objectives for correlation clustering present a fundamental trade-off between minimizing total disagreements (the ℓ_1-norm) and ensuring fairness to individual nodes (the ℓ_∞-norm). Surprisingly, in the offline setting it is possible to simultaneously approximate all ℓ_p-norms with a single clustering. Can this powerful guarantee be achieved in an online setting? This paper provides the first affirmative answer. We present a single algorithm for the online-with-a-sample (AOS) model that, given a small constant fraction of the input as a sample, produces one clustering that is *simultaneously* $O(\log^4 n)$-competitive for all ℓ_p-norms with high probability, $O(\log n)$-competitive for the ℓ_∞-norm with high probability, and $O(1)$-competitive for the ℓ_1-norm in expectation. This work successfully translates the offline "all-norm" guarantee to the online world.

Our setting is motivated by a new hardness result that demonstrates a fundamental separation between these objectives in the standard random-order (RO) online model. Namely, while the ℓ_1-norm is trivially $O(1)$-approximable in the RO model, we prove that any algorithm in the RO model for the fairness-promoting ℓ_∞-norm must have a competitive ratio of at least $\Omega(n^{1/3})$. This highlights the necessity of a different beyond-worst-case model. We complement our algorithm with lower bounds, showing our competitive ratios for the ℓ_1- and ℓ_∞- norms are nearly tight in the AOS model.

2012 ACM Subject Classification Theory of computation → Online algorithms

Keywords and phrases Online algorithms, correlation clustering, all-norms objective, beyond-worst-case analysis

Digital Object Identifier 10.4230/LIPIcs.ICALP.2026.73

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2510.15076

1 Introduction

Clustering is a fundamental task in unsupervised learning. In this paper, we study *correlation clustering*, where the goal is to partition a set of n items based on pairwise similarity $(+)$ and dissimilarity $(-)$ labels. Given a *complete* graph where each edge has a label of positive or negative, a clustering is a partition of the n vertices. A positive edge is a *disagreement* if its endpoints are in different clusters, and a negative edge is a disagreement if its endpoints are in the same cluster. The goal is to find a partition that minimizes some function of these disagreements.

The objective function controls the balance between total cost and individual equity. The most well-studied among the objectives for correlation clustering is minimizing the ℓ_1-norm of the *disagreement vector*, which is the vector of length n where the ith coordinate indicates the number of disagreeing edges incident to node i. This objective is equivalent to minimizing

53rd International Colloquium on Automata, Languages, and Programming (ICALP 2026).
Editors: Sayan Bhattacharya, Danupon Nanongkai, Michael Benedikt, and Gabriele Puppis;
Article No. 73; pp. 73:1–73:23


Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the total number of disagreements. However, the ℓ_1-norm can be highly "unfair," leaving some nodes with a disproportionately large number of incident disagreements. On the other hand, the ℓ_∞-norm is a fair objective, as it minimizes the maximum number of disagreements incident to any single node. The ℓ_2-norm and other finite ℓ_p-norms, for $p \in (1, \infty)$, interpolate between the two, allowing one to balance total error against local equity. It is natural to wonder whether it is possible to study correlation clustering (with any of these objectives) in an online setting, when nodes arrive sequentially and must be irrevocably assigned to a cluster.

However, the lower bounds for correlation clustering are pessimistic in the standard online model, where nodes arrive one at a time in adversarial order – a simple construction due to Mathieu, Sankur, and Schudy shows it is impossible to achieve any sublinear competitive ratio for any ℓ_p-norm objective while maintaining cluster consistency [39]. To circumvent this, prior works have explored relaxed, *beyond-worst-case* models, but these have focused exclusively on the ℓ_1-norm. These models include allowing limited recourse (changing a node's cluster assignment) [19] or, as we study here, providing the algorithm with a small, random sample of the input upfront [37], where the *remaining* part of the instance then arrives *adversarially* online. This latter model, which we call the *online-with-a-sample* model (AOS[1] for short), is motivated by applications where historical data can inform decisions on new arrivals. The model has been used to study, for example, the Secretary problem [33], online bipartite matching [34], and online set cover [27]. Lattanzi et. al. [37] showed that the AOS model can be used to overcome the strong lower bounds for online ℓ_1-norm correlation clustering; they gave an $O(1/\varepsilon)$-competitive algorithm when a random sample of the nodes of size $\varepsilon \cdot n$ is revealed upfront.

Despite the importance and large body of literature on fair clustering, e.g., [18, 7, 10, 1, 8, 2, 42, 24], the online landscape for objectives other than the ℓ_1-norm for correlation clustering has remained entirely unexplored. In particular, there has been no work on the fairness-promoting ℓ_∞-norm, or general ℓ_p-norms, in an online setting. While the problem is natural and well-motivated, all previous approximation algorithms for the ℓ_∞-norm, and in fact for any ℓ_p-norm with $p > 1$, were, up until recently, based on convex program rounding, making them more difficult to adapt to the online setting than new combinatorial algorithms (see subsection 1.1 for more details).

As the aforementioned lower bounds show that a beyond-worst-case approach is needed, we choose to study ℓ_p-norm correlation clustering online using the AOS model. One motivation for this choice is the following. While the ℓ_1-norm admits an $O(1)$-competitive algorithm in the standard *random-order* (RO) model[2] via the Pivot algorithm [3], we prove that for ℓ_∞-norm correlation clustering, every algorithm in the RO model is $\Omega(n^{1/3})$-competitive. In other words, the ℓ_∞-norm is much more difficult than the ℓ_1-norm in the RO model. This hardness highlights a crucial limitation of the RO model, and provides strong motivation for using the AOS model for ℓ_∞-norm (and more generally ℓ_p-norm) correlation clustering. Moreover, from the perspective of comparing beyond-worst-case models, it is very interesting to understand the class of problems for which the RO and AOS models lead to roughly the same competitiveness, versus when one model is provably stronger than the other. These models are seemingly incomparable,[3] but for some problems (e.g., ℓ_1-norm correlation

[1] We note the abbreviation AOS stands for adversarial-order model with a sample.
[2] In the RO model, vertices arrive online in uniformly random order.
[3] While the RO model is fully online, it does not contain any adversarial component. The AOS model is not fully online, but it does contain an adversarial component.

clustering) the RO and AOS models admit algorithms with roughly the same competitive ratio, while for other problems (e.g., Steiner tree [4]) the AOS model can do better than what is possible in the RO model.[4] We ask the following:

Can a small sample in the AOS model provide enough structural information to break the online hardness barrier for ℓ_∞-norm correlation clustering?

We then take things a step further and ask whether we can achieve this goal without sacrificing global efficiency (the ℓ_1-norm) and, for that matter, any intermediate ℓ_p-norm. In particular, we ask whether we can replicate – in the online setting – a recent result of Davies, Moseley, and Newman [22] showing that for any instance of correlation clustering, there exists a single clustering that is simultaneously $O(1)$-approximate for all ℓ_p-norms, and moreover that it can be found efficiently in the offline setting. This is known as the *all-norms* objective. Formally introduced by Azar et. al. [6], the all-norms objective has received a steady stream of interest for various problems, including load balancing, set cover, k-clustering, and facility location, and in various settings (offline, online, distributed, and parallel)[35, 25, 11, 5, 15, 28, 13, 29]. The all-norms guarantee is particularly powerful because it is *parameter-free*: the algorithm is oblivious to the specific fairness versus total cost trade-off desired by the user, yet it returns a clustering that is simultaneously near-optimal for all $p \in [1, \infty]$. An ambitious goal would be to achieve a similar guarantee in the online setting:

Is it possible to approximate the all-norms objective for correlation clustering in the AOS model?

Our Contributions

We answer the above questions affirmatively, showing that in the AOS model, there is an algorithm that is locally fair (ℓ_∞-norm) without sacrificing global efficiency (ℓ_1-norm), and that our guarantees are nearly best possible for both norms. Moreover, we obtain a robustness guarantee for all norms in between, thus translating the offline all-norms guarantee into the online world.

Let OPT_p denote the cost of an optimal clustering for the ℓ_p-norm objective. We assume in the AOS model that an adversary fixes the online input. However, we see upfront a random sample S of εn nodes and the induced subgraph $G[S]$, where each node is sampled uniformly and independently into S with probability ε. The parameter $0 < \varepsilon < 1$ can be anything in our upper bound results (Theorem 1), and as small as $n^{-1/4}$ in our lower bound results (Theorem 2).

▶ **Theorem 1.** *Given $0 < \varepsilon < 1$, there is an algorithm in the AOS model that produces a* single *clustering that is:*

1. ***(Fairness)*** *$O\left(\frac{1}{\varepsilon^6} \cdot \log n\right)$-competitive for the ℓ_∞-norm with probability at least $1 - 1/n$.*
2. ***(Simultaneous Robustness)*** *$O\left(\frac{1}{\varepsilon^8} \cdot \log^4 n\right)$-competitive for the ℓ_p-norms, $1 \leqslant p < \infty$, with probability at least $1 - 1/n$.*
3. ***(Global Efficiency)*** *$O\left(\frac{1}{\varepsilon^6}\right)$-competitive in expectation for the ℓ_1-norm.*

[4] The RO and AOS models (as shown in [37]) both admit $O(1)$-competitive algorithms for ℓ_1-norm correlation clustering when ε is a constant. On the other hand, online Steiner tree has a lower bound of $\Omega(\log n)$ in the RO model, but in the AOS model there is an algorithm with competitive ratio $O(\log(1/\varepsilon))$.

Complementing our upper bounds, we show that the competitive ratios of our algorithm are nearly optimal for the ℓ_1- and ℓ_∞- norms. In particular, the logarithmic factor for the ℓ_∞-norm and a dependence on $1/\varepsilon$ for the ℓ_1- and ℓ_∞- norms are necessary. While the lower bound for the ℓ_1-norm is known [37], we contribute the new lower bound for the ℓ_∞-norm.

▶ **Theorem 2.** *For any $1/n^{1/4} \leqslant \varepsilon \leqslant 3/4$, any* randomized *algorithm in the AOS model is $\Omega(1/\varepsilon \cdot \log n)$-competitive for the ℓ_∞-norm and $\Omega(1/\varepsilon)$-competitive for the ℓ_1-norm.*

Finally, we give a hardness result for the ℓ_∞-norm in the RO model, justifying our focus on the AOS model.

▶ **Theorem 3.** *Any* randomized *algorithm for ℓ_∞-norm correlation clustering in the RO model is $\Omega(n^{1/3})$-competitive.*

This result establishes a fundamental separation between the ℓ_1- and ℓ_∞- norm objectives in the RO model, where an $O(1)$-competitive ratio for the ℓ_1-norm trivially follows from the Pivot algorithm. Recent work [27] shows that for a general class of minimization problems (called *augmentable integer programs*), there is a reduction from the AOS model to the RO model with loss of a factor at most $1/\varepsilon$ in the competitive ratio, i.e., if there is an algorithm with competitive ratio Δ in the RO model, then there is an algorithm with competitive ratio Δ/ε in the AOS model. In contrast, ℓ_∞-norm correlation clustering is an example of a minimization problem where the AOS model is actually *much stronger* at breaking through worst-case instances than the RO model. Thus in this regard, ℓ_∞-norm correlation clustering is more similar to online Steiner tree, where the AOS model can achieve competitive ratio $O(\log(1/\varepsilon))$, even though the RO model has a lower bound of $\Omega(\log n)$.

1.1 Related work

Prior work offline. Bansal, Blum, and Chawla [9] proposed correlation clustering for the goal of minimizing the ℓ_1-norm of the disagreement vector. The problem is NP-hard, and numerous approximation algorithms have been developed [3, 17, 20, 12]. A 1.437-approximation is known for the ℓ_1-norm [12], which improves upon the work that beat the threshold of 2 [20]. Puleo and Milenkovic [40] proposed the ℓ_p-norm objective for $p > 1$ and for each fixed p they gave a 48-approximation. This factor has since been improved in a series of works, first to 7 [16], and then to 5 [32]. Notably, this entire line of work on ℓ_p-norm objectives relies on rounding solutions to convex programs.

Davies, Moseley, and Newman [21] introduced the first *combinatorial* $O(1)$-approximation algorithm for the ℓ_∞-norm. Heidrich, Irmai, and Andres [30] built off of the techniques in [21] to prove a combinatorial 4-approximation for the ℓ_∞-norm, and this was later improved by Cao, Roche, and Su [14] to a combinatorial 3-approximation. Then, Davies, Moseley, and Newman [22] offered a new combinatorial algorithm proving there exists a single clustering that is an $O(1)$-approximation for all ℓ_p-norms simultaneously (also known as the all-norms objective). Cao, Li, and Ye [13] modified their algorithm in order to improve the factor for the all-norms objective, as well as to run in near-linear time in the MPC model in polylogarithmic rounds.

Prior work online. In the online setting, nodes arrive over time and reveal the signs of all of their edges to nodes that have previously arrived. Upon arrival of a node, an algorithm must irrevocably assign the node to a cluster. In the popular *competitive analysis* framework, the algorithm's clustering cost is compared to that of the optimal offline algorithm that is aware of the entire instance in advance. Recall that no constant-competitive algorithm exists in the

purely online setting for any ℓ_p-norm objective of correlation clustering [39]. Clustering in general has received much attention in the online and streaming settings. For the popular class of k-clustering problems (including k-median, k-means, and k-center), which likewise face pessimistic lower bounds in the online setting, a popular remedy is recourse [38, 31, 23, 26]. Likewise, to the best of our knowledge, the only previous beyond-worst-case results on correlation clustering in the online setting, besides the AOS model [37], allow recourse [19, 8]. We note that the work of Balkanski, Chatzitheodorou, and Maggiori [8] is similar in spirit to ours, as they also study fair correlation clustering in the online setting, but they use the traditional ℓ_1-norm for the objective while adding fairness constraints, and allow recourse as their beyond-worst-case approach.

Prior work in the AOS model. The AOS model was initiated by Kaplan, Naori, and Raz [33] in the context of the secretary problem.[5] Lattanzi et al. [37] then used the AOS model for ℓ_1-norm correlation clustering. They showed that the classic Pivot algorithm [3], modified to be seeded with an offline sample of size εn, gives an $O(1/\varepsilon)$-competitive algorithm, and this guarantee matches the lower bound up to constant factors. Since the works of Kaplan, Naori, and Raz [33] and Lattanzi et. al. [37] in the AOS model, the model has been applied to Steiner tree, load balancing, and facility location [4]; bipartite matching [34]; and set cover [27]. A similar "semi-online" setting also appears in [36] for bipartite matching; like the AOS model, the semi-online model described there also contains predicted and adversarial parts of the input, but the predicted part is not necessarily a random sample, and further, the parts may be interleaved in an arbitrary manner.

1.2 Technical overview

Our primary challenge is to adapt an offline algorithm that requires complete, global knowledge of the graph to an online setting with limited information. The offline algorithm for all-norms correlation clustering due to Davies, Moseley and Newman [22] relies on two offline-only steps. Step (1) is to compute a semi-metric d^* over all vertex pairs; d^* is an (almost) feasible solution to the canonical convex relaxation for the problem, but can be computed using explicit combinatorial properties of the graph. Step (2) is to, in place of an optimal solution x^* to the relaxation, feed d^* into the convex program rounding algorithm by Kalhan, Makarychev, and Zhou [32] (from now on, the *KMZ algorithm*). This is a ball-cutting procedure that iteratively cuts out clusters based on "suggestions" from d^* (i.e., if d^*_{uv} is small, then nodes u and v "want" to be clustered together). As in Step (1), the whole graph is required to determine the order in which clusters are cut out. So, both steps require significant changes in order to be adapted to the online setting.

To replace step (1), we compute a semi-metric $\tilde{d}$, only using the sample S, as a proxy for d^*. This will imply that when a node v arrives online, its distances $\tilde{d}_{uv}$ can immediately be computed for all u that have already arrived. To replace step (2), we note that the offline all-norms result holds (up to constants) when d^* is fed into *any* constant-approximate convex program rounding algorithm. We adapt the offline rounding algorithm of Charikar, Gupta, and Schwartz [16] (from now on, the *CGS algorithm*) instead of the KMZ algorithm, as we find the former easier to adapt to our online setting. The CGS algorithm is as follows: choose the unclustered node v that has the most unclustered vertices in the ball of radius r around

[5] The analysis, however, differs from that presented here, in that we define competitiveness on the whole instance, whereas they restrict to the online portion of the input.

it, then let v and all unclustered nodes in its ball of radius $3r$ form a new cluster (where the balls are w.r.t the semi-metric that is the solution to the convex program). Notably, the algorithm is dynamic in that the sizes of the balls changes at each iteration, because nodes are removed when they are clustered.

We next discuss some of the main technical challenges to computing a semi-metric based on the offline sample S, developing an online algorithm for ℓ_p-norm correlation clustering, and analyzing our algorithm.

Estimating a semi-metric via sampling

One key insight is that the semi-metric d^* in [22] – which, crucially, is defined using explicit combinatorial properties of the graph – can be approximated by computing it only on $G[S]$, the graph we see upfront on the random sample S. In the full version, we show this estimate, $\tilde{d}$, is of high quality . Analyzing $\tilde{d}$ is non-trivial, as $\tilde{d}$ is defined based on non-linear calculations as well as on thresholds, both of which are highly sensitive to error.

More specifically, in the offline world, d^* is computed in two steps. Let N_u^+ be the set of positive neighbors of u. First, intermediate distances, $d_{uv} = 1 - |N_u^+ \cap N_v^+|/|N_u^+ \cup N_v^+|$, are computed, and then some of these are rounded up to 1 based on thresholds. We estimate these intermediate distances using S, so we must work with quotients of correlated random variables which are prone to high variance and bias, especially for nodes with small positive neighborhoods. We then do threshold rounding with respect to these erroneous intermediate distances. This is technically problematic because, unlike the ℓ_1 objective where local estimation errors tend to average out across the instance, ℓ_p and ℓ_∞ objectives are brittle. The ideal scenario would be to obtain pointwise bounds of the form $\mathbb{E}[\tilde{d}_{uv}] \approx d^*_{uv}$, which would in turn enable us to black-box previous results from [22] bounding the cost of d^* against optimal. This does not seem possible, so instead we develop a novel charging scheme.

We pause to note that our technique, of using the sample S to estimate combinatorial quantities of interest, is quite different from that for ℓ_1-norm correlation clustering, which recall was previously studied in the AOS model [37]. That work adapts the Pivot algorithm, a 3-competitive algorithm in the RO model. In some sense, the Pivot algorithm is more readily adaptable to the AOS setting, since it is already an online algorithm, and the distribution of the (first ε-fraction of) input matches that of the random sample. Our present work provides an example of how the AOS sample can be useful in other situations. In particular, we believe this general strategy of estimating quantities for an algorithm using the sample, and then charging the cost of an algorithm's objective to these estimates, is of broader interest for other problems in this model.

Preprocessing the sample

As discussed above, the sample S is used to estimate d^* with a proxy metric $\tilde{d}$. It is also used to adapt the CGS algorithm. The CGS algorithm treats all of V as a set of candidate centers v, which are used to cut out clusters in the ball-cutting procedure. These are cut out in decreasing order of $|\text{Ball}_{x*}(v, r)|$. The challenge in adapting this is that V arrives online and adversarially. So, we restrict our centers, and the count of the ball sizes (computed now w.r.t. $\tilde{d}$) around those centers, to S. Importantly, to avoid correlational issues arising from the fact that several sets / quantities are computed on S, we show how to simulate four independent subsamples on S. Then, we estimate different random variables of interest using different subsamples, rather than on the whole common sample S. The subsamples are seemingly essential for making the analysis tractable.

Feeding the proxy metric into a static, online version of the CGS algorithm

The algorithm has two phases. The first is an online version of the offline ball-cutting CGS algorithm. Vertices clustered in this phase are said to be *pre-clustered*. The second phase handles the remaining vertices.

Pre-clustering phase: When a vertex v arrives, we first determine if the sample S is trustworthy for v via two checks. We first check whether the distances $\tilde{d}$ are sufficiently accurate for edges incident to v. If so, we check if v is close to one of the pre-selected centers.

- If both checks pass, v is assigned to a center that it is close to, particularly the earliest in the ordering of centers (see above). We note that to take the CGS algorithm online, this ordering is static, meaning it is pre-computed before the vertices arrive, unlike in the offline CGS algorithm, which dynamically orders the centers using the unclustered vertices at each iteration.
- If the test fails, the algorithm falls back to one of two versions of the Pivot algorithm.

(Modified) Pivot phases: We perform the classic Pivot algorithm of [3] on the vertices that fail the first check ($\tilde{d}$ is a poor estimator), and a modified version of the Pivot algorithm on the vertices that only fail the second check ($\tilde{d}$ is a good estimator, but the cluster centers are poor). While in the classic Pivot algorithm pivots cluster their positive neighbors, our modified version restricts to positive neighbors that are close (w.r.t. $\tilde{d}$) to the pivot.

In both versions of the Pivot algorithm, our analysis looks different from that of classic Pivot. This is for two reasons. First, Pivot has previously only been used for the ℓ_1-norm. It is not hard to find instances in which Pivot is $\Omega(n)$-approximate for the ℓ_∞ norm offline (see, e.g., Appendix A in [41]). This is because the analysis of Pivot for the ℓ_1-norm hinges on aggregating disagreements over bad triangles (triangles with two positive edges and one negative edge), which are configurations on which any clustering (thus an optimal one) must make a disagreement. But because ℓ_p-norm objectives in general are node-wise rather than edge-wise, our charging to bad triangles is necessarily nonlocal.

Second, Pivot crucially uses random order to ensure that disagreements in the optimal are not overcharged as a result of several bad triangles sharing the same edge. With adversarial order instead in our case, we show that overcharging can be controlled for a different reason, namely, that vertices that fail the first check have bounded positive degree.

For vertices that fail the second check, the argument is a bit more involved. Now, we have a generalization of bad triangles based not only on signs of edges but on $\tilde{d}$, so we need to take a hybrid approach and charge to both vanilla bad triangles and to the distances $\tilde{d}$. The rule that pivots grab *nearby* positive neighbors is crucial for preventing overcharging: now, in lieu of the positive degree of vertices being bounded as is the case above, we can use that the number of vertices near a pivot must be bounded. (Otherwise, a nearby vertex would have been sampled into S, and the pivot would have been preclustered!)

Lower Bounds

Mathieu, Sankur, and Schudy showed that any strictly online algorithm for the ℓ_1-norm objective must be at least $\Omega(n)$-competitive [39]. It is not hard to see that this lower bound carries over to all ℓ_p-norms, including ℓ_∞. To establish our lower bounds for the RO model (Theorem 3) and the AOS model (Theorem 2), we consider k gadgets of the lower bound instance above, each of size $\frac{n}{k}$ (with k set differently for each result). Crucially, for the ℓ_∞ objective, the offline optimal cost does not grow with k, unlike for finite ℓ_p-norms. For the RO model, we show that with constant probability there exists a gadget where the analogous

u, v in that gadget arrive first; then, the argument above can be repeated. For the AOS model, we show that, with constant probability, there is some gadget that arrives completely online, i.e., that is not touched by the sample; again, we can then repeat the argument above.

1.3 Organization

In Section 2, we discuss the AOS model and define the correlation metric and adjusted correlation metric. In Section 3, we define Algorithm 1, whose output satisfies Theorem 1. We prove item 2 of Theorem 1 (the statement for all finite p) in Section 4, though many proofs of lemmas and constructions stated are deferred to the full version of the paper. The proofs of items 1 and 3 of Theorem 1, and the lower bound results, Theorems 2 and 3, can be found in the full version.

2 Preliminaries

Let $G = (V, E)$ be a complete graph, where E is partitioned into positive edges (E^+) and negative edges (E^-). Let N_u^+ and N_u^- denote the positive and negative neighborhoods, respectively, of vertex u. That is, $N_u^+ = \{v \in V : uv \in E^+\}$ and $N_u^- = \{v \in V : uv \in E^-\}$. For convenience, assume that each vertex has a positive self-loop, i.e., for all $u \in V$, $u \in N_u^+$.

Recall that OPT_p is the optimal objective value of an integral solution for the ℓ_p-norm objective, where here $p \in [1, \infty]$. We set some parameters. Let $\delta = 10/7$, and define $c = c(\delta) := 2\delta^2 + \delta = 270/49$ and $r = r(\delta) := \frac{1}{2c\delta^2} = \frac{2401}{54000}$.

2.1 The AOS model

In the AOS model, an adversary fixes the online input, then we are given a sample S, where each element of the universe V is in S independently with probability $\varepsilon > 0$. We assume ε is known.[6] So for our problem, we see the induced subgraph on S a priori.

We will estimate various quantities using the sample S, and, for the analysis to be tractable, these estimations should not be correlated with each other. To this end, we show how to "split" the sample S into four independent subsamples S_p, S_d, S_b, S_r (where vertices in S may be in more than one subsample). We define this notion formally in the full version of the paper, but it will suffice to think of these subsamples as having the same joint distribution as four random subsets of vertices, each obtained by taking a uniformly random sample of (expected) size $\Theta(\varepsilon^2 n)$, and repeating this procedure *independently* four times.[7]

▶ **Definition 4.** *We call S_d the* distance sample*, S_p the* pre-clustering sample*, S_b the* counting sample*, and S_r the* rounding sample*. We call the vertices in S_p* centers*.*

Since the four samples are constructed from S, edges in the (complete) subgraph induced by $S_d \cup S_p \cup S_b \cup S_r$ are known to the online algorithm a priori. Then, the remaining vertices arrive one by one in adversarial order. When a vertex arrives, it reveals the signs of its incident edges to all vertices that have previously arrived (including all of S), and it must be irrevocably assigned a cluster.

We define $q(\varepsilon) := \mathbb{P}[v \in S_i] = \varepsilon^2/2$ for any subsample $S_i \in \{S_d, S_p, S_b, S_r\}$.

---

[6] One can immediately remove this assumption if we consider an "online-with-samples" model, suggested in the next footnote.

[7] Alternatively, one could consider a type of "online-with-samples" model, where an algorithm is given access to a constant number k of independent random samples of size $\varepsilon_i \cdot n$ of V, where $\sum_{i \in [k]} \varepsilon_i = \varepsilon$. We find this to be a perfectly reasonable model; the reader may find it simpler to assume this model.

2.2 Correlation metric, adjusted correlation metric, and their estimates

The *correlation metric* is a near-optimal feasible solution to the canonical convex program[8] for ℓ_p-norm correlation clustering. The correlation metric d is feasible for this convex program, meaning here that it satisfies the triangle inequality. So we may think of d_{uv} as specifying a distance between u and v, where the smaller the distance, the more that u and v would like to be clustered together. Relevant theorems on the (adjusted) correlation metric from [21, 22] are in the full version.

Below, we generalize the definition of the correlation metric by defining it on a subgraph of G. Taking $U = V$ below recovers the correlation metric in [21].

▶ **Definition 5** (Correlation metric). *Let $G = (V, E)$ be a complete, signed graph, and let $U \subseteq V$. For every (unordered) pair $u, v \in V$, define the* correlation metric *on U, denoted $d^U : V \times V \to [0, 1]$, by*

$$d^U_{uv} := 1 - \frac{|N^+_u \cap N^+_v \cap U|}{|(N^+_u \cup N^+_v) \cap U|}.$$

To make this well-defined, we take $d^U_{uv} = 1$ if $(N^+_u \cup N^+_v) \cap U = \varnothing$ for $u \neq v$, and $d^U_{uu} = 0$ always. When $U = V$, we simply write d for d^V and refer to d as the correlation metric.

The correlation metric distances are near-optimal for the convex program with the ℓ_∞-norm objective, but the metric must be adjusted in order to be simultaneously near-optimal for all ℓ_p-norms. We likewise generalize the definition of the so-called *adjusted correlation metric* in [22]; taking $U = W = V$ recovers their definition.

▶ **Definition 6** (Adjusted correlation metric). *Let $G = (V, E)$ be a complete, signed graph, and let $U, W \subseteq V$. Let d^U be the correlation metric on U as in Definition 5. Compute the* adjusted correlation metric *on U and W, denoted $d^{U,W} : E \to [0, 1]$, as follows:*

- *If $uv \in E^-$ and $d^U_{uv} > 7/10$, set $d^{U,W}_{uv} = 1$. (We say uv is* rounded up.*)*
- *For $u \in V$ such that $|\{v \in N^-_u : d^U_{uv} \leqslant 7/10\} \cap W| \geqslant 10/3 \cdot |N^+_u \cap U|$, set $d^{U,W}_{uv} = 1$ for all $v \in V \backslash \{u\}$. (We say u is* isolated *by $d^{U,W}$.)*

When $U = W = V$, we write d^* for $d^{V,V}$ and refer to d^* as the adjusted correlation metric.

A key takeaway is that the correlation metric and the adjusted correlation metric, while intended as surrogates for *non-combinatorial* optimal solutions to a convex program, are based solely on *combinatorial* properties of the graph – thus making them more amenable to the online setting. In the AOS model, we obviously cannot exactly compute the correlation metric or the adjusted correlation metric as we go, as the full positive neighborhood of a vertex is not necessarily known upon its arrival. But, for any $u, v \in V$, we *can* compute, e.g., d^U_{uv} in the case that U is a subset of S, *as soon as u, v have arrived*, since S is known to the algorithm upfront! The hope is that d^U and $d^{U,W}$ should be good approximations of $d = d^V$ and $d^* = d^{V,V}$, respectively, since S is a random sample of V – while also being usable online.

To estimate the adjusted correlation metric, we proceed in two steps. First, we estimate the correlation metric using the distance sample S_d; call this $\bar{d}$. Then, using the rounding sample S_r, we round $\bar{d}$ to estimate the adjusted correlation metric; call this $\tilde{d}$.

[8] See [22] for more details on this convex program which, while motivating, is not necessary for understanding the work herein.

▶ **Definition 7** (Estimated correlation metric). *For every (unordered) pair $u, v \in V$, define the* estimated correlation metric $\bar{d} : V \times V \to [0,1]$ *by* $\bar{d}_{uv} := d^{S_d}_{uv}$.

▶ **Definition 8** (Estimated adjusted correlation metric). *Let $\bar{d}$ be the estimated correlation metric as in Definition 7. Define the* estimated adjusted correlation metric $\tilde{d} : E \to [0,1]$ *by* $\tilde{d}_{uv} := d^{S_d,S_r}_{uv}$.
Moreover, we let R_1 be the (random) subset of V for which bullet 2 of Definition 6 applies (i.e., the set of vertices that are isolated by $\tilde{d}$), $R_2 = V \backslash R_1$, and $R_1(u) := \{v \in N^-_u : \bar{d}_{uv} \leqslant 7/10\}$.

The next observation ensures our algorithm for the AOS model is an online algorithm.

▶ **Observation 9.** *As soon as both u and v have arrived (including if one or both is in S), $\bar{d}_{uv}$ and $\tilde{d}_{uv}$ can be computed.*

If u has no positive neighbors in the sample S_d, then $\tilde{d}$ isolates u from all other vertices.

▶ **Fact 1.** *Fix $u \in V$ such that $N^+_u \cap S_d = \varnothing$. Then $\tilde{d}_{uv} = 1$ for all $v \neq u$.*

Both $\bar{d}$ and $\tilde{d}$ enjoy similar properties to d and d^*, respectively, in that they are a semi-metric and near semi-metric, respectively. Thus, we can still view $\bar{d}$ and $\tilde{d}$ as specifying distances between vertices.

▶ **Definition 10.** *We say a symmetric function $f : V \times V \to \mathbb{R}_+$ is a* δ-semi-metric *if $f_{uv} \leqslant \delta \cdot (f_{uw} + f_{wv})$ for all $u, v, w \in V$ (along with the usual requirement that $f_{uu} = 0$). We say in this case that f satisfies an* approximate triangle inequality*, or f is a* near semi-metric*.*

▶ **Lemma 11.** *Let $\bar{d}$ and $\tilde{d}$ be as in Definitions 7 and 8, respectively. Then*
- *$\bar{d} : V \times V \to [0,1]$ is a 1-semi-metric, that is, $\bar{d}$ satisfies the triangle inequality.*
- *$\tilde{d} : V \times V \to [0,1]$ is a $\frac{10}{7}$-semi-metric.*

2.3 Ordering the centers

Given a map $f : V \times V \to [0,1]$ on the vertices of G, for $c \in V$, $U \subseteq V$, and $\rho \geqslant 0$ we define: $\mathrm{Ball}_f(c,\rho) := \{v \in V : f_{cv} \leqslant \rho\}$ and $\mathrm{Ball}^U_f(c,\rho) := \{v \in U : f_{cv} \leqslant \rho\}$.

▶ **Definition 12** (Density). *Given a semi-metric $f : V \times V \to [0,1]$ and vertex $c \in V$, define $|Ball_f(c,r)|$ to be the* density of c *w.r.t f and r.*

A subroutine of our algorithm will be an adaptation of the CGS algorithm (see Section 1.2). This algorithm takes as input a metric f on the vertices and orders the vertices in decreasing order of their densities with respect to f and some radius r. Due to our online setting, we will only be able to estimate these densities, which we do using the counting sample S_b. We note that $|\mathrm{Ball}^{S_b}_{\tilde{d}}(c,r)|$ depends on the randomness of S_b, S_d, and S_r. The following observation will ensure our algorithm is well-defined.

▶ **Observation 13.** *For any center $c \in S_p$, the density $|Ball^{S_b}_{\tilde{d}}(c,r)|$ can be computed using only the information given* a priori *in the AOS model, i.e., the sample S.*

Lastly, we order the centers S_p based on their estimated densities.

▶ **Definition 14** (Ordered center sample). *Let $S_p = \{u_1, \ldots, u_{|S_p|}\}$. We assume the u_i are labeled so that $|Ball^{S_b}_{\tilde{d}}(u_1,r)| \geqslant \cdots \geqslant |Ball^{S_b}_{\tilde{d}}(u_{|S_p|},r)|$ (with ties broken arbitrarily).*

3 Algorithm Description

We describe our main algorithm (Algorithm 1) for the AOS model. Note that the for loop of Algorithm 1 considers the vertices in S in arbitrary order. So, we assume the algorithm considers the vertices in S first, and then the vertices in $V \backslash S$ as they arrive online.

Algorithm 1 is an online version of the offline CGS algorithm [16] (see the first two paragraphs in Section 1.2). The CGS algorithm must be adapted in several important ways in order to be taken online. First, the CGS algorithm takes as input an optimal solution to the convex program for the ℓ_p-norm objective of interest. Since we are not given the graph upfront, this is impossible to compute on the fly. Moreover, the convex program requires specifying an ℓ_p-norm objective, whereas we would like to optimize for all ℓ_p-norms simultaneously. In place of this optimal solution, we use $\tilde{d}$, which *can* be computed on the fly (Observation 9), and does *not* depend on p.

Algorithm 1 Main Algorithm.

Input: $G = (V, E)$, S_d, $\tilde{d}$, *ordered* $S_p = \{u_1, \dots, u_{|S_p|}\}$, radius r, constant c
Initialize: empty clusters $\mathcal{C}_{\mathsf{ALG}} = \{C_1, \dots, C_{|S_p|}\}$, sets $V_0 = V$ and $V' = \varnothing$
for each arriving $v \in V$ **do**
  // Pre-clustering phase
  if $|N_v^+ \cap S_d| \neq \varnothing$ and there exists $u_i \in S_p$ such that $\tilde{d}_{u_i v} \leqslant c \cdot r$ **then**
    Let i^* be the earliest in the ordering among all such u_i
    Add v to cluster C_{i^*}
  // Pivot phase
  else
    Add v to V'
    if v has $|N_v^+ \cap S_d| = \varnothing$ **then**
      Remove v from V_0, order $\overline{V}_0 = V \backslash V_0$ by arrival order
      Set $E_c := E^+(G[\overline{V}_0])$
      Add v to $\mathcal{C}_{\mathsf{ALG}}$ by running ModifiedPivot$(G[\overline{V}_0], \mathcal{C}_{\mathsf{ALG}}, \overline{V}_0 \backslash \{v\}, E_c)$
    else
      Order $V' \backslash \overline{V}_0$ by arrival order
      Set $E_c := E^+(G[V' \backslash \overline{V}_0]) \cap \{uv \in E : \tilde{d}_{uv} < c \cdot r\}$
      Add v to $\mathcal{C}_{\mathsf{ALG}}$ by running ModifiedPivot$(G[V' \backslash \overline{V}_0], \mathcal{C}_{\mathsf{ALG}}, V' \backslash (\overline{V}_0 \cup \{v\}), E_c)$
Output: Clustering $\mathcal{C}_{\mathsf{ALG}}$

The CGS algorithm requires an ordering on *all* of V based on the ball densities around the vertices. We are only able to estimate these densities for vertices in the offline sample, so we use the ordered sample of centers S_p (Definition 14), and $V \backslash S_p$ remains unordered. In the offline CGS algorithm, each vertex in V is clustered by the earliest vertex in the ordering that is nearby (thus by itself if necessary). In our setting, we cannot guarantee every vertex will be clustered because only S_p is ordered. Further, $\tilde{d}_{uv}$ is meaningless as a distance when u or v does not have positive neighbors in S_d (see Definition 5). Thus, we will not cluster every vertex in this way, and instead throw the vertices that are *not* "pre-clustered" (for either of these two reasons) to a subroutine called ModifiedPivot (Algorithm 2).

ModifiedPivot is a generalization of the Pivot algorithm from [3]. Taking $E_c = E^+$ in Algorithm 2 (as we do in the first call to ModifiedPivot in Algorithm 1) recovers the original Pivot algorithm. In the second call to ModifiedPivot in Algorithm 1, we further restrict E_c to edges that are "short" according to $\tilde{d}$; this is not simply an optimization, but seemingly needed in the analysis.

Algorithm 2 ModifiedPivot$(H, \mathcal{C}_{\mathsf{ALG}}, P, E_c)$.

Input: $H = (V_H, E_H)$ where V_H is *ordered*, $\mathcal{C}_{\mathsf{ALG}}$ is a clustering on $P \subseteq V_H$, and $E_c \subseteq E_H$ is a set of *clusterable* edges

for each $v_i \in V_H \backslash P$ (in order) **do**

  if there exists v_j that is before v_i in the ordering with $v_j v_i \in E_c$ **then**

    Let j^* be the earliest in the ordering among all such v_j

    Add v_i to the same cluster as v_{j^*}

  else

    Open a new cluster and add v_i to it

Output: Clustering $\mathcal{C}_{\mathsf{ALG}}$

Terminology

If the else statement in Algorithm 2 holds, we say v_i is a *pivot*, and that v_i's pivot is itself, v_i. If the if statement holds, we refer to v_{j^*} as v_i's *pivot*.

We say v is *pre-clustered* by u_i if v is added to C_i. Otherwise, $v \in V'$, and we say v is *unclustered* or *not pre-clustered*. Note that a vertex v may be unclustered for one of two reasons: either v has no positive neighbors in S_d ($v \notin V_0$ or as written in Algorithm 1, $v \in \overline{V}_0$, where $\overline{V}_0 = V \backslash V_0$), or $v \in V_0$ but v is not close to any vertex in S_p with respect to $\tilde{d}$. We call the vertices in V_0 *eligible for pre-clustering*, or simply *eligible*, and otherwise *ineligible*.

If v is pre-clustered by s, we denote s by $s^*(v)$. We may refer to $s^*(v)$ as v's *center*. We say u *is clustered after* v or v *is clustered before* u if either both u and v are pre-clustered, but $s^*(v)$ is before $s^*(u)$ (w.r.t the ordering of S_p), or if v is pre-clustered but u is not. Notationally, this will denoted as $v \succ u$.

The "good" event. For the cost analysis of Algorithm 1 when $p \neq 1$, we condition on a certain *good event*, denoted B^c, that occurs with high probability (see the full version). Informally, the event B^c ensures via concentration that the size of sufficiently large sets of vertices can be well-estimated based on observing their intersection with the subsamples. It also ensures that for pairs of nodes u, v with large combined positive neighborhood, $\bar{d}_{uv}$ is a good estimate of d_{uv}.

4 Cost of Algorithm 1 for Finite p

We bound the cost of Algorithm 1 for $p \in [1, \infty)$. We still use OPT_p to refer to the value of an optimal solution for the ℓ_p-norm objective, and additionally we often fix such an optimal clustering $\mathcal{C}_{\mathsf{OPT}}$.

At a high-level, we charge disagreements made by Algorithm 1 to the disagreements in $\mathcal{C}_{\mathsf{OPT}}$. Recall Algorithm 1 has several subroutines:

- a Pre-clustering phase that clusters nodes v that have (i) some of their positive neighborhood in S_d, i.e. $v \in V_0$, and (ii) are close (w.r.t. $\tilde{d}$) to a cluster center S_p;
- a subroutine that runs the standard Pivot algorithm on nodes $v \in \overline{V_0}$, which are the nodes that did not have any positive neighbor sampled into S_d;
- and a subroutine that runs a modified version of the Pivot algorithm on nodes $v \in V_0'$, which are the nodes that have a positive neighbor sampled into S_d, but were far away from all cluster centers.

We note the last two subroutines are part of the Pivot phase of Algorithm 1. Further, the clusters output by each subroutine are totally disjoint. The disagreements incurred by

Algorithm 1 can be partitioned into the disagreements made *within* each subroutine (e.g., u and v are both clustered in the Pre-clustering phase, but uv forms a disagreement in the solution output by Algorithm 1), and the disagreements *between* each subroutine (e.g., u is clustered in the Pre-clustering phase and v is clustered in the Pivot phase, but $uv \in E^+$). Therefore, we partition our analysis into the cost of the disagreements incurred during the Pre-clustering phase (Section 4.1), the cost of the disagreements incurred during the Pivot phase (see Section 4.2), and the cost incurred between these two phases (Section 4.3). Note the cost of the disagreements incurred during the Pivot phase includes the cost from running the standard Pivot algorithm on nodes $G[\overline{V_0}]$, the cost from running the modified Pivot algorithm on nodes $G[V_0']$, and the cost of disagreements between the two subroutines of the Pivot phase. See Figure 1 for references to the lemmas for each type of disagreement.

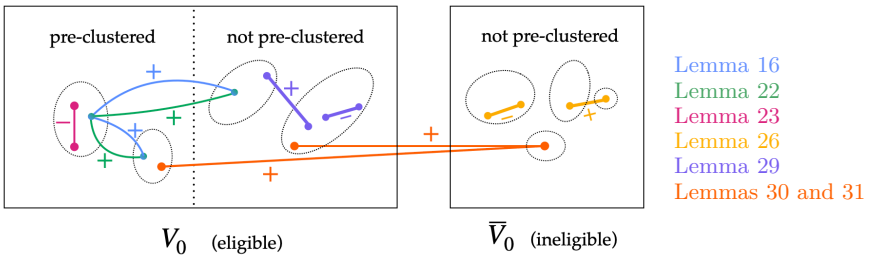

Figure 1 Overview of lemmas for the analysis of Algorithm 1 for finite p. Solid edges are disagreements, and dashed ovals are clusters. Vertices are partitioned into three sets based on whether or not they are eligible and pre-clustered, eligible and not pre-clustered, or ineligible. Charging the cost of a disagreement then depends on which set its endpoints belong to, its sign, and potentially which endpoint was higher with respect to the partial ordering $\succ$. Edges are partitioned by color, with edge types of the same color bounded by the correspondingly colored lemma. Note Lemmas 16 and 22 correspond to the same uv pair, but which lemma is relevant depends on (from the perspective of u's disagreements) whether $u \succ v$ or $v \succ u$.

Sometimes we are able to directly charge disagreements Algorithm 1 makes to $\mathcal{C}_{\mathsf{OPT}}$. More often, we use the estimated adjusted correlation metric, $\tilde{d}$, as an intermediary – specifically, we charge disagreements made by Algorithm 1 to $\tilde{d}$, then charge the cost of $\tilde{d}$ to OPT_p. The latter charging arguments (as in the following lemmas) can be found in the full version:

▶ **Lemma 15.** *Let $1 \leqslant p < \infty$. Conditioned on the event B^c, the estimated adjusted correlation metric $\tilde{d}$ satisfies $\sum_{u \in V_0} \left(\sum_{v \in N_u^+ \cap V_0} \tilde{d}_{uv} \right)^p \leqslant O\left(\left(1/\varepsilon^6 \cdot \log^3 n \right)^p \right) \cdot OPT_p^p$.*

Recall that the good event B^c (defined in Section 3) occurs with high probability.

4.1 Cost of Pre-clustering phase

Throughout, we use the choices of δ, c, r in Algorithm 1, and let $t := r/(2\delta)$ be a threshold parameter, which will be used in our analysis.

Recall that for nodes u assigned to clusters during the Pre-clustering phase, there is some node in S_p that has close $\tilde{d}$ distance to u. The highest ordered, with respect to the ordering of S_p, is said to pre-cluster u and is denoted by $s^*(u)$. We may refer to $s^*(u)$ as u's *center*. Recall we say v *is clustered before* u (or u *is clustered after* v) if either both u and v are pre-clustered, but $s^*(v)$ is before $s^*(u)$ (with respect to the ordering of S_p), or if v is pre-clustered but u is not. For shorthand, we write $v \succ u$ when v is clustered before u.

Fix a node u that is assigned a cluster during the Pre-clustering phase of Algorithm 1. Consider all nodes $v \in V_0$, so that uv is a disagreement in the output of Algorithm 1.

4.1.1 Cost of positive edges

We begin by bounding the cost of positive edges where at least one endpoint is pre-clustered, and both endpoints are eligible.

Fix a vertex $u \in V_0$. We partition the cost of positive disagreements incurred within the Pre-clustering phase based on whether $u \prec v$ or $v \prec u$. Lemma 16 handles the cost of disagreements uv incident to u when $v \succ u$, while Lemma 22 handles the cost of disagreements uv incident to u when $u \succ v$ and $v \in V_0$. In the proofs of both lemmas, we will see that it is easy to charge a disagreement uv to $\tilde{d}$ when $\tilde{d}_{uv}$ is sufficiently large, as we can then charge the ℓ_p-norm cost of $\tilde{d}$ to OPT_p. On the other hand, the difficult settings for both lemmas are for edges uv where $\tilde{d}_{uv}$ is small and both $u, v \in V_0$, so this distance is actually a reliable indicator that u and v do have many positive neighbors in common. The key is that even though $\tilde{d}_{uv}$ is small, the fact that u and v are not clustered together indicates there must be some other vertices we can charge to that do have large distance from u.

Some of the future claims will use that for the choices of δ, c, r as in the algorithm,

$$\max\left\{\frac{1}{cr/\delta - r}, \frac{1}{1-(\delta\cdot r + \delta^2\cdot c\cdot r + \delta\cdot r/2)}\right\} \leqslant 8 \quad \text{and} \quad \max\left\{\frac{1}{r/2 + c\cdot\delta\cdot r}, \frac{1}{c\cdot r}\right\} \leqslant 5. \tag{1}$$

▶ **Lemma 16.** *Condition on the good event B^c and fix $1 \leqslant p < \infty$. The ℓ_p-cost for $u \in V_0$ of the edges $uv \in E^+$ in disagreement with respect to $\mathcal{C}_{\mathsf{ALG}}$, where $v \succ u$, is bounded by $O\left(\frac{1}{\varepsilon^8}\cdot\log^4 n\right)\cdot \mathsf{OPT}_p$.*

Proof of Lemma 16. Note by definition of $\succ$ that each v in the statement of the lemma is necessarily pre-clustered, thus also $v \in V_0$. We partition the set $\{v \in N_u^+ : v \succ u\}$ depending on whether $\tilde{d}_{uv} > t$ or $\tilde{d}_{uv} \leqslant t$. Define $E_1(u)$ to be the random set of u's close, positive neighbors that are clustered before u: $E_1(u) := \{v \in N_u^+ : v \succ u\} \cap \mathrm{Ball}_{\tilde{d}}(u,t)$. Define $E_2(u)$ to be the remaining positive neighbors of u that are clustered before u: $E_2(u) := \{v \in N_u^+ : v \succ u\} \backslash E_1(u)$.

We partition the sum we wish to bound using Jensen's inequality to see

$$\sum_{u\in V_0} \left|\{v \in N_u^+ : v \succ u\}\right|^p \leqslant 2^{p-1}\cdot\sum_{u\in V_0}|E_1(u)|^p + 2^{p-1}\cdot\sum_{u\in V_0}|E_2(u)|^p. \tag{2}$$

As we alluded to before the beginning of the proof, it is straightforward to bound the cost of disagreeing edges uv when $\tilde{d}_{uv}$ is large. In particular, we can bound the latter sum:

$$\begin{aligned}\sum_{u\in V_0}|E_2(u)|^p = \sum_{u\in V_0}|\{v\in N_u^+\cap V_0 : \tilde{d}_{uv} \geqslant t\}|^p &\leqslant \frac{1}{t^p}\cdot\sum_{u\in V_0}\left(\sum_{v\in N_u^+\cap V_0}\tilde{d}_{uv}\right)^p \\ &\leqslant \left((7215\cdot C^3\cdot\log^3 n)/\varepsilon^6\right)^p\cdot\mathsf{OPT}_p^p. \end{aligned}\tag{3}$$

where the third inequality is from Lemma 15 and subbing in the values of δ and r.

Bounding $\sum_{u\in V_0}|E_1(u)|^p$ in line (2) is the more involved piece. We begin by partitioning $u \in V_0$ based on whether u has a close neighbor sampled by the center sample S_p; overall, we need to bound E_{1a} and E_{1b} where

$$\sum_{u\in V_0}|E_1(u)|^p = \underbrace{\sum_{u\in V_0:\mathrm{Ball}_{\tilde{d}}^{S_p}(u,t)=\varnothing}|E_1(u)|^p}_{E_{1a}} + \underbrace{\sum_{u\in V_0:\mathrm{Ball}_{\tilde{d}}^{S_p}(u,t)\neq\varnothing}|E_1(u)|^p}_{E_{1b}}$$

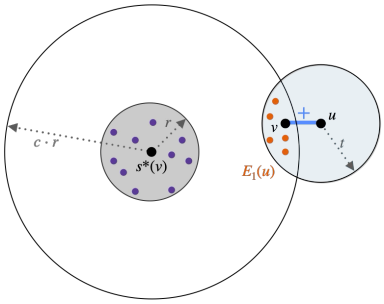

Figure 2 Bounding E_{1b} in the proof of Lemma 16, where $v \in N_u^+$ with $v \succ u$ and $\tilde{d}_{uv} \leqslant t$. We charge the disagreements between $v \in E_1(u)$ and u to the purple nodes in $\text{Ball}_{\tilde{d}}^{S_b}(s^*(v), r)$.

Bounding E_{1a}. Intuitively, the term E_{1a} will be easier to bound than E_{1b}, because, conditioned on B^c, the fact that $\text{Ball}_{\tilde{d}}^{S_p}(u,t) = \varnothing$ implies that $|\text{Ball}_{\tilde{d}}(u,t)|$ is small. So even though there are some nodes $v \in N_u^+$ that are close to u but assigned a different cluster than u, there cannot be that many of them. We use this insight together with a claim proving that there is sufficient fractional cost incident to $u \in V_0$. In turn, the small number of disagreements incident to u can be charged to this fractional cost, via Lemma 15.

The proof of the following claim may be found in the full version.

▷ Claim 17. Condition on the good event B^c. For $t = \frac{r}{2\delta}$, it is the case that

$$E_{1a} := \sum_{u \in V_0 : \text{Ball}_{\tilde{d}}^{S_p}(u,t) = \varnothing} |E_1(u)|^p \leqslant O\left(\left(1/\varepsilon^8 \cdot \log^4 n\right)^p\right) \cdot \mathsf{OPT}_p^p.$$

Bounding E_{1b}. Recall

$$E_{1b} := \sum_{u \in V_0 : \text{Ball}_{\tilde{d}}^{S_p}(u,t) \neq \varnothing} |E_1(u)|^p = \sum_{u \in V_0 : \text{Ball}_{\tilde{d}}^{S_p}(u,t) \neq \varnothing} |\{v \in N_u^+ : v \succ u, \tilde{d}_{uv} \leqslant t\}|^p.$$

Intuitively, because u is both pre-clustered and has a close neighbor sampled in S_p, we are now closer to the offline setting. In particular, since $v \succ u$, we have $|\text{Ball}_{\tilde{d}}^{S_b}(s^*(u), r)| \leqslant |\text{Ball}_{\tilde{d}}^{S_b}(s^*(v), r)|$. The idea is that u lies in an annulus around $\text{Ball}_{\tilde{d}}^{S_b}(s^*(v), r)$, and so the fractional cost of u can be lower bounded by (a constant factor times) the $|\text{Ball}_{\tilde{d}}^{S_b}(s^*(v), r)|$. The subtlety is that the inequality above lower bounding $|\text{Ball}_{\tilde{d}}^{S_b}(s^*(v), r)|$ in turn only holds when the balls are restricted to S_b, unlike in the offline case, where $S_b = V$. So it is not a priori clear that there will be enough fractional cost to which to charge $|E_1(u)|^p$. See Figure 2 for an illustration.

For each $u \in V_0$ with $\text{Ball}_{\tilde{d}}^{S_p}(u,t) \neq \varnothing$, choose a fixed but arbitrary $z(u) \in \text{Ball}_{\tilde{d}}^{S_p}(u,t)$. Note that because $z(u) \in S_p$ and $\tilde{d}_{u\,z(u)} \leqslant t \leqslant c \cdot r$, we know that $z(u)$ is a *candidate* for clustering u, so in particular $s^*(u)$ exists. Note $z(u)$ is a random variable depending on S_p, S_d, and S_r.

Recall that $E_1(u) := \{v \in N_u^+ : v \succ u, \tilde{d}_{uv} \leqslant t\}$. Define $B(u) := \bigcup_{v \in E_1(u)} \text{Ball}_{\tilde{d}}^{S_b}(s^*(v), r)$, that is, $B(u)$ is the union of balls in S_b, cut out around the vertices that cluster the vertices in $E_1(u)$. We will show that we can charge $|E_1(u)|^p$ to $|B(u)|^p$. Further, the set $B(u)$ is constructed so that every node $b \in B(u)$ lies in an annulus around u, so we can in turn charge $|B(u)|^p$ to the ℓ_p-cost of $\tilde{d}$.

The interesting case is when $|\text{Ball}_{\tilde{d}}(z(u), r)|$ is large. Here, we use Claims 18 and 19 to bound $|E_1(u)|$ in terms of $|B(u)|$. Then we use Claim 20 to relate $|B(u)|$ to the ℓ_p-cost of $\tilde{d}$. The proofs of Claims 18 and 19 are in the full version.

▷ Claim 18. If $\text{Ball}_{\tilde{d}}^{S_p}(u, t) \neq \varnothing$, then $|E_1(u)| \leqslant |\text{Ball}_{\tilde{d}}(z(u), r)|$.

▷ Claim 19. If $u \in V_0$ and $\text{Ball}_{\tilde{d}}^{S_p}(u, t) \neq \varnothing$, then $|\text{Ball}_{\tilde{d}}^{S_b}(z(u), r)| \leqslant |B(u)|$.

▷ Claim 20. If $u \in V_0$, then $|B(u)| \leqslant 8 \cdot \tilde{D}_0(u)$.

Proof of Claim 20. If $E_1(u) = \varnothing$, then $B(u) = \varnothing$, so the claim holds. Thus we may assume that $E_1(u) \neq \varnothing$. It then suffices to show that $\tilde{d}_{ub} \geqslant 1/8$ and $1 - \tilde{d}_{ub} \geqslant 1/8$ for every $b \in B(u)$. Then we will have (by Fact 1) that $b \in V_0$ (because $\tilde{d}_{ub} < 1$) for every $b \in B(u)$, and thus the claim follows.

To see that $\tilde{d}_{ub} \geqslant 1/8$ for any $b \in B(u)$, let $v \in E_1(u)$ be such that $b \in \text{Ball}_{\tilde{d}}(s^*(v), r)$ (such v exists by the definition of $B(u)$). Since $v \in E_1(u)$, u is clustered after v, so, using also that $u \in V_0$, we have that $\tilde{d}_{u\, s^*(v)} > c \cdot r$. Also, by choice of v, $\tilde{d}_{b\, s^*(v)} \leqslant r$. So by the approximate triangle inequality (Lemma 11), $\tilde{d}_{ub} \geqslant cr/\delta - r$, which is lower bounded by $1/8$ by line (1).

To see that $1 - \tilde{d}_{ub} \geqslant 1/8$ for any $b \in B(u)$, observe that $\tilde{d}_{b\, s^*(v)} \leqslant r$ (by choice of v), $\tilde{d}_{v\, s^*(v)} \leqslant c \cdot r$ (by definition of the algorithm and of $s^*(v)$), and $\tilde{d}_{vu} \leqslant t$ (since $v \in E_1(u)$). So by the approximate triangle inequality (Lemma 11), we have $\tilde{d}_{ub} \leqslant \delta \cdot [\tilde{d}_{b\, s^*(v)} + \delta(\tilde{d}_{v\, s^*(v)} + \tilde{d}_{vu})] \leqslant \delta \cdot r + \delta^2 \cdot c \cdot r + \delta^2 \cdot t$, so $1 - \tilde{d}_{ub} \geqslant 1 - (\delta \cdot r + \delta^2 \cdot c \cdot r + \delta^2 \cdot t)$, which is lower bounded by $1/8$ by line (1). ◁

▷ Claim 21. Condition on the good event B^c. For $t = \frac{r}{2\delta}$, it is the case that

$$E_{1b} := \sum_{u \in V_0 : \text{Ball}_{\tilde{d}}^{S_p}(u,t) \neq \varnothing} |E_1(u)|^p \leqslant O\left(\left(1/\varepsilon^8 \cdot \log^4 n\right)^p\right) \cdot \mathsf{OPT}_p^p.$$

The proof of this claim (which may be found in the full version) follows by combining the previous claims.

Combining the bounds on the sums and using that C is large (which is required by the good event), we conclude $\sum_{u \in V_0} \left|\{v \in N_u^+ : v \succ u\}\right|^p \leqslant \left((2900 \cdot C' \cdot C^4 \cdot \log^4 n)/\varepsilon^8\right)^p \cdot \mathsf{OPT}_p^p$. ◀

The proof of Lemma 22 is similar in spirit to that of Lemma 16, but must nonetheless be handled separately (except in the case of $p = 1$, where we can sum over disagreements edge-wise rather than node-wise). It's proof is fully deferred to the full version.

▶ **Lemma 22.** *Condition on the good event B^c and fix $1 \leqslant p < \infty$. The ℓ_p-cost for u that are pre-clustered (thus are necessarily in V_0) of the edges $uv \in E^+$ in disagreement with respect to $\mathcal{C}_{ALG}$, where $u \succ v$, is bounded by*

$$\sum_{u \in V_0} \left|\{v \in V_0 \cap N_u^+ : u \succ v\}\right|^p \leqslant O\left(\left(1/\varepsilon^8 \cdot \log^4 n\right)^p\right) \cdot \mathit{OPT}_p^p.$$

4.1.2 Cost of negative edges

The edges $uv \in E^-$, where at least one endpoint is pre-clustered, that are in disagreement with respect to $\mathcal{C}_{\mathsf{ALG}}$ are those where u is clustered with v. The proof follows easily once we have an analogue of Lemma 15 for negative edges, and can be found in the full version.

▶ **Lemma 23.** *Condition on the good event B^c and fix $1 \leqslant p < \infty$. The ℓ_p-cost for u in V_0 of the negative edges in the Pre-clustering phase is bounded by*

$$\sum_{u \in V_0} \left| \left\{ v \in N_u^- : v \text{ clustered with } u \right\} \right|^p \leqslant O\left((1/\varepsilon^2 \cdot \log n)^p \right) \cdot OPT_p^p.$$

4.2 Cost of Pivot phase

Let $G' = (V', E')$ be the subgraph induced by the vertices that are *not* pre-clustered in Algorithm 1. In this section, we bound the cost of disagreements in G'. Recall that V_0 is the set of eligible vertices, i.e., those vertices $v \in V$ such that $|N_v^+ \cap S_d| \neq \varnothing$. So V' contains the vertices that are *not* eligible (those in $\overline{V_0} = V \backslash V_0$), as well as vertices that are eligible but that are far from all vertices in S_p:

$$V' := \overline{V_0} \cup V_0', \text{ where } V_0' = V_0 \cap \{ v \in V : \tilde{d}_{vu_i} > c \cdot r \text{ for all } u_i \in S_p \}.$$

Algorithm 1 runs the standard Pivot algorithm on $G[\overline{V_0}]$, and runs Modified Pivot on $G[V_0']$. In Lemma 26, we bound the disagreements incurred by Pivot on $G[\overline{V_0}]$, and in Lemma 29 we bound the disagreements incurred by Modified Pivot on $G[V_0']$.

We note the arguments in this section may look rather different than those in the Pre-clustering phase. This is a consequence of the fact that we are using totally different clustering subroutines in each phase. In this Pivot phase, we use two versions of the (modified) Pivot algorithm. Therefore, the analysis is more combinatorial; often the charging arguments use "bad triangles" as intermediaries:

▶ **Definition 24.** *A* bad triangle *is a triple uvw such that $uv, uw \in E^+$ and $vw \in E^-$.*

Note every clustering incurs a disagreement on at least one edge in a bad triangle.

▶ **Definition 25.** *In Algorithm 2, for $v_i \in \overline{V_0}$ (analogously, $v_i \in V_0'$), we define v_i*'s pivot *to be v_j^* if the* **if** *statement holds, and to be v_i otherwise.*

4.2.1 Disagreements in $G[\overline{V_0}]$

To bound disagreements incident to $u \in \overline{V_0}$, ideally we would relate the set of bad triangles that contain u to the disagreements incident to u in $\mathcal{C}_{\mathsf{ALG}}$. However, while the optimal must make a disagreement on each bad triangle it does not necessarily have any disagreements incident u. So in effect, we have to charge some of our disagreements incident to u to the optimal solution's disagreements on other vertices.

As before, fix an optimal clustering $\mathcal{C}_{\mathsf{OPT}}$ (for the entire graph G) for any fixed ℓ_p-norm. Let $\mathsf{OPT}(u)$ be the (positive or negative) neighbors inducing disagreements with u in $\mathcal{C}_{\mathsf{OPT}}$: $\mathsf{OPT}(u) := \{ v \in V \mid uv \text{ a disagreement in } \mathcal{C}_{\mathsf{OPT}} \}$.

Analogously, for $u \in \overline{V_0}$, define $\mathsf{Pivot}(u)$ to be the (positive or negative) neighbors inducing disagreements with u, *restricted to the clusters formed by the Pivot phase of Algorithm 1 on* $G[\overline{V_0}]$: $\mathsf{Pivot}(u) := \{ v \in \overline{V_0} \mid uv \text{ a disagreement in } \mathcal{C}_{\mathsf{ALG}} \}$.

▶ **Lemma 26.** *Condition on the good event B^c and fix $1 \leqslant p < \infty$. The ℓ_p-cost for $u \in \overline{V_0}$ of the edges uv in disagreement with respect to $\mathcal{C}_{ALG}$ for $v \in \overline{V_0}$, is bounded by*

$$\sum_{u \in \overline{V_0}} |Pivot(u)|^p \leqslant O\left((1/\varepsilon^2 \cdot \log n)^p \right) \cdot OPT_p^p.$$

Proof of Lemma 26. Let $\mathcal{T}$ denote the set of bad triangles in $G[\overline{V_0}]$, and let $\mathcal{T}(u)$ denote the set of bad triangles in $G[\overline{V_0}]$ that contain vertex u. As mentioned before the start of the proof, to bound disagreements uv where both u and v are in $\overline{V_0}$, we would like to relate $|\mathcal{T}(u)|$ to $|\mathsf{Pivot}(u)|$. However, this is not possible, so instead we charge some of the disagreements incident to u incurred by our Pivot phase to disagreements the optimal solution incurs on other vertices.

Let $\mathcal{T}_{\mathsf{OPT}}(u) \subseteq \mathcal{T}(u)$ be the bad triangles T for which $\mathcal{C}_{\mathsf{OPT}}$ has a disagreement incident to u in T, and let $\overline{\mathcal{T}_{\mathsf{OPT}}(u)} = \mathcal{T}(u) \backslash \mathcal{T}_{\mathsf{OPT}}(u)$ be the remaining bad triangles in $G[\overline{V_0}]$ containing u. We observe that for every $u \in \overline{V_0}$, every disagreement in $G[\overline{V_0}]$ incident to u can be mapped to some $T \in \mathcal{T}(u)$ – namely, the unique bad triangle containing the disagreement and u's pivot. Moreover, this mapping is injective because Pivot incurs *exactly* one disagreement on each bad triangle. Therefore we can bound the disagreements that our algorithm makes in $G[\overline{V_0}]$ that are incident to u by applying Jensen's inequality,

$$\sum_{u \in \overline{V_0}} |\mathsf{Pivot}(u)|^p \leqslant \sum_{u \in \overline{V_0}} |\mathcal{T}(u)|^p \leqslant 2^{p-1} \cdot \underbrace{\sum_{u \in \overline{V_0}} |\mathcal{T}_{\mathsf{OPT}}(u)|^p}_{S_1} + 2^{p-1} \cdot \underbrace{\sum_{u \in \overline{V_0}} \left|\overline{\mathcal{T}_{\mathsf{OPT}}(u)}\right|^p}_{S_2}. \tag{4}$$

Recall that $u \in \overline{V_0}$ because $N_u^+ \cap S_d = \varnothing$. So by the good event, it must be that $|N_u^+| < C \log n/\varepsilon^2$. This bound on $|N_u^+|$ will repeatedly be used.

First we bound S_1, which will be simpler to bound since these triangles directly correspond to a disagreement that $\mathcal{C}_{\mathsf{OPT}}$ has on u. Its proof is in the full version.

▷ Claim 27. $\sum_{u \in \overline{V_0}} |\mathcal{T}_{\mathsf{OPT}}(u)|^p \leqslant O\left((1/\varepsilon^2 \cdot \log n)^p\right) \cdot \mathsf{OPT}_p^p$.

It remains to bound S_2, and this sum contains the bad triangles which we will charge to disagreements not incident to u.

▷ Claim 28. $\sum_{u \in \overline{V_0}} \left|\overline{\mathcal{T}_{\mathsf{OPT}}(u)}\right|^p \leqslant O\left((1/\varepsilon^2 \cdot \log n)^p\right) \cdot \mathsf{OPT}_p^p$.

Proof of Claim 28. Fix $u \in \overline{V_0}$. By definition, for every $T \in \overline{\mathcal{T}_{\mathsf{OPT}}(u)}$, $\mathcal{C}_{\mathsf{OPT}}$ has an edge in disagreement on the *unique* edge of T not incident to u. So, no other triangle in $\overline{\mathcal{T}_{\mathsf{OPT}}(u)}$ contains this edge as a disagreement. Moreover, by the definition of a bad triangle, one of the endpoints of this disagreeing edge is a positive neighbor of u in $G[\overline{V_0}]$. So we have by the discussion above that

$$\begin{aligned}
S_2 = \sum_{u \in \overline{V_0}} \left|\overline{\mathcal{T}_{\mathsf{OPT}}(u)}\right|^p &\leqslant \sum_{u \in \overline{V_0}} \left(\sum_{\overline{V_0} \cap N_u^+} |\mathsf{OPT}(v) \cap \overline{V_0}| \right)^p \\
&\leqslant \sum_{u \in \overline{V_0}} |N_u^+|^{p-1} \sum_{v \in \overline{V_0} \cap N_u^+} |\mathsf{OPT}(v) \cap \overline{V_0}|^p \qquad (5) \\
&\leqslant \sum_{v \in \overline{V_0}} |\mathsf{OPT}(v) \cap \overline{V_0}|^p \sum_{u \in \overline{V_0} \cap N_v^+} |N_u^+|^{p-1} \\
&\leqslant (C/\varepsilon^2 \cdot \log n)^p \cdot \mathsf{OPT}_p^p. \qquad (6)
\end{aligned}$$

Line (5) follows from Jensen's inequality. Line (6) follows from the fact that because we conditioned on the good event B^c, the maximum positive degree of any $u \in \overline{V_0}$ is $\frac{C}{\varepsilon^2} \cdot \log n$ and for $v \in \overline{V_0}$, there are at least $|\mathsf{OPT}(v) \cap \overline{V_0}|$ disagreements incident to v in $\mathcal{C}_{\mathsf{OPT}}$ by definition of $\mathsf{OPT}(v)$. ◁

The lemma statement follows from the claims and line (4). ◀

4.2.2 Disagreements in $G[V_0']$

Next we will bound the disagreements in $G[V_0']$. Recall that the vertices in V_0' are those that have sufficiently large positive neighborhood sampled in S_d, but were far from all cluster centers. As has been the case for other disagreement types, the disagreements whose cost is most difficult to bound are on the positive edges uv, where $\tilde{d}_{uv}$ is quite small. Here, we are able to charge uv to some other edges uw, for uvw a bad triangle.

For $u \in V_0'$, define $\mathsf{Pivot}(u)$ to be the (positive or negative) disagreements incident to u, *restricted to the clusters formed by the (Modified) Pivot phase of Algorithm 1 on* $G[V_0']$: $\mathsf{Pivot}(u) := \{uv \mid v \in V_0', uv \text{ a disagreement in } \mathcal{C}_{\mathsf{ALG}}\}$.

Note that both sets are sets of *edges*, unlike in Lemma 26 where the analogous sets are sets of vertices. We write Pivot for brevity, but recall that the algorithm on $G[V_0']$ is actually a modified version of the classic Pivot algorithm, as the pivots in our algorithm grab positive neighbors that are additionally required to be nearby with respect to $\tilde{d}$ (see the definition of E_c in the **else** statement for $V_0' = V' \backslash \overline{V_0}$ in Algorithm 1).

The proof of the following lemma may be found in the full version.

▶ **Lemma 29.** *Condition on the good event B^c and fix $1 \leqslant p < \infty$. The ℓ_p-cost for $u \in V_0'$ of the edges uv in disagreement with respect to $\mathcal{C}_{ALG}$ for $v \in V_0'$ is bounded by*

$$\sum_{u \in V_0'} \left|\mathit{Pivot}(u)\right|^p \leqslant O\left(\left(1/\varepsilon^8 \cdot \log^4 n\right)^p\right) \cdot \mathit{OPT}_p^p.$$

4.2.3 Disagreements between $G[\overline{V_0}]$ and $G[V_0']$

The only disagreements occurring on edges going between V_0' and $\overline{V_0}$ are from positive edges. We bound the cost of disagreements uv incident to $u \in \overline{V_0}$, for $v \in V_0 \cap N_u^+$, in Lemma 30, then we bound the cost of disagreements uv incident to $u \in V_0$, for $v \in \overline{V_0} \cap N_u^+$, in Lemma 31. Since $V_0' \subseteq V_0$, these lemmas immediately bound the cost of disagreements between $G[V_0']$ and $G[\overline{V_0}]$.

4.3 Cost between the Pre-clustering phase and the Pivot phase

It remains to bound the cost of edges that go between the Pre-clustering and Pivot phases. The disagreements uv incident to u can take several forms: u is pre-clustered and $v \in V_0'$, $u \in V_0'$ and v is pre-clustered, $u \in \overline{V_0}$ and v is pre-clustered, u is pre-clustered and $v \in \overline{V_0}$. We discuss each disagreement type in order of the above list.

For u pre-clustered and $v \in V_0'$, both u and v are in V_0, but $u \succ v$; recall we use the notation $u \succ v$ to mean u is clustered before v, or in other words u is either pre-clustered to a higher ordered center than v, or u is pre-clustered and v is not. These disagreements are already accounted for in Lemma 22.

Similarly, $u \in V_0'$ and v pre-clustered, both nodes are in V_0 again. Though this time, $v \succ u$. These disagreements are already accounted for in Lemma 16.

The cost of the next type of disagreement, when $u \in \overline{V_0}$ and v is pre-clustered, will be bounded in Lemma 30. Then the cost of disagreements where u is pre-clustered (thus $u \in V_0$) and $v \in \overline{V_0}$ will be bounded in Lemma 31 (these proofs can be found in the full version.)

▶ **Lemma 30.** *Condition on the good event B^c and fix $1 \leqslant p < \infty$. The ℓ_p-cost for $u \in \overline{V_0}$ of edges uv in disagreement with respect to $\mathcal{C}_{ALG}$, for $v \in V_0$, is bounded by*

$$\sum_{u \in \overline{V_0}} |V_0 \cap N_u^+|^p \leqslant O\left(\left(1/\varepsilon^4 \cdot \log^2 n\right)^p\right) \cdot OPT_p^p.$$

As two consequences, we have that $\sum_{u \in \overline{V_0}} |\{v \in N_u^+ \mid v \text{ pre-clustered}\}|^p \leqslant O\left(\left(1/\varepsilon^4 \cdot \log^2 n\right)^p\right) \cdot OPT_p^p$ and $\sum_{u \in \overline{V_0}} |\{v \in N_u^+ \cap V_0'\}|^p \leqslant O\left(\left(1/\varepsilon^4 \cdot \log^2 n\right)^p\right) \cdot OPT_p^p$.

▶ **Lemma 31.** *Condition on the good event B^c and fix $1 \leqslant p < \infty$. The ℓ_p-cost for $u \in V_0$ of edges uv in disagreement with respect to $\mathcal{C}_{ALG}$, for $v \in \overline{V_0}$, is bounded by*

$$\sum_{u \in V_0} |\overline{V_0} \cap N_u^+|^p \leqslant O\left(\left(1/\varepsilon^4 \cdot \log^2 n\right)^p\right) \cdot OPT_p^p.$$

As two consequences, we have that $\sum_{u \text{ pre-clustered}} |\{v \in N_u^+ \cap \overline{V_0}\}|^p \leqslant O\left(\left(1/\varepsilon^4 \cdot \log^2 n\right)^p\right) \cdot OPT_p^p$ and $\sum_{u \in V_0'} |\{v \in N_u^+ \cap \overline{V_0}\}|^p \leqslant O\left(\left(1/\varepsilon^4 \cdot \log^2 n\right)^p\right) \cdot OPT_p^p$.

4.4 Proof of item 2 for Theorem 1

We are ready to combine the results proven so far in this section.

Proof of item 2 for Theorem 1. Let $\mathcal{C}_{\mathsf{ALG}}$ be the clustering output by Algorithm 1, and let $\text{cost}_p(\mathcal{C}_{\mathsf{ALG}})$ be the ℓ_p-norm of the disagreement vector of $\mathcal{C}_{\mathsf{ALG}}$. We further partition the edges in disagreement based on whether they are positive or negative, which phase in Algorithm 1 they are clustered in, and (if at least one endpoint of an edge is pre-clustered) whether or not $u \succ v$. These cases are exhaustive; see Figure 1. Combining the terms from Lemmas 16, 22, 23, 26, 29, 30, and 31, and then applying Jensen's inequality and taking the p^{th} root, we see that with high probability (as we recall the good event B^c occurs with high probability) $||y_{\mathcal{C}_{\mathsf{ALG}}}||_p \leqslant O\left(1/\varepsilon^8 \cdot \log^4 n\right) \cdot \mathsf{OPT}_p$. ◀

5 Conclusion

We develop an algorithm for online correlation clustering which, given a sample of ε-fraction of the nodes from the underlying instance, returns a clustering that is simultaneously $O(1/\varepsilon^6)$-competitive for the ℓ_1-norm objective in expectation and $O(\log n/\varepsilon^6)$-competitive for the ℓ_∞-norm objective with high probability. This is the first positive result for the ℓ_∞-norm in the online setting. We also prove lower bounds that match our upper bounds up to constants and powers of $1/\varepsilon$ for either norm. Finally, we show that our algorithm is also $O(\log^4 n/\varepsilon^8)$-competitive for each finite ℓ_p-norm with high probability. Thus, we successfully translate the all-norms result of [22] to the online setting.

Our work highlights two key insights. First, it demonstrates the robustness of the adjusted correlation metric: even an estimated version suffices to guide near-optimal decisions in the AOS model. Second, it identifies structural properties that make problems amenable to this online model. Specifically, the ability to estimate key quantities from a small but uniformly sampled subset of the input is crucial for solving problems in the AOS model.

Overall, our results suggest that the AOS model is a promising framework for problems where limited but well-distributed information allows for effective decision-making. In particular, ℓ_∞-norm clustering is an example of a problem when the AOS model is much

stronger than the popular RO model. We remark that the model is still relatively new, and we believe the techniques from this paper can be of use to understand a wider range of problems in this setting. For instance, our idea of leveraging different subsamples independently to help mitigate correlation effects across estimating different quantities may be useful. It is of further interest to determine which problems with strong lower bounds on the competitive ratio in the strictly online setting and/or the random-order (RO) model admit small competitive ratios in the AOS model.

References

1 Saba Ahmadi, Sainyam Galhotra, Barna Saha, and Roy Schwartz. Fair correlation clustering. *arXiv preprint arXiv:2002.03508*, 2020. arXiv:2002.03508.

2 Sara Ahmadian, Alessandro Epasto, Ravi Kumar, and Mohammad Mahdian. Fair correlation clustering. In *International Conference on Artificial Intelligence and Statistics*, pages 4195–4205. PMLR, 2020. URL: http://proceedings.mlr.press/v108/ahmadian20a.html.

3 Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM*, 55(5):1–27, 2008. doi:10.1145/1411509.1411513.

4 CJ Argue, Alan Frieze, Anupam Gupta, and Christopher Seiler. Learning from a sample in online algorithms. *Advances in Neural Information Processing Systems*, 35:13852–13863, 2022. URL: https://dl.acm.org/doi/10.5555/3600270.3601277.

5 Sepehr Assadi, Aaron Bernstein, and Zachary Langley. Improved bounds for distributed load balancing. *arXiv preprint arXiv:2008.04148*, 2020. arXiv:2008.04148.

6 Yossi Azar, Leah Epstein, Yossi Richter, and Gerhard J Woeginger. All-norm approximation algorithms. *Journal of Algorithms*, 52(2):120–133, 2004. doi:10.1016/J.JALGOR.2004.02.003.

7 Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. Scalable fair clustering. In *International Conference on Machine Learning*, pages 405–413. PMLR, 2019. URL: http://proceedings.mlr.press/v97/backurs19a.html.

8 Eric Balkanski, Jason Chatzitheodorou, and Andreas Maggiori. Cost-free fairness in online correlation clustering. In *36th International Conference on Algorithmic Learning Theory*, pages 167–203. PMLR, 2025. URL: https://proceedings.mlr.press/v272/balkanski25b.html.

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

10 Suman Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. *Advances in Neural Information Processing Systems*, 32:4955–4966, 2019. URL: https://dl.acm.org/doi/abs/10.5555/3454287.3454733.

11 Aaron Bernstein, Tsvi Kopelowitz, Seth Pettie, Ely Porat, and Clifford Stein. Simultaneously load balancing for every p-norm, with reassignments. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, pages 51:1–51:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ITCS.2017.51.

12 Nairen Cao, Vincent Cohen-Addad, Euiwoong Lee, Shi Li, Alantha Newman, and Lukas Vogl. Understanding the cluster linear program for correlation clustering. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1605–1616, 2024. doi:10.1145/3618260.3649749.

13 Nairen Cao, Shi Li, and Jia Ye. Simultaneously Approximating All Norms for Massively Parallel Correlation Clustering. In *52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025)*, pages 40:1–40:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.ICALP.2025.40.

14 Nairen Cao, Steven Roche, and Hsin-Hao Su. Min-Max Correlation Clustering via Neighborhood Similarity. In *33rd Annual European Symposium on Algorithms (ESA 2025)*, pages 41:1–41:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.ESA.2025.41.

15 Deeparnab Chakrabarty and Chaitanya Swamy. Approximation algorithms for minimum norm and ordered optimization problems. In *Proceedings of the 51st Annual ACM Symposium on Theory of Computing*, pages 126–137, 2019. doi:10.1145/3313276.3316322.

16 Moses Charikar, Neha Gupta, and Roy Schwartz. Local guarantees in graph cuts and clustering. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 136–147. Springer, 2017. doi:10.1007/978-3-319-59250-3_12.

17 Shuchi Chawla, Konstantin Makarychev, Tselil Schramm, and Grigory Yaroslavtsev. Near optimal lp rounding algorithm for correlation clustering on complete and complete k-partite graphs. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*, pages 219–228, 2015. doi:10.1145/2746539.2746604.

18 Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. *Advances in Neural Information Processing Systems*, 30:5036–5044, 2017. URL: https://dl.acm.org/doi/abs/10.5555/3295222.3295256.

19 Vincent Cohen-Addad, Silvio Lattanzi, Andreas Maggiori, and Nikos Parotsidis. Online and consistent correlation clustering. In *International Conference on Machine Learning*, pages 4157–4179. PMLR, 2022. URL: https://proceedings.mlr.press/v162/cohen-addad22a.html.

20 Vincent Cohen-Addad, Euiwoong Lee, and Alantha Newman. Correlation clustering with sherali-adams. *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 651–661, 2022. doi:10.1109/FOCS54457.2022.00068.

21 Sami Davies, Benjamin Moseley, and Heather Newman. Fast combinatorial algorithms for min max correlation clustering. In *International Conference on Machine Learning*, pages 7205–7230. PMLR, 2023. URL: https://proceedings.mlr.press/v202/davies23a.html.

22 Sami Davies, Benjamin Moseley, and Heather Newman. Simultaneously approximating all lp-norms in correlation clustering. In *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*, pages 52:1–52:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ICALP.2024.52.

23 Hendrik Fichtenberger, Silvio Lattanzi, Ashkan Norouzi-Fard, and Ola Svensson. Consistent k-clustering for general metrics. In *Proceedings of the 2021 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2660–2678. SIAM, 2021. doi:10.1137/1.9781611976465.158.

24 Zachary Friggstad and Ramin Mousavi. Fair correlation clustering with global and local guarantees. In *Workshop on Algorithms and Data Structures*, pages 414–427. Springer, 2021. doi:10.1007/978-3-030-83508-8_30.

25 Daniel Golovin, Anupam Gupta, Amit Kumar, and Kanat Tangwongsan. All-norms and all-l_p-norms approximation algorithms. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 199–210. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2008. doi:10.4230/LIPIcs.FSTTCS.2008.1753.

26 Xiangyu Guo, Janardhan Kulkarni, Shi Li, and Jiayi Xian. Consistent k-median: Simpler, better and robust. In *International Conference on Artificial Intelligence and Statistics*, pages 1135–1143. PMLR, 2021. URL: https://proceedings.mlr.press/v130/guo21a.html.

27 Anupam Gupta, Gregory Kehne, and Roie Levin. Set covering with our eyes wide shut. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4530–4553. SIAM, 2024. doi:10.1137/1.9781611977912.160.

28 Swati Gupta, Jai Moondra, and Mohit Singh. Which lp norm is the fairest? approximations for fair facility location across all "p". In *Proceedings of the 24th ACM Conference on Economics and Computation*, page 817, 2023. doi:10.1145/3580507.3597664.

29 Swati Gupta, Jai Moondra, and Mohit Singh. Balancing notions of equity: Trade-offs between fair portfolio sizes and achievable guarantees. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1136–1165. SIAM, 2025. doi:10.1137/1.9781611978322.33.

30 Holger SG Heidrich, Jannik Irmai, and Bjoern Andres. A 4-approximation algorithm for min max correlation clustering. In *International Conference on Artificial Intelligence and Statistics,*, pages 1945–1953. PMLR, 2024. URL: https://proceedings.mlr.press/v238/heidrich24a.html.

31 Mohammad Reza Karimi Jaghargh, Andreas Krause, Silvio Lattanzi, and Sergei Vassilvtiskii. Consistent online optimization: Convex and submodular. In *International Conference on Artificial Intelligence and Statistics*, pages 2241–2250. PMLR, 2019. URL: https://proceedings.mlr.press/v89/jaghargh19a.html.

32 Sanchit Kalhan, Konstantin Makarychev, and Timothy Zhou. Correlation clustering with local objectives. *Advances in Neural Information Processing Systems*, 32:9346–9355, 2019. URL: https://dl.acm.org/doi/10.5555/3454287.3455125.

33 Haim Kaplan, David Naori, and Danny Raz. Competitive analysis with a sample and the secretary problem. In *Proceedings of the 2020 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2082–2095. SIAM, 2020. doi:10.1137/1.9781611975994.128.

34 Haim Kaplan, David Naori, and Danny Raz. Online weighted matching with a sample. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1247–1272. SIAM, 2022. doi:10.1137/1.9781611977073.52.

35 Jon Kleinberg, Yuval Rabani, and Éva Tardos. Fairness in routing and load balancing. In *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*, pages 568–578. IEEE, 1999. doi:10.1109/SFFCS.1999.814631.

36 Ravi Kumar, Manish Purohit, Aaron Schild, Zoya Svitkina, and Erik Vee. Semi-Online Bipartite Matching. In *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124, pages 50:1–50:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ITCS.2019.50.

37 Silvio Lattanzi, Benjamin Moseley, Sergei Vassilvitskii, Yuyan Wang, and Rudy Zhou. Robust online correlation clustering. *Advances in Neural Information Processing Systems*, 34:4688–4698, 2021. URL: https://proceedings.neurips.cc/paper/2021/hash/250dd56814ad7c50971ee4020519c6f5-Abstract.html.

38 Silvio Lattanzi and Sergei Vassilvitskii. Consistent k-clustering. In *International Conference on Machine Learning*, pages 1975–1984. PMLR, 2017. URL: http://proceedings.mlr.press/v70/lattanzi17a.html.

39 Claire Mathieu, Ocan Sankur, and Warren Schudy. Online correlation clustering. In *27th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 573–584. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2010. doi:10.4230/LIPIcs.STACS.2010.2486.

40 Gregory J Puleo and Olgica Milenkovic. Correlation clustering with constrained cluster sizes and extended weights bounds. *SIAM Journal on Optimization*, 25(3):1857–1872, 2015. doi:10.1137/140994198.

41 Gregory J. Puleo and Olgica Milenkovic. Correlation clustering and biclustering with locally bounded errors. In *International Conference on Machine Learning*, pages 869–877. PMLR, 2016. URL: http://proceedings.mlr.press/v48/puleo16.html.

42 Roy Schwartz and Roded Zats. Fair correlation clustering in general graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022)*, pages 37:1–37:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.APPROX/RANDOM.2022.37.