OPTIMAL TRADEOFFS FOR CORRELATION CLUSTERING COST ESTIMATION IN NODE-ARRIVAL STREAMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large collections of embeddings induce similarity graphs whose edges can be computed on demand rather than stored explicitly. We study estimation of the optimal correlation clustering cost in the node-arrival streaming model. Here, the nodes themselves are the data objects arriving sequentially, and edges can be implicitly derived between stored nodes. We present ONCE and TWICE, our one- and two-pass algorithms, respectively. For an n-node graph, additive error Δ, and constant $0 < \epsilon < 1$, ONCE gives a $(3+\epsilon, \Delta)$-approximation using $O(n^2/\Delta)$ words, and TWICE gives a $(5+\epsilon, \Delta)$-approximation using $\widetilde{O}(n/\sqrt{\Delta})$ words. We prove matching lower bounds that characterize the tradeoff between space and additive error in the one- and constant-pass regimes, up to polylogarithmic factors and the representation size of the nodes. Experiments on 100 million image embeddings show that ONCE and TWICE give accurate estimates while storing less than 2% of the nodes.

1 INTRODUCTION

In the fundamental Correlation Clustering (Bansal et al., 2004; Ailon et al., 2008) problem, the input is a set V of n distinctly identified objects, and a symmetric binary *similarity oracle* which, upon query, declares nodes u and v *similar*, written $u \sim v$, or *dissimilar*, written $u \not\sim v$. The aim is to output a partition $\mathcal{C}$ of V into *clusters* that minimizes the number of *disagreements*: pairs that are similar but clustered separately (*false positives*), or dissimilar but clustered together (*false negatives*).

Correlation clustering turns pairwise similarity judgments into a globally consistent grouping without specifying the number of clusters in advance (Bansal et al., 2004). This is useful when similarities come from noisy classifiers or thresholded embeddings: two objects can each be similar to a third without being similar to one another. Minimizing disagreements is NP-hard and even APX-hard (Charikar et al., 2005), but admits constant-factor approximations. In particular, the randomized PIVOT algorithm of Ailon et al. (2008) produces a clustering whose expected cost is at most three times the optimal cost.

We are interested in massive similarity graphs defined implicitly by the data, such as those induced by datasets of embeddings. Yet, the vast majority of work in graph streams focuses on the edge-arrival streaming model, in which the edges appear sequentially. Building the explicit set of edges requires $\Theta(n^2)$ similarity queries, and even worse, could require $\Theta(n^2)$ *disk space* to store if the graph is dense. For example, the 256GB SimSearchNet++ dataset Big ANN Benchmarks (2021) has one billion vectors, so writing down all pairs of similar vectors could take up $\binom{10^9}{2}$ bits $= 62.5$PB on disk; over $244,000\times$ larger than the original dataset! These sorts of costly precomputations are exactly what the streaming model seeks to avoid.

We study the node-arrival streaming model of Liu et al. (2026). During a pass, the objects of V arrive one at a time, in the same order across passes. We can query similarity between any two objects we currently have in memory. Our goal is to minimize the *space complexity* of our algorithms. By convention, space is measured in words, where each object fits into a single word. We assume n is known before the stream.

Rather than producing a clustering, which could require an $\Omega(n)$ output, we study how much space is needed to *estimate* the minimum clustering cost, i.e. the clusterability of V under the similarity oracle (Assadi et al., 2023; Liu et al., 2026). The optimal cost is the fewest pairwise judgments that

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Space–error tradeoffs for arbitrary $c \geq 1$, fixed $0 < \epsilon < 1$, and constant success probability. Our lower bounds hold for $1 \leq \Delta \leq n^2/64$.

Result	Passes	Approximation	Space
Liu et al. (2026, Theorem 3.1)	1	$(c, n/3)$	$\Omega(n)$ bits
ONCE (Theorem 2.1)	1	$(3+\epsilon, \Delta)$	$O(\min\{n, n^2/\Delta\})$ words
This work (Theorem 4.1)	1	(c, Δ)	$\Omega(\min\{n, n^2/\Delta\})$ bits
C^{4}APPROX (Liu et al., 2026, Theorem 2.11)	$k+4$	$(9+\frac{24}{k-1}, \Delta)$	$\widetilde{O}(n/\Delta^{1/4})$ words
Liu et al. (2026, Theorem 3.2)	$O(1)$	$(c, 0)$	$\Omega(n)$ bits
TWICE (Theorem 3.1)	2	$(5+\epsilon, \Delta)$	$\widetilde{O}(n/\sqrt{\Delta})$ words
This work (Theorem 4.2)	$O(1)$	(c, Δ)	$\Omega(n/\sqrt{\Delta})$ bits

must be changed to make the similarity relation agree with a partition. It therefore measures the global inconsistency of the similarity rule: a small cost means that most judgments can be explained by a single clustering, while cost zero means that the similarity graph is a disjoint union of cliques.

Let OPT denote the optimal correlation clustering cost. We call Z a (c, Δ)-*approximation* of OPT if it satisfies $\mathrm{OPT} \leq Z \leq c\,\mathrm{OPT} + \Delta$, and if $\Delta = 0$, Z is a c-*approximation* of OPT. We call Z a $(1 \pm \epsilon, \pm\Delta)$-*approximation* of M if $|Z - M| \leq \epsilon M + \Delta$.

The C^{4}APPROX algorithm of Liu et al. (2026) demonstrates that it is possible to obtain $(O(1), o(n))$-approximations using sublinear space and a constant number of passes.

In this work, we are interested in the relationship between pass count, space, and *additive* error for this problem. How accurately can we estimate OPT using only one pass, and how much space does it require? If we allow more passes, allowing us to revisit discarded objects, how much space can we save? How many more passes are sufficient to attain those savings?

1.1 OUR CONTRIBUTIONS

We characterize the tradeoff between space and additive error for correlation clustering cost estimation in the one- and constant-pass regimes, up to polylogarithmic factors and word size. Table 1 summarizes our guarantees and prior work. Throughout this comparison, $0 < \epsilon < 1$ is fixed, success probabilities are 0.99, and $c \geq 1$ is arbitrary.

One-pass regime. Our algorithm ONCE uses one node-arrival pass and $O(\min\{n, n^2/\Delta\})$ words of space to obtain a $(3+\epsilon, \Delta)$-approximation (Theorem 2.1). We prove a matching $\Omega(\min\{n, n^2/\Delta\})$ bit lower bound for any one-pass (c, Δ)-approximation (Theorem 4.1).

Constant-pass regime. Our algorithm TWICE uses only two node-arrival passes and $\widetilde{O}(n/\sqrt{\Delta})$ words of space to obtain a $(5+\epsilon, \Delta)$-approximation (Theorem 3.1). We prove a matching $\Omega(n/\sqrt{\Delta})$ bit lower bound for any constant-pass (c, Δ)-approximation (Theorem 4.2).

For $n \leq \Delta \leq n^2/64$, a second pass improves the space bound from $O(n^2/\Delta)$ to $\widetilde{O}(\sqrt{n^2/\Delta})$, a quadratic improvement up to logarithmic factors. For example, when $\Delta = \Theta(n)$, ONCE uses $O(n)$ words while TWICE uses $\widetilde{O}(\sqrt{n})$ words.

We also show that prior knowledge of n is necessary to achieve the one-pass tradeoff (Theorem 4.3), ruling out the existence of "doubling" algorithms which dynamically grow with the input size.

Experiments We evaluate ONCE, TWICE, and C^{4}APPROX on similarity graphs from ImageNet and SimSearchNet++ embeddings, examining how their outputs vary with the retained-object budget. We compare fixed-priority trials and independent-priority repetitions against exact clustering costs, distinguishing sampling error from differences between the target clusterings.

Ideas behind TWICE TWICE uses the minimum-neighbor clustering of Behnezhad et al. (2023). Each node receives a random priority and chooses the minimum-priority node in its closed neighborhood. A node choosing itself is a pivot and forms a cluster with all nodes choosing it. A small

reference sample determines many of the cluster labels outright. For the remaining nodes, we partition the disagreements based on their relative ordering in the stream. If a sampled node's chosen neighbor occurs after it, we collect it during the first pass and eagerly check if it's a pivot in the second while simultaneously verifying the candidate. This fails when a pair shares a chosen neighbor which occurs before both endpoints in the stream, shown in Figure 1. For these remaining disagreements we instead sample the center. Doing so naively recovers the $\widetilde{O}(n/\Delta^{1/4})$ behavior of C^{4}APPROX, as a sampled center can require a large neighborhood record and can have a large rescaled contribution. TWICE avoids both losses by maintaining a constant-size local record for each center and sampling centers inversely proportional to their random priorities. We introduce a priority bound that allows us to relate these sampling priorities to degrees, allowing us to control the overall contribution.

1.2 RELATED WORK

Edge-stream algorithms construct approximate clusterings (Ahn et al., 2015; Assadi & Wang, 2022; Behnezhad et al., 2023) or estimate the optimal cost in sublinear space with multiplicative and additive error (Assadi et al., 2023); our node-arrival model follows Liu et al. (2026). Garcia-Soriano et al. (2020) study query-efficient correlation clustering and supply ONCE's pivot-sample construction, while Garg et al. (2026) prove query lower bounds under memory constraints, including for cost estimation with random queries. These query models differ from our restriction to objects currently in memory. Implicit vertex streams also arise in independent-set algorithms (Cormode et al., 2019).

Related streaming statistics include similarity-join and self-join sizes (Rafiei & Deng, 2020), which count similar pairs, and frequency moments under inconsistent similarity judgments (Liu & Zhang, 2026). The latter connect correlation clustering cost to the fewest pairwise mismatches over possible ground truths.

Notation We write $N(u) = \{v \in V \setminus \{u\} \mid u \sim v\}$ and $\deg(u) = |N(u)|$, and use the convention $u \sim u$. For a clustering $\mathcal{C}$, let $E_\oplus$ be its set of unordered pairs of distinct nodes that disagree, so $\mathrm{cost}(\mathcal{C}) = |E_\oplus|$, and let $\deg_\oplus(u)$ count the disagreements incident to u. For either clustering construction below, $\mathrm{pivot}(u)$ denotes the pivot labeling u's cluster, with $\mathrm{pivot}(u) = u$ for singletons. Thus, two nodes share a cluster exactly when their pivot labels agree, and $(u, v) \in E_\oplus$ exactly when $(u \sim v) \oplus (\mathrm{pivot}(u) = \mathrm{pivot}(v))$.

2 A ONE-PASS ALGORITHM

Here, we present our algorithm ONCE (**O**ne-pass **N**ode-arrival **C**lustering cost **E**stimator), which provides a $(3+\epsilon, \Delta)$-approximation using $O(n^2/\Delta)$ words, for any constant $0 < \epsilon < 1$.

Theorem 2.1. *Let $0 < \epsilon < 1$ and $0 < \Delta < n^2$. There exists an algorithm that, with probability at least 0.99, gives a $(3+\epsilon, \Delta)$-approximation of OPT in one node-arrival pass, using $O(n^2/\epsilon^3\Delta)$ words of space.*

Consider the following clustering due to Garcia-Soriano et al. (2020).

Definition 2.2 (Algorithm 3 of Garcia-Soriano et al. (2020))**.** *Fix a uniformly random ordered sample $R = (w_1, \ldots, w_t)$ of t distinct nodes from V, and let P be the (initially empty) ordered sequence of pivots. Initially, all nodes are unclustered. Process $w_1, \ldots, w_t$ in order: if w_i is unclustered, mark it as a pivot by appending it to P, and form its cluster with all of its currently unclustered neighbors. All remaining unclustered nodes are singletons. We call this clustering $\mathcal{C}_{\mathrm{QECC}}(R)$. For $u \in V$, define $\mathrm{pivot}(u)$ as the first node of P that is adjacent to u, or u if $P \cap N(u) = \varnothing$.*

Theorem 2.3 (Theorem 3.4 of Garcia-Soriano et al. (2020))**.** *For a uniformly random ordered sample R of $\min\{n, \lceil n^2/\Delta \rceil\}$ nodes, the expected cost of $\mathcal{C}_{\mathrm{QECC}}(R)$ is at most $3\,\mathrm{OPT} + \Delta$.*

During the pass, we maintain two independent samples R and S. R determines the clustering $\mathcal{C}_{\mathrm{QECC}}(R)$, while S estimates its cost. After the pass, we recover the pivots P from R, and use them to assign cluster labels to each node in S. With the cluster labels in hand, we directly count the disagreements in the subgraph induced by S and rescale the result to obtain our estimate. A variance calculation (given in Appendix B.1) shows that S is sufficiently large to estimate $|E_{\oplus}|$ during a single instance of Algorithm 1.

Lemma 2.4. *Let $0 < \epsilon < 1$ and $0 < \Delta < n^2$, and fix $R \subseteq V$. With probability at least $1 - \epsilon$ over the sampling of S, Algorithm 1 outputs a $(1 \pm \epsilon, \pm\Delta)$-approximation of $\mathrm{cost}(\mathcal{C}_{\mathrm{QECC}}(R))$ in one node-arrival pass, using $O(n^2/\epsilon^2\Delta)$ words of space.*

Algorithm 1 A single run of ONCE(ϵ, Δ)

$t \leftarrow \min\{n, \lceil n^2/\Delta \rceil\}$
$p \leftarrow \min\{1, 2n/(\epsilon^2\Delta)\}$
During pass: concurrently
reservoir-sample t nodes $R \subseteq V$
independently Bernoulli-sample each node into $S \subseteq V$ with probability p
Postprocessing
$R \leftarrow \mathrm{shuffle}(R), P \leftarrow \varnothing$
for $w_i \in R$ **do**
  if $N(w_i) \cap P = \varnothing$ **then**
    append w_i to P
$X \leftarrow \frac{1}{2} \sum_{u,v \in S} \mathbb{1}\{(u,v) \in E_{\oplus}\}$
return X/p^2

We prove Theorem 2.1 in Appendix B.2 by combining $O(1/\epsilon)$ instances of Algorithm 1. With constant probability, at least one of the clusterings has cost at most $(3 + O(\epsilon))\,\mathrm{OPT} + O(\Delta)$ by Theorem 2.3, and all of the cost estimates succeed, so taking the minimum of the upwards-adjusted estimates gives the desired approximation to OPT. We amplify this to 0.99 by taking the median of a constant number of independent batch outputs, executed in parallel.

3 A TWO-PASS ALGORITHM

Now, we present our algorithm TWICE (**Two**-pass **I**nverse-priority **C**lustering cost **E**stimator), which provides a $(5 + \epsilon, \Delta)$-approximation using $\tilde{O}(n/\sqrt{\Delta})$ words, for any constant $0 < \epsilon < 1$.

Theorem 3.1. *Let $0 < \epsilon < 1$ and $0 < \Delta < n^2$. There exists an algorithm which, with probability at least 0.99, outputs a $(5 + \epsilon, \Delta)$-approximation of OPT in two node-arrival passes and using $O(n \log^2 n/\epsilon^3\sqrt{\Delta})$ words of space.*

Throughout this section, let the *priority function* $\kappa : V \to (0, 1)$ be a fully independent hash function on the node identifiers, which assigns each node a *priority* sampled from $\mathrm{Unif}(0, 1)$. The priorities are distinct with probability 1. Let π_κ be the induced rank permutation. That is, $\pi_\kappa(u) < \pi_\kappa(v)$ is equivalent to $\kappa(u) < \kappa(v)$. In analyzing a single run, we fix κ and take probabilities and expectations over the remaining sampling randomness, unless stated otherwise.

We will employ the following clustering due to Behnezhad et al. (2023). This clustering rule is local, in the sense that two hops are enough to decide a node's fate.

Definition 3.2 (Algorithm 1 of Behnezhad et al. (2023))**.** *Fix a uniformly random rank permutation $\pi : V \to [n]$, and let $\eta(u) = \arg\min_\pi(\{u\} \cup N(u))$ be the* minimum neighbor *of $u \in V$.*

Each node w with $\eta(w) = w$ is a pivot. A node u is clustered with $\eta(u)$ if $\eta(u)$ is a pivot, i.e. if $\eta(u) = \eta(\eta(u))$. Otherwise, if $\eta(u) \neq \eta(\eta(u))$, u is a singleton. We call this clustering $\mathcal{C}_{\mathrm{BCMT}}(\pi)$.

Theorem 3.3 (Theorem 3.1 of Behnezhad et al. (2023))**.** $\mathbb{E}_\pi[\mathrm{cost}\,(\mathcal{C}_{\mathrm{BCMT}}(\pi))] \leq 5\,\mathrm{OPT}$.

We study the clustering $\mathcal{C}_{\mathrm{BCMT}}(\pi_\kappa)$ induced by κ. Since π_κ orders nodes by their priorities,

$$\eta(u) = \arg\min_\kappa(\{u\} \cup N(u)).$$

Intuitively, a minimum neighbor should have small priority, especially when the node has many neighbors, as each additional neighbor is an additional opportunity to see a small priority. The following bound makes this precise for all nodes simultaneously. Its proof is in Appendix B.6.

Lemma 3.4 (Priority bound)**.** *With probability at least $1 - 1/n$ over κ, simultaneously for all $u \in V$,*

$$\kappa(\eta(u)) \leq \frac{2 \log n}{\deg(u) + 1}. \tag{1}$$

Now, we will fix κ, fixing $\mathcal{C}_{\text{BCMT}}(\pi_\kappa)$ and $E_\oplus$, and condition on priority bound (1) holding. We first describe a two-pass estimator for $\text{cost}\,(\mathcal{C}_{\text{BCMT}}(\pi_\kappa))$, for a given $0 < \epsilon < 1$ and $0 < \Delta < n^2$. Our target space complexity for a single instance is $O(n \log^2 n/\epsilon^2\sqrt{\Delta})$, which we will control with our global sampling parameter $p = \min\{1, \frac{40\log(n)}{\epsilon^2\sqrt{\Delta}}\}$. It will be useful to interpert $\sqrt{\Delta}$ as a degree threshold, so we will write $D = \sqrt{\Delta}$, so $p = \widetilde{\Theta}(1/D)$. Our final algorithm combines $\Theta(1/\epsilon)$ independent runs of this estimator to obtain the guarantee of Theorem 3.1.

3.1 Reference partition

We first collect a *reference sample* R during the first pass to recover a subset of cluster assignments. Similar to ONCE, our goal is to be do cluster label recovery "offline", where holding u and R in memory is enough to recover $\text{pivot}(u)$. Given that $\text{pivot}(u)$ depends on only $\eta(u)$ and $\eta(\eta(u))$, and that $\kappa(u) \geq \kappa(\eta(u)) \geq \kappa(\eta(\eta(u)))$, simply taking R to be all nodes with priority below a cutoff p is sufficient to recover $\text{pivot}(u)$ for all u with $\kappa(\eta(u)) < p$. That is to say, we can recover $\text{pivot}(u)$ for any node with a (closed) neighbor in R.

Definition 3.5 (Reference partition)**.** *Let $R = \{u \in V \mid \kappa(u) < p\}$ be the* reference sample*, and let $A \uplus B$ be the* reference partition*, given by $A = R \cup N(R)$ and $B = V \setminus A$.*

Having the cluster labels for A allows us to detect disagreements between pairs inside of A, but what about pairs with only one endpoint in A? We prove in Appendix B.4 that all such pairs must be clustered separately, as nodes $u \in B$ have no neighbors in R, and there are no pivots in $A \setminus R$, so we must have $\text{pivot}(u) \in B$. Using this, we show in Appendix B.5 that we can indeed detect disagreements for any pair touching A.

Lemma 3.6. *Each cluster is entirely contained within A or B.*

Lemma 3.7. *When R and the queried nodes are stored, we can test membership in A, compute $\text{pivot}(u)$ for $u \in A$, and test disagreements for pairs touching A.*

Let E_A contain the disagreements with at least one endpoint in A, and let $E_B = E_\oplus \setminus E_A$.

3.2 Estimating residual disagreements

We'll first consdier the disagreements E_B among nodes in B, whose cluster labels are not recovered by R. This part of the graph is necessarily sparse, given that it is disconnected from R, because R is a random node sample when κ is sampled uniformly. We show this in Appendix B.6 using the priority bound (1), which gives a degree bound of D for any node u with $\kappa(\eta(u)) \geq p = \widetilde{\Theta}(1/D)$.

Corollary 3.8. *When (1) holds, for all nodes $u \in B$, $\deg(u) < \epsilon^2 D \leq D$ and $\deg_\oplus(u) < 2D$.*

This degree bound makes endpoint sampling look promising! Let's sample $u, v \in B$ in the first pass, and determine $\eta(u)$ and $\eta(v)$ in the second pass. Right off the bat, if $\eta(u) \neq \eta(v)$, clearly u and v are clustered separately. Otherwise, if $\eta(u) = \eta(v)$, we need to determine if $\eta(u)$ is a pivot.

If $\eta(u) \in A$, Lemma 3.6 guarantees that u and v are not clustered with $\eta(u)$, so they must be singletons, clustered separately. But if $\eta(u) \in B$, we do indeed need to know if $\eta(u)$ is a pivot or not, meaning we need to find $\eta(\eta(u))$. Yet by the time we've determined $\eta(u)$ in the second pass, some of $\eta(u)$'s neighbors may have already occured, which might include $\eta(\eta(u))$, shown in Figure 1.

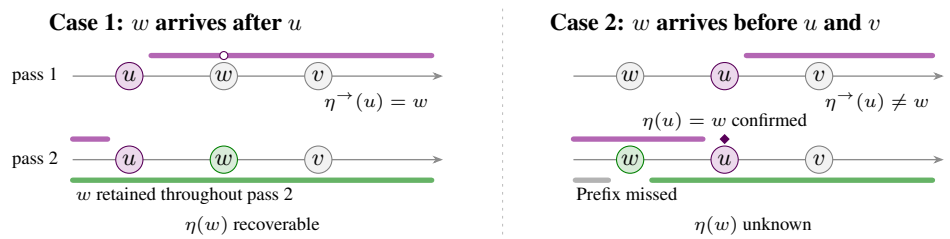

Figure 1: Arrival order determines whether a disagreement $(u, v) \in E_B$ belongs to E_{pair} or E_{center}, when they share a minimum neighbor $w = \eta(u) = \eta(v)$.

While *determining* $\eta(u)$ may require looking at the entire stream, if we happen to *collect* $\eta(u)$ into memory by the end of the first pass, we can begin computing $\eta(\eta(u))$ before we've determined the final value of $\eta(u)$. This motivates the following definition.

Definition 3.9 (Forward neighbors)**.** *Define $\tau_u \in [n]$ to be the position in the stream where u appears. Let $N^{\to}(u) = \{v \in N(u) \mid \tau_u < \tau_v\}$ be u's* forward neighborhood*, $\deg^{\to}(u) = |N^{\to}(u)|$ be its* forward degree*, and $\eta^{\to}(u) = \arg\min_{\kappa}(\{u\} \cup N^{\to}(u))$ its analogous* minimum forward neighbor*.*

For each sampled $u \in B$, we can maintain $\eta^{\to}(u)$ in the first pass immediately after sampling u, and in the second pass, we simultaneously determine $\eta(u)$ and $\eta(\eta^{\to}(u))$. In cases where $\eta^{\to}(u)$ is indeed $\eta(u)$, we've recovered $\eta(\eta(u))$ without a third pass!

Order the endpoints $u, v \in B$ so that $\tau_u < \tau_v$. If their common minimum neighbor $w \in B$ arrives at or after u ($\tau_u \le \tau_w$), then $\eta^{\to}(u) = w$, and the record above determines whether w is a pivot. The only remaining case is when w appears before both endpoints ($\tau_w < \tau_u < \tau_v$). For these, we flip the sampling order: we'll sample the centers w during the first pass, so we encounter $u, v \in N^{\to}(w)$ among w's forward neighbors, and find each of their minimum neighbors in the second pass.

We therefore partition E_B into $E_{\text{center}} \uplus E_{\text{pair}}$, each given by

$$E_{\text{center}} = \{(u,v) \in E_B \mid \eta(u) = \eta(v) \in B, \tau_{\eta(u)} < \min\{\tau_u, \tau_v\}\}, \qquad E_{\text{pair}} = E_B \setminus E_{\text{center}}.$$

We estimate E_{pair} by sampling the endpoints first, and conversely, we estimate E_{center} by sampling the centers first. It seems we've cracked the case!

3.2.1 Space-efficient center sampling

The casework, at least. A center $w \in B$ can still have $O(D)$ forward neighbors. If we retain them all, as in C^{4}Approx, we pay up to $O(D)$ additional space for each sampled center. Worse still, there can be $O(D^2) = O(\Delta)$ possible disagreements among one center's forward neighbors. Sampling centers uniformly at rate p gives only an $O(D^2/p)$ rescaled contribution bound: too weak for our analysis at $\Delta = D^2$. Center samples with constant-size records and bounded rescaled contributions are the two ingredients behind the space improvement from C^{4}Approx's $\widetilde{O}(n/\Delta^{1/4})$ to Twice's $\widetilde{O}(n/\sqrt{\Delta})$.

Suppose the center $w \in B$ was sampled with probability $q(w)$, and that w is a pivot. We can sample a uniform pair $\{a_w, b_w\}$ from $N^{\to}(w)$ in the first pass, ensure that $\eta(a_w) = \eta(b_w) = w$ and $a_w \not\sim b_w$ in the second pass, and output the rescaled contribution $\binom{\deg^{\to}(w)}{2}/q(w) = O(\deg(w)^2/q(w))$. This is a constant-size record for pivots, provided that we can control the term $\deg(w)^2/q(w)$.

At the time we encounter w, we only have its priority. Since a pivot is its own minimum neighbor, the priority bound guarantees that high degree pivots have small priorities. We therefore define the *center-sampling rate*

$$q(w) = \frac{p}{\kappa(w)}, \qquad w \notin R. \tag{2}$$

For a pivot $w \in B$, we have $q(w) = \Omega(\deg(w)/D)$, giving the desired $\deg(w)^2/q(w) = O(D^2) = O(\Delta)$ rescaled contribution bound.

This argument doesn't carry over to the false positive case, where w is not a pivot, since $\eta(w) \neq w$. But for an endpoint u with $\eta(u) = w$, we do still have $\deg(u)/q(w) = O(D)$. We can introduce the term $\deg(u)$ into the rescale weight by sampling out of $N^{\to}(u)$, for $u \in N^{\to}(w)$.

Because false positives are adjacent, we employ a two-level sampling procedure: sample one of w's forward neighbors $c_w \in N^{\to}(w)$, then sample one of c_w's forward neighbors $d_w \in N^{\to}(c_w)$. (c_w, d_w) forms a false positive centered at w exactly when $\eta(c_w) = \eta(d_w) = w$, so the priority bound controls $\deg(c_w)/q(w) = O(D)$. We still have $\deg(w) = O(D)$ as $w \in B$, giving our desired $\deg(w)\deg(c_w)/q(w) = O(D^2)$ bound.

Both cases use the following consequence of the priority bound, proved in Appendix B.6.

Corollary 3.10. *When (1) holds, for every $u \in B$,*

$$\frac{1}{q(\eta(u))} \le \frac{\epsilon^2 D}{\deg(u)+1}.$$

Definition 3.11 (Center record)**.** *For each $w \notin R$, the* center record $\textsc{Rec}(w) = (a_w, b_w, c_w, d_w)$ *consists of four node random variables sampled as follows:*

- $\{a_w, b_w\}$ *sampled uniformly from* $\binom{N^{\rightarrow}(w)}{2}$,
- c_w *sampled uniformly from* $N^{\rightarrow}(w)$,
- d_w *sampled uniformly from* $N^{\rightarrow}(c_w)$, *conditional on* c_w,

If a required sample cannot be drawn, assign it $\perp$.

During the first pass, we sample each node $w \notin R$ into C with probability $q(w)$, and maintain the center record $\textsc{Rec}(w)$ and counters for $\deg^{\rightarrow}(w)$ and $\deg^{\rightarrow}(c_w)$ for each $w \in C$. In the second pass, we find the minimum neighbor for each $w \in C$ and all four record endpoints in $\textsc{Rec}(w)$. Afterwards, we know whether or not w is a pivot, so we test the appropriate endpoint pair.

For $w \in B$, define the false negative estimator Z_w^- and false positive estimator Z_w^+ by

$$\begin{aligned} Z_w^- &= \mathbb{1}\{\eta(w) = w, \eta(a_w) = \eta(b_w) = w, a_w \not\sim b_w\}\binom{\deg^{\rightarrow}(w)}{2}, \\ Z_w^+ &= \mathbb{1}\{\eta(w) \neq w, \eta(c_w) = \eta(d_w) = w, c_w \sim d_w\}\deg^{\rightarrow}(w)\deg^{\rightarrow}(c_w). \end{aligned} \tag{3}$$

In Appendix B.10, we show that these rescaled contributions are indeed bounded by $O(D^2)$.

Lemma 3.12. *For all $w \in B$, $(Z_w^+ + Z_w^-)/q(w) \leq \epsilon^4 D^2$.*

Then, the overall, unbiased E_{center} estimator is

$$Z_{\text{center}} = \sum_{w \in C \cap B} \frac{Z_w^- + Z_w^+}{q(w)}.$$

We use Lemma 3.12 to control the variance to prove the concentration guarantee of Z_{center}, also found in Appendix B.10.

Lemma 3.13. *Z_{center} is a $(1 \pm \epsilon, \pm\epsilon\Delta)$-approximation of $|E_{\text{center}}|$ with probability at least $1 - \epsilon^2/4$.*

3.2.2 Estimating endpoint disagreements

Now, we consider the disagreements E_{pair}. First, orient the pairs $(u, v) \in E_{\text{pair}}$ such that $\tau_u < \tau_v$. We estimate these using two endpoint samples, S and T, where we count $(u, v) \in E_{\text{pair}}$ when $u \in S$ and $v \in T$. During the first pass, we independently include each node u in S with probability p, and independently do the same for T. For each $u \in S$, we also maintain $\eta^{\rightarrow}(u)$ during the first pass. During the second pass, we find the minimum neighbors of all nodes retained in the first pass. The following lemma shows these endpoint records are sufficient to recover $\mathbb{1}\{(u, v) \in E_{\text{pair}}\}$, which we prove in Appendix B.8.

Lemma 3.14 (Pair recoverability)**.** *For $u, v \in V$ with $\tau_u < \tau_v$, suppose that $\eta(u), \eta(v), \eta^{\rightarrow}(u)$, and $\eta(\eta^{\rightarrow}(u))$ have all been computed and that u, v, and R also are stored. Then, membership of (u, v) in E_{pair} can be determined.*

Now, we define the E_{pair} estimator

$$Z_{\text{pair}} \leftarrow \frac{1}{p^2} \sum_{u \in S} \sum_{v \in T} \mathbb{1}\{\tau_u < \tau_v, (u, v) \in E_{\text{pair}}\}.$$

Weighting by $1/p^2$ makes Z_{pair} an unbiased estimate of $|E_{\text{pair}}|$, and by Corollary 3.8, each node in B is incident to only $O(D)$ disagreements, yielding the following concentration guarantee, proven in Appendix B.9.

Lemma 3.15. *With probability at least $1 - 5/n$, Z_{pair} is a $(1 \pm \epsilon, \pm\epsilon\Delta)$-approximation of $|E_{\text{pair}}|$.*

Algorithm 2 A single run of TWICE$(\kappa, \epsilon, \Delta)$

$D \leftarrow \sqrt{\Delta}$
$p \leftarrow \min\{1, 40\log(n)/(\epsilon^2 D)\}$
During pass 1: concurrently
collect $R \leftarrow \{u \in V; \kappa(u) < p\}$
sample each node independently into each of S, T with probability p
maintain $\eta^{\rightarrow}(u)$ for every $u \in S$
sample each $w \notin R$ independently into C with probability $q(w)$
for each $w \in C$, maintain REC(w) and counters $\deg^{\rightarrow}(w), \deg^{\rightarrow}(c_w)$
During pass 2: concurrently
find $\eta(u)$ for all nodes kept from pass 1
accumulate $X \leftarrow p^{-1}\sum_{u\in V} X_u \mathbb{1}\{X_u \geq pD\}$
accumulate $Y \leftarrow p^{-1}\sum_{u\in V} Y_u$
Postprocessing
$Z_A \leftarrow (X+Y)/2$
$Z_{\text{pair}} \leftarrow p^{-2}|\{(u,v) \in S \times T \mid \tau_u < \tau_v,\ (u,v) \in E_{\text{pair}}\}|$
$Z_{\text{center}} \leftarrow \sum_{w\in C\cap B}(Z_w^- + Z_w^+)/q(w)$
return $Z_A + Z_{\text{pair}} + Z_{\text{center}}$

3.3 ESTIMATING REFERENCE-RECOVERED DISAGREEMENTS

Our last task is to estimate $|E_A|$, reusing S and T. Unlike the residual disagreements E_B, we have no comparable bound for a node's E_A-degree. We'll use S to estimate the large E_A-degrees, and T to estimate the total contribution of small E_A-degree nodes.

For a node u, let $\deg_A(u)$ denote u's E_A-degree, and define X_u and Y_u as

$$X_u = |\{v \in S \mid (u,v) \in E_A\}|, \qquad Y_u = |\{v \in T \mid (u,v) \in E_A, X_v < pD\}|. \tag{4}$$

We'll use S to estimate $\deg_A(u)$ via the pilot estimates X_u/p, and partition the nodes into $H = \{u \in V \mid X_u \geq pD\}$ and $L = \{u \in V \mid X_u < pD\}$ based to their estimated E_A-degree. With high probability, the pilot estimates X_u/p are simultaneously accurate for all $u \in H$, while the remaining nodes $u \in L$ with small pilot estimates X_u/p have bounded $\deg_A(u) = O(D)$.

After the first pass, we have R, S, and T in memory, so Lemma 3.7 allows us to calculate X_u and thus Y_u for any u in memory. During the second pass, we accumulate the sum X of large pilot estimates,

$$X = \frac{1}{p}\sum_{u\in V} X_u \cdot \mathbb{1}\{X_u \geq pD\} = \sum_{u\in H}\frac{X_u}{p},$$

to estimate the contribution $\sum_{u\in H}\deg_A(u)$. We estimate the remainder, $\sum_{u\in L}\deg_A(u)$ using the accumulated sum Y, where

$$Y = \frac{1}{p}\sum_{u\in V} Y_u = \frac{1}{p}\sum_{u\in L}\deg_A(u)\cdot\mathbb{1}\{u \in T\}.$$

Crucially, T is sampled during the first pass, before L is known. After the first pass, R and S are determined, so Lemma 3.7 allows us to filter out $T \cap L$. Since L depends on S, not T, this has the same distribution as sampling directly from L, saving us a sampling pass.

Each edge is counted twice in the degree sum, so our estimate is $(X+Y)/2$. All that remains is the concentration, which we prove in Appendix B.7.

Lemma 3.16. *With probability at least* $1 - 5/n$, $\frac{X+Y}{2}$ *is a* $(1 \pm \epsilon, \pm\epsilon\Delta)$*-approximation of* $|E_A|$.

3.4 STREAMING IMPLEMENTATION

It remains to bound how many records we retain. Taking expectation over both the priorities κ and the sampling randomness, we have $\mathbb{E}[|R|] = \mathbb{E}[|S|] = \mathbb{E}[|T|] = np, \mathbb{E}[|C|] = np\log(1/p)$. Thus, our center sampling rate $q(w) = p/\kappa(w)$ costs only a $\log(1/p) = O(\log n)$ factor. Concentration bounds over the sample sizes gives the target space usage below.

Combining this space bound with the three estimator guarantees gives the full single instance guarantee, which we show in Appendix B.11.

Lemma 3.17 (Two-pass implementation)**.** *Let* $0 < \epsilon < 1$ *and* $0 < \Delta < n^2$*, and let* κ *assign independent uniform priorities to the nodes. With probability at least* $1 - \epsilon^2/4 - 15/n$ *over* κ *and the sampling randomness, Algorithm 2 outputs a* $(1 \pm \epsilon, \pm 3\epsilon\Delta)$*-approximation of* $\text{cost}(\mathcal{C}_{\text{BCMT}}(\pi_\kappa))$ *in two node-arrival passes, using* $O(n\log^2 n/\epsilon^2\sqrt{\Delta})$ *words of space.*

We can now prove Theorem 3.1 by combining $O(1/\epsilon)$ instances of Algorithm 2. With constant probability, at least one of the clusterings has cost at most $(5 + O(\epsilon))\,\mathrm{OPT}$ by Theorem 3.3, and all of the cost estimates succeed, so taking the minimum of the upwards-adjusted estimates gives the desired approximation to OPT. We amplify this to 0.99 by taking the median of a constant number of independent batch outputs, executed in parallel. We give the detailed proof in Appendix B.12.

4 Lower bounds

In this section, we present two space lower bounds matching the bounds of Once and Twice, up to polylogarithmic factors and the representation size of the nodes. Taken together, these complete the characterization of the tradeoff between space and additive error in the one- and constant-pass regimes for the correlation clustering cost estimation problem in the node-arrival streaming setting. The constructions and proofs appear in Appendix B.13.

Theorem 4.1 (One-pass space lower bound)**.** *For any $c \geq 1$ and $\Delta \in [1, n^2/64]$, any one-pass 0.49-error (c, Δ)-approximation streaming algorithm for estimating the optimal correlation clustering cost on an n-node graph in the node-arrival model needs $\Omega\big(\min\{n, n^2/\Delta\}\big)$ bits of space.*

Theorem 4.2 (Multi-pass space lower bound)**.** *For any $c \geq 1$, $\Delta \in [1, n^2/64]$, and integer $p \geq 1$, any p-pass 0.49-error (c, Δ)-approximation streaming algorithm for estimating the optimal correlation clustering cost on an n-node graph in the node-arrival model needs $\Omega\Big(n/(p\sqrt{\Delta})\Big)$ bits of space.*

A natural question is whether there exists a one-pass algorithm achieving the same tradeoff without advance knowledge of n. We rule out this possibility with the following corollary. In the unknown-length setting, the algorithm is given the absolute additive budget Δ but *not* the stream length, and must satisfy its approximation guarantee for every admissible final stream length.

Corollary 4.3 (One-pass unknown stream length)**.** *For any $c \geq 1$ and $\Delta \in [1, n^2/64]$, any one-pass 0.49-error (c, Δ)-approximation streaming algorithm for estimating the optimal correlation clustering cost on an n-node graph in the unknown-length node-arrival model needs $\Omega(n)$ bits of space.*

5 Experiments

We compare Once, Twice, and C^{4}Approx (Liu et al., 2026) ($k = 5$) on ImageNet (3,994,197 vectors, cosine threshold 0.70) and SSN++ (10^8 vectors, Euclidean radius 400). Figure 2 aggregates 500 independent-priority trials into 20 median-of-means outputs (protocol and fixed-priority results in Appendix C). Mean Once outputs remain close to Pivot, Twice tracks BCMT, and C^{4}Approx approaches PrunedPivot as space grows.

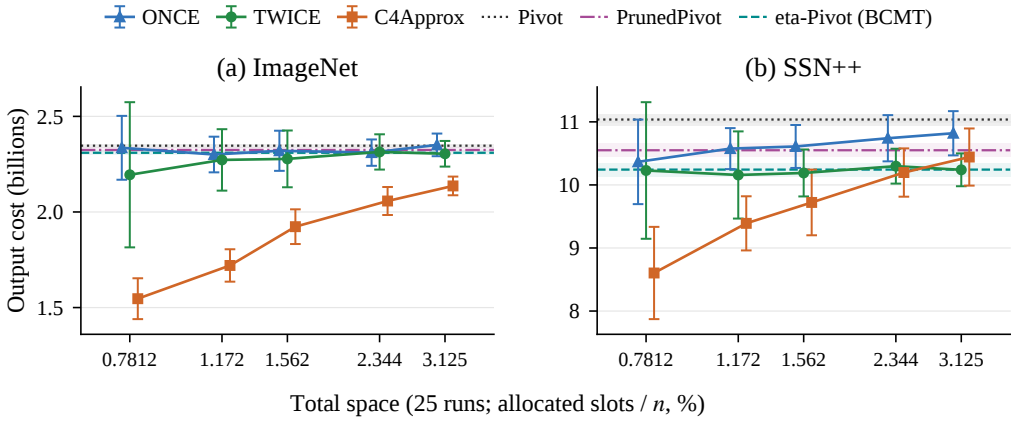

Figure 2: Mean ± sample SD for estimates and exact-cost references. Space sums 25 runs' vector allocations; metadata and workspace excluded. All trials retained.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the writing, searching for related literature, implementing the experiments, and drafting the experimental section. We have not used generative AI tools for drafting sections of the paper (except the experimental section), and generating synthetic datasets is not applicable to this work. We also used generative AI tools to assist with checking and correcting constant factors in the proofs regarding concentration inequalities. We reviewed all AI-assisted work and verified the experimental implementation against previously known benchmarks. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Kook Jin Ahn, Graham Cormode, Sudipto Guha, Andrew McGregor, and Anthony Wirth. Correlation clustering in data streams. In *ICML*, pp. 2237–2246. JMLR.org, 2015.

Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *J. ACM*, 55(5):23:1–23:27, 2008.

Sepehr Assadi and Chen Wang. Sublinear time and space algorithms for correlation clustering via sparse-dense decompositions. In Mark Braverman (ed.), *ITCS*, volume 215 of *LIPIcs*, pp. 10:1–10:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

Sepehr Assadi, Vihan Shah, and Chen Wang. Streaming algorithms and lower bounds for estimating correlation clustering cost. In *NeurIPS*, 2023.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Mach. Learn.*, 56(1-3): 89–113, 2004.

Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.*, 68(4):702–732, 2004.

Soheil Behnezhad, Moses Charikar, Weiyun Ma, and Li-Yang Tan. *Single-Pass Streaming Algorithms for Correlation Clustering*, pp. 819–849. Society for Industrial and Applied Mathematics, January 2023. ISBN 9781611977554. doi: 10.1137/1.9781611977554.ch33. URL http://dx.doi.org/10.1137/1.9781611977554.ch33.

Big ANN Benchmarks. Billion-scale approximate nearest neighbor search challenge, 2021. URL https://big-ann-benchmarks.com/neurips21.html.

Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. *J. Comput. Syst. Sci.*, 71(3):360–383, 2005.

Graham Cormode, Jacques Dark, and Christian Konrad. Independent sets in vertex-arrival streams. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi (eds.), *ICALP*, volume 132 of *LIPIcs*, pp. 45:1–45:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

David Garcia-Soriano, Konstantin Kutzkov, Francesco Bonchi, and Charalampos Tsourakakis. Query-efficient correlation clustering. In *Proceedings of The Web Conference 2020*, WWW'20, pp. 1468–1478. ACM, April 2020. doi: 10.1145/3366423.3380220. URL http://dx.doi.org/10.1145/3366423.3380220.

Sumegha Garg, Songhua He, and Periklis A. Papakonstantinou. Query lower bounds for correlation clustering under memory constraints. In *17th Innovations in Theoretical Computer Science Conference (ITCS 2026)*, volume 362 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 67:1–67:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026. doi: 10.4230/LIPIcs.ITCS.2026.67.

Jiong Guo. A more effective linear kernelization for cluster editing. *Theoretical Computer Science*, 410(8-10):718–726, March 2009. ISSN 0304-3975. doi: 10.1016/j.tcs.2008.10.021. URL http://dx.doi.org/10.1016/j.tcs.2008.10.021.

Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, Cambridge, 1996. ISBN 9780521560672. doi: 10.1017/CBO9780511574948.

Kaiwen Liu and Qin Zhang. Frequency moments in noisy streaming and distributed data under mismatch ambiguity, 2026. URL https://arxiv.org/abs/2603.11216.

Kaiwen Liu, Seba Daniela Villalobos, and Qin Zhang. Estimating correlation clustering cost in node-arrival stream. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=LplfRFhduF.

Davood Rafiei and Fan Deng. Similarity join and similarity self-join size estimation in a streaming environment. *IEEE Trans. Knowl. Data Eng.*, 32(4):768–781, 2020.

A MATH TOOLS

Lemma A.1 (Chernoff bound). *Let $X_1, \dots, X_n$ be independent random variables such that $X_i \in [0,1]$ for all $i \in [n]$, and let $X = \sum_{i=1}^n X_i$. Then, for any $0 < \epsilon \le 1$,*

$$\mathbb{P}[X \ge (1+\epsilon)\,\mathbb{E}[X]] \le \exp\left(-\frac{\epsilon^2\,\mathbb{E}[X]}{3}\right),$$

$$\mathbb{P}[X \le (1-\epsilon)\,\mathbb{E}[X]] \le \exp\left(-\frac{\epsilon^2\,\mathbb{E}[X]}{2}\right),$$

$$\mathbb{P}[|X - \mathbb{E}[X]| \ge \epsilon\,\mathbb{E}[X]] \le 2\exp\left(-\frac{\epsilon^2\,\mathbb{E}[X]}{3}\right).$$

Lemma A.2 (Bernstein inequality). *Let $X_1, \dots, X_n$ be independent random variables such that $X_i \in [0, M]$ for all $i \in [n]$, and let $X = \sum_{i=1}^n X_i$. Then, for any $t > 0$,*

$$\mathbb{P}[|X - \mathbb{E}[X]| \ge t] \le 2\exp\left(-\frac{t^2}{2M(\mathbb{E}[X] + t/3)}\right)$$

We will also frequently use the following corollary of the Bernstein inequality.

Corollary A.3 (Mixed guarantee Bernstein). *With the same notations as Lemma A.2, for $0 < \epsilon \le 1$ and $\Delta > 0$, we have*

$$\mathbb{P}[|X - \mathbb{E}[X]| \ge \epsilon(\mathbb{E}[X] + \Delta)] \le 2\exp\left(-\frac{\epsilon^2\Delta}{2M}\right)$$

B DELAYED PROOFS

B.1 PROOF OF LEMMA 2.4

Proof. Note that if $\Delta \le 2n/\epsilon^2$, $p = 1$, so $S = V$ and X is exactly $|E_\oplus|$. In this case, $n^2/(\epsilon^2\Delta) \ge n/2$, so storing all nodes fits the space bound and the claim holds. Thus, we assume $\Delta > 2n/\epsilon^2$ and $p = 2n/(\epsilon^2\Delta)$. We will consider the approximation claim and the space claim separately.

Let $Y = X/p^2$ be the algorithm's output, and for an edge $e \in E_\oplus$, let $X_e = \mathbb{1}\{e \subseteq S\}$. Each node is sampled independently, so for any edge $(u,v) \in E_\oplus$, we know $\mathbb{P}[u, v \in S] = p^2$. Thus,

$$X = \frac{1}{2}\sum_{u,v \in S} \mathbb{1}\{(u,v) \in E_\oplus\} = \sum_{(u,v)\in E_\oplus} \mathbb{1}\{u, v \in S\} = \sum_{e \in E_\oplus} X_e,$$

giving that $\mathbb{E}[X \mid R] = p^2|E_\oplus|$ and that Y is unbiased.

Now, we analyze the variance. Each $X_e \in \{0,1\}$ for each $e \in E_\oplus$, so $\mathrm{Var}(X_e \mid R) = p^2(1-p^2) \le p^2$. Consider a pair of distinct edges $e, e' \in E_\oplus$. If e and e' share an endpoint, we have that $\mathrm{Cov}(X_e, X_{e'} \mid R) \le \mathbb{E}[X_e X_{e'}] = p^3$. For a fixed edge e, there are at most $2n$ other edges e' that can share an endpoint with e. On the other hand, if e and e' are disjoint, we have $\mathrm{Cov}(X_e, X_{e'} \mid R) = 0$. Combining these bounds gives

$$\mathrm{Var}(X \mid R) = \sum_{e \in E_\oplus} \mathrm{Var}(X_e \mid R) + \sum_{e,e' \in E_\oplus \mid e \ne e'} \mathrm{Cov}(X_e, X_{e'} \mid R) \le p^2|E_\oplus| + 2np^3|E_\oplus|.$$

Since $np > 2$, we have $\mathrm{Var}(Y \mid R) \le |E_\oplus|/p^2 + 2n|E_\oplus|/p \le 4n|E_\oplus|/p$. We finally apply Chebyshev's inequality,

$$\mathbb{P}[|Y - |E_\oplus|| \ge \epsilon|E_\oplus| + \Delta \mid R] \le \frac{\mathrm{Var}(Y \mid R)}{(\epsilon|E_\oplus| + \Delta)^2} \le \frac{n}{p\epsilon\Delta} \le \frac{\epsilon}{2}.$$

Thus, Y is a $(1 \pm \epsilon, \pm\Delta)$-approximation of $|E_\oplus|$ with probability at least $1 - \epsilon/2$.

The space bound is always respected for R. We know that $\mathbb{E}[|S|] = np = 2n^2/(\epsilon^2\Delta) > 2/\epsilon^2$, so by a Chernoff bound,

$$\mathbb{P}[|S| \ge 3np] \le e^{-2np/3} \le e^{-4/(3\epsilon^2)} \le \epsilon/2.$$

Thus, $|S| = O(n^2/(\epsilon^2\Delta))$ with probability at least $1 - \epsilon/2$.

Both claims hold with probability at least $1 - \epsilon$. □

For a worst-case space guarantee, one may abort a run if storing the next sampled node would make $|S|$ exceed $\lceil 3np \rceil$. This event is covered by the space-failure bound above, so the overall failure bound remains at most ϵ.

B.2 Proof of Theorem 2.1

Proof of Theorem 2.1. Let $\gamma = \epsilon/6$, $k = \lceil 7 \log 8/\epsilon \rceil \leq 16/\epsilon$, $\epsilon_0 = \epsilon/64$, and $\Delta_0 = \Delta/4$.

We will run Algorithm 1 k times in parallel. For the ith independent execution of Once, let R_i denote its ordered node sample and $O_i = \textsc{Once}(\epsilon_0, \Delta_0)$ its output. Let $M_i = \text{cost}\left(\mathcal{C}_{\text{QECC}}(R_i)\right)$, and let $\widehat{M}_i = (O_i + \Delta_0)/(1 - \epsilon_0)$. By a union bound over Lemma 2.4, all $\widehat{M}_i$ are $(\frac{1+\epsilon_0}{1-\epsilon_0}, \frac{2}{1-\epsilon_0}\Delta_0)$-approximations of their respective M_i, and all runs respect their space bounds, with probability at least $1 - k\epsilon_0 \geq 3/4$. The total space usage is $k \cdot O\left(n^2/\epsilon_0^2\Delta_0\right) = O\left(n^2/\epsilon^3\Delta\right)$.

However, all k of the clusterings may have costs that are too high. Let $A = 3\,\text{OPT} + \Delta/4 \geq \mathbb{E}[M_i]$ for all $i \in [k]$. Applying Markov's inequality gives $\mathbb{P}[M_i \geq (1+\gamma)A] \leq \mathbb{E}[M_i]/((1+\gamma)A) \leq 1/(1+\gamma)$, for each $i \in [k]$. Now, all of the k clusterings have cost exceeding $(1+\gamma)A$ with probability at most $\mathbb{P}[\min_{i\in[k]} M_i \geq (1+\gamma)A] \leq (1+\gamma)^{-k} \leq \exp(-k\gamma/(1+\gamma)) \leq 1/8$. Therefore, there exists some $i^* \in [k]$ where $M_{i^*} \leq (1+\epsilon/6)(3\,\text{OPT} + \Delta/4) \leq (3+\epsilon/2)\,\text{OPT} + \Delta/3$ with probability at least $7/8$.

Taking both of these together, we have that

$$\text{OPT} \leq \min_{i\in[k]} \widehat{M}_i \leq \frac{1+\epsilon_0}{1-\epsilon_0}\left((3+\epsilon/2)\,\text{OPT} + \Delta/3\right) + \frac{2}{1-\epsilon_0}\Delta_0 \leq (3+\epsilon)\,\text{OPT} + \Delta$$

with probability at least $5/8$. We boost this probability to 0.99 by taking the median of a constant number of independent batch outputs, all executed in parallel. □

B.3 Minimum neighbors in the reference partition

Fact B.1. *For every $u \in A$, $\eta(u) \in R$.*

Proof. Consider a node $u \in A$. By definition of A, either $u \in R$ or u has a neighbor in R, so $(\{u\} \cup N(u)) \cap R$ is nonempty. Let $v \in (\{u\} \cup N(u)) \cap R$. By definition of η and R, we have $\kappa(\eta(u)) \leq \kappa(v) < p$, so $\eta(u) \in R$. □

B.4 Proof of Lemma 3.6

Proof. It suffices to show that both u and $\text{pivot}(u)$ lie on the same side of the partition for each $u \in V$.

Consider a node $u \in A$. By Fact B.1, $\eta(u) \in R$. Now, because both $u \in A$ and $\eta(u) \in R \subseteq A$, it must be the case that $\text{pivot}(u) \in A$.

Consider a node $u \in B$. If $\eta(u) \in B$, then we immediately have $\text{pivot}(u) \in B$. Suppose $\eta(u) \in A$. By construction of B, u is not in R and has no neighbors in R, so $\eta(u) \notin R$. However, $\eta(\eta(u)) \in R$, so $\eta(u) \neq \eta(\eta(u))$, giving $\text{pivot}(u) = u \in B$. □

B.5 Proof of Lemma 3.7

Proof. Any node u belongs to A exactly when $R_u = R \cap (\{u\} \cup N(u))$ is nonempty, giving a membership test for A. Because R has the smallest priority nodes, we can recover $\eta(u)$ for each $u \in A$ with $\eta(u) = \arg\min_\kappa R_u$. Moreover, $\eta(u) \in R \subseteq A$, so we can similarly recover $\eta(\eta(u))$ and hence $\text{pivot}(u) \in A$.

Similarly, for $u \in B$, $R_u = \varnothing$, so $\eta(u) \notin R$. If $\eta(u) \in A$, $\eta(\eta(u)) \in R$, making u a singleton. Otherwise, $\eta(u) \in B$, In any case, $\text{pivot}(u) \in B$ when $u \in B$, so each cluster is entirely in A or B.

Finally, consider a pair $u, v \in V$ for membership in E_A. If both are in B, then $(u, v) \notin E_A$. If exactly one is in A, u and v are clustered separately, so $(u, v) \in E_A$ exactly when $u \sim v$. Otherwise, we recover $\text{pivot}(u)$ and $\text{pivot}(v)$ and manually check for a disagreement. □

B.6 Proof of priority bound and corollaries

For these proofs, write $K = 40 \log n$.

Proof of Lemma 3.4. For a node u, we will consider the case where $\deg(u) + 1 > 2 \log n$, as otherwise the condition trivially holds. Because $\eta(u)$ is u's minimum neighbor, when $\kappa(\eta(u)) > 2 \log n/(\deg(u) + 1)$, it must be the case that $\kappa(v) > 2 \log n/(\deg(u) + 1)$ for all $v \in N(u) \cup \{u\}$. Thus,

$$\mathbb{P}_\kappa\left[\kappa(\eta(u)) \geq \frac{2 \log n}{\deg(u)+1}\right] = \left(1 - \frac{2 \log n}{\deg(u)+1}\right)^{\deg(u)+1} \leq e^{-2 \log n} \leq 1/n^2.$$

Taking a union bound over all nodes gives the desired claim. □

Proof of Corollary 3.8. When $p = 1$, $R = V$ and $B = \varnothing$, so the condition is vacuously true. Otherwise, we know $p = K/\epsilon^2 D$. For a node $u \in B$, $\eta(u) \notin R$, as u has no neighbors in R. Thus, $\kappa(\eta(u)) \geq p$, so

$$\deg(u) < \deg(u) + 1 \leq \frac{2 \log n}{\kappa(\eta(u))} \leq \frac{K}{p} \leq \epsilon^2 D \leq D.$$

Finally, observe that for all $v \in V$ such that $(u, v) \in E_{\text{mis}}$, $v \in N(u) \cup N(\text{pivot}(u))$. Because $\text{pivot}(u) \in B$ by Lemma 3.6, there can be at most $\deg(u) + \deg(\text{pivot}(u)) < 2D$ such v. □

Proof of Corollary 3.10. $u \in B$, so $\eta(u) \notin R$. We know that $p \leq \kappa(\eta(u)) < 1$, giving $p = K/\epsilon^2 D$. By Lemma 3.4,

$$q(\eta(u)) = p \cdot \frac{1}{\kappa(\eta(u))} \geq \frac{K}{\epsilon^2 D} \cdot \frac{\deg(u)+1}{2 \log n} \geq \frac{\deg(u)+1}{\epsilon^2 D}.$$ □

B.7 Proof of Lemma 3.16

Proof. If $p = 1$, then $S = T = V$, and the estimator is exact. Hence assume $p < 1$.

Note that $\mathbb{E}_S[X_u] = p \deg_A(u)$. When $\deg_A(u) < D/2$, we have $\mathbb{E}_S[X_u] < pD/2$, so by a Chernoff bound,

$$\mathbb{P}_S[X_u \geq pD] \leq \exp\left(-\frac{1}{3} \cdot 1^2 \cdot pD/2\right). \leq e^{-pD/6} \leq 1/n^2.$$

Similarly, when $\deg_A(u) \geq 2D$, we have $\mathbb{E}_S[X_u] \geq 2pD$, so

$$\mathbb{P}_S[X_u < pD] \leq \exp\left(-\frac{1}{2} \cdot (1/2)^2 \cdot 2pD\right) \leq e^{-pD/4} \leq 1/n^2$$

By taking a union bound over all nodes, we have with probability at least $1 - 2/n$,

$$\forall u \in L, \deg_A(u) < 2D \text{ and } \forall u \in H, \deg_A(u) \geq D/2. \tag{5}$$

First, we will bound X. Now, consider any $u \in V$ such that $\deg_A(u) \geq D/2$. We have by a Chernoff bound

$$\mathbb{P}_S[|X_u - \mathbb{E}_S[X_u]| > \epsilon \mathbb{E}_S[X_u]] \leq 2 \exp\left(-\frac{\epsilon^2}{3} \cdot pD/2\right) \leq 2e^{-\epsilon^2 pD/6} \leq 2/n^2$$

Thus, X_u/p is a $(1 \pm \epsilon, 0)$-approximation of $\deg_A(u)$ with probability at least $1 - 2/n^2$. By a union bound, this holds for all such nodes with probability at least $1 - 2/n$.

Intersect this event with (5), and fix S in their intersection, whose probability is at least $1 - 4/n$. Then all $u \in H$ satisfy $\deg_A(u) \geq D/2$, and X is a $(1 \pm \epsilon, 0)$-approximation of $\sum_{u \in H} \deg_A(u)$.

Finally, we bound Y. Observe that by reordering the sum,

$$Y = \frac{1}{p} \sum_{u \in V} Y_u, = \frac{1}{p} \sum_{u \in L} \deg_A(u) \cdot \mathbb{1}\{u \in T\}$$

so we have that

$$\mathbb{E}_T[Y \mid S] = \sum_{u \in L} \deg_A(u).$$

Let $\mu = \mathbb{E}_T[Y \mid S]$. Note that for $u \in L$, $\deg_A(u)/p \leq 2D/p$, so by the Bernstein inequality (Corollary A.3),

$$\mathbb{P}_T[|Y - \mu| > \epsilon\,(\mu + \Delta) \mid S] \leq 2e^{-p\epsilon^2\Delta/4D} \leq 1/n$$

Therefore, with probability at least $1 - 1/n$, Y is a $(1 \pm \epsilon, \pm\epsilon\Delta)$-approximation of $\mathbb{E}_T[Y \mid S] = \sum_{u \in L} \deg_A(u)$.

Because L and H form a partition, we can combine these two results to show that $(X + Y)/2$ is a $(1 \pm \epsilon, \epsilon\Delta)$-approximation of $\frac{1}{2}\sum_{u \in V} \deg_A(u) = |E_A|$ with probability at least $1 - 5/n$. □

B.8 Proof of Lemma 3.14

Proof. Using Lemma 3.7, reject the pair if either endpoint lies in A, and otherwise assume $u, v \in B$. We also reject the pair if $\eta(u) = \eta(v) \in B$ but $\eta^{\rightarrow}(u) \neq \eta(u)$, because here $\eta(u)$ arrived before u and v, so (u, v) would be a candidate for E_{center}, not E_{pair}.

Now, we want to find if u and v belong to the same cluster, because afterwards we can directly test $\mathbb{1}\{(u, v) \in E_{\text{pair}}\}$. If $\eta(u) \neq \eta(v)$, clearly u and v must belong to different clusters. Now, let $w = \eta(u) = \eta(v)$, and we simply need to know if w is a pivot. If $w \in A$, as determined by Lemma 3.7, then w is not a pivot, as Lemma 3.6 forbids clusters spanning A and B. The final case is $w \in B$ and $\eta^{\rightarrow}(u) = w$. Here, we hold $\eta(\eta^{\rightarrow}(u)) = \eta(w)$, so we can check if w is a pivot. □

B.9 Proof of Lemma 3.15

Proof. Fix κ satisfying the priority bound (1).

If $p = 1$, then $R = V$ and $B = \varnothing$, so the pair estimator is identically zero. Hence assume $p < 1$.

For convenience, we will orient all of the edges in E_{pair} such that for $(u, v) \in E_{\text{pair}}$, $\tau_u < \tau_v$.

Each $(u, v) \in E_{\text{pair}}$ has $\mathbb{P}[(u, v) \in S \times T] = p^2$, so we have that $\mathbb{E}[Z_{\text{pair}}] = |E_{\text{pair}}|$. Let's define the estimators

$$P_u = \sum_{v \in V} \mathbb{1}\{(u, v) \in E_{\text{pair}}\}$$

We have

$$Z_{\text{pair}} = \frac{1}{p^2} \sum_{u \in S} \sum_{v \in T} \mathbb{1}\{\tau_u < \tau_v, (u, v) \in E_{\text{pair}}\} = \frac{1}{p^2} \sum_{v \in V} \mathbb{1}\{v \in T\} \sum_{u \in S} \mathbb{1}\{(u, v) \in E_{\text{pair}}\}$$

Thus,

$$\mathbb{E}_T[Z_{\text{pair}} \mid S] = \frac{1}{p^2} \sum_{v \in V} p \sum_{u \in S} \mathbb{1}\{(u, v) \in E_{\text{pair}}\} = \frac{1}{p} \sum_{u \in S} P_u = \frac{1}{p} \sum_{u \in V} P_u \cdot \mathbb{1}\{u \in S\}$$

Let $P = \mathbb{E}_T[Z_{\text{pair}} \mid S]$, so $\mathbb{E}_S[P] = \mathbb{E}_{S,T}[Z_{\text{pair}}] = |E_{\text{pair}}|$. By Corollary 3.8, $P_u/p < 2D/p$, so by the Bernstein inequality (Corollary A.3), we have

$$\mathbb{P}_S[|P - |E_{\text{pair}}|| > \frac{\epsilon}{3}(|E_{\text{pair}}| + \Delta)] \leq 2e^{-p\epsilon^2\Delta/36D} \leq 2/n.$$

Therefore, with probability at least $1 - 2/n$ over S, P is a $(1 \pm \epsilon/3, \epsilon\Delta/3)$-approximation of $|E_{\text{pair}}|$.

Now, define Q_v as

$$Q_v = \sum_{u \in S} \mathbb{1}\{(u, v) \in E_{\text{pair}}\}.$$

Again by Corollary 3.8, we have $Q_v < 2D$, so $\mathbb{E}_S[Q_v] < 2pD$. By a Chernoff bound, we also have that $\mathbb{P}_S[Q_v > 4pD] \leq \exp(-2pD/3) \leq 1/n^2$. Taking a union bound, we have $Q_v \leq 4pD$ for all $v \in V$ simultaneously with probability at least $1 - 1/n$. Fix S satisfying both this event and the approximation bound on P.

Now, we have

$$Z_{\text{pair}} = \frac{1}{p^2} \sum_{v \in V} Q_v \cdot \mathbb{1}\{v \in T\}.$$

Keeping S fixed and using $Q_v/p^2 \leq 4D/p$, we can apply the Bernstein inequality (Corollary A.3) over the randomness of T,

$$\mathbb{P}_T[|Z_{\text{pair}} - P| > \frac{\epsilon}{2}(P + \Delta) \mid S] \leq 2e^{-p\epsilon^2\Delta/32D} \leq 2/n.$$

Therefore, with conditional probability at least $1 - 2/n$ over T, Z_{pair} is a $(1 \pm \epsilon/2, \epsilon\Delta/2)$-approximation of P.

Writing $m = |E_{\text{pair}}|$, these events give $|Z_{\text{pair}} - m| \leq (5\epsilon/6 + \epsilon^2/6)(m + \Delta) \leq \epsilon(m + \Delta)$. A union bound gives total failure probability at most $5/n$. □

B.10 Proof of Lemma 3.13

For the analysis, we extend record sampling $\textsc{Rec}(w) = (a_w, b_w, c_w, d_w)$ taken in Algorithm 2 to all $w \in V$. Label the sampled pair such that a_w appears before b_w in the stream, i.e. $\tau_{a_w} < \tau_{b_w}$. Defining $\textsc{Rec}(w)$ for all $w \in V$ also defines $Z_w^{\pm}$. If a required sample is undefined, its corresponding contribution is zero.

Fix κ satisfying the priority bound (1).

First, we show that Z_{center} is indeed unbiased.

Lemma B.2. $\mathbb{E}[Z_{\text{center}}] = |E_{\text{center}}|$.

Proof. We'll partition E_{center} into $E^+_{\text{center}} \uplus E^-_{\text{center}}$ for the false positive edges across clusters and the false negative edges missing inside a cluster, respectively. That is,

$$E^+_{\text{center}} = \{(u, v) \in E_{\text{center}} \mid u \sim v\}, \qquad E^-_{\text{center}} = \{(u, v) \in E_{\text{center}} \mid u \not\sim v\}.$$

Note that for all $(u, v) \in E^+_{\text{center}}$, $\eta(u)$ is not a pivot, and for all $(u, v) \in E^-_{\text{center}}$, $\eta(u)$ is a pivot. We also define the sets E^+_w and E^-_w of mismatch edges centered at w for each $w \in B$ as

$$E^{\pm}_w = \{(u, v) \in E^{\pm}_{\text{center}} \mid \eta(u) = \eta(v) = w\},$$

satisfying $E^{\pm}_{\text{center}} = \biguplus_{w \in B} E^{\pm}_w$.

We'll also define

$$Z^{\pm}_{\text{center}} = \sum_{w \in B} Z^{\pm}_w \frac{\mathbb{1}\{w \in C\}}{q(w)} \qquad \text{such that} \qquad Z_{\text{center}} = Z^+_{\text{center}} + Z^-_{\text{center}}.$$

Consider some false negative $(u, v) \in E^-_w$ with $\tau_u < \tau_v$. By construction, (u, v) is a pair of forward neighbors of w, which were sampled uniformly from $N^{\rightarrow}(w)$, so $\mathbb{P}[(u, v) = (a_w, b_w)] = 1/\binom{\deg^{\rightarrow}(w)}{2}$.

Now, consider some false positive $(u, v) \in E^+_w$ with $\tau_u < \tau_v$. c_w was sampled uniformly from $N^{\rightarrow}(w)$, giving $\mathbb{P}[u = c_w] = 1/\deg^{\rightarrow}(w)$. Moreover, in this case $u \sim v$ always, so $v \in N^{\rightarrow}(u)$. d_w was sampled uniformly from $N^{\rightarrow}(c_w)$, so $\mathbb{P}[v = d_w \mid u = c_w] = 1/\deg^{\rightarrow}(u)$. Combining these two, $\mathbb{P}[(u, v) = (c_w, d_w)] = 1/(\deg^{\rightarrow}(w) \deg^{\rightarrow}(u))$.

We say $(u, v) \in E^-_w$ is selected at w when $(u, v) = (a_w, b_w)$, and that $(u, v) \in E^+_w$ is selected when $(u, v) = (c_w, d_w)$. $(u, v) \in E^{\pm}_w$ being selected at w is independent of the inclusion of w in C, which occurs with probability $q(w)$. Thus, $\mathbb{P}[w \in C, (u, v) \text{ is selected at } w] = q(w) \cdot \mathbb{P}[(u, v) \text{ is selected at } w]$. By the definitions of both Z^+_w and Z^-_w, we have

$$\mathbb{E}[Z^{\pm}_w] = \sum_{(u,v) \in E^{\pm}_w} \frac{1}{\mathbb{P}[(u, v) \text{ is selected at } w]} \cdot \mathbb{P}[(u, v) \text{ is selected at } w] = \left|E^{\pm}_w\right|.$$

We similarly have that for all $w \in B$, $\mathbb{E}[\mathbb{1}\{w \in C\}/q(w)] = \mathbb{P}[w \in C]/q(w) = 1$. Finally, $E^{\pm}_w$ forms a partition of $E^{\pm}_{\text{center}}$, so we conclude that $\mathbb{E}[Z^{\pm}_{\text{center}}] = \left|E^{\pm}_{\text{center}}\right|$. □

Now we prove the concentration result.

Proof of Lemma 3.13. By Lemma B.2, we have that $\mathbb{E}[Z_{\text{center}}] = |E_{\text{center}}|$. We also have that $Z_{\text{center}} = \sum_{w \in B} (Z_w^+ + Z_w^-) \cdot \frac{\mathbb{1}\{w \in C\}}{q(w)}$: the sum of independent estimators which, by Lemma 3.12, are each bounded by $\epsilon^4 D^2$. Thus, $\text{Var}(Z_{\text{center}}) \le \epsilon^4 D^2 |E_{\text{center}}|$. Applying Chebyshev's inequality,

$$\mathbb{P}[|Z_{\text{center}} - |E_{\text{center}}|| \ge \epsilon(|E_{\text{center}}| + \Delta)] \le \frac{\text{Var}(Z_{\text{center}})}{\epsilon^2(|E_{\text{center}}| + \Delta)^2} \le \frac{\epsilon^2}{4}. \qquad \square$$

Next, we prove Lemma 3.12.

Proof. Let's first consider Z_w^- for all $w \in B$. We necessarily have $\eta(w) = w$ when $Z_w^- > 0$. We apply the Priority bound twice: Corollary 3.10 on w to bound $1/q(\eta(w)) = 1/q(w)$, and Corollary 3.8 to bound $\deg(w)$. All together, we have

$$\frac{Z_w^-}{q(w)} \le \frac{1}{q(w)} \binom{\deg^{\to}(w)}{2} \le \frac{\epsilon^2 D}{\deg(w) + 1} \cdot \deg^{\to}(w)^2 \le \epsilon^4 D^2.$$

We turn our attention to Z_w^+. If $Z_w^+ > 0$, we have found a false positive (c_w, d_w) with $\eta(c_w) = \eta(d_w) = w$ and $c_w \sim d_w$. This time, we apply Corollary 3.10 on c_w and Corollary 3.8 on w, so

$$\frac{Z_w^+}{q(w)} \le \frac{1}{q(w)} \deg^{\to}(w) \deg^{\to}(c_w) \le \frac{\epsilon^2 D}{\deg(c_w) + 1} \deg^{\to}(w) \deg^{\to}(c_w) \le \epsilon^4 D^2.$$

We cannot simultaneously have $Z_w^+ > 0$ and $Z_w^- > 0$, proving $(Z_w^+ + Z_w^-)/q(w) < \epsilon^4 D^2$. $\square$

B.11 Proof of Lemma 3.17

Before we prove Lemma 3.17, we must prove that Algorithm 2 correctly implements the estimators we defined.

Lemma B.3. *Algorithm 2 computes the estimators X, Y, Z_{pair}, and Z_{center} for given κ, ϵ, Δ in two node-arrival passes while storing at most $O(|R| + |S| + |T| + |C|)$ nodes and counters.*

Proof. First, we discuss Pass 1. For each incoming $u \in V$, we flip two fresh p-biased coins to determine inclusion in S and T. Next, if $\kappa(u) < p$, we add u to R. Otherwise, we flip a fresh $q(u)$-biased coin to determine inclusion in C.

We maintain $\eta^{\to}(u)$ for each $u \in S$ during Pass 1 as follows. When u is added to S, initialize $\eta^{\to}(u) \leftarrow u$. For each subsequent $v \sim u$, update $\eta^{\to}(u) \leftarrow v$ if $\kappa(v) < \kappa(\eta^{\to}(u))$. Thus, $\eta^{\to}(S)$ requires $O(|S|)$ nodes to be stored.

Last, we must maintain $\text{REC}(w)$ for each $w \in C$. Upon insertion of w_0 into C, we initialize $a_{w_0}, b_{w_0}, c_{w_0}, d_{w_0} \leftarrow \perp$, $\deg^{\to}(w_0) \leftarrow 0$, and $\deg^{\to}(c_{w_0}) \leftarrow 0$. Note that if any of these samples are $\perp$ at the end of the stream, their respective estimators are 0. For each stream item $u \sim w_0$, we update $\deg^{\to}(w_0) \leftarrow \deg^{\to}(w_0) + 1$. For the current state of $\deg^{\to}(w_0)$, u is the $\deg^{\to}(w_0)$th item of $N^{\to}(w_0)$, so we can use that to update the reservoir sample $\{a_{w_0}, b_{w_0}\}$ and uniform sample c_{w_0}. If we do update $c_{w_0} \leftarrow u$, we also reset $d_{w_0} \leftarrow \perp$ and $\deg^{\to}(c_{w_0}) \leftarrow 0$. In parallel, for each stream item u, we consider all $w \in C$ such that $u \sim c_w$. Again, we update $\deg^{\to}(c_w) \leftarrow \deg^{\to}(c_w) + 1$ and uniformly update the sample d_w (except if $u = c_w$). The final first-stage sample c_w was selected when it arrived and was never subsequently replaced. The second-stage reservoir therefore processes exactly $N^{\to}(c_w)$, so the final sample d_w is indeed uniform over that set. In conclusion, $\text{REC}(C)$ requires $O(|C|)$ nodes and counters to be stored.

Between Pass 1 and 2, we have the final states of R, S, and T. For any u in memory, we can compute $X_u \leftarrow |\{v \in S \mid (u, v) \in E_A \text{ or } (v, u) \in E_A\}|$, as membership in E_A can be tested via Lemma 3.7. Thus, we can test u's membership to L or H by comparing X_u with pD, and build $T_L \leftarrow \{u \in T \mid u \in L\}$. Finally, we can now compute $Y_u \leftarrow |\{v \in T_L \mid (u, v) \in E_A \text{ or } (v, u) \in E_A\}|$. There is no

need to retain L, H, X_u, Y_u, T_L in memory, as the membership and values can be found live using R, S, and T.

During Pass 2, we maintain $\eta(u)$ for every u found in Pass 1 ($R, S, T, \eta^{\rightarrow}(S), C, \textsc{Rec}(C)$) using the same method as maintaining $\eta^{\rightarrow}$. We initialize $\eta(u) \leftarrow u$ before the pass, and update it throughout the pass. We also accumulate $X \leftarrow X + X_u \cdot \mathbb{1}\{u \in H\}/p$ and $Y \leftarrow Y + Y_u/p$ for each incoming stream item u. This stores at most $O(|R| + |S| + |T| + |C|)$ more nodes.

After Pass 2, for each $(u, v) \in S \times T$ with $\tau_u < \tau_v$, we now have $\eta(u), \eta(v), \eta^{\rightarrow}(u)$, and $\eta(\eta^{\rightarrow}(u))$, so by Lemma 3.14, we can test membership of $(u, v) \in E_{\text{pair}}$. We thus compute $Z_{\text{pair}} \leftarrow |\{(u, v) \in S \times T \mid \tau_u < \tau_v, (u, v) \in E_{\text{pair}}\}|/p^2$.

Finally, for each $w \in C$, we have the samples $\textsc{Rec}(w)$ along with the values $\eta(\textsc{Rec}(w))$ and $\eta(w)$, so we are able to test membership of the sampled pairs (a_w, b_w) and (c_w, d_w) to $E_w^{\pm}$. Using the final values of $\deg^{\rightarrow}(w)$ and $\deg^{\rightarrow}(c_w)$, we can compute $Z_w^{\pm}$. Thus, we compute $C \cap B$ using Lemma 3.7 and accumulate $Z_{\text{center}} \leftarrow \sum_{w \in C \cap B} (Z_w^+ + Z_w^-)/q(w)$. □

We now prove Lemma 3.17.

Proof. We have three bad events to consider. First, the Priority bound (1) may not hold, which occurs with probability at most $1/n$ by Lemma 3.4. Second, the algorithm's output may fail to be a good approximation of the clustering cost. Third, the space used by the algorithm may exceed the prescribed limit.

By Lemma B.3, Algorithm 2 implements the estimators $X, Y, Z_{\text{pair}}, Z_{\text{center}}$. Condition on the priority bound for the accuracy analysis. $(X+Y)/2$ gives a $(1 \pm \epsilon, \pm\epsilon\Delta)$-approximation of $|E_A|$ with failure probability at most $5/n$ by Lemma 3.16. Z_{pair} gives a $(1 \pm \epsilon, \pm\epsilon\Delta)$-approximation of $|E_{\text{pair}}|$ with failure probability at most $5/n$ by Lemma 3.15. Z_{center} gives a $(1 \pm \epsilon, \pm\epsilon\Delta)$-approximation of $|E_{\text{center}}|$ with failure probability at most $\epsilon^2/4$ by Lemma 3.13. Putting all three of these together, the output of Algorithm 2, $(X + Y)/2 + Z_{\text{pair}} + Z_{\text{center}}$, is a $(1 \pm \epsilon, \pm 3\epsilon\Delta)$-approximation of $|E_A| + |E_{\text{pair}}| + |E_{\text{center}}| = \text{cost}(\mathcal{C}_{\text{BCMT}}(\pi_\kappa))$ with conditional probability at least $1 - \epsilon^2/4 - 10/n$.

To analyze the space, we only need to consider the sizes of R, S, T, and C, as Lemma B.3 guarantees that the space is at most $O(|R| + |S| + |T| + |C|)$ stored nodes and additional words. For this analysis, take probability over both κ and the sampling randomness, without conditioning on the priority bound. We assume $p < 1/e$, as otherwise $O(np) = O(n)$ space is allowed. We know that $\mathbb{E}_\kappa[|R|] = np$, so a Chernoff bound gives $\mathbb{P}_\kappa[|R| \geq 2np] \leq \exp(-np/3) \leq 1/n$. $|S|$ and $|T|$ both have the same expectation np regardless of the choice of κ, so the same Chernoff bound gives $\mathbb{P}[|T| \geq 2np] = \mathbb{P}[|S| \geq 2np] \leq 1/n$, over the randomness of the algorithm.

Now, we consider $|C|$. For any $w \in V$, we have

$$\mathbb{P}[w \in C] = \mathbb{P}[\kappa(w) \geq p]\, \mathbb{E}[q(w) \mid \kappa(w) \geq p] = (1-p) \cdot \frac{1}{1-p} \int_p^1 \frac{p}{x}\, dx = p \log(1/p).$$

Thus, $\mathbb{E}[|C|] = np \log(1/p)$. Another Chernoff bound gives $\mathbb{P}[|C| \geq 2np \log(1/p)] \leq 1/n$.

A union bound gives that the probability of any of these bad events occurring is at most $\epsilon^2/4 + 15/n = \epsilon^2/4 + O(1/n)$. Abort a run if its prescribed storage budget would be exceeded, counting this event as a failure and returning 0. □

B.12 Proof of Theorem 3.1

Proof of Theorem 3.1. Let $\gamma = \epsilon/2$, $k = \lceil ((5+\gamma)/\gamma) \log 8 \rceil \leq 24/\epsilon$, and $\epsilon_0 = \epsilon/25$.

Independently sample k priority functions κ_i, and write $M_i = \text{cost}\,(\mathcal{C}_{\text{BCMT}}(\pi_{\kappa_i}))$. We will run Algorithm 2 k times in parallel and write each output as $O_i = \textsc{Twice}(\kappa_i, \epsilon_0, \Delta)$. We will adjust these estimates upwards and write $\widehat{M}_i = (O_i + 3\epsilon_0\Delta)/(1-\epsilon_0)$. Thus, we have $M_i \leq \widehat{M}_i \leq \frac{1+\epsilon_0}{1-\epsilon_0} M_i + \frac{6\epsilon_0}{1-\epsilon_0}\Delta$ with probability at least $1 - \epsilon_0^2/4 - 15/n$. A union bound gives that the probability of any estimate failing is at most $k(\epsilon_0^2/4 + 15/n) \leq 24\epsilon/2500 + 360/(n\epsilon) < 1/12$ when $p < 1$, since then $n\epsilon^2 > 25000 \log n$; when $p = 1$, the costs are evaluated exactly.

If $\mathrm{OPT} = 0$, then $M_i = 0$ almost surely. Otherwise, using Markov's inequality with Theorem 3.3, for each $i \in [k]$ we have $\mathbb{P}[M_i \le (5+\gamma)\,\mathrm{OPT}] \ge \gamma/(5+\gamma)$. Thus, $\mathbb{P}[\min_{i\in[k]} M_i > (5+\gamma)\,\mathrm{OPT}] \le \exp(-k\gamma/(5+\gamma)) \le 1/8$.

Now, a union bound gives both events with probability at least $1 - 1/12 - 1/8 > 2/3$. All that remains is the final arithmetic,

$$\frac{1+\epsilon_0}{1-\epsilon_0}(5+\gamma) = \frac{1+\epsilon/25}{1-\epsilon/25}(5+\epsilon/2) \le 5+\epsilon \qquad \text{and} \qquad \frac{6\epsilon_0}{1-\epsilon_0} = \frac{6\epsilon}{25-\epsilon} \le 1.$$

Now, we have that $\min_{i\in[k]} \widehat{M}_i$ is a $(5+\epsilon, \Delta)$-approximation of OPT with probability at least $2/3$. Taking the median of independent batches run in parallel boosts this to probability 0.99 with a constant number of repetitions. The final space complexity is $O\left(1/\epsilon\right) \cdot O\left(n \log^2 n/\epsilon_0^2\sqrt{\Delta}\right) = O\left(n \log^2 n/\epsilon^3\sqrt{\Delta}\right)$. □

B.13 Proofs of lower bounds

We will make use of the following well-studied problems in communication complexity.

Definition B.4 (INDEX). *In the* INDEX *problem, Alice holds an N-bit vector $\boldsymbol{x} = (x_1, \ldots, x_N)$, and Bob holds an index $j \in [N]$. The goal is for Alice to send a message to Bob that allows him to determine x_j.*

Lemma B.5 (cf. Kushilevitz & Nisan (1996)). *Any one-way randomized algorithm that solves* INDEX *with success probability at least 0.51 requires $\Omega(N)$ bits of communication.*

Definition B.6 (DISJ). *In the* DISJ *problem, Alice holds an N-bit vector $\boldsymbol{x} = (x_1, \ldots, x_N)$, and Bob holds an N-bit vector $\boldsymbol{y} = (y_1, \ldots, y_N)$. Their goal is to exchange messages so that they can determine if there exists an index $i_0 \in [N]$ with $x_{i_0} = y_{i_0} = 1$.*

Lemma B.7 (Bar-Yossef et al. (2004)). *Any randomized algorithm that solves* DISJ *with success probability at least 0.51 requires $\Omega(N)$ bits of communication.*

We also use the standard critical-clique lemma from the cluster-editing literature.

Definition B.8 (Critical clique). *A pair of nodes $u, v \in V$ are called* true twins *when they share the same positive closed neighborhood, $N(u) \cup \{u\} = N(v) \cup \{v\}$. A maximal set of pairwise true twins is called a* critical clique.

Lemma B.9 (Guo (2009)). *In any optimal (unweighted* MINDISAGREE*) clustering, each critical clique is entirely contained in a single cluster.*

Proof. Suppose a clustering splits a critical clique $K \subseteq V$. Observe that the cost of this clustering can be improved by moving all of K to an existing cluster that minimizes disagreements between K and $V \setminus K$. The number of external disagreements cannot increase, as all of K has the same neighborhood, but the number of internal disagreements strictly decreases. □

In our lower bound proofs, we work with vectors in $\mathbb{R}^d$ under the ℓ_1 distance, treating each vector as its corresponding node. We write the i-th standard basis vector in $\mathbb{R}^d$ as $\boldsymbol{e}_i^d$. The similarity function is defined for two points $\boldsymbol{p}$ and $\boldsymbol{q}$ such that $\boldsymbol{p} \sim \boldsymbol{q}$ if and only if $\ell_1(\boldsymbol{p}, \boldsymbol{q}) \le 1$. We allow distinct nodes to have identical feature vectors. Although d might be large, the vectors we use will be explicitly constructed and contain $O(1)$ non-zero entries, and the coordinate indices and values each use $O(\log n)$ bits, so each vector admits an $O(1)$-word representation.

For convenience, we suppress integer rounding and pad our constructions with isolated vertices as necessary.

B.13.1 Proof of Theorem 4.1

We reduce from INDEX on $N \ge 2$ bits. Let $q \ge 1$ be the blow-up parameter to be chosen later, and let $r = Nq$ and $d = r + 1$. Alice will encode $\boldsymbol{x}$ using N q-cliques, and Bob will recover x_j using a selector of r nodes.

For each $i \in [N]$, Alice creates a q-clique A_i by placing q identical points at $(4i + x_i)\boldsymbol{e}_1^d \in \mathbb{R}^d$. Bob creates the selector B for index j by placing r points, one at $4j\boldsymbol{e}_1^d + \boldsymbol{e}_{k+1}^d$ for each $k \in [r]$. Thus, B is an independent set of r nodes, and there are never edges between A_i and B for all $i \neq j$.

In the case $x_j = 1$, there are no edges between A_j and B, so the graph consists of the N cliques $A_1, \ldots, A_N$ and the r independent nodes of B. Hence, $\mathrm{OPT} = 0$.

In the case $x_j = 0$, each node in A_j is connected to all of B. Each A_i for $i \neq j$ can once again be freely placed in its own cluster. By the critical-clique lemma (Lemma B.9), all of A_j must be contained within a single cluster, and any node in B that is not clustered with A_j must be a singleton. If we let k be the number of nodes in B clustered with A_j, then the cost is

$$\mathrm{OPT} = \min_{0 \le k \le r} \left(q(r-k) + \binom{k}{2} \right) = qr - \binom{q+1}{2} \ge \frac{qr}{2},$$

using $q < r$ and that the minimum is attained when $k = q + 1$.

Now, the total number of nodes is $n = Nq + r = 2Nq$. Let $\mathcal{A}$ be a one-pass 0.49-error (c, Δ)-approximation algorithm, for any $c \ge 1$ and $\Delta < Nq^2/2$. This gives a randomized one-way protocol for INDEX on N bits. By Lemma B.5, $\mathcal{A}$ must use at least $\Omega(N)$ bits of space.

Finally, to create a construction for a target n, Δ with $\Delta \le n^2/64$, we can choose $q = 8(n + \Delta)/n$ and take $N = n/2q$, so the required space is $\Omega(n^2/(n + \Delta)) = \Omega(\min\{n, n^2/\Delta\})$. □

B.13.2 Proof of Corollary 4.3

Consider the INDEX construction above, with some minor modifications. First, decouple the size of Bob's selector r from Alice's encoding Nq and instead let it vary as an independent parameter. Set $q = 1$. Fix the INDEX instance size N and the additive error budget $1 \le \Delta \le N^2/16$, both of which are known to Alice. However, the stream length $n = N + r$ is controlled by Bob and is *not* known to Alice. Finally, we will use the space $\mathbb{R}^{N^2}$ to encode the points.

Let's consider two instances for which Alice's construction is identical. Consider two possible continuations from Bob,

$$r_{\mathrm{short}} = N \qquad\qquad r_{\mathrm{long}} = \max\{N, \Delta + 2\}$$

In the long continuation, $\mathrm{OPT} = 0$ when $x_j = 1$, and $\mathrm{OPT} = r_{\mathrm{long}} - 1 \ge \Delta + 1$ when $x_j = 0$. Thus, a one-pass algorithm $\mathcal{A}$ supposing the hypotheses admits a randomized one-way protocol for INDEX on N bits, which by Lemma B.5 requires $\Omega(N)$ bits.

$\mathcal{A}$ sees the same additive error budget Δ and the same prefix regardless of which continuation occurs. Thus, the distribution of states of $\mathcal{A}$ after Alice's prefix is identical, and $\mathcal{A}$ requires $\Omega(N)$ bits in the short continuation as well. The short stream has length $2N$, so taking $N = n/2$ shows that the required space is $\Omega(n)$. □

B.13.3 Proof of Theorem 4.2

We reduce from DISJ on N bits. Let $q \ge 1$ be the blow-up parameter to be chosen later. The graph will consist of N disconnected gadgets, where the ith gadget encodes $x_i y_i$ in its optimal clustering cost. We will consider each of these separately and sum their clustering costs.

The ith gadget G_i consists of three q-cliques: S_i, A_i, and B_i. Alice places q identical points at $8i + 2$ to build S_i and places another q identical points at $8i + x_i$ to build A_i. Bob builds B_i using q identical points at $8i + 4 - y_i$.

When $x_i = 1$, A_i and S_i are connected and form a clique, and when $x_i = 0$, A_i and S_i are disconnected. Similarly, B_i and S_i are connected exactly when $y_i = 1$. Note that A_i and B_i are always disconnected.

Suppose $x_i y_i = 0$. S_i is connected to at most one of A_i or B_i. Thus, G_i consists of either two (if $x_i + y_i = 1$) or three (if $x_i = y_i = 0$) disjoint cliques. Each of these cliques can be clustered alone without disagreements, so $\mathrm{OPT}(G_i) = 0$.

Otherwise, $x_i y_i = 1$, so A_i and B_i are both connected to S_i but not to each other. By the critical-clique lemma (Lemma B.9), we only need to consider clusterings which leave A_i, B_i, and S_i intact.

Clustering all of $A_i \cup S_i \cup B_i$ together has cost q^2 due to the missing edges between A_i and B_i. Otherwise, at least q^2 edges must be cut: either $A_i \times S_i$ or $B_i \times S_i$. Hence, $\mathrm{OPT}(G_i) = q^2$.

Now, we consider the two DISJ cases. If $x_i y_i = 0$ for all $i \in [N]$, then each G_i can be perfectly clustered, giving $\mathrm{OPT} = 0$. If there exists some $i_0 \in [N]$ such that $x_{i_0} y_{i_0} = 1$, then $\mathrm{OPT} \geq \mathrm{OPT}(G_{i_0}) = q^2$.

The total number of nodes is $n = 3Nq$. Let $\mathcal{A}$ be a p-pass 0.49-error (c, Δ)-approximation algorithm, for any $c \geq 1$ and $\Delta < q^2$. This gives a randomized two-way 0.49-error protocol for DISJ on N bits using $2p - 1$ messages. By Lemma B.7, $\mathcal{A}$ must use at least $\Omega(N/p)$ bits of space.

Finally, we create a construction for a target n, Δ with $1 \leq \Delta \leq n^2/64$ by choosing $q = \sqrt{2\Delta}$ and taking $N = n/3q$, which produces the desired space bound $\Omega(n/(p\sqrt{\Delta}))$ for a p-pass algorithm. □

C ADDITIONAL EXPERIMENTAL RESULTS

Retained vectors use FP32 coordinates: 1,024 dimensions for ImageNet and 256 for SSN++. The displayed per-run budgets therefore represent 5.11–20.45 MB and 32–128 MB of vector payload, respectively. The vector implementations of ONCE, TWICE, and C^{4}APPROX ($k = 5$) use one, two, and nine passes. Our capacity-bounded C^{4}APPROX implementation retains only its dense contribution when its sampled-center pool is empty.

For Figure 2, we shuffle 500 independent-priority trials per method and budget into 20 blocks of 25; each returns the median of five means of five estimates. Per-run budgets are 0.03125%–0.125% of n vector slots. This median-of-means aggregation differs from the minimum-over-clusterings wrapper used in our guarantees. Priorities are reused across budgets within each method, with a separate shuffle at each budget. Exact-reference costs use the same aggregation, giving 40 outputs from 1,000 priorities for ImageNet and 12 from 300 for SSN++. The summed allocations (0.78125%–3.125% of n) need not be resident simultaneously.

Figure 3 fixes priority seed 1 and varies sampling randomness over 500 trials per method and budget. Mean ONCE and TWICE outputs are within 10% of Pivot across the displayed budgets; at the smallest, C^{4}APPROX averages 31% and 18% below Pivot on ImageNet and SSN++, respectively. Differences from Pivot include both clustering and estimation effects.

To keep the fixed-priority error bars readable, we exclude SSN++ C^{4}APPROX outputs exceeding 10 times the exact Pivot cost. Only six of 3,500 runs across all seven budgets exceed this cutoff: two each at 24 and 64 MB, and one each at 96 and 128 MB; none occur at 16, 32, or 48 MB. These runs account for 87.83%–99.75% of the variance at the affected budgets. The 96 MB outlier is 285 times Pivot's cost, with 99.7% from its E_B estimate; omitting it reduces the Pivot-normalized standard deviation from $1{,}270.9\%$ to 28.5%. The other five outliers are also dominated by E_B. The displayed budgets omit 24 MB, so only four exclusions affect Figure 3, leaving 498–500 trials per point; no ImageNet runs are excluded. These are diagnostic display exclusions: the raw data are unchanged, and the main median-of-means figure is unfiltered.

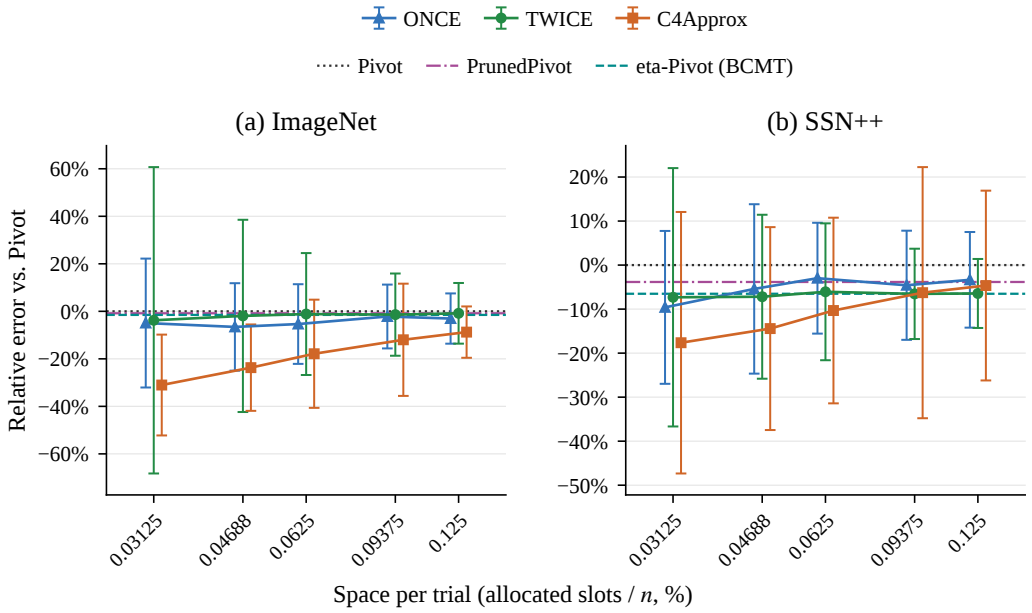

Figure 3: Fixed priorities: mean ± sample standard deviation of $(\widehat{M} - M_{\text{Pivot}})/M_{\text{Pivot}}$ after the diagnostic exclusions above. Horizontal lines mark exact clustering costs on this scale; bars show trial dispersion, not confidence intervals.