On Estimating Maximum Matching Size in Graph Streams

Sepehr Assadi[*]   Sanjeev Khanna[*]   Yang Li[*]

Abstract

We study the problem of estimating the maximum matching *size* in graphs whose edges are revealed in a streaming manner. We consider both *insertion-only* streams, which only contain edge insertions, and *dynamic* streams that allow both insertions and deletions of the edges, and present new upper and lower bound results for both cases.

On the upper bound front, we show that an α-approximate estimate of the matching size can be computed in dynamic streams using $\widetilde{O}(n^2/\alpha^4)$ space, and in insertion-only streams using $\widetilde{O}(n/\alpha^2)$-space. These bounds respectively shave off a factor of α from the space necessary to compute an α-approximate matching (as opposed to only size), thus proving a non-trivial separation between approximate estimation and approximate computation of matchings in data streams.

On the lower bound front, we prove that any α-approximation algorithm for estimating matching size in dynamic graph streams requires $\Omega(\sqrt{n}/\alpha^{2.5})$ bits of space, *even* if the underlying graph is both *sparse* and has *arboricity* bounded by $O(\alpha)$. We further improve our lower bound to $\Omega(n/\alpha^2)$ in the case of *dense* graphs. These results establish the first non-trivial streaming lower bounds for *super-constant* approximation of matching size.

Furthermore, we present the first *super-linear* space lower bound for computing a $(1 + \varepsilon)$-approximation of matching size *even* in insertion-only streams. In particular, we prove that a $(1 + \varepsilon)$-approximation to matching size requires $\mathsf{RS}(n) \cdot n^{1-O(\varepsilon)}$ space; here, $\mathsf{RS}(n)$ denotes the maximum number of edge-disjoint *induced matchings* of size $\Theta(n)$ in an n-vertex graph. It is a major open problem with far-reaching implications to determine the value of $\mathsf{RS}(n)$, and current results leave open the possibility that $\mathsf{RS}(n)$ may be as large as $n/\log n$. Moreover, using the best known lower bounds for $\mathsf{RS}(n)$, our result already rules out any $O(n \cdot \mathrm{poly}(\log n/\varepsilon))$-space algorithm for $(1 + \varepsilon)$-approximation of matchings. We also show how to avoid the dependency on the parameter $\mathsf{RS}(n)$ in proving lower bound for dynamic streams and present a near-optimal lower bound of $n^{2-O(\varepsilon)}$ for $(1 + \varepsilon)$-approximation in this model.

Using a well-known connection between matching size and *matrix rank*, all our lower bounds also hold for the problem of estimating matrix rank. In particular our results imply a near-optimal $n^{2-O(\varepsilon)}$ bit lower bound for $(1 + \varepsilon)$-approximation of matrix ranks for dense matrices in dynamic streams, answering an open question of Li and Woodruff (STOC 2016).

---

1 Introduction

Recent years have witnessed tremendous progress on solving graph optimization problems in the *streaming* model of computation, formally introduced in the seminal work of [6]. In this model, a graph is presented as a stream of edge insertions (*insertion-only streams*) or edge insertions and deletions (*dynamic streams*), and the goal is to solve the given problem with minimum space requirement (see a survey by McGregor [48] for a summary).

One of the most extensively studied problems in the streaming literature is the classical problem of finding a *maximum matching* [46]. Although significant advances have been made on understanding the space needed to compute a maximum matching in the streaming model [1–3, 9, 14, 16, 17, 20–24, 29, 31, 36–39, 47–49, 56], some important problems remain wide open. In particular, not much is known about the tradeoff between space and approximation for the problem of estimating the *size* of a maximum matching in the streaming model.

In this paper, we obtain new upper and lower bounds for the matching size problem. Our results show that while the problem of matching size estimation is provably easier than the problem of finding an approximate matching, the space complexity of the two problems starts to converge together as the accuracy desired in the computation approaches near-optimality. In particular, we establish the first super-linear space lower bound (in n) for the matching size estimation problem. A well-known connection between matching size and matrix rank allows us to carry our lower bound results to the problem of estimating rank of a matrix in the streaming model, and we show that essentially quadratic space is necessary to obtain a near-optimal approximation of matrix rank. In what follows, we first briefly review the previous work, and then present our results and techniques in detail.

1.1 Models and Previous Work

Two types of streams are generally considered in the literature, namely insertion-only streams and dynamic streams. In insertion-only streams, edges are only inserted, and in dynamic streams, edges can be both inserted and deleted. In the following, we briefly summarize previous results for *single-pass* algorithms (i.e., algorithms that only make one pass over the steam) in both insertion-only streams and dynamic streams.

Insertion-only streams. It is easy to compute a 2-approximate matching using $\widetilde{O}(n)$ space in insertion-only streams: simply maintain a *maximal* matching during the stream; here n denotes the number of vertices in the input graph. This can be done similarly for computing an α-approximate matching in $\widetilde{O}(n/\alpha)$ space for any $\alpha \geq 2$. On the lower bound side, it is shown in [29, 36] that computing better than a $e/(e-1)$-approximate matching requires $n^{1+\Omega(1/\log\log n)}$ space.

For the seemingly easier problem of estimating the maximum matching *size* (the focus of this paper), the result of [29, 36] can be modified to show that computing better than a $e/(e-1)$-approximation for matching size requires $n^{\Omega(1/\log\log n)}$ space (see also [37]). It was shown later in [23] that computing better than a $3/2$-approximation requires $\Omega(\sqrt{n})$ bits of space. More recently, this lower bound was extended by [14] to show that computing a $(1+\varepsilon)$-estimation requires $n^{1-O(\varepsilon)}$ space. On the other hand, the only existing non-trivial algorithm is a folklore that an $O(\sqrt{n})$-approximation can be obtained in polylog(n) space even in dynamic streams (for completeness, we provide a self-contained proof of this result in Appendix A). We note that other algorithms that use $o(n)$ space for this problem also exist, but they only work under certain conditions on the input: either the edges are presented in a *random order* [37] or the input graph has *bounded arboricity* [14, 16, 23, 49].

Dynamic streams. Space complexity of finding an α-approximate matching in dynamic graph streams is essentially resolved: it is shown in [9] that $\tilde{\Theta}(n^2/\alpha^3)$ space is *necessary* and in [9, 16], that it is also *sufficient* (see also [38]). However, the space complexity of estimating the matching size (the focus of this paper) is far from being settled in this model. For example, it is not even known if α-approximating matching size is strictly easier than finding an α-approximate matching (for any $\alpha = o(\sqrt{n})$). Moreover, no better lower bounds are known for estimating matching size in dynamic streams than the ones in [14, 23], which already hold even for insertion-only streams.

This state-of-the-art in both insertion-only and dynamic streams highlights the following natural question: *How well can we approximate the maximum matching size in a space strictly smaller that what is needed for finding an approximate matching? In general, what is the space-approximation tradeoff for estimating the matching size in graph streams?*

Indeed, this question (and its closely related variants) has already been raised in the literature [23, 37, 49]. In this paper, we make progress on this question from both upper bound and lower bound ends.

1.2 Our Results

Upper bounds. We prove that computing an α-approximate estimate of matching size is strictly easier than finding an α-approximate matching. Formally,

Theorem 1. *There exist single-pass streaming algorithms that for any $2 \leq \alpha \leq \sqrt{n}$, w.h.p.*[1]*, output an α-approximation of the maximum matching size in dynamic streams using $\widetilde{O}(n^2/\alpha^4)$ and in insertion-only streams using $\widetilde{O}(n/\alpha^2)$ space, respectively.*

The algorithms in Theorem 1 are the first algorithms that outperform (by a factor of α), respectively, the *optimal* $\widetilde{O}(n^2/\alpha^3)$-space algorithm in dynamic streams, and the *optimal* $\widetilde{O}(n/\alpha)$-space algorithm in insertion-only streams for finding an α-approximate matching. This provides the first non-trivial separation between approximate estimation and approximate computation of matchings in both dynamic and insertion-only streams.

Lower bounds. Our first lower bound result concerns computing an α-approximation of the maximum matching size in dynamic streams for any $\alpha \geq 1$, *not necessarily a constant*.

Theorem 2. *Any (randomized) single-pass streaming algorithm that computes an α-approximation of maximum matching size with a constant probability in dynamic streams requires $\Omega(\sqrt{n}/\alpha^{2.5})$ bits of space. This bound holds even if the input graph is both* sparse *and has* arboricity[2] *$O(\alpha)$. Moreover, if the input graph is allowed to be* dense*, then $\Omega(n/\alpha^2)$ bits of space is necessary.*

The lower bounds in Theorem 2 are the first non-trivial space lower bounds for *super-constant* approximation algorithms for matching size estimation. Obtaining space lower bounds for polylog(n)-approximation of matching size has been posed as an open problem by Kapralov *et al.* [37], who also mentioned that "existing techniques do not seem to lend easily to answer this question and it will be very useful (quite possibly for other related problems) to develop tools needed to make progress on this front". Our results in Theorem 2 make progress on this question in dynamic streams.

An interesting aspect of our lower bound in Theorem 2 is that it holds even for bounded arboricity graphs. There is an active line of research on estimating matching size of bounded arboricity graphs in graph streams [14, 16, 23, 49], initiated by Esfandiari *et al.* [23]. The state-of-the-art is an $O(1)$-approximation in $\widetilde{O}(n^{4/5})$ space for dynamic streams in bounded-arboricity graphs [14, 16, 49].

---

[1]We use w.p. and w.h.p. to abbreviate with probability and with high probability, respectively.

[2]A graph G has arboricity ν if the set of edges in G can be partitioned into at most ν forests.

Our second lower bound result concerns computing a $(1 + \varepsilon)$-approximation of the maximum matching size in both insertion-only streams and in dynamic streams. In the following, let $\mathsf{RS}(n)$ denote the maximum number of edge-disjoint *induced matchings* of size $\Theta(n)$ in any n-vertex graph (see Section 2.1).

Theorem 3. *Any (randomized) single-pass streaming algorithm that with a constant probability outputs a $(1 + \varepsilon)$-approximation of the maximum matching size in insertion-only streams requires $\mathsf{RS}(n) \cdot n^{1-O(\varepsilon)}$ space. The lower bound improves to $n^{2-O(\varepsilon)}$ for dynamic streams.*

Since $\mathsf{RS}(n)$ is known to be at least $n^{\Omega(1/\log\log n)}$ [25], Theorem 3 immediately implies that no $\widetilde{O}(n \cdot \mathrm{poly}(1/\varepsilon))$-space algorithm can output a $(1 + \varepsilon)$-approximation of matching size in insertion-only streams. Interestingly, it is known that by allowing *multiple passes* over the stream, a $(1 + \varepsilon)$-approximate matching (as opposed to only its size) can be found in $\widetilde{O}(n \cdot \mathrm{poly}(1/\varepsilon))$ space, even in dynamic streams and even for the weighted version of the problem [1, 2] (see also [48]).

Our lower bounds in Theorem 3 are the first *super linear* (in n) space lower bounds for estimating matching size in graph streams. An interesting implication of these lower bounds is that while the problem of matching size estimation is provably easier than the problem of finding an approximate matching (by Theorem 1), the space complexity of the two problems starts to converge together as the accuracy desired in the computation approaches near-optimality.

Schatten p-norms. The *Schatten p-norm* of a matrix A is defined as the ℓ_p-norm of the vector of the singular values of A (see [44] for more detail); in particular, the case of $p = 0$ corresponds to the *rank* of the matrix A. Schatten norms and rank computation have been previously studied in the streaming and sketching models [14, 18, 41, 43–45]. It is shown that exact computation of matrix rank in data streams requires $\Omega(n^2)$ space [18, 43] (even allowing multiple passes), and $(1 + \varepsilon)$-approximation requires $n^{1-O(\varepsilon)}$ space [14]; the latter result was recently extended to all Schatten p-norms for *odd* values of p [44].

It is well-known that computing the maximum matching size is equivalent to computing the rank of the Tutte matrix [46, 54]. Consequently, all our lower bounds stated for matching size estimation also hold for matrix rank computation. This in particular implies an $\Omega(\sqrt{n})$ space lower bound for *any constant* approximation of rank in *sparse* matrices and a near-optimal $n^{2-O(\varepsilon)}$ space lower bound for $(1 + \varepsilon)$-approximation in *dense* matrices, answering an open question of Li and Woodruff [44].

2 Preliminaries

Notation. For any graph G, $\mathrm{opt}(G)$ denotes the maximum matching *size* in G. We use bold face letters to represent random variables. For any random variable $\boldsymbol{X}$, $\mathrm{SUPP}(\boldsymbol{X})$ denotes its support set. We define $|\boldsymbol{X}| := \log|\mathrm{SUPP}(\boldsymbol{X})|$. For any k-dimensional tuple $X = (X_1, \ldots, X_k)$ and any $i \in [k]$, we define $X^{<i} := (X_1, \ldots, X_{i-1})$, and $X^{-i} := (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_k)$.

Total Variation Distance. For any two distributions μ and ν with the same support Ω where $|\Omega|$ is finite, the *total variation distance* between μ and ν, denoted by $\|\mu - \nu\|_{tvd}$, is given by $\max_{\Omega' \subseteq \Omega} (\mu(\Omega') - \nu(\Omega')) = \frac{1}{2} \sum_{x \in \Omega} |\mu(x) - \nu(x)|$. We use the following well-known fact in our proofs.

Fact 2.1. *Suppose we want to distinguish between two probability distributions μ and ν given one sample from one of the two; then the best probability of success is $\frac{1}{2} + \frac{\|\mu - \nu\|_{tvd}}{2}$.*

2.1 Ruzsa-Szemerédi graphs

For any graph G, a matching M of G is an *induced matching* iff for any two vertices u and v that are matched in M, if u and v are not matched to each other, then there is no edge between u and v in G.

Definition 1 (Ruzsa-Szemerédi graph). *A graph G is an (r,t)-Ruzsa-Szemerédi graph (or (r,t)-RS graph for short), iff the set of edges in G consists of t pairwise disjoint induced matchings $M_1, \ldots, M_t$, each of size r.*

RS graphs, first introduced by Ruzsa and Szemerédi [52], have been extensively studied as they arise naturally in property testing, PCP constructions, additive combinatorics, etc. (see, e.g., [5, 7, 8, 13, 25, 27, 29, 32, 53]). These graphs are of interest typically when r and t are large relative to the number of vertices in the graph.

One particularly interesting range of the parameters is when $r = \Theta(n)$ [25–27], i.e., when the induced matchings are of linear size. We use the notation $\mathsf{RS}(n)$ to denote the *largest* possible value for the parameter t such that an (r,t)-RS graph on n vertices with $r = \Theta(n)$ exists. It is a major open problem to determine the asymptotic of $\mathsf{RS}(n)$ [26, 27, 30], but currently there is a huge gap between existing upper and lower bounds for $\mathsf{RS}(n)$. In particular, it is known that for any constant $c < 1/4$, a $(c \cdot n, t)$-RS graph with $t = n^{\Omega(1/\log \log n)}$ exists [25] (see also [29]). However, the best known upper bound only shows that for $(c \cdot n, t)$-RS graphs, where c is any constant less than $1/4$, t is upper bounded by $\frac{n}{\log^{(x)} n}$, with $x = O(\log \frac{1}{c})$, $(\log^{(x)}(n)$ denotes the x-fold iterative logarithm of n) [26]. Slightly better upper bounds are known for large values of c; in particular, it is shown in [27] that for $1/5 < c < 1/4$, $t = O(n/\log n)$. We refer the interested reader to [7, 27] for more on the history of Ruzsa-Szemerédi graphs and to [7, 29] for their application to different areas of computer science, including proving lower bounds for streaming algorithms.

Obtaining (r,t)-RS graphs for $r = \Theta(n)$ and $t = n^{\varepsilon}$ (for some constant $\varepsilon > 0$) seems to be out of the scope of the state-of-the-art techniques; however, Alon *et al.* [7] provide a surprising construction of (very) dense RS graphs when we allow r to be just *slightly sublinear*: there are (r,t)-RS graphs on n vertices with parameters $r = n^{1-o(1)}$ and $r \cdot t = \binom{n}{2} - o(n^2)$ [7]. While our lower bound for insertion-only streams requires the use of (r,t)-RS graphs with $r = \Theta(n)$ (hence naturally leads to a dependence on $\mathsf{RS}(n)$), for our lower bound for dynamic streams it suffices to work with RS graphs with $r = n^{1-o(1)}$ and hence we can directly use the construction of [7] (hence avoiding dependency on $\mathsf{RS}(n)$).

2.2 Tools from Information Theory

We briefly review some basic concepts from information theory needed for establishing our lower bounds. For a broader introduction to the field, we refer the reader to the excellent text by Cover and Thomas [19].

In the following, we denote the *Shannon Entropy* of a random variable $\boldsymbol{A}$ by $H(\boldsymbol{A})$ and the *mutual information* of two random variables $\boldsymbol{A}$ and $\boldsymbol{B}$ by $I(\boldsymbol{A}; \boldsymbol{B}) = H(\boldsymbol{A}) - H(\boldsymbol{A} \mid \boldsymbol{B}) = H(\boldsymbol{B}) - H(\boldsymbol{B} \mid \boldsymbol{A})$. If the distribution $\mathcal{D}$ of the random variables is not clear from the context, we use $H_{\mathcal{D}}(\boldsymbol{A})$ (resp. $I_{\mathcal{D}}(\boldsymbol{A}; \boldsymbol{B})$). We use H_2 to denote the binary entropy function where for any real number $0 < \delta < 1$, $H_2(\delta) = \delta \log \frac{1}{\delta} + (1-\delta) \log \frac{1}{1-\delta}$. We know that $0 \leq H(\boldsymbol{A}) \leq |\boldsymbol{A}|$ and equality holds iff $\boldsymbol{A}$ is uniform on its support. Similarly, $I(\boldsymbol{A}; \boldsymbol{B}) \geq 0$ and equality holds iff $\boldsymbol{A}$ and $\boldsymbol{B}$ are independent of each other.

We use the following basic properties of entropy and mutual information (proofs can be found in [19], Chapter 2).

4

Fact 2.2. *Let A, B, C be random variables.*

1. Conditioning reduces the entropy: $H(A \mid B, C) \leq H(A \mid B)$; equality holds iff A and C are independent conditioned on B.

2. Chain rule for entropy: $H(A, B) = H(A) + H(B \mid A)$.

3. Chain rule for mutual information: $I(A, B; C) = I(A; C) + I(B; C \mid A)$.

4. Conditional sub-additivity of mutual information: if $A_1, A_2, \ldots, A_t$ are conditionally independent given B, then $I(A_1, A_2, \ldots, A_t; B) \leq \sum_{i=1}^{t} I(A_i; B)$.

5. Conditional super-additivity of mutual information: if $A_1, A_2, \ldots, A_t$ are conditionally independent given C, then $I(A_1, A_2, \ldots, A_t; B \mid C) \geq \sum_{i=1}^{t} I(A_i; B \mid C)$.

The following claim (Fano's inequality) states that if a random variable A can be used to estimate the value of another random variable B, then A should "consume" most of the entropy of B.

Claim 2.3 (Fano's inequality). *For any binary random variable B and any (possibly randomized) function f that predicts B based on A, if $\Pr(f(A) \neq B) = \delta$, then $H(B \mid A) \leq H_2(\delta)$.*

Finally, we prove the following auxiliary lemma that allows us to decompose any random variable with high entropy to a convex combination of relatively small number of near uniform distributions plus a low probability "noise term".

Lemma 2.4. *Let $X \sim \mathcal{D}$ be a random variable on $\{0, 1\}^n$ such that $H(X) \geq n - \Delta$. For any $\varepsilon > 0$, there exist $k + 1$ distributions $\mu_0, \mu_1, \ldots, \mu_k$ on $\{0, 1\}^n$, along with $k + 1$ probabilities $p_0, p_1, \ldots, p_k$ ($\sum_i p_i = 1$) for some $k = O(n/\varepsilon)$, such that $\mathcal{D} = \sum_i p_i \cdot \mu_i$, $p_0 = O(\varepsilon)$, and for any $i \geq 1$,*

1. $\log |\mathrm{SUPP}(\mu_i)| \geq n - \frac{\Delta}{\varepsilon} - \log \Theta(\frac{n}{\varepsilon})$.

2. $\|\mu_i - U_i\|_{tvd} = O(\varepsilon)$, where U_i denotes the uniform distribution on $\mathrm{SUPP}(\mu_i)$.

Proof. Partition the support of $\mathcal{D}$ into k' sets $S_0, S_1, \ldots, S_{k'}$ for $k' = \Theta(n/\varepsilon)$ such that S_0 contains every element $a \in \{0, 1\}^n$ where $\Pr(X = a) < 2^{-2n}$, and for each $i \geq 1$, S_i contains every element a where $(1 + \varepsilon)^{-(i+1)} \leq \Pr(X = a) < (1 + \varepsilon)^{-i}$. We say that a set S_i is *large* if $|S_i| \geq 2^{(n - \frac{\Delta}{\varepsilon} - \log \Theta(\frac{n}{\varepsilon}))}$ and is otherwise *small*. Let $\mathcal{L}$ (resp. $\mathcal{S}$) denote the set of all elements that belong to a large set (resp. a small set). Moreover let k be the number of large sets, and, without loss of generality, assume $S_1, \ldots, S_k$ are these large sets.

We define the $k + 1$ distributions in the lemma statement as follows. Let μ_0 be the distribution $\mathcal{D}$ conditioned on X being in $S_0 \cup \mathcal{S}$ (i.e., S_0 and elements from small sets), and let $p_0 = \Pr_{\mathcal{D}}(X \in \mathcal{S} \cup S_0)$; for each $i \geq 1$, let μ_i be the distribution $\mathcal{D}$ conditioned on X being in S_i (i.e., the i-th large set) and let $p_i = \Pr_{\mathcal{D}}(X \in S_i)$.

By construction, the described distributions satisfy $\mathcal{D} = \sum_i p_i \cdot \mu_i$. Moreover, for each $i \geq 1$, since the support S_i of μ_i is a large set, we have $\log |\mathrm{SUPP}(\mu_i)| \geq n - \frac{\Delta}{\varepsilon} - \log \Theta(\frac{n}{\varepsilon})$; since each element a in $\mathrm{SUPP}(\mu_i)$ has $\Pr_{\mathcal{D}}(X = a) \in [(1 + \varepsilon)^{-(i+1)}, (1 + \varepsilon)^{-i})$, it is straightforward to verify that $\|\mu_i - U_i\|_{tvd} = O(\varepsilon)$. Hence it only remains to argue that $p_0 = O(\varepsilon)$.

It is easy to see that $\Pr_{\mathcal{D}}(X \in S_0) = o(1)$ and therefore in the following we prove that $\Pr_{\mathcal{D}}(X \in \mathcal{S}) = O(\varepsilon)$. Let $Z \in \{0, 1\}$ be a random variable that denotes whether X chosen from $\mathcal{D}$ belongs to $\mathcal{L}$ or $\mathcal{S}$. We have,

$$H(X \mid Z) \geq H(X) - H(Z) \geq H(X) - 1 \geq n - \Delta - 1 \tag{1}$$

where the first inequality is by chain rule of entropy (Fact 2.2-(2)). Moreover, since the total number of elements belonging to small sets is at most $\Theta(n/\varepsilon) \cdot 2^{\left(n - \frac{\Delta}{\varepsilon} - \log \Theta(\frac{n}{\varepsilon})\right)} = 2^{n - \frac{\Delta}{\varepsilon}}$,

$$
\begin{aligned}
H(\boldsymbol{X} \mid \boldsymbol{Z}) &= \Pr(\boldsymbol{Z} = 0) \cdot H(\boldsymbol{X} \mid \boldsymbol{Z} = 0) + (1 - \Pr(\boldsymbol{Z} = 0)) \cdot H(\boldsymbol{X} \mid \boldsymbol{Z} = 1) \\
&\leq \Pr(\boldsymbol{Z} = 0) \cdot \log\left(2^{n - \frac{\Delta}{\varepsilon}}\right) + (1 - \Pr(\boldsymbol{Z} = 0)) \cdot \log\left(2^n\right) \\
&\qquad\qquad\qquad\qquad \text{(since } H(\boldsymbol{A}) \leq |\boldsymbol{A}| \text{ for any random variable } \boldsymbol{A}) \\
&= n - \Pr(\boldsymbol{Z} = 0) \cdot \left(\frac{\Delta}{\varepsilon}\right)
\end{aligned}
$$

and consequently, if $\Pr(\boldsymbol{Z} = 0) > 2\varepsilon$, then $H(\boldsymbol{X} \mid \boldsymbol{Z}) < n - 2\Delta < n - \Delta - 1$, a contradiction to Eq (1). This finalizes the proof that $p_0 = O(\varepsilon)$. ∎

2.3 Communication Complexity and Information Complexity

Communication complexity and information complexity play an important role in our lower bound proofs. We now provide necessary definitions for completeness.

Communication complexity. Our lowers bounds for streaming algorithms are established through communication complexity lower bounds. Here, we briefly provide some relevant background; for a more detailed treatment of communication complexity, we refer the reader to the excellent text by Kushilevitz and Nisan [40].

We focus on two models of communication, namely, the *two-player one-way communication* model, and the *multi-party number-in-hand simultaneous message passing model* (SMP).

One-way Communication Model. Let P be a relation with domain $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$. Alice receives an input $X \in \mathcal{X}$ and Bob receives $Y \in \mathcal{Y}$, where (X, Y) are chosen from a joint distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. In addition to private randomness, the players also have an access to a shared public tape of random bits R. Alice sends a single message $M(X, R)$ and Bob needs to output an answer $Z := Z(M(X, R), Y, R)$ such that $(X, Y, Z) \in P$.

We use Π to denote a protocol used by the players. Unless specified otherwise, we always assume that the protocol Π can be randomized (using both public and private randomness), *even against a prior distribution $\mathcal{D}$ of inputs*. For any $0 < \delta < 1$, we say Π is a δ-error protocol for P, if the probability that for *any* input (X, Y), Bob outputs some Z where $(X, Y, Z) \in P$ is at least $1 - \delta$ (over the randomness of the protocol Π).

The *communication cost* of a one-way protocol Π for a problem P on an input distribution $\mathcal{D}$, denoted by $\|\Pi\|$, is the worst-case size of the message sent from Alice to Bob in the protocol Π, when the inputs are chosen from the distribution $\mathcal{D}$. *Communication complexity* $\mathsf{CC}^{\delta}_{1\text{-way}, \mathcal{D}}(P)$ of a problem P with respect to a distribution $\mathcal{D}$ is the minimum communication cost of any one-way protocol Π that is required to solve P on *every instance* w.p. at least $1 - \delta$.

SMP Communication Model. Let P be a $(k + 1)$-ary relation with domain $\mathcal{X}_1 \times \ldots \times \mathcal{X}_k \times \mathcal{Z}$. In the SMP communication model, k players $P^{(1)}, \ldots, P^{(k)}$ recieve inputs $X_1, \ldots, X_k$, jointly distributed according to a prior distribution $\mathcal{D}$ over $\mathcal{X}_1 \times \ldots \times \mathcal{X}_k$. In addition to private randomness, the players also have an access to a shared public tape of random bits R. Each of the players simultaneously sends a single message $M_j(X_j, R)$ to an external party called the *referee* and referee needs to output an answer $Z := Z(M_1(X_1, R), \ldots, M_k(X_k, R), R)$ such that $(X_1, \ldots, X_k, Z) \in P$.

6

Similar to the one-way communication model, we let Π denote the protocol used by the players and define δ-error protocols for P over a distribution $\mathcal{D}$ analogously. The *communication cost* of a SMP protocol Π for a problem P on an input distribution $\mathcal{D}$, denoted by $\|\Pi\|$, is the sum of the worst-case size of the messages sent by players to the referee, i.e., $\|\Pi\| := \sum_{i \in [k]} |M_i|$, when the inputs are chosen from the distribution $\mathcal{D}$. *Communication complexity* $\mathsf{CC}^{\delta}_{\mathrm{SMP},\mathcal{D}}(P)$ in SMP model is defined the same as in the one-way communication model.

Remark 2.5. *To facilitate our proofs, we sometimes need to give the referee an auxiliary input as well, which is jointly distributed with the input of the k players. The referee's answer then would be a function of the k messages he receives as well as his input. As a convention, we typically ignore this artificial feature of the model and only include it implicitly.*

Information Complexity. There are several possible definitions of information complexity of a communication problem that have been considered depending on the application (see, e.g., [10–12, 15]). In this paper, we use the notion of (external) *information cost* of a protocol. Roughly speaking, information cost of a one-way or SMP protocol is the average amount of information one can learn about the input of the players that are sending the messages by observing the transcript of the protocol.

More formally, the *information cost* of a one-way protocol Π with respect to a distribution $\mathcal{D}$ is $\mathsf{ICost}_{\mathcal{D}}(\Pi) = I_{\mathcal{D}}(\boldsymbol{\Pi}; \boldsymbol{X})$, where $\boldsymbol{X} \sim \mathcal{D}$ is the random variable for the input to Alice, $\boldsymbol{\Pi} := \boldsymbol{\Pi}(\boldsymbol{X})$ is the random variable denoting the message sent from Alice to Bob in the protocol Π, *concatenated* with the *public* randomness $\boldsymbol{R}$ used by Π. The *information complexity* $\mathsf{IC}^{\delta}_{\text{1-way},\mathcal{D}}(P)$ of P with respect to a distribution $\mathcal{D}$ is the minimum $\mathsf{ICost}_{\mathcal{D}}(\Pi)$ taken over all one-way protocols Π that are required to solve P on *every instance* w.p. at least $1 - \delta$.

Similarly, the information cost of a SMP protocol is defined as $\sum_{j=1}^{k} I_{\mathcal{D}}(\boldsymbol{\Pi}_j; \boldsymbol{X}_1, \dots, \boldsymbol{X}_k)$, where $\boldsymbol{X}_i$ denotes the input of the player $P^{(i)}$ and $\boldsymbol{\Pi}_i := \boldsymbol{\Pi}_i(X_i)$ denotes the message sent from the player $P^{(i)}$ to the referee *concatenated* with the public randomness $\boldsymbol{R}$ used by Π. *Information complexity* $\mathsf{IC}^{\delta}_{\mathrm{SMP},\mathcal{D}}(P)$ of a problem P in the SMP communication model can also be defined analogous to the one-way communication model.

Remark 2.6. *The requirement in the above definitions that Π is correct* everywhere, *even outside the support of the distribution $\mathcal{D}$ is crucial: we analyze our lower bounds on distributions that are "trivial" and the only reason that these lower bounds are meaningful (i.e., are non-zero) is that these protocols are required to succeed* uniformly.

The following well-known proposition (see, e.g., [15]) relates communication complexity and information complexity.

Proposition 2.7. *For every $0 < \delta < 1$ and every distribution $\mathcal{D}$:*

$$(i)\ \mathsf{CC}^{\delta}_{\text{1-way},\mathcal{D}}(P) \geq \mathsf{IC}^{\delta}_{\text{1-way},\mathcal{D}}(P) \qquad\qquad (ii)\ \mathsf{CC}^{\delta}_{\mathrm{SMP},\mathcal{D}}(P) \geq \mathsf{IC}^{\delta}_{\mathrm{SMP},\mathcal{D}}(P)$$

Proof. We only prove this for the SMP model; the result for the one-way model can be proven similarly. Let Π be a SMP protocol with the minimum communication complexity for P on $\mathcal{D}$ and $\boldsymbol{R}$ denote the public randomness of Π; we have,

$$\mathsf{IC}^{\delta}_{\mathrm{SMP},\mathcal{D}}(P) \leq \mathsf{ICost}_{\mathcal{D}}(\Pi) = \sum_{i=1}^{k} I_{\mathcal{D}}(\boldsymbol{\Pi}_i; \boldsymbol{X}) = \sum_{i=1}^{k} I_{\mathcal{D}}(\boldsymbol{\Pi}_i, \boldsymbol{R}; \boldsymbol{X})$$
$$(\boldsymbol{\Pi}_i \text{ contains both the message of player } P^{(i)} \text{ and the public randomness } \boldsymbol{R})$$

7

$$= \sum_{i=1}^{k} I_{\mathcal{D}}(\boldsymbol{R}; \boldsymbol{X}) + I_{\mathcal{D}}(\boldsymbol{\Pi}_i; \boldsymbol{X}, \boldsymbol{R}) \quad \text{(chain rule of mutual information (Fact 2.2-(3)))}$$

$$= \sum_{i=1}^{k} I_{\mathcal{D}}(\boldsymbol{\Pi}_i; \boldsymbol{X}, \boldsymbol{R}) = \underset{R \sim \boldsymbol{R}}{\mathbb{E}} \Big[\sum_{i=1}^{k} I_{\mathcal{D}}(\boldsymbol{\Pi}_i; \boldsymbol{X} \mid \boldsymbol{R} = R) \Big]$$

$$(\boldsymbol{R} \perp \boldsymbol{X} \text{ and hence } I(\boldsymbol{R}; \boldsymbol{X}) = 0)$$

$$\leq \underset{R \sim \boldsymbol{R}}{\mathbb{E}} \Big[\sum_{i=1}^{k} H_{\mathcal{D}}(\boldsymbol{\Pi}_i; \boldsymbol{X} \mid \boldsymbol{R} = R) \Big] \leq \underset{R \sim \boldsymbol{R}}{\mathbb{E}} \Big[\sum_{i=1}^{k} |\boldsymbol{\Pi}_i^R| \Big]$$

$$(\boldsymbol{\Pi}_i^R \text{ is the message sent from } P^{(i)} \text{ and is equal to } \boldsymbol{\Pi}_i \text{ conditioned on } \boldsymbol{R} = R)$$

$$\leq \|\Pi\| = \mathsf{CC}_{\mathrm{SMP}, \mathcal{D}}^{\delta}(P)$$

∎

Connection to Streaming: We conclude this section by pointing out the connection between the communication models defined in this section and the streaming setting. It is a standard fact that any streaming algorithm directly works as a one-way communication protocol and hence lower bounds in the one-way communication model also imply the same bounds on the space complexity of streaming algorithms in *insertion-only* streams. Recent results of [4, 42] prove a similar situation for the SMP model and *dynamic* streams: communication complexity lower bounds in SMP model imply space lower bounds for dynamic streams. In particular, communication complexity of a *k-player* problem in the SMP model is at most *k times* the space complexity of the same problem in dynamic streams.

2.4 The Boolean Hidden Hypermatching Problem

We shall use the following communication problem first studied by [55] in proving our lower bounds.

Definition 2 (Boolean Hidden Hypermatching, $\mathsf{BHH}_{n,t}$). *The* Boolean Hidden Hypermatching problem *is a one-way communication problem in which Alice is given a boolean vector* $x \in \{0,1\}^n$ *where* $n = 2kt$ *(for some integer* $k \geq 1$*) and Bob gets a* perfect t-hypermatching M *on* n *vertices, and a boolean vector* $w \in \{0,1\}^{n/t}$*. Let* Mx *denote the length* n/t *boolean vector* $(\bigoplus_{1 \leq i \leq t} x_{M_{1,i}}, \ldots, \bigoplus_{1 \leq i \leq t} x_{M_{n/t,i}})$ *where* $\{M_{1,1}, \ldots, M_{1,t}\}, \ldots, \{M_{n/t,1}, \ldots, M_{n/t,t}\}$ *are the edges of* M*. It is promised that either* $Mx = w$ *or* $Mx = \overline{w}$*. The goal of the problem is for Bob to output* Yes *when* $Mx = w$ *and* No *when* $Mx = \overline{w}$ *(* $\oplus$ *stands for addition modulo 2).*

The special case of this problem where $t = 2$ is referred to as the *Boolean Hidden Matching* problem, BHM_n, and was originally introduced by Gavinsky *et al.* [28] who established an $\Omega(\sqrt{n})$ lower bound on its one-way communication complexity. This lower bound was extended to $\Omega(n^{1-1/t})$ for the more general $\mathsf{BHH}_{n,t}$ problem by Verbin and Yu [55] (see Section 4 for more details). We further extend this result and establish a matching lower bound on the *information complexity* of $\mathsf{BHH}_{n,t}$ (see Theorem 5).

For our purpose, it is more convenient to work with a special case of the $\mathsf{BHH}_{n,t}$ problem, namely $\mathsf{BHH}_{n,t}^0$ where the vector $w = 0^{n/t}$ and hence the goal of Bob is simply to decide whether $Mx = 0^{n/t}$ (Yes case) or $Mx = 1^{n/t}$ (No case). We define $\mathsf{BHM}_n^0 := \mathsf{BHH}_{n,2}^0$ (similar to BHM_n). It is known that (see, e.g. [14, 44, 55]) any instance of the original $\mathsf{BHH}_{n,t}$ problem can be reduced to an instance of $\mathsf{BHH}_{2n,t}^0$ *deterministically* and with *no communication* between the players. This allows for extending the communication and information complexity lower bounds of $\mathsf{BHH}_{n,t}$ to $\mathsf{BHH}_{2n,t}^0$ problem (see Corollary 6).

BHH$_{n,t}^0$ and Matching Size Estimation. The BHH$_{n,t}^0$ problem has been used previously in [14, 23] to prove lower bounds for estimating matching size in data streams. We now briefly describe this connection.

The following reduction was first proposed by [14]. Given an instance $(x, \mathcal{M})^3$ of BHH$_{n,t}^0$, we create a graph $G(V \cup W, E)$ with $|V| = |W| = n$ as follows:

- For any $x_i = 1$, Alice adds an edge between v_i and w_i to E.

- For any hyperedge e in the t-hypermatching $\mathcal{M}$, Bob adds to E a clique between the vertices w_i where i is incident on e.

The following claim, proven originally by [14], establishes the correctness of this reduction. For the sake of completeness, we provide a simple proof this claim here.

Claim 2.8 ([14]). *Suppose $G(V \cup W, E)$ is the graph obtained from an instance $(x, \mathcal{M})$ of BHH$_{n,t}^0$ (for an even integer t) with the property that $\|x\|_0 = n/2$;*

- *if $Mx = 0^{n/t}$ (i.e., Yes case), then $\mu(G) = \frac{3n}{4}$.*

- *if $Mx = 1^{n/t}$ (i.e., No case), then $\mu(G) = \frac{3n}{4} - \frac{n}{2t}$.*

Proof. Denote by $M^\star$ a maximum matching in G. Since the vertices in V all have degree one, without loss of generality, we can assume all edges in $V \times W$ belong to $M^\star$, and we only need to consider the maximum matching size between the remaining vertices. Since the remaining vertices in V all have degree 0, we only need to consider the remaining vertices in W (and $n/2$ vertices in W remains since $\|x\|_0 = \frac{n}{2}$).

In the Yes case, for each hyperedge e, the clique created by e has t vertices, and even number of these vertices will be matched by edges in $V \times W$. Since t is even, *even* number of the vertices of the clique remain. Since there is still a clique between these remaining vertices, there is a matching that matches all of them. Therefore, the total matching size is $\frac{n}{2} + \frac{1}{2} \cdot \frac{n}{2} = \frac{3n}{4}$.

In the No case, for each hyeredge e, the clique created by e has *odd* number of vertices remained. Therefore, for every hyperedge, one vertex will be left unmatched. Since there are $\frac{n}{t}$ hyperedges, $\frac{n}{t}$ of the remaining vertices will be left unmatched, hence the total matching size is $\frac{n}{2} + \frac{1}{2}\left(\frac{n}{2} - \frac{n}{t}\right) = \frac{3n}{4} - \frac{n}{2t}$. ∎

3 Technical Overview

3.1 Lower Bounds

Our lower bounds are obtained by establishing communication complexity lower bounds in the one-way model (for insertion-only streams) and in the SMP model (for dynamic streams). We prove our lower bounds for sparse graphs (first part of Theorem 2) and dense graphs (Theorem 3 and second part of Theorem 2) using conceptually different techniques; we elaborate below on each case separately.

---

[3]In order to distinguish between matchings and hypermatchings, when not clear from the context, we use $\mathcal{M}$ instead of M to denote a hypermatching.

Sparse graphs. We prove this lower bound by analyzing the following k-player problem, referred to as the *sparse matching size estimation* (SMS) problem, in the SMP model: each player $P^{(i)}$ (for $i \in [k]$) is given a matching $M_i \subseteq E$ in a *sparse* graph $G(V_S \cup V_P, E)$ with $|V_P| = \Theta(k) \cdot |V_S|$; think of vertices in V_S as *shared* vertices that appear in the input of every player and vertices in V_P as *private* vertices that appear in the input of only a single player (the partition V_S and V_P is *not* known to the players). In the Yes case, the end-points of every edge are either both shared or both private such that $\mathrm{opt}(G) = \Theta(V_P)$, and in the No case, every edge has one shared end-point and one private end-point, hence $\mathrm{opt}(G) = \Theta(V_S)$.

We can interpret the setup in the SMS problem as follows. For any player $P^{(i)}$ with the matching M_i, define a binary vector x_i over the set $V(M_i)$ of vertices incident on M_i: for any $v \in V(M_i)$, $x_i(v) = 1$ if the vertex v is a shared vertex and $x_i(v) = 0$ otherwise. The vector x_i for player $P^{(i)}$ can be identified uniquely given the set of vertices in $V(M_j)$ of any other player $j \neq i$. Now, in the Yes case (resp. the No case) of the SMS problem, for any matching M_i and any two vertices $u, v \in V(M_i)$, $x_i(u) \oplus x_i(v) = 0$ (resp. $x_i(u) \oplus x_i(v) = 1$). One may notice that this setup is quite similar to the BHM^0 problem in the one-way model described in Section 2. Indeed, we ultimately prove a lower bound on the simultaneous communication complexity of our SMS problem using the $\Omega(\sqrt{n})$ lower bound of BHM^0 problem [28]. However, there is an inherent difficulty in performing such a reduction that we elaborate on next. Addressing this challenge results in a rather non-standard and protocol-specific reduction of a simultaneous multi-player problem to a two-player one-way problem, which is one of our central technical contributions.

A standard technique in proving communication lower bounds for multi-player problems is *symmetrization* [51]; here, one reduces a 2-player problem to a k-player problem by letting Bob play the role of one of the k players and Alice play the role of the remaining $(k-1)$ players. This technique is used (both explicitly and implicitly) in many known lower bounds for *finding* approximate matchings in different multi-player communication models [9, 33, 36, 38]. The success of this technique in these cases can be mostly attributed to the fact that in finding an approximate matching, every player is responsible for communicating the set of edges in *his input* that belongs to a maximum matching in the final graph; in other words, the message communicated by a player is typically *not* helping in finding the edges of another player.

In contrast, in matching *size* estimation, the players only need to (together) convey a *signal* about whether their common input is a Yes instance or a No instance. In particular, a small number of players already have enough information to distinguish between the large and small matching size cases; for example, in the SMS problem, any two players together have sufficient information to solve the problem completely. Indeed, the two players $P^{(i)}$ and $P^{(j)}$ can identify the set of shared vertices (and hence the vector x_i) and then simply check the parity of one arbitrary edge in M_i using x_i, to distinguish between the two cases. This implies that no matter how we split the role of the k players between Alice and Bob, Alice already gains enough information from the distribution to solve the underlying BHM^0 instance.

To circumvent this issue, we consider the internals of any fixed protocol Π_{SMS} for the SMS problem. We prove that for *any* protocol Π_{SMS}, there exists *some* index $i \in [k]$, such that Π_{SMS} is solving the BHM^0 instance encoded by the matching M_i of player $P^{(i)}$ and the vector x_i, defined by the inputs of players $P^{(j)}$ for $j \neq i$. In order to prove this, we need to analyze the protocol Π_{SMS} on distributions other than the ones defined above for SMS. Interestingly, in these distributions, there is *no* large gap between the matching size (in Yes and No cases) and hence a priori it is not even clear why Π_{SMS} should perform any non-trivial task over them. Having proved this, we can then embed any instance of BHM^0 into an instance of SMS for the *specific* protocol Π_{SMS}, using a careful reduction, in which we have to crucially use the fact that Π_{SMS} is a simultaneous protocol (as opposed to one-way) to obtain a one-way protocol for BHM^0.

Dense graphs. The starting point of our approach in Theorem 3 is [14] (itself based on a prior result of [23]) that establishes a reduction for estimating matching size from the BHH^0 problem in the one-way model (as mentioned in Section 2). The setup here is as follows: Alice is given a matching M, Bob is given a collection of cliques of size $\Theta(1/\varepsilon)$ (denoted by E_B) and depending on the answer of the "embedded" BHH^0 problem in the reduction, the maximum matching in $M \cup E_B$ differs by a factor of $(1 + \varepsilon)$. This reduction then implies a lower bound of $n^{1-O(\varepsilon)}$ by the known lower bounds on the communication complexity of BHH^0 [55].

To "boost" this lower bound from $n^{1-O(\varepsilon)}$ to the *super-linear* regime, a natural idea is to provide Alice not with a single matching M, but a collection of t *independently chosen* matchings $M_1, \ldots, M_t$, and then ask Alice and Bob to solve the problem for a *uniformly at random* chosen matching $M_{j^\star}$ and a single collection of $\Theta(1/\varepsilon)$-cliques (provided to Bob as before). Intuitively, Alice now needs to solve t different instances of the BHH^0 problem (as index $j^\star$ is not known to Alice) and this should make the new problem t *times harder* than the original one.

There are three main obstacles in implementing this idea: (i) the matchings $M_1, \ldots, M_t$ should be "supported" on $\Theta(n)$ vertices, as opposed to the trivial $\Theta(t \cdot n)$ vertices (or otherwise the lower bound would be too weak in terms of size of the final graph), (ii) the matchings should be chosen independently even though they are supported on essentially the same set of vertices (or otherwise we cannot argue that the new problem is indeed t times harder), and finally (iii), the reduction should ensure that Alice and Bob still need to solve the specific embedded BHH^0 instance for a uniformly at random chosen matching $M_{j^\star}$ and the $\Theta(1/\varepsilon)$-cliques (as otherwise we do not obtain a valid reduction).

We bypass these obstacles by using RS graphs defined in Section 2. Intuitively, we use RS graphs to "pack" the matchings $M_1, \ldots, M_t$ in the aforementioned reduction over $\Theta(n)$ vertices and use the fact that these matchings are *induced* to ensure the independence between the different matchings. Our reduction can be interpreted as "embedding" multiple instances of the BHH^0 problem into a single graph. RS graphs have been used previously in [9, 29, 36, 38] for proving lower bounds for *finding* approximate matchings. While it was possible to analyze the hard instances in [9, 29, 36, 38] using simple counting arguments that crucially exploited the requirement on *outputting a valid matching*, we now need to prove the lower bound using *information complexity* to reduce the original problem (i.e., matching size estimation) to multiple instances of a simpler problem (i.e., t instances of BHH^0), using a direct-sum style argument. This introduces new challenges, including designing a reduction from a two-player one-way problem like BHH^0 to a multi-player simultaneous problem that does not "leak" much information. En route, we also establish a lower bound on the *information complexity* of BHH^0 that matches the best known lower bound on its communication complexity (see Theorem 5).

3.2 Upper bounds

The main idea behind our algorithms in Theorem 1 is the following structural result that we show: if we sample each *vertex* in a graph G w.p. (essentially) $1/\alpha$, then the maximum matching size in the subgraph G' induced by the sampled vertices can be used to obtain an α-approximate estimate of matching size in G. Using this result, we design an algorithm that samples the vertices of G at a rate $1/\alpha$ to obtain an induced subgraph G', and maintains a sufficiently large matching in G' to estimate $\mathsf{opt}(G)$.

For insertion-only streams, a large matching (up to 2 approximation) can be computed simply by maintaining a maximal matching in G'. For dynamic streams, we use existing results of [9, 16] that allow finding a (large) matching with size at most k (in our case, $k = \Theta(n/\alpha^2)$) in $\widetilde{O}(k^2)$ space in dynamic streams.

11

4 An Information Complexity Lower Bound for BHH

In this section, we state the known communication complexity results for the BHH problem defined in Section 2.4 and then prove an information complexity lower bound for this problem.

The following is a hard distribution for $\mathsf{BHH}_{n,t}$ used in [55]:

The distribution $\mathcal{D}$ for $\mathsf{BHH}_{n,t}$.

- **Alice**: The input to Alice is a boolean vector $x \in \{0,1\}^n$ chosen *uniformly at random*.

- **Bob**: The input to Bob is a perfect t-hypermatching M chosen *uniformly at random* and a boolean vector w such that, w.p. $1/2$, $w = Mx$ and w.p. $1/2$, $w = \overline{Mx}$.

The result in [55] can now be stated as follows.

Theorem 4 (Communication Complexity of $\mathsf{BHH}_{n,t}$ [55]). *For any $t \geq 2$, suppose $n = 2kt$ for some integer $k \geq 1$, and $\delta \in (0, 1/2)$. Let $\gamma := \frac{1}{2} - \delta$; then,*

$$\mathsf{CC}^{\delta}_{\text{1-way},\mathcal{D}}(\mathsf{BHH}_{n,t}) = \Omega\left(\gamma \cdot n^{1-1/t}\right)$$

This bound also holds for the communication cost *of the protocols that are* only *required to be correct w.p. $1 - \delta$ on the distribution $\mathcal{D}$ (not necessarily on all inputs).*

We point out that the communication lower bound for $\mathsf{BHH}_{n,t}$ stated in [55] (and similarly for BHM_n stated in [28]), has a dependence of γ^2 instead of γ; however, obtaining the linear dependence on γ is straightforward (see the proof of Theorem 5 in this paper). This dependence is crucial for our results in Section 5.1 since we are analyzing protocols which outperforms random guessing only by a very small probability.

For our application, we need a (stronger) lower bound on the *information complexity* of $\mathsf{BHH}_{n,t}$ rather than its communication complexity. Since this result does not follow directly from those of [28, 55], for completeness, we provide a proof of this result in this section following the approach in [28, 55]. We remark that one can also use the message compression technique of [34] for bounded round communication protocols to prove this result.

Theorem 5 (Information Complexity of $\mathsf{BHH}_{n,t}$). *For any $t \geq 2$, any $n = 2kt$ for some integer $k \geq 1$, and any constant $\delta < 1/2$,*

$$\mathsf{IC}^{\delta}_{\text{1-way},\mathcal{D}}(\mathsf{BHH}_{n,t}) = \Omega\left(n^{1-1/t}\right)$$

This bound also holds for the information cost *of the protocols that are* only *required to be correct w.p. $1 - \delta$ on the distribution $\mathcal{D}$ (not necessarily on all inputs).*

The general idea in the proof of [28, 55] is as follows: in any protocol Π, Alice's message partitions the set $\{0,1\}^n$ into 2^c sets $A_1, \ldots, A_{2^c}$ (here $c := \|\Pi\|$) and hence for a typical message of Alice, Bob knows that the random variable $\boldsymbol{X}$ (of Alice's input) is chosen uniformly at random from some set A_i with $|A_i| \geq 2^{n-c}$. Now consider the hypermatching M of Bob, and the distributions $M\boldsymbol{X}$ and $\overline{M\boldsymbol{X}}$ for $\boldsymbol{X}$ uniform on A_i; the main technical result in [28, 55] proves that for any sufficiently large set A_i (size essentially $2^{n-(n^{1-1/t})}$), if $\boldsymbol{X}$ is chosen uniformly at random from A_i, then the distribution of $M\boldsymbol{X}$ and $\overline{M\boldsymbol{X}}$ look identical to Bob. Consequently, since Bob's task is to, given

a vector w, decide whether w was chosen from $M\boldsymbol{X}$ or $\overline{M\boldsymbol{X}}$, the advantage of Bob over random guessing would be negligible.

To prove Theorem 5, we also follow the same approach described above. The main difference here is that to prove an information complexity lower bound we need to work with protocols that are randomized *even* on the distribution $\mathcal{D}$. This makes the problem more challenging since unlike deterministic protocols, randomized protocols do *not* split the input into disjoint distributions that are uniform (i.e., the sets $A_1, \ldots, A_{2^c}$ described above). To overcome this, we use Lemma 2.4 proved in Section 2.2 to partition the inputs conditioned on Alice's message into several near uniform parts and then apply the aforementioned technical result of [28, 55] on each part separately to finalize the proof.

We now provide the formal proof. We say that a set $A \subseteq \{0, 1\}^n$ is a *single parity* set iff the parity (i.e., the $\oplus$ summation) of each string in A is the same. The following lemma is the main ingredient of the proof in [55] (see Theorem 3.1 in [55]).

Lemma 4.1 ([55])**.** *Suppose $n = 2kt$ for some integer $k \geq 1$, $A \subseteq \{0, 1\}^n$ is a single parity set of size $|A| \geq 2^{n-c}$ for some $c \geq 1$, and x is a vector drawn uniformly at random from A (denote the distribution by U_A). Let M be a perfect t-hypermatching on $[n]$ chosen uniformly at random; there exists an* absolute *constant $\ell > 0$, such that for all $\varepsilon \in (0, 1]$, if $c \leq \ell \cdot \varepsilon \cdot n^{1-1/t}$, then,*

$$\mathbb{E}_{M} \left[\|p_M - q_M\|_{tvd} \right] \leq \varepsilon$$

where p_M and q_M are distributions over $\{0, 1\}^{n/t}$ whose p.d.f are (for any $z \in \{0, 1\}^{n/t}$)

$$p_M(z) := \Pr_{\boldsymbol{X} \sim U_A} (M\boldsymbol{X} = z)$$

and

$$q_M(z) := \Pr_{\boldsymbol{X} \sim U_A} (M\boldsymbol{X} = \overline{z})$$

respectively. In other words, $p_M = MU_A$ and $q_M = \overline{MU_A}$.

We are now ready to prove Theorem 5.

Proof of Theorem 5. Define $C_1 := \left(2\varepsilon^3 \cdot \ell \cdot n^{1-1/t} - 1 \right)$ for a constant ε (depending on δ) to be determined later[4]. Suppose towards a contradiction that $\mathsf{IC}^\delta_{\text{1-way}, \mathcal{D}}(\mathsf{BHH}_{n,t}) \leq C_1$ and let Π be a δ-error protocol for $\mathsf{BHH}_{n,t}$ on the distribution $\mathcal{D}$ with information cost C_1. Let $\boldsymbol{\Pi}$ be the random variable denoting the message sent from Alice to Bob using Π, and let $\boldsymbol{R}$ be the random variable denoting the public coins used in the protocol. We have,

$$\mathsf{IC}^\delta_{\text{1-way}, \mathcal{D}}(\mathsf{BHH}_{n,t}) = I(\boldsymbol{\Pi}; \boldsymbol{X} \mid \boldsymbol{R}) = H(\boldsymbol{X} \mid \boldsymbol{R}) - H(\boldsymbol{X} \mid \boldsymbol{R}, \boldsymbol{\Pi}) = n - H(\boldsymbol{X} \mid \boldsymbol{R}, \boldsymbol{\Pi})$$

where the last equality is because the input $\boldsymbol{X}$ is uniform on $\{0, 1\}^n$ in $\mathcal{D}$ and is chosen independently of the public coins $\boldsymbol{R}$. Consequently, we have $H(\boldsymbol{X} \mid \boldsymbol{R}, \boldsymbol{\Pi}) = n - C_1$. We further define a random variable $\boldsymbol{P} \in \{0, 1\}$ that indicates the *parity* of the input vector $\boldsymbol{X}$. We have,

$$H(\boldsymbol{X} \mid \boldsymbol{R}, \boldsymbol{\Pi}, \boldsymbol{P}) \geq H(\boldsymbol{X} \mid \boldsymbol{R}, \boldsymbol{\Pi}) - H(\boldsymbol{P}) = n - C_1 - 1$$

---

[4]Here, unlike the case in Theorem 4, we are *not* particularly interested in achieving the best dependence on the parameter δ, since in our proofs that require this theorem we always work with constant values of δ; this allows us to simplify the proof significantly.

where the inequality is by chain rule of entropy (Fact 2.2-(2)). Hence,

$$\mathbb{E}_{R,\pi,P}\Big[H(\boldsymbol{X} \mid \boldsymbol{R} = R, \boldsymbol{\Pi} = \pi, \boldsymbol{P} = P)\Big] \ge n - C_1 - 1$$

For brevity, we denote the event $(\boldsymbol{R} = R, \boldsymbol{\Pi} = \pi, \boldsymbol{P} = P)$ by Z. Define $C_2 := \frac{C_1+1}{\varepsilon} = 2\varepsilon^2 \cdot \ell \cdot n^{1-1/t}$. By Markov inequality,

$$\Pr_Z \left(n - H(\boldsymbol{X} \mid Z) \ge C_2\right) \le \frac{\mathbb{E}_{R,\pi,P}\Big[n - H(\boldsymbol{X} \mid \boldsymbol{R} = R, \boldsymbol{\Pi} = \pi, \boldsymbol{P} = P)\Big]}{C_2} \le \frac{C_1+1}{C_2} = \varepsilon$$

Hence, w.p. at least $1 - \varepsilon$, $H(\boldsymbol{X} \mid Z) \ge n - C_2$. Assuming this event happens, let $\boldsymbol{X}_Z$ denote the random variable $\boldsymbol{X}$ conditioned on Z and hence $H(\boldsymbol{X}_Z) \ge n - C_2$. We emphasize here that the randomness in $\boldsymbol{X}_Z$ lies both in the distribution $\mathcal{D}$ and in the *private* randomness used by the protocol.

Now consider the task of Bob for solving $\mathsf{BHH}_{n,t}$. Bob is given a perfect t-hypermatching M, a vector w, a message $\boldsymbol{\Pi} = \pi$ (from Alice), and the public coins $\boldsymbol{R} = R$. Suppose, additionally, we provide the parity of the input $\boldsymbol{X}$ to Bob for free , i.e., Bob knows $\boldsymbol{P} = P$. Conditioned on the aforementioned event happening (w.p. $1-\varepsilon$), Bob knows that the input of Alice is chosen from the distribution of the random variable $\boldsymbol{X}_Z$ with $H(\boldsymbol{X}_Z) \ge n - C_2$. For the hypermatching M of Bob, Bob is given a vector w chosen from the distribution of either $M\boldsymbol{X}_Z$ or $\overline{M\boldsymbol{X}_Z}$ and his task is to distinguish between these two cases. In the rest of the proof, we show that total variation distance of these two distributions is small and hence, by Fact 2.1, Bob will not able to distinguish them using a single sample.

Note that since the protocol is *not* deterministic, the distribution of $\boldsymbol{X}_Z$ is not necessarily uniform over its support and hence we cannot directly apply Lemma 4.1 to bound the total variation distance between the distributions $M\boldsymbol{X}_Z$ and $\overline{M\boldsymbol{X}_Z}$. To bypass this, we first apply Lemma 2.4 on the random variable $\boldsymbol{X}_Z$ with the parameter $\Delta = C_2$ and ε to obtain a sequence of $k + 1$ distributions $\mu_0, \ldots, \mu_k$ where $k = O(n/\varepsilon)$. For any $i \ge 1$, let U_i be the uniform distribution over the support of μ_i. By Lemma 2.4, $\|\mu_i - U_i\|_{tvd} = O(\varepsilon)$. Since M is chosen independently of $\boldsymbol{X}_Z$, this implies that $\|M\mu_i - MU_i\|_{tvd} = O(\varepsilon)$ and $\|\overline{M\mu_i} - \overline{MU_i}\|_{tvd} = O(\varepsilon)$. Furthermore, by Lemma 2.4, $|\mathrm{SUPP}(U_i)| = |\mathrm{SUPP}(\mu_i)| \ge 2^{n-\frac{\Delta}{\varepsilon}-\log\Theta(n/\varepsilon)} \ge 2^{n-\varepsilon\ell n^{1-1/t}}$, and hence by Lemma 4.1, $\mathbb{E}_M\Big[\|MU_i - \overline{MU_i}\|_{tvd}\Big] = O(\varepsilon)$. Finally, by triangle inequality, $\mathbb{E}_M\Big[\|M\mu_i - \overline{M\mu_i}\|_{tvd}\Big] = O(\varepsilon)$.

Again fix an $i \ge 1$. Suppose we further specify to Bob that $\boldsymbol{X}_Z$ is chosen from μ_i. We say that Bob is *successful* if for the given matching M and the vector w, he can correctly identify whether w is chosen from $M\mu_i$ or $\overline{M\mu_i}$; in other words, use one sample (i.e., w) to distinguish between $M\mu_i$ or $\overline{M\mu_i}$. Denote the event that Bob is successful by $\mathsf{success}$. For the following equations, let the summation over x ranges over all possible values of $\|M\mu_i - \overline{M\mu_i}\|_{tvd}$ (since there are only $n!$ matchings, there are at most $n!$ choices); we have,[5]

$$\Pr(\mathsf{success}) = \sum_x \Pr\left(\|M\mu_i - \overline{M\mu_i}\|_{tvd} = x\right) \cdot \Pr\left(\mathsf{success} \mid \|M\mu_i - \overline{M\mu_i}\|_{tvd} = x\right)$$

$$\le \sum_x \Pr\left(\|M\mu_i - \overline{M\mu_i}\|_{tvd} = x\right) \cdot \left(\frac{1}{2} + \frac{x}{2}\right) \qquad \text{(by Fact 2.1)}$$

---

[5]As stated after Theorem 4, to obtain a dependence of γ instead of γ^2 in the communication complexity lower bound proofs of [28,55], simply replace the Markov bound argument at the end of the their proofs with the slightly more careful argument based on probability of being successful presented here.

$$= \frac{1}{2} + \frac{1}{2} \cdot \sum_{x \in [0,1]} \Pr\left(\|M\mu_i - \overline{M\mu_i}\|_{tvd} = x\right) \cdot x$$

$$= \frac{1}{2} + \frac{1}{2} \cdot \mathbb{E}\left[\|M\mu_i - \overline{M\mu_i}\|_{tvd}\right] = \frac{1}{2} + O(\varepsilon)$$

To summarize, the advantage of Bob over randomly guessing the output is at most ε (for the unlikely event that $H(\boldsymbol{X}_Z) < n - C_1$) plus $O(\varepsilon)$ (for the unlikely event that $\boldsymbol{X}$ is chosen from μ_0 in Lemma 2.4) plus $O(\varepsilon)$ (for the advantage over random guessing, i.e., when event success happens in μ_i for $i \geq 1$). In summary, the probability of success of Bob is at most $\frac{1}{2} + O(\varepsilon) < 1 - \delta$, by choosing ε small enough in compare to δ. This means that the protocol succeeds w.p. strictly less that $1 - \delta$, a contradiction. ∎

For our purpose, it would be more convenient to work with a special case of the $\mathsf{BHH}_{n,t}$ problem, namely $\mathsf{BHH}^0_{n,t}$ in which the vector $w = 0^{n/t}$ and hence the goal of Bob is simply to decide whether $Mx = 0^{n/t}$ (Yes case) or $Mx = 1^{n/t}$ (No case). We define $\mathsf{BHM}^0_n := \mathsf{BHH}^0_{n,2}$ (similar to BHM_n). It is known that (see, [14, 44, 55]) any instance of the original $\mathsf{BHH}_{n,t}$ problem can be reduced to an instance of $\mathsf{BHH}^0_{2n,t}$ *deterministically* and with *no communication* between the players.

The following corollary summarizes the results in this section.

Corollary 6. *For any $n = 2kt$ (for some integer $k \geq 1$), there exists a distribution $\mathcal{D}_{\mathsf{BHH}}$ for $\mathsf{BHH}^0_{n,t}$ such that:*

- *For any $\delta \in (0,1)$ and $\gamma := \frac{1}{2} - \delta$, $\mathsf{CC}^\delta_{\text{1-way},\mathcal{D}_{\mathsf{BHH}}}(\mathsf{BHH}^0_{n,t}) = \Omega(\gamma \cdot n^{1-1/t})$.*

- *For any constant $\delta < 1/2$, $\mathsf{IC}^\delta_{\text{1-way},\mathcal{D}_{\mathsf{BHH}}}(\mathsf{BHH}^0_{n,t}) = \Omega(n^{1-1/t})$.*

- *Alice's input $\boldsymbol{X} \sim \mathcal{D}_{\mathsf{BHH}}$ is supported on boolean vectors $x \in \{0,1\}^n$ with $\|x\|_0 = \frac{n}{2}$.*

Moreover, these bounds also hold for, respectively, the communication cost and information cost of the protocols that are only *required to be correct w.p. $1 - \delta$ on the distribution $\mathcal{D}_{\mathsf{BHH}}$ (not necessarily on all inputs).*

We remark that this distribution satisfy the requirement of the Claim 2.8 (i.e., $\|x\|_0 = \frac{n}{2}$) and hence can be used in the reduction for the matching size problem mentioned in Section 2.

5 Space Lower Bounds for α-Approximating Matching Size

In this section, we present our space lower bounds for α-approximation algorithms in dynamic streams. As already remarked in Section 2, by the results of [4, 42], it suffices to prove the lower bound in the SMP model.

5.1 An $\Omega(\sqrt{n}/\alpha^{2.5})$ Lower Bound for Sparse Graphs

We consider the sparse graphs case in this section (i.e., Part (1) of Theorem 2), and show that any single-pass streaming algorithm that computes an α-approximation of matching size must use $\Omega(\sqrt{n}/\alpha^{2.5})$ bits of space even if the input graph only have $O(n)$ edges.

Define the *sparse matching size estimation* problem, $\mathsf{SMS}_{n,k}$, as the following k-player communication problem in the SMP model: each player $P^{(i)}$ is given a matching M_i over a set V of $n + \frac{n}{k}$ vertices[6] and the goal of the players is to approximate the maximum matching size of

---

[6]To simplify the exposition, we use $n + \frac{n}{k}$ instead of the usual n as the number of vertices.

$G(V, \bigcup_{i \in [k]} M_i)$ to within a factor *better than* $\frac{k+1}{2}$. We prove the following lower bound on the communication complexity of $\mathsf{SMS}_{n,k}$.

Theorem 7. *For any sufficiently large n, and $k \geq 2$, there exists a distribution $\mathcal{D}$ for $\mathsf{SMS}_{n,k}$ such that for any constant $\delta < 1/2$:* $\mathsf{CC}^{\delta}_{\mathrm{SMP},\mathcal{D}}(\mathsf{SMS}_{n,k}) = \Omega\left(\frac{\sqrt{n}}{k\sqrt{k}}\right)$

Part (1) of Theorem 2 immediately follows from Theorem 7.

Proof of Theorem 2, Part (1). Any SMP protocol for estimating matching size to within a factor of $\alpha < \frac{k+1}{2}$ can be used to solve the $\mathsf{SMS}_{n,k}$ problem. Moreover, as stated in Section 2.3, SMP communication complexity of a k-player problem is at most k times the space complexity of any single-pass streaming algorithm in dynamic streams [4, 42]; this finalizes the first part of the proof.

To see that the space complexity holds even when the input graph is both sparse and having bounded arboricity, notice that any graph G in $\mathsf{SMS}_{n,k}$ has exactly $k \cdot \frac{n}{k} = n$ edges (hence sparse); furthermore, since each player is given a matching (which is always a forest), the arboricity of G is at most $k \leq 2\alpha$. ∎

In the following, we focus on proving Theorem 7. This theorem is ultimately proved by a reduction from the BHM^0 problem defined in Section 4. However, this reduction is non-standard in the sense that it is *protocol-dependent*: given any protocol Π for SMS, we create a protocol for BHM^0 by *embedding* an instance of BHM^0 in the input of SMS, whereby the embedding is designed specifically for the protocol Π. It is worth mentioning that BHM^0 is a hard problem even in the one-way model, while the distribution that we create for SMS is only hard in the SMP model, meaning that if any player is allowed to send a single message to any other player (instead of the referee), then $\widetilde{O}(1)$ bits of communication suffices to solve the problem. Therefore, a key technical challenge here is to design a reduction from a one-way problem to a problem that is "inherently" simultaneous, or in other words, is easy to solve in the one-way model.

5.1.1 A Hard Input Distribution for $\mathsf{SMS}_{n,k}$

Let $\mathcal{D}_{\mathsf{BHM}}$ be the hard input distribution of $\mathsf{BHM}^0_{\frac{2n}{k}}$ in Corollary 6 (for $t = 2$) and $\mathcal{D}^{\mathsf{Y}}_{\mathsf{BHM}}$ and $\mathcal{D}^{\mathsf{N}}_{\mathsf{BHM}}$ be, respectively, the distribution on Yes and No instances of $\mathcal{D}_{\mathsf{BHM}}$.

The distribution $\mathcal{D}_{\mathsf{SMS}}$ for $\mathsf{SMS}_{n,k}$:

1. For each $i \in [k]$, independently draw a $\mathsf{BHM}^0_{\frac{2n}{k}}$ instance $(M^{\mathsf{B}}_i, x^{\mathsf{B}}_i) \sim \mathcal{D}_{\mathsf{BHM}}$.

2. Draw a *random* permutation $\sigma : \left[n + \frac{n}{k}\right] \to \left[n + \frac{n}{k}\right]$.

3. For each player $i \in [k]$, we define a mapping $\sigma_i : [\frac{2n}{k}] \to [n + \frac{n}{k}]$ as follows:

   - For each $j \in [\frac{2n}{k}]$ with $x^{\mathsf{B}}_i(j) = 1$, if $x^{\mathsf{B}}_i(j)$ is the ℓ-th smallest index with value 1, let $\sigma_i(j) := \sigma(\ell)$[a].
   - For each $j \in [\frac{2n}{k}]$ with $x^{\mathsf{B}}_i(j) = 0$, if $x^{\mathsf{B}}_i(j)$ is the ℓ-th smallest index with value 0, let $\sigma_i(j) := \sigma(i \cdot \frac{n}{k} + \ell)$.

4. The input to each player $P^{(i)}$ is a matching $M_i := \left\{ (\sigma_i(u), \sigma_i(v)) \mid (u, v) \in M^{\mathsf{B}}_i \right\}$.

---

[a]Here, we use the fact that $\|x^{\mathsf{B}}_i\|_0 = \frac{n}{k}$ in $\mathcal{D}_{\mathsf{BHM}}$ by Corollary 6

Observe that the distribution $\mathcal{D}_{\mathsf{SMS}}$ is defined by k instances of $\mathsf{BHM}^0_{\frac{2n}{k}}$, i.e., $(M_i^{\mathsf{B}}, x_i^{\mathsf{B}})$ (for $i \in [k]$), along with a mapping σ. The mapping σ relates the vectors x_i^{B} to the set of vertices in the final graph G while ensuring that across the players, for any $j \in [\frac{2n}{k}]$ where $x_i^{\mathsf{B}}(j) = 1$, the vertex that j maps to is *shared*, while the vertices with $x_i^{\mathsf{B}}(j) = 0$ are *unique* to each player. Moreover, the mapping σ_i provided to each player effectively describes the set of vertices (denoted by V_i) that the edges of $P^{(i)}$ will be incident on, and the matching M_i^{B} describes the edges between V_i. Hence, we can *uniquely* define the input of each player $P^{(i)}$ by the pair $(M_i^{\mathsf{B}}, \sigma_i)$, and from now on, without loss of generality, we assume the input given to each player $P^{(i)}$ is the pair $(M_i^{\mathsf{B}}, \sigma_i)$.

We should note right away that the distribution $\mathcal{D}_{\mathsf{SMS}}$ is not a "hard" distribution for $\mathsf{SMS}_{n,k}$ in the traditional sense: it is not hard to verify that for any graph $G \sim \mathcal{D}_{\mathsf{SMS}}$, $\mathrm{opt}(G)$ is concentrated around its expectation, and hence it is trivial to design a protocol when instances are promised to be *only* sampled from $\mathcal{D}_{\mathsf{SMS}}$: always output $\mathbb{E}_{G \sim \mathcal{D}_{\mathsf{SMS}}}[\mathrm{opt}(G)]$, which requires no communication from the players.

Nevertheless, the way we use the distribution $\mathcal{D}_{\mathsf{SMS}}$ as a hard distribution is to consider any protocol Π_{SMS} that succeeds *uniformly*, i.e., on *any* instance of $\mathsf{SMS}_{n,k}$; we then execute Π_{SMS} on $\mathcal{D}_{\mathsf{SMS}}$ and argue that in order to perform well on every instance of $\mathcal{D}_{\mathsf{SMS}}$, Π_{SMS} must convey a non-trivial amount of information about the input of the players in *some sub-distribution* of $\mathcal{D}_{\mathsf{SMS}}$. To continue, we need the following definitions.

Definition 3 (Input Profile). *For each graph $G \sim \mathcal{D}_{\mathsf{SMS}}$, we define the input profile of G to be a vector $f \in \{\mathsf{Yes}, \mathsf{No}\}^k$, where $f(i) = \mathsf{Yes}$ iff the i-th BHM instance $(M_i^{\mathsf{B}}, x_i^{\mathsf{B}})$ in G is a Yes instance and otherwise $f(i) = \mathsf{No}$.*

The 2^k different possible input profiles partition $\mathcal{D}_{\mathsf{SMS}}$ into 2^k different distributions. For any input profile f, we use the notation $\mathcal{D}_{\mathsf{SMS}} \mid f$ to denote the distribution of $\mathcal{D}_{\mathsf{SMS}}$ *conditioned* on its input profile being f. Two particularly interesting profiles for our purpose are the *all-equal* profiles, i.e., $f_{\mathsf{Yes}} := (\mathsf{Yes}, \ldots, \mathsf{Yes})$ and $f_{\mathsf{No}} := (\mathsf{No}, \ldots, \mathsf{No})$, due to the following claim.

Claim 5.1. *For any graph $G \sim (\mathcal{D}_{\mathsf{SMS}} \mid f_{\mathsf{Yes}})$, $\mathrm{opt}(G) \geq \frac{n}{2} + \frac{n}{2k}$, and for any graph $G \sim (\mathcal{D}_{\mathsf{SMS}} \mid f_{\mathsf{No}})$, $\mathrm{opt}(G) \leq \frac{n}{k}$.*

Proof. In $(\mathcal{D}_{\mathsf{SMS}} \mid f_{\mathsf{Yes}})$, each BHM instance $(M_i^{\mathsf{B}}, x_i^{\mathsf{B}})$ (for $i \in [k]$) is drawn from $\mathcal{D}_{\mathsf{SMS}}^{\mathsf{Y}}$, meaning that for every edge $(u, v) \in M_i^{\mathsf{B}}$, $x_i^{\mathsf{B}}(u) \oplus x_i^{\mathsf{B}}(v) = 0$. Therefore, either $x_i^{\mathsf{B}}(u) = x_i^{\mathsf{B}}(v) = 0$ or $x_i^{\mathsf{B}}(u) = x_i^{\mathsf{B}}(v) = 1$. Since M_i^{B} is a perfect matching over the set $[\frac{2n}{k}]$ and the hamming weight of x_i^{B} is $\frac{n}{k}$ (by Corollary 6), for half of the edges in M_i^{B}, we must have $x_i^{\mathsf{B}}(u) = x_i^{\mathsf{B}}(v) = 0$. Moreover, as $\mathcal{D}_{\mathsf{SMS}}$ maps every vertex with $x_i^{\mathsf{B}}(j) = 0$ to a distinct vertex in G, these $\frac{1}{2} \cdot |M_i^{\mathsf{B}}| = \frac{n}{2k}$ edges are vertex-disjoint with any other edge in the final graph G. Hence, between the k players, these edges together form a matching of size $k \cdot \frac{n}{2k} = \frac{n}{2}$. Finally, there is also a matching of size $\frac{n}{2k}$ between the shared vertices: simply use the edges corresponding to a matching M_i^{B} of an arbitrary player $P^{(i)}$ that are incident on shared vertices. This means that in this case, $\mathrm{opt}(G) \geq \frac{n}{2} + \frac{n}{2k}$.

In $(\mathcal{D}_{\mathsf{SMS}} \mid f_{\mathsf{No}})$, each BHM instance $(M_i^{\mathsf{B}}, x_i^{\mathsf{B}})$ (for $i \in [k]$) is drawn from $\mathcal{D}_{\mathsf{SMS}}^{\mathsf{N}}$, meaning that for every edge $(u, v) \in M_i^{\mathsf{B}}$, $x_i^{\mathsf{B}}(u) \oplus x_i^{\mathsf{B}}(v) = 1$. Therefore, exactly one of $x_i^{\mathsf{B}}(u)$ or $x_i^{\mathsf{B}}(v)$ is equal to 1. In $\mathcal{D}_{\mathsf{SMS}}$, for every player, the vertices where $x_i^{\mathsf{B}}(j) = 1$ are all mapped to the (same) set of vertices $\{\sigma(1), \sigma(2), \ldots, \sigma(\frac{n}{k})\}$ (denoted by V_0). Therefore, in the final graph G, *every* edge of *every* player is incident on some vertex in V_0, and hence the maximum matching size in G is at most $|V_0| = \frac{n}{k}$. ■

In the following, we fix any δ-error protocol Π_{SMS} for $\mathsf{SMS}_{n,k}$. By Claim 5.1, Π_{SMS} is also a δ-error protocol for distinguishing between the two distributions $(\mathcal{D}_{\mathsf{SMS}} \mid f_{\mathsf{Yes}})$ and $(\mathcal{D}_{\mathsf{SMS}} \mid f_{\mathsf{No}})$:

simply output Yes if the estimate of $\mathrm{opt}(G)$ is strictly larger than $\frac{n}{k}$ and output No otherwise. From here on, with a slight abuse of notation, we say that Π_{SMS} outputs Yes whenever it estimates $\mathrm{opt}(G)$ strictly larger than $\frac{n}{k}$ and outputs No otherwise (this notation is defined over *any* input, not necessarily chosen from $(\mathcal{D}_{\mathsf{SMS}} \mid f_{\mathsf{Yes}})$ or $(\mathcal{D}_{\mathsf{SMS}} \mid f_{\mathsf{No}})$).

Intuitively, to distinguish between $(\mathcal{D}_{\mathsf{SMS}} \mid f_{\mathsf{Yes}})$ and $(\mathcal{D}_{\mathsf{SMS}} \mid f_{\mathsf{No}})$, one should solve (at least one of) the BHM^0 instances embedded in the distribution. This naturally suggests the possibility of performing a reduction from BHM^0 and arguing that the distribution on $(\mathcal{D}_{\mathsf{SMS}} \mid f_{\mathsf{Yes}})$ and $(\mathcal{D}_{\mathsf{SMS}} \mid f_{\mathsf{No}})$ is a hard distribution for $\mathsf{SMS}_{n,k}$. However, in the case of these two distributions, the k BHM^0 instances are highly correlated and hence it is hard to reason about which BHM^0 instance is "actually being solved". To get around this, we try Π_{SMS} on other input profiles, with, informally speaking, less correlation across the BHM instances. An immediate issue here is that, unlike the case for the distributions $(\mathcal{D}_{\mathsf{SMS}} \mid f_{\mathsf{Yes}})$ and $(\mathcal{D}_{\mathsf{SMS}} \mid f_{\mathsf{No}})$, the matching sizes for graphs drawn from the other input profiles do not have a large gap. Hence, a priori it is not even clear what the actual task of Π_{SMS} is, or why Π_{SMS} should be able to distinguish them. However, we show that there are special pairs of input profiles (other than f_{Yes} and f_{No}) with our desired property (i.e., "low" correlation between the BHM^0 instances) that Π_{SMS} is still able to distinguish. These pairs are ultimately connected to the (property of) protocol Π_{SMS} itself and hence vary across different choices for the protocol Π_{SMS}; this is the main reason that we perform a protocol-dependent reduction in our proof.

For any input profile f, define p_f^{Y} (resp. p_f^{N}) as the probability that Π_{SMS} outputs Yes (resp. No) when its input is sampled from $\mathcal{D}_{\mathsf{SMS}} \mid f$. We define the notation of *informative index* for the protocol Π_{SMS}.

Definition 4 (Informative Index). *We say that an index $i \in [k]$ is γ-informative for the protocol Π_{SMS} iff there exist two input profiles f and g where $f(i) = \mathsf{Yes}$, $g(i) = \mathsf{No}$, and $f(j) = g(j)$ for all $j \neq i$, such that $p_f^{\mathsf{Y}} + p_g^{\mathsf{N}} \geq 1 + 2\gamma$. In this case, the input profiles f and g are called the witness of i.*

Informally speaking, if Π_{SMS} has a γ-informative index i, then Π_{SMS} can distinguish whether the i-th BHM^0 instance is a Yes or No instance w.p. at least $\frac{1}{2} + \gamma$ (i.e., Π_{SMS} solves the i-th BHM^0 instance). In the rest of this section, we prove that indeed every protocol Π_{SMS} has an informative index.

Lemma 5.2. *Any δ-error protocol Π_{SMS} for SMS has a γ-informative index for $\gamma = \frac{1-2\delta}{2k}$.*

Proof. Suppose towards a contradiction that for any two input profiles f and g that differ only on one entry (say i, and $f(i) = \mathsf{Yes}$, $g(i) = \mathsf{No}$), we have, $p_f^{\mathsf{Y}} + p_g^{\mathsf{N}} < 1 + 2\gamma$ for $\gamma = \frac{1-2\delta}{2k}$.

Consider the following sequence of $(k+1)$ input profiles:

$$(f_{\mathsf{Yes}} =)(\mathsf{Yes}, \mathsf{Yes}, \ldots, \mathsf{Yes}), (\mathsf{No}, \mathsf{Yes}, \ldots, \mathsf{Yes}), (\mathsf{No}, \mathsf{No}, \ldots, \mathsf{Yes}), \ldots, (\mathsf{No}, \mathsf{No}, \ldots, \mathsf{No})(= f_{\mathsf{No}})$$

whereby, for the j-th input profile of this sequence (denoted by f_j), the first $j - 1$ entries of f_j are all No, and the rest are all Yes.

Observe that for any $j \in [k]$, the input profiles f_j and f_{j+1} differ in exactly one entry j, and $f_j(j) = \mathsf{Yes}$, while $f_{j+1}(j) = \mathsf{No}$. Hence, by our assumption, we have $p_{f_j}^{\mathsf{Y}} + p_{f_{j+1}}^{\mathsf{N}} < 1 + 2\gamma$, which implies

$$p_{f_j}^{\mathsf{Y}} < 1 + 2\gamma - p_{f_{j+1}}^{\mathsf{N}} = p_{f_{j+1}}^{\mathsf{Y}} + 2\gamma \qquad\qquad (p_{f_{j+1}}^{\mathsf{Y}} + p_{f_{j+1}}^{\mathsf{N}} = 1)$$

Therefore,

$$p_{f_1}^{\mathsf{Y}} < p_{f_2}^{\mathsf{Y}} + 2\gamma < p_{f_3}^{\mathsf{Y}} + 2\gamma \cdot 2 < \cdots < p_{f_{k+1}}^{\mathsf{Y}} + 2\gamma \cdot k$$

which implies (by adding $p^{\mathsf{N}}_{f_{k+1}}$ to both sides of the inequality)

$$p^{\mathsf{Y}}_{f_1} + p^{\mathsf{N}}_{f_{k+1}} < p^{\mathsf{Y}}_{f_{k+1}} + p^{\mathsf{N}}_{f_{k+1}} + 2\gamma \cdot k = 1 + 2\gamma \cdot k = 2 \cdot (1 - \delta) \tag{2}$$

by our choice of γ. However, since Π_{SMS} is a δ-error protocol for $\mathsf{SMS}_{n,k}$, by Claim 5.1, the probability that Π_{SMS} succeeds in distinguishing $(\mathcal{D}_{\mathsf{SMS}} \mid f_{\mathsf{Yes}})$ from $(\mathcal{D}_{\mathsf{SMS}} \mid f_{\mathsf{No}})$ on the distribution $\frac{1}{2}(\mathcal{D}_{\mathsf{SMS}} \mid f_{\mathsf{Yes}}) + \frac{1}{2}(\mathcal{D}_{\mathsf{SMS}} \mid f_{\mathsf{No}})$ is at least $1 - \delta$. Therefore, $\frac{1}{2} \cdot (p^{\mathsf{Y}}_{f_1} + p^{\mathsf{N}}_{f_{k+1}}) \geq 1 - \delta$, a contradiction to Eq (2). ∎

In the next section, we use existence of a γ-informative index in any protocol Π_{SMS} for $\mathsf{SMS}_{n,k}$ to obtain a protocol for $\mathsf{BHM}^0_{\frac{2n}{k}}$ w.p. of success at least $\frac{1}{2} + \gamma$, based Π_{SMS}.

5.1.2 The Reduction From the $\mathsf{BHM}^0_{\frac{2n}{k}}$ Problem

Recall that Π_{SMS} is a δ-error protocol for the distribution $\mathcal{D}_{\mathsf{SMS}}$. Let $i^\star$ be a γ-informative index of Π_{SMS} (as in Lemma 5.2), and let input profiles $f_{i^\star}$ and $g_{i^\star}$ be the witness of $i^\star$.

Protocol Π_{BHM}. A protocol for reducing $\mathsf{BHM}^0_{\frac{2n}{k}}$ to $\mathsf{SMS}_{n,k}$

Input: An instance $(M, x) \sim \mathcal{D}_{\mathsf{BHM}}$ of $\mathsf{BHM}^0_{\frac{2n}{k}}$.
Output: Yes if $Mx = 0^{\frac{n}{k}}$ and No if $Mx = 1^{\frac{n}{k}}$.

---

1. Bob creates the input $(M^{\mathsf{B}}_{i^\star}, \sigma_{i^\star})$ for the player $P^{(i^\star)}$ as follows:

   - Let $M^{\mathsf{B}}_{i^\star} = M$.
   - Using *public randomness*, Bob picks $\sigma_{i^\star}$ to be a *uniformly random* injection from $[\frac{2n}{k}]$ to $[n + \frac{n}{k}]$.
   - Let $V_{i^\star}$ be the image of $\sigma_{i^\star}$ (i.e., $V_{i^\star} = \{\sigma_{i^\star}(j) \mid j \in [\frac{2n}{k}]\}$).

2. Alice generates the inputs for all other players. Using *private randomness*, Alice first randomly partitions the set $[n + \frac{n}{k}] \setminus V_{i^\star}$ into $(k-1)$ sets $\{V'_i\}_{i \in [k] \setminus \{i^\star\}}$, where each V'_i has size $\frac{n}{k}$. She then generates the input of each player $P^{(i)}$ $(i \neq i^\star)$ as follows:

   - If $f_{i^\star}(i) = \mathsf{Yes}$ (resp. $f_{i^\star}(i) = \mathsf{No}$), Alice draws a $\mathsf{BHM}^0_{\frac{2n}{k}}$ instance $(M^{\mathsf{B}}_i, x^{\mathsf{B}}_i)$ from $\mathcal{D}^{\mathsf{Y}}_{\mathsf{BHM}}$ (resp. from $\mathcal{D}^{\mathsf{N}}_{\mathsf{BHM}}$).
   - The mapping $\sigma_i : [\frac{2n}{k}] \to [n + \frac{n}{k}]$ is defined as follows. For the $\frac{n}{k}$ entries in $[\frac{2n}{k}]$ where x_i is 0, Alice assigns a *uniformly random* bijection to V'_i. For each entry j in $[\frac{2n}{k}]$ where $x^{\mathsf{B}}_i(j) = 1$, suppose $x^{\mathsf{B}}_i(j)$ is the ℓ-th 1 of x_i, Alice assigns $\sigma_i(j) = \sigma_{i^\star}(j')$ where j' is the index such that $x(j')$ is the ℓ-th 1 of x.[a]

3. Bob runs Π_{SMS} for the $i^\star$-th player and Alice runs Π_{SMS} for all other players and sends the messages of all other players to Bob.

4. After receiving the messages from Alice, Bob runs the referee part of the protocol Π_{SMS}, and outputs the same answer as Π_{SMS}.

---

[a]Recall that x is the input vector to Alice in a BHM^0 instance.

It is relatively straightforward to verify that the distribution of the instances created by this reduction and the original distributions $(\mathcal{D}_{\mathsf{SMS}} \mid f_{i^\star})$ and $(\mathcal{D}_{\mathsf{SMS}} \mid g_{i^\star})$ are identical. Formally,

Claim 5.3. *Suppose (M, x) is a* Yes *(resp.* No*) BHM instance; then the SMS instance constructed by Alice and Bob in the given reduction is sampled from $\mathcal{D}_{\mathsf{SMS}} \mid f_{i^\star}$ (resp. $\mathcal{D}_{\mathsf{SMS}} \mid g_{i^\star}$).*

The proof of this claim is deferred to the end of this section.

Proof of Theorem 7. Let $\gamma = \frac{1-2\delta}{2k}$; we first argue that Π_{BHM} outputs a correct answer for $\mathsf{BHM}^0_{\frac{2n}{k}}$ w.p. at least $\frac{1}{2} + \gamma$. If the input BHM^0 instance (M, x) is a Yes (resp. No) instance, then by Claim 5.3, the distribution of the SMS instance created in Π_{BHM} is exactly $\mathcal{D}_{\mathsf{SMS}} \mid f_{i^\star}$ (resp. $\mathcal{D}_{\mathsf{SMS}} \mid g_{i^\star}$); consequently, Π_{SMS} outputs the correct answer w.p. $\frac{1}{2} \cdot \left(p^{\mathsf{Y}}_{f_{i^\star}} + p^{\mathsf{N}}_{g_{i^\star}} \right)$. Since $i^\star$ is a $\frac{1-2\delta}{2k}$-informative instance, we have $\frac{1}{2} \cdot \left(p^{\mathsf{Y}}_{f_{i^\star}} + p^{\mathsf{N}}_{g_{i^\star}} \right) \geq \frac{1}{2} + \frac{1-2\delta}{2k} = \frac{1}{2} + \gamma$ and hence the protocol Π_{BHM} outputs the correct answer w.p. at least $\frac{1}{2} + \gamma$.

Now notice that in Π_{BHM}, Alice is sending messages of $k-1$ players in Π_{SMS} to Bob and hence communication cost of Π_{BHM} is at most the communication cost of Π_{SMS}. Since solving $\mathsf{BHM}_{\frac{2n}{k}}$ on $\mathcal{D}_{\mathsf{BHM}}$ w.p. of success $\frac{1}{2} + \gamma$ requires at least $\Omega(\gamma \cdot \sqrt{\frac{n}{k}})$ bits of communication by Corollary 6, we have $\|\Pi_{\mathsf{SMS}}\| = \Omega(\gamma \cdot \sqrt{\frac{n}{k}})$. Moreover, $\gamma = \frac{\varepsilon}{k}$ for some constant ε bounded away from 0 (since δ is a constant bounded away from $1/2$), hence we obtain that $\mathsf{CC}^{\delta}_{\mathsf{SMP}, \mathcal{D}}(\mathsf{SMS}_{n,k}) = \Omega\left(\frac{\sqrt{n}}{k\sqrt{k}} \right)$ for $\mathcal{D} := \frac{1}{2} \left(\mathcal{D}_{\mathsf{SMS}} \mid f_{i^\star} \right) + \frac{1}{2} \left(\mathcal{D}_{\mathsf{SMS}} \mid g_{i^\star} \right)$. ∎

It only remains to prove Claim 5.3.

Proof of Claim 5.3. Suppose the input BHM^0 instance (M, x) is a Yes instance. We need to prove that the distribution of the SMS instance created by the protocol Π_{BHM} is the same as the distribution $\mathcal{D}_{\mathsf{SMS}} \mid f_{i^\star}$ (the case where (M, x) is a No instance is similar). In the following, we will go through the construction of $\mathcal{D}_{\mathsf{SMS}}$ conditioned on $f_{i^\star}$ step by step and explain how the reduction captures each step of the construction.

Firstly, in the distribution $\mathcal{D}_{\mathsf{SMS}} \mid f_{i^\star}$, we draw k instances of BHM^0, where the i-th instance $(M^{\mathsf{B}}_i, x^{\mathsf{B}}_i)$ is drawn from the Yes (resp. No) BHM^0 instances if $f_{i^\star}(i) = $ Yes (resp. $f_{i^\star}(i) = $ No). It is straightforward to verify that for any $i \neq i^\star$, in the reduction, the BHM^0 instance created by Alice is drawn following $f_{i^\star}(i)$. The $i^\star$-th instance corresponds to the original input of Alice and Bob and since we assume it is a Yes instance, this instance is also sampled following $f_{i^\star}(i)$. This implies that the first part of the input of every player (i.e., the matchings over $[\frac{2n}{k}]$) are drawn the way in the reduction as in the original distribution.

Secondly, in the original distribution, we draw a random permutation $\sigma : [n + \frac{n}{k}] \to [n + \frac{n}{k}]$ and use σ to define the mapping σ_i of each player i: map the vertices j ($j \in [\frac{2n}{k}]$) where $x_i(j) = 1$ (resp. $x_i(j) = 0$) to the same set of vertices $\{\sigma(1), \sigma(2), \ldots, \sigma(\frac{n}{k})\}$ (resp. to a private set of vertices $\{\sigma(i \cdot \frac{n}{k} + 1), \sigma(i \cdot \frac{n}{k} + 2), \ldots, \sigma(i \cdot \frac{n}{k} + \frac{n}{k})\}$) in the final graph G.

In the reduction, Bob picks a random injection $\sigma_{i^\star}$, and defines the image of $\sigma_{i^\star}$ by $V_{i^\star}$; Alice randomly partition $[n + \frac{n}{k}] \setminus V_{i^\star}$ into $k-1$ sets of size $\frac{n}{k}$ each. For each of the other player $i \neq i^\star$, Alice assigns the entries in $[\frac{2n}{k}]$ where x_i is 1 to V^* following the same procedure as $\mathcal{D}_{\mathsf{SMS}}$. For the entries in $[\frac{2n}{k}]$ where x_i is 0, Alice assigns a random bijection to V_i.

To see that the two ways of defining the mappings σ_i's are equivalent in the original distribution and in the reduction, simply note that one can decompose the choice of the random permutation σ into the following steps:

1. Pick a random subset of size $\frac{2n}{k}$ (i.e., $V_{i^\star}$).

20

2. Partition the remaining universe into $k-1$ sets of size $\frac{n}{k}$ each.

3. Pick a random permutation for each set (which is equivalent to picking a random bijection as used in the reduction).

Therefore, the mappings σ_i's created by the reduction induce the same distribution as the mappings created by $\mathcal{D}_{\mathsf{SMS}}$. Hence, the distribution of the inputs of all players created by the protocol Π_{BHM} is the same as the distribution $\mathcal{D}_{\mathsf{SMS}} \mid f_1$. ∎

5.2 An $\Omega(n/\alpha^2)$ Lower Bound for Dense Graphs

We switch to the dense graphs case in this section (i.e., Part (2) of Theorem 2), and establish a better lower bound of $\Omega(n/\alpha^2)$ for computing an α-approximate matching size in dynamic streams.

We define $\mathsf{Matching}_{n,k,\alpha}$ as the k-*player simultaneous communication* problem of estimating the matching size to within a factor of α, when edges of an n-vertex input graph $G(V, E)$ are partitioned across the k-players. In this section, we prove the following lower bound on the information complexity of $\mathsf{Matching}_{n,k,\alpha}$ in the SMP communication model.

Theorem 8 (Lower bound for $\mathsf{Matching}_{n,k,\alpha}$)**.** *For any sufficiently large n and α, there exists some $k = \alpha \cdot \left(\frac{n}{\alpha}\right)^{o(1)}$ and a distribution $\mathcal{D}_{\mathsf{M}}$ for $\mathsf{Matching}_{n,k,\alpha}$ such that for any constant $\delta < \frac{1}{2}$:*

$$\mathsf{IC}^{\delta}_{\mathrm{SMP}, \mathcal{D}_{\mathsf{M}}}(\mathsf{Matching}_{n,k,\alpha}) = \Omega(nk/\alpha^2)$$

Theorem 8, combined with Proposition 2.7, immediately gives the same lower bound on the SMP communication complexity of $\mathsf{Matching}_{n,k,\alpha}$. Since SMP communication complexity of a k-player problem is at most k times the space complexity of any single-pass streaming algorithm in dynamic streams [4, 42], this immediately proves the $\Omega(n/\alpha^2)$ lower bound in Part (2) of Theorem 3. We now prove Theorem 8.

The hard distribution $\mathcal{D}_{\mathsf{M}}$ for $\mathsf{Matching}_{n,k,\alpha}$:

  Parameters: $r = N^{1-o(1)}$, $t = \frac{\binom{N}{2} - o(N^2)}{r}$, $k = \frac{(\alpha+1)N}{r}$, $n = N + k \cdot r$.

1. Fix an (r, t)-RS graph G^{RS} on N vertices with induced matchings $M_1^{\mathsf{RS}}, \ldots, M_t^{\mathsf{RS}}$.

2. Pick $j^\star \in [t]$ and $\theta \in \{0, 1\}$ independently and uniformly at random.

3. For each player $P^{(i)}$ independently,

   (a) Denote by G_i the input graph of $P^{(i)}$, initialized to be a copy of G^{RS} with vertices $V_i = [N]$. Moreover, define V_i^* as the set of vertices incident on the matching $M_{j^\star}^{\mathsf{RS}}$.

   (b) Let $x^{(i)}$ be a t-dimensional vector, whereby $x^{(i)}(j^\star) = \theta$ and for any $j \neq j^\star$, $x^{(i)}(j)$ is chosen uniformly at random from $\{0, 1\}$.

   (c) For any $j \in [t]$, if $x^{(i)}(j) = 0$ *remove* the matching $M_{j^\star}^{\mathsf{RS}}$ from G_i (otherwise, do nothing).

4. Pick a random permutation σ of $[n]$. For every player $P^{(i)}$, for each vertex v in $V_i \setminus V_i^*$ with label j ($\in [N]$), *relabel* v to $\sigma(j)$. Enumerate the vertices in V_i^* (from the one with the smallest label to the largest), and relabel the j-th vertex to $\sigma(N + (i-1) \cdot 2r + j)$. In the final graph, the vertices with the same label correspond to the same vertex.

The vertices whose labels belong to $\sigma([N])$ are referred to as *shared* vertices since they belong to the input graph of *every* player, and the vertices V_i^* are referred to as the *private* vertices of the player $P^{(i)}$ since they only appear in the input graph of $P^{(i)}$ (in the final graph, i.e., after relabeling). We point out that, in general, the final graph constructed by this distribution is a multi-graph with n vertices and $O(kN^2) = O(n^2/\alpha)$ edges (counting the multiplicities); the multiplicity of each edge is also at most k. Finally, the existence of an (r,t)-RS graph G^{RS} with the parameters used in this distribution is guaranteed by a result of [7] (see Section 2.1).

Claim 5.4. *Let:*

$$\mathrm{opt}_1 := \min_G \left(\mathrm{opt}(G) \mid G \text{ is chosen from } \mathcal{D}_{\mathsf{M}} \text{ conditioned on } \theta = 1 \right)$$

$$\mathrm{opt}_0 := \max_G \left(\mathrm{opt}(G) \mid G \text{ is chosen from } \mathcal{D}_{\mathsf{M}} \text{ conditioned on } \theta = 0 \right).$$

then, $\mu_1 > \alpha \cdot \mu_0$.

Proof. Notice that in each graph G_i, except for the matching $M_{j^\star}^{\mathsf{RS}}$, all other matching edges are incident on the set of shared vertices. This implies that across the players, the total contribution of all matchings except for $M_{j^\star}^{\mathsf{RS}}$'s is at most N. Consequently, when $\theta = 0$, i.e., when the matching $M_{j^\star}^{\mathsf{RS}}$ of each player is removed, $\mathrm{opt}(G) \leq N$. On the other hand, when $\theta = 1$, since the matching $M_{j^\star}^{\mathsf{RS}}$ of each player is incident on a unique set of vertices of G (i.e., private vertices), they form a matching of size $k \cdot r = (\alpha + 1) \cdot N$. Hence, $\mathrm{opt}(G) \geq (\alpha + 1) \cdot N$ in this case. ∎

Claim 5.4 shows that any δ-error protocol Π_{Matching} for $\mathsf{Matching}_{n,k,\alpha}$ can determine the value of the parameter θ in the distribution $\mathcal{D}_{\mathsf{M}}$ (also with error prob. δ). We use this fact to prove a lower bound on the mutual information between the parameter θ and the message of the players. Define $\boldsymbol{\theta}$, $\boldsymbol{\sigma}$, and $\boldsymbol{J}$, as random variables for, respectively, the parameter θ, the random permutation σ, and the index $j^\star$ in the distribution. We have the following simple claim.

Claim 5.5. *For any $\delta < 1/2$ and δ-error protocol Π_{Matching}, $I(\boldsymbol{\theta}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{\sigma}, \boldsymbol{J}, \boldsymbol{R}) = \Omega(1)$.*

Proof. As proven in Claim 5.4, protocol Π_{Matching} can be used directly to determine the value of $\boldsymbol{\theta}$ w.p. $1 - \delta$. Hence, by Fano's inequality (Claim 2.3), $H(\boldsymbol{\theta} \mid \boldsymbol{\Pi}_{\mathsf{Matching}}, \boldsymbol{R}) \leq H_2(\delta)$, since Π_{Matching} uses the message $\boldsymbol{\Pi}_{\mathsf{Matching}}$ together with the public coins $\boldsymbol{R}$ to output the answer. We further have,

$$H_2(\delta) \geq H(\boldsymbol{\theta} \mid \boldsymbol{\Pi}_{\mathsf{Matching}}, \boldsymbol{R}) \geq H(\boldsymbol{\theta} \mid \boldsymbol{\Pi}_{\mathsf{Matching}}, \boldsymbol{R}, \boldsymbol{\sigma}, \boldsymbol{J})$$
$$\text{(conditioning reduces the entropy (Fact 2.2-(1)))}$$
$$= H(\boldsymbol{\theta} \mid \boldsymbol{\sigma}, \boldsymbol{R}, \boldsymbol{J}) - I(\boldsymbol{\theta}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{R}, \boldsymbol{\sigma}, \boldsymbol{J}) = 1 - I(\boldsymbol{\theta}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{R}, \boldsymbol{\sigma}, \boldsymbol{J})$$

where the last equality is because $\boldsymbol{\theta}$ is chosen uniformly at random from $\{0, 1\}$ independent of $\boldsymbol{\sigma}, \boldsymbol{R}$ and $\boldsymbol{J}$. Consequently, we have $I(\boldsymbol{\theta}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{R}, \boldsymbol{\sigma}, \boldsymbol{J}) \geq 1 - H_2(\delta) = \Omega(1)$ (since $\delta < 1/2$ is a constant). ∎

We now use the bound in Claim 5.5 to lower bound the information cost of any δ-error protocol Π_{Matching}.

Lemma 5.6. *For any $\delta < 1/2$ and δ-error protocol Π_{Matching}, $\mathsf{ICost}_{\mathcal{D}_{\mathsf{M}}}(\Pi_{\mathsf{Matching}}) = \Omega(t)$.*

Proof. By Claim 5.5, $I(\boldsymbol{\theta}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{R}, \boldsymbol{\sigma}, \boldsymbol{J}) = \Omega(1)$; in the following, we prove that for this to happen, information cost of Π_{Matching} needs to be $\Omega(t)$. We have,

$$I(\boldsymbol{\theta}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{R}, \boldsymbol{\sigma}, \boldsymbol{J}) = \mathop{\mathbb{E}}_{j \in [t]} I(\boldsymbol{\theta}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{R}, \boldsymbol{\sigma}, \boldsymbol{J} = j)$$

$$= \frac{1}{t} \sum_{j=1}^{t} I(\boldsymbol{\theta}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{\sigma}, \boldsymbol{R}, \boldsymbol{J} = j)$$

$$= \frac{1}{t} \sum_{j=1}^{t} I(\boldsymbol{Y}_j; \boldsymbol{\Pi}_{\mathsf{Matching}}^{(1)}, \dots, \boldsymbol{\Pi}_{\mathsf{Matching}}^{(k)} \mid \boldsymbol{\sigma}, \boldsymbol{R}, \boldsymbol{J} = j)$$

Here, for any $j \in [t]$, $\boldsymbol{Y}_j := (\boldsymbol{X}_{1,j}, \boldsymbol{X}_{2,j}, \dots, \boldsymbol{X}_{k,j})$, where $\boldsymbol{X}_{i,j}$ (for any $i \in [k]$) is the random variable denoting $x^{(i)}(j)$. This equality holds since conditioned on $\boldsymbol{J} = j$, for any $i \in [k]$, each $x^{(i)}(j)$ is assigned to be θ (i.e., $\boldsymbol{\theta} = \boldsymbol{X}_{i,j}$ conditioned on $\boldsymbol{J} = j$). Moreover,

$$I(\boldsymbol{\theta}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{R}, \boldsymbol{\sigma}, \boldsymbol{J}) \leq \frac{1}{t} \sum_{j=1}^{t} \sum_{i=1}^{k} I(\boldsymbol{Y}_j; \boldsymbol{\Pi}_{\mathsf{Matching}}^{(i)} \mid \boldsymbol{\sigma}, \boldsymbol{R}, \boldsymbol{J} = j)$$

by conditional sub-additivity of mutual information (Fact 2.2-(4)) since for any $i \in [k]$, $\boldsymbol{\Pi}_{\mathsf{Matching}}^{(i)}$ and $\boldsymbol{\Pi}_{\mathsf{Matching}}^{-i}$ are *independent* conditioned on $\boldsymbol{Y}_j, \boldsymbol{\sigma}, \boldsymbol{R}$ and $\boldsymbol{J} = j$. We can also drop the conditioning on the event $\boldsymbol{J} = j$ and have,

$$I(\boldsymbol{\theta}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{R}, \boldsymbol{\sigma}, \boldsymbol{J}) \leq \frac{1}{t} \sum_{j=1}^{t} \sum_{i=1}^{k} I(\boldsymbol{Y}_j; \boldsymbol{\Pi}_{\mathsf{Matching}}^{(i)} \mid \boldsymbol{\sigma}, \boldsymbol{R})$$

since $\boldsymbol{\Pi}_{\mathsf{Matching}}^{(i)}$ is a function of $(\boldsymbol{X}_i, \boldsymbol{\sigma}_i)$ where $\boldsymbol{X}_i := (\boldsymbol{X}_{i,1}, \dots, \boldsymbol{X}_{i,t})$ is a random variable for the vector $x^{(i)}$. Moreover, $\boldsymbol{X}_i$ defines the graph G_i without the labels, i.e., over the set of vertices $V_i := [N]$ and $\boldsymbol{\sigma}_i$ is the random variable denoting how the vertices of the player $P^{(i)}$ map to G, i.e., specify the labels of vertices. Therefore $(\boldsymbol{X}_i, \boldsymbol{\sigma}_i)$ is independent of $\boldsymbol{J} = j$ (given the input graph G_i, each matching has the same probability of being the chosen matching for $j^\star$); hence it is easy to see that all four random variables in above term are independent of the event $\boldsymbol{J} = j$. Moreover, since $\boldsymbol{Y}_j$ and $\boldsymbol{Y}^{-j}$ are independent of each other, conditioned on $\boldsymbol{\sigma}$ and $\boldsymbol{R}$, by conditional super-additivity of mutual information (Fact 2.2-(5)),

$$I(\boldsymbol{\theta}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{R}, \boldsymbol{\sigma}, \boldsymbol{J}) \leq \frac{1}{t} \sum_{i=1}^{k} I(\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_t; \boldsymbol{\Pi}_{\mathsf{Matching}}^{(i)} \mid \boldsymbol{\sigma}, \boldsymbol{R})$$

$$= \frac{1}{t} \sum_{i=1}^{k} I(\boldsymbol{X}_1, \dots, \boldsymbol{X}_k; \boldsymbol{\Pi}_{\mathsf{Matching}}^{(i)} \mid \boldsymbol{\sigma}, \boldsymbol{R})$$

$$(\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_t \text{ uniquely determines } \boldsymbol{X}_1, \dots, \boldsymbol{X}_k \text{ and vice versa})$$

$$\leq \frac{1}{t} \sum_{i=1}^{k} I(\boldsymbol{X}_1, \dots, \boldsymbol{X}_k, \boldsymbol{\sigma}_1, \dots, \boldsymbol{\sigma}_k; \boldsymbol{\Pi}_{\mathsf{Matching}}^{(i)}, \boldsymbol{R})$$

$$(\text{by chain rule of mutual information (Fact 2.2-(3)))}$$

$$= \frac{1}{t} \cdot \mathsf{ICost}_{\mathcal{D}_{\mathsf{M}}}(\Pi_{\mathsf{Matching}})$$

where the last equality is because $(\boldsymbol{X}_i, \boldsymbol{\sigma}_i)$ uniquely determines the input to the player $P^{(i)}$ for $i \in [k]$ and vice versa. Since $I(\boldsymbol{\theta}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{R}, \boldsymbol{\sigma}, \boldsymbol{J}) = \Omega(1)$, we obtain that $\mathsf{ICost}_{\mathcal{D}_{\mathsf{M}}}(\Pi_{\mathsf{Matching}}) = \Omega(t)$. ∎

Theorem 8 now follows from Lemma 5.6 by noticing that $n = (2\alpha + 1) \cdot N$, $r = 2\alpha N/k$ and $t \geq \frac{N^2}{2r} = \frac{N \cdot k}{2\alpha} = \Omega(nk/\alpha^2)$.

6 Space Lower Bounds for $(1+\varepsilon)$-Approximating Matching Size

In this section, we present our space lower bounds for algorithms that compute a $(1+\varepsilon)$-approximation of the maximum matching size in graph streams. We first introduce some notation which will be used throughout this section.

Notation. Fix any (r,t)-RS graph $G^{\mathsf{RS}}(V,E)$ (for any parameters r,t) with induced matchings $M_1^{\mathsf{RS}}, \ldots, M_t^{\mathsf{RS}}$. For each matching M_i^{RS}, we assume an arbitrary ordering of the edges in M_i^{RS}, denoted by $e_{i,1}, \ldots, e_{i,r}$, and further denote $e_{i,j} := (u_{i,j}, v_{i,j})$ for all $j \in [r]$. Let $L(M_i^{\mathsf{RS}}) := \{u_{i,1}, \ldots, u_{i,r}\}$ and $R(M_i^{\mathsf{RS}}) := \{v_{i,1}, \ldots, v_{i,r}\}$. We emphasize that we do not require $G^{\mathsf{RS}}(V,E)$ to be necessarily a *bipartite* graph; each bipartition $L(M_i^{\mathsf{RS}})$ and $R(M_i^{\mathsf{RS}})$ (for $i \in [t]$) is defined locally for the matching itself and hence a vertex v is allowed to belong to, say, $L(M_i^{\mathsf{RS}})$ and $R(M_j^{\mathsf{RS}})$ for $i \neq j$, simultaneously.

Furthermore, for each matching M_i^{RS} and any boolean vector $x \in \{0,1\}^r$, we define the matching $M_i^{\mathsf{RS}}|_x$ as the subset of (the edges) of M_i^{RS} obtained by retaining the edge $e_{i,j} \in M_i^{\mathsf{RS}}$ (for any $j \in [r]$) iff $x(j) = 1$. In addition, for the vertex set $R(M_i^{\mathsf{RS}})$ and any perfect p-hypermatching[7] $\mathcal{M}$ on $[r]$, we define the *p-clique family* of $\mathcal{M}$ on $R(M_i^{\mathsf{RS}})$ to be a set of $|\mathcal{M}|$ cliques where the vertices $C_{\mathbf{e}}$ of each clique is defined by a distinct hyperedge $\mathbf{e} \in \mathcal{M}$: $C_{\mathbf{e}} := \{v_{i,k} \mid k \in \mathbf{e}\}$.

6.1 Insertion-Only Streams

We define $\mathsf{Matching}_{n,\varepsilon}$ as the *two-player one-way communication* problem of estimating the matching size to within a factor of $(1+\varepsilon)$, when Alice and Bob are each given a subset of the edges of an n-vertex input graph $G(V,E)$. In this section, we prove the following lower bound on the information complexity of $\mathsf{Matching}_{n,\varepsilon}$.

Theorem 9 (Lower bound of $\mathsf{Matching}_{n,\varepsilon}$). *For any sufficiently large n and sufficiently small $\varepsilon < \frac{1}{2}$, there exists a distribution $\mathcal{D}_{\mathsf{M}}$ for $\mathsf{Matching}_{n,\varepsilon}$ such that for any constant $\delta < \frac{1}{2}$:*

$$\mathsf{IC}_{1\text{-way}, \mathcal{D}_{\mathsf{M}}}^{\delta}(\mathsf{Matching}_{n,\varepsilon}) = \mathsf{RS}(n) \cdot n^{1 - O(\varepsilon)}$$

The lower bound of theorem 9, together with Proposition 2.7, implies the same lower bound on the one-way communication complexity of $\mathsf{Matching}_{n,\varepsilon}$. Since one-way communication complexity is a lower bound on the space complexity of any single-pass streaming algorithm in insertion-only streams, this immediately proves Part (1) of Theorem 3.

In the following, we focus on proving Theorem 9. Suppose the maximum value for $\mathsf{RS}(n)$ is achieved by an (r,t)-RS graph with the parameter $r = \mathsf{c}_{\mathsf{rs}} \cdot n$. We propose the following (hard) input distribution $\mathcal{D}_{\mathsf{M}}$ for $\mathsf{Matching}_{n,\varepsilon}$.

---

[7]Throughout this section, we use p instead of the usual parameter t for hypermatchings in order to avoid confusion with the parameter t in RS graphs

---

The hard distribution $\mathcal{D}_M$ for $\mathsf{Matching}_{n,\varepsilon}$:

    Parameters: $N := \frac{n}{2-2c_{rs}}$, $r := c_{rs} \cdot N$, $t := \mathsf{RS}(N)$, and $p := \left\lfloor \frac{c_{rs}}{2\varepsilon} \right\rfloor$.

- The input to the players is a graph $G(V, E_A \cup E_B)$ where E_A is given to Alice and E_B is given to Bob.

- **Alice:**

  1. Let V_1 $(\subset V)$ and $V_2 := V \setminus V_1$ be, respectively, a set of N and $n - N$ vertices.
  2. Let H be any fixed (r,t)-RS graph with $V(H) = V_1$.
  3. Draw r-dimensional binary vectors $x^{(1)}, \ldots, x^{(t)}$ *independently* following the distribution $\mathcal{D}_{\mathsf{BHH}}$ for $\mathsf{BHH}^0_{r,p}$.
  4. The input to Alice is the edge-set $E_A := M_1 \cup \ldots \cup M_t$, where $M_j := M_j^{\mathsf{RS}}|_{x^{(j)}}$.

- **Bob:**

  1. Pick $j^\star \in [t]$ uniformly at random.
  2. For the vector $x^{(j^\star)}$, draw a perfect p-hypermatching $\mathcal{M}$ following the distribution $\mathcal{D}_{\mathsf{BHH}}$ conditioned on $x^{(j^\star)}$; consequently, $(x^{(j^\star)}, \mathcal{M})$ is a $\mathsf{BHH}^0_{r,p}$ instance drawn from the distribution $\mathcal{D}_{\mathsf{BHH}}$.
  3. Let $E_{B,1}$ be an arbitrary perfect matching between $V_1 \setminus V(M_{j^\star}^{\mathsf{RS}})$ and V_2.
  4. Let $E_{B,2}$ be the edges of the p-clique family of $\mathcal{M}$ on $R(M_{j^\star}^{\mathsf{RS}})$.
  5. The input to Bob is the edge-set $E_B := E_{B,1} \cup E_{B,2}$.

We say that the instance $(x^{(j^\star)}, \mathcal{M})$ of $\mathsf{BHH}^0_{r,p}$ in the distribution (denoted by I_{BHH}) is *embedded* inside $\mathcal{D}_{\mathsf{MM}}$. The following claim established the connection between I_{BHH} and maximum matching size in G.

Claim 6.1. *Let:*

$$\mathsf{opt}_{\mathsf{Yes}} := \min_G \Big(\mathsf{opt}(G) \mid G \text{ is chosen from } \mathcal{D}_M \text{ conditioned } on\ I_{\mathsf{BHH}} \text{ being a } \mathsf{Yes} \text{ instance} \Big)$$

$$\mathsf{opt}_{\mathsf{No}} := \max_G \Big(\mathsf{opt}(G) \mid G \text{ is chosen from } \mathcal{D}_M \text{ conditioned } on\ I_{\mathsf{BHH}} \text{ being a } \mathsf{No} \text{ instance} \Big)$$

then, $(1 - \varepsilon) \cdot \mathsf{opt}_{\mathsf{Yes}} > \mathsf{opt}_{\mathsf{No}}$.

Proof. Let $M^\star$ be a maximum matching in G. Since all vertices in V_2 have degree 1, without loss of generality, we can assume $M^\star$ contains the matching $E_{B,1}$ between $V_1 \setminus V(M_{j^\star}^{\mathsf{RS}})$ and V_2. Consequently the size of $M^\star$ only depends on how many vertices in $V(M_{j^\star}^{\mathsf{RS}})$ can be matched with each other.

Consider the subgraph $H := G[L(M_{j^\star}^{\mathsf{RS}}) \cup R(M_{j^\star}^{\mathsf{RS}})]$ of G; by Claim 2.8, if I_{BHH} is a Yes instance, then $\mathsf{opt}(H) = \frac{3r}{4}$. Hence, in this case,

$$\mathsf{opt}(G) = |V_2| + \mathsf{opt}(H) = N - 2r + \frac{3r}{4} = N - \frac{5c_{rs}N}{4} = \frac{4 - 5c_{rs}}{4} \cdot N$$

If I_{BHH} is a No instance, then $\mathsf{opt}(H) = \frac{3r}{4} - \frac{r}{2p}$. Hence, in this case,

$$\mathsf{opt}(G) = |V_2| + \mathsf{opt}(H) \leq N - 2r + \frac{3r}{4} - \frac{r}{2p} = N - \frac{5c_{rs}N}{4} - \frac{c_{rs}}{2p}N = \frac{4 - 5c_{rs}}{4} \cdot N - \frac{c_{rs}}{2p}N$$

25

The bound on $\mathsf{opt}_{\mathsf{Yes}}$ and $\mathsf{opt}_{\mathsf{No}}$ now follows from the fact that $p \leq \frac{\mathsf{c}_{\mathsf{rs}}}{2\varepsilon}$ and therefore $\frac{\mathsf{c}_{\mathsf{rs}}}{2p}N \geq \varepsilon N > \varepsilon \cdot \left(\frac{4-5\mathsf{c}_{\mathsf{rs}}}{4} \cdot N\right)$. ∎

Fix any δ-error protocol Π_{Matching} for $\mathsf{Matching}_{n,\varepsilon}$ on $\mathcal{D}_{\mathsf{M}}$; Claim 6.1 implies that Π_{Matching} is also a δ-error protocol for solving the embedded instance I_{BHH}: simply return Yes whenever the estimate is larger than $\mathsf{opt}_{\mathsf{No}}$ and return No otherwise. We now use this fact to design a protocol Π_{BHH} for solving $\mathsf{BHH}^0_{r,p}$ on $\mathcal{D}_{\mathsf{BHH}}$, and prove that the information cost of Π_{Matching} is t times the information cost of $\mathsf{BHH}^0_{r,p}$.

The protocol Π_{BHH} for reducing $\mathsf{BHH}^0_{r,p}$ to $\mathsf{Matching}_{n,\varepsilon}$:

1. Let $(x, \mathcal{M})$ be the input $\mathsf{BHH}^0_{r,p}$ instance (x is given to Alice and $\mathcal{M}$ is given to Bob).

2. Using *public randomness*, Alice and Bob sample an index $j^\star \in [t]$ uniformly at random.

3. Let $x^{(1)}, \ldots, x^{(t)}$ be t vectors in $\{0,1\}^r$ whereby $x^{(j^\star)} = x$ and for any $j \neq j^\star$, $x^{(j)}$ is sampled by Alice using *private randomness* as in the distribution $\mathcal{D}_{\mathsf{M}}$. Alice creates the edges E_A following the distribution $\mathcal{D}_{\mathsf{M}}$ using these vectors.

4. Given the p-hypermatching $\mathcal{M}$ as input, Bob creates $E_{B,1}$ as an arbitrary perfect matching between $V_1 \setminus V(M_{j^\star}^{\mathsf{RS}})$ and V_2. He also creates $E_{B,2}$ as the edges of the p-clique family of $\mathcal{M}$ on $R(M_{j^\star}^{\mathsf{RS}})$ (V_1, V_2, and $M_{j^\star}^{\mathsf{RS}}$ are defined exactly as in $\mathcal{D}_{\mathsf{M}}$).

5. The players then run Π_{Matching} on the graph $G(V, E_A \cup E_B)$ and Bob outputs Yes if the output is larger than $\mathsf{opt}_{\mathsf{No}}$ and No otherwise.

The correctness of the protocol follows immediately from Claim 6.1. We now bound the information cost of this new protocol.

Lemma 6.2. $\mathsf{ICost}_{\mathcal{D}_{\mathsf{BHH}}}(\Pi_{\mathsf{BHH}}) \leq \frac{1}{t} \cdot \mathsf{ICost}_{\mathcal{D}_{\mathsf{M}}}(\Pi_{\mathsf{Matching}})$.

Proof. We have,

$$\mathsf{ICost}_{\mathcal{D}_{\mathsf{BHH}}}(\Pi_{\mathsf{BHH}}) = I_{\mathcal{D}_{\mathsf{BHH}}}(\boldsymbol{X}; \boldsymbol{\Pi}_{\mathsf{BHH}}, \boldsymbol{R}) = I_{\mathcal{D}_{\mathsf{BHH}}}(\boldsymbol{X}; \boldsymbol{\Pi}_{\mathsf{BHH}}^R \mid \boldsymbol{R})$$

(by chain rule of mutual information (Fact 2.2-(3)) and since $I(\boldsymbol{X}; \boldsymbol{R}) = 0$ as $\boldsymbol{X} \perp \boldsymbol{R}$)

$$= I_{\mathcal{D}_{\mathsf{BHH}}}(\boldsymbol{X}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{J})$$

($\boldsymbol{R} = \boldsymbol{J}$ and the message of Π_{BHH} is the same as Π_{Matching} after fixing the index $j^\star$)

$$= \underset{j \in [t]}{\mathbb{E}}\left[I_{\mathcal{D}_{\mathsf{BHH}}}(\boldsymbol{X}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{J} = j)\right] = \frac{1}{t} \cdot \sum_{j=1}^{t} I_{\mathcal{D}_{\mathsf{BHH}}}(\boldsymbol{X}_j; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{J} = j)$$

$$= \frac{1}{t} \cdot \sum_{j=1}^{t} I_{\mathcal{D}_{\mathsf{M}}}(\boldsymbol{X}_j; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{J} = j)$$

(joint distribution of $\boldsymbol{\Pi}_{\mathsf{Matching}}$ and $\boldsymbol{X}_j$, conditioned on $\boldsymbol{J} = j$, is the same under $\mathcal{D}_{\mathsf{M}}$ and $\mathcal{D}_{\mathsf{BHH}}$)

$$= \frac{1}{t} \cdot \sum_{j=1}^{t} I_{\mathcal{D}_{\mathsf{M}}}(\boldsymbol{X}_j; \boldsymbol{\Pi}_{\mathsf{Matching}})$$

where the last equality is true since the random variables $\boldsymbol{X}_j$ and $\boldsymbol{\Pi}_{\mathsf{Matching}}$ are both independent of the event $\boldsymbol{J} = i$ (by definition of the distribution $\mathcal{D}_{\mathsf{M}}$). Finally,

$$
\begin{aligned}
\mathsf{ICost}_{\mathcal{D}_{\mathsf{BHH}}}(\Pi_{\mathsf{BHH}}) &= \frac{1}{t} \cdot \sum_{j=1}^{t} I_{\mathcal{D}_{\mathsf{M}}}(\boldsymbol{X}_j ; \boldsymbol{\Pi}_{\mathsf{Matching}}) \\
&\leq \frac{1}{t} \cdot I_{\mathcal{D}_{\mathsf{M}}}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_t ; \boldsymbol{\Pi}_{\mathsf{Matching}})
\end{aligned}
$$

(by conditional super-additivity of mutual information (Fact 2.2-(5)) since $\boldsymbol{X}_j \perp \boldsymbol{X}^{<j}$)

$$
= \frac{1}{t} \cdot I_{\mathcal{D}_{\mathsf{M}}}(\boldsymbol{E}_A ; \boldsymbol{\Pi}_{\mathsf{Matching}}) = \frac{1}{t} \cdot \mathsf{ICost}_{\mathcal{D}_{\mathsf{M}}}(\Pi_{\mathsf{Matching}})
$$

where the second last inequality is because the set of edges in E_A can be determined uniquely by the vectors $x^{(1)}, \ldots, x^{(t)}$ and vice versa. ∎

Theorem 9 now follows from Lemma 6.2, lower bound of $\Omega(r^{1-1/p}) = n^{1-O(\varepsilon)}$ for $\mathsf{BHH}_{r,p}^0$ in Corollary 6, and the choice of $t = \mathsf{RS}(n)$.

6.2 Dynamic Streams

We define $\mathsf{Matching}_{n,k,\varepsilon}$ as the *k-player simultaneous communication* problem of estimating the maximum matching size to within a factor of $(1+\varepsilon)$, when edges of an *n*-vertex input graph $G(V, E)$ are partitioned across the *k*-players and the referee (see Remark 2.5). In this section, we prove the following lower bound on the information complexity of $\mathsf{Matching}_{n,k,\varepsilon}$ in the SMP communication model.

Theorem 10 (Lower bound for $\mathsf{Matching}_{n,k,\varepsilon}$)**.** *For any sufficiently large n and sufficiently small $\varepsilon < \frac{1}{2}$, there exists some $k = n^{o(1)}$ and a distribution $\mathcal{D}_{\mathsf{M}}$ for $\mathsf{Matching}_{n,k,\varepsilon}$ such that for any constant $\delta < \frac{1}{2}$:*

$$
\mathsf{IC}_{\mathsf{SMP}, \mathcal{D}_{\mathsf{M}}}^{\delta}(\mathsf{Matching}_{n,k,\varepsilon}) = n^{2-O(\varepsilon)}
$$

Theorem 10, combined with Proposition 2.7, immediately proves the same lower bound on the SMP communication complexity of $\mathsf{Matching}_{n,k,\varepsilon}$. Since SMP communication complexity of a *k*-player problem is at most *k* times the space complexity of any single-pass streaming algorithm in dynamic streams [4, 42] (and $k = n^{o(1)}$), this immediately proves Part (2) of Theorem 3.

In the following, we focus on proving Theorem 10. We propose the following (hard) distribution $\mathcal{D}_{\mathsf{M}}$ for $\mathsf{Matching}_{n,k,\varepsilon}$. Intuitively, the distribution $\mathcal{D}_{\mathsf{M}}$ can be seen as imposing the hard distribution for matching size estimation in [14] on each induced matching in the hard instance of [9] for finding approximate matchings.

The hard distribution $\mathcal{D}_M$ for Matching$_{n,k,\varepsilon}$:

    Parameters: $r = N^{1-o(1)}$, $t = \frac{\binom{N}{2} - o(N^2)}{r}$, $k = \frac{N}{\varepsilon \cdot r}$, $n = N + k \cdot r$, and $p := \left\lfloor \frac{1}{8\varepsilon} \right\rfloor$.

1. Fix an (r, t)-RS graph G^{RS} on N vertices.

2. Pick $j^\star \in [t]$ uniformly at random and draw a $\mathsf{BHH}^0_{r,p}$ instance $(x^{(j^\star)}, \mathcal{M})$ from the distribution $\mathcal{D}_{\mathsf{BHH}}$.

3. For each player $P^{(i)}$ independently,

   (a) Denote by G_i the input graph of $P^{(i)}$, initialized to be a copy of G^{RS} with vertices $V_i = [N]$.

   (b) Let V_i^* be the set of vertices matched in the $j^\star$-th induced matching of G_i. Change the induced matching $M_{j^\star}^{\mathsf{RS}}$ of G_i to $M_{j^\star} := M_{j^\star}^{\mathsf{RS}}|_{x^{(j^\star)}}$.

   (c) For any $j \in [t] \setminus \{j^\star\}$, draw a vector $x^{(i,j)} \in \{0,1\}^r$ following the distribution $\mathcal{D}_{\mathsf{BHH}}$ for $\mathsf{BHH}^0_{r,p}$, and change the induced matching M_j^{RS} of G_i to $M_j := M_j^{\mathsf{RS}}|_{x^{(j)}}$.

   (d) Create the p-clique family of $\mathcal{M}$ on the vertices $R(M_{j^\star}^{\mathsf{RS}})$, and give the edges of the p-clique family to the referee.

4. Pick a random permutation σ of $[n]$. For every player $P^{(i)}$, for each vertex v in $V_i \setminus V_i^*$ with label j ($\in [N]$), *relabel* v to $\sigma(j)$. Enumerate the vertices in V_i^* (from the one with the smallest label to the largest), and relabel the j-th vertex to $\sigma(N + (i-1) \cdot 2r + j)$. In the final graph, the vertices with the same label correspond to the same vertex.

    The vertices whose labels belong to $\sigma([N])$ are referred to as *shared* vertices since they belong to the input graph of *every* player, and the vertices V_i^* are referred to as the *private* vertices of the player $P^{(i)}$ since they only appear in the input graph of $P^{(i)}$ (in the final graph, i.e., after relabeling). We point out that, in general, the final graph constructed by this distribution is a multi-graph with n vertices and $O(kN^2) = O(n^2)$ edges (counting the multiplicities); the multiplicity of each edge is also at most k. Finally, the existence of an (r, t)-RS graph G^{RS} with the parameters used in this distribution is guaranteed by a result of [7] (see Section 2.1).

    Similar to the lower bound in Section 6.1, let I_{BHH} be the *embedded* $\mathsf{BHH}^0_{r,p}$ instance $(x^{(i)}, \mathcal{M})$. The following claim is analogous to Claim 6.1 in Section 6.1.

Claim 6.3. *Let:*

$$\mathrm{opt}_{\mathsf{Yes}} := \min_G \left(\mathrm{opt}(G) \mid G \text{ is chosen from } \mathcal{D}_M \text{ conditioned on } I_{\mathsf{BHH}} \text{ being a } \mathsf{Yes} \text{ instance} \right)$$

$$\mathrm{opt}_{\mathsf{No}} := \max_G \left(\mathrm{opt}(G) \mid G \text{ is chosen from } \mathcal{D}_M \text{ conditioned on } I_{\mathsf{BHH}} \text{ being a } \mathsf{No} \text{ instance} \right)$$

then, $(1 - \varepsilon) \cdot \mathrm{opt}_{\mathsf{Yes}} > \mathrm{opt}_{\mathsf{No}}$.

Proof. We partition the edges of G into $k + 1$ groups: for any $i \in [k]$, group i contains the edges that are between the private vertices V_i^* of player $P^{(i)}$, and group $k + 1$ contains the edges incident on at least one shared vertex. Let $H_i := G[V_i^*]$, i.e., the subgraph of G induced on the vertices V_i^*.

    If I_{BHH} is a Yes instance, then for any $i \in [k]$, $\mathrm{opt}(H_i) = \frac{3r}{4}$ by Claim 2.8. Since V_i^* are private vertices, one can choose *any* matching from each H_i, and the collection of the chosen edges form a

matching of G. Therefore,

$$\text{opt}(G) > \sum_{i=1}^{k} \text{opt}(H_i) = \frac{3kr}{4} = \frac{3N}{\varepsilon}$$

Note that, $\text{opt}(G)$ is *strictly* larger than $\frac{3N}{\varepsilon}$ since one can add (any) edge between the public vertices to the matching.

If I_{BHH} is a No instance, then $\text{opt}(H_i) = \frac{3r}{4} - \frac{r}{2p}$. Since the maximum matching size in G is at most the summation of the maximum matching size in each group, we have

$$\text{opt}(G) \leq \sum_{i=1}^{k} \text{opt}(H_i) + N \leq \frac{3kr}{4} - \frac{kr}{2p} + N \leq \frac{3N}{\varepsilon} - 3N$$

and the gap between $\text{opt}_{\mathsf{Yes}}$ and opt_{No} follows. $\blacksquare$

Fix any δ-error protocol Π_{Matching} for $\mathsf{Matching}_{n,k,\varepsilon}$ on $\mathcal{D}_{\mathsf{M}}$; Claim 6.3 implies that Π_{Matching} is also a δ-error protocol for solving the embedded instance I_{BHH}: simply return Yes whenever the estimate is larger than opt_{No} and return No otherwise. In the following, we use this fact to design a protocol Π_{BHH} for solving $\mathsf{BHH}_{r,p}^0$ on $\mathcal{D}_{\mathsf{BHH}}$, and then prove that the information cost of Π_{Matching} is t times the information cost of $\mathsf{BHH}_{r,p}^0$.

In the protocol Π_{BHH}, Alice will simulate all k players of $\mathsf{Matching}_{n,k,\varepsilon}$ and Bob will simulate the referee; Alice and Bob will use public coins to draw the special index $j^\star$ and the permutation σ. Together with the input from $\mathcal{D}_{\mathsf{BHH}}$, Alice and Bob will be able to create a $\mathsf{Matching}_{n,k,\varepsilon}$ instance. The reduction is formally defined as follows (the parameters used in the reduction are exactly the same as that in the definition of $\mathcal{D}_{\mathsf{M}}$).

The protocol Π_{BHH} for reducing $\mathsf{BHH}_{r,p}^0$ to $\mathsf{Matching}_{n,k,\varepsilon}$:

1. Let $(x, \mathcal{M})$ be the input $\mathsf{BHH}_{r,p}^0$ instance (x is given to Alice and $\mathcal{M}$ is given to Bob).

2. Using *public randomness*, Alice and Bob sample an index $j^\star \in [t]$, and a permutation σ on $[n]$ uniformly at random.

3. For any player $P^{(i)}$, let $x^{(i,1)}, \ldots, x^{(i,t)}$ be t vectors in $\{0,1\}^r$ whereby $x^{(i,j^\star)} = x$ (i.e., Alice's input in the $\mathsf{BHH}_{r,p}^0$ problem) and for any $j \neq j^\star$, $x^{(i,j)}$ is sampled by Alice using *private randomness* as in the distribution $\mathcal{D}_{\mathsf{M}}$. Alice then uses these vector together with permutation σ to create the input graph G_i for each player $P^{(i)}$ for $i \in [k]$ following how G_i is created in the distribution $\mathcal{D}_{\mathsf{M}}$ for $\mathsf{Matching}_{n,k,\varepsilon}$.

4. The vertices $R(M_{j^\star}^{\mathsf{RS}})$ of each player will be mapped (by σ) to a different set of vertices in G. Since Bob knows σ and $j^\star$, and the (input) p-hypermatching $\mathcal{M}$, Bob can create the p-clique families of each player (following the input of the referee in $\mathcal{D}_{\mathsf{M}}$).

5. The players then run Π_{Matching} on the $\mathsf{Matching}_{n,k,\varepsilon}$ that they created, and Bob outputs Yes if the matching size estimate is larger than opt_{No} and No otherwise.

It is straightforward to verify that the distribution of the $\mathsf{Matching}_{n,k,\varepsilon}$ instance created by the protocol Π_{BHH} is identical to the distribution $\mathcal{D}_{\mathsf{M}}$. The correctness of the protocol now follows immediately from Claim 6.1. In the remainder of this section, we bound the information cost of this protocol.

Lemma 6.4. $\mathsf{ICost}_{\mathcal{D}_{\mathsf{BHH}}}(\Pi_{\mathsf{BHH}}) \leq \frac{1}{t} \cdot \mathsf{ICost}_{\mathcal{D}_{\mathsf{M}}}(\Pi_{\mathsf{Matching}})$.

Proof. We have,

$$\mathsf{ICost}_{\mathcal{D}_{\mathsf{BHH}}}(\Pi_{\mathsf{BHH}}) = I_{\mathcal{D}_{\mathsf{BHH}}}(\boldsymbol{X}; \boldsymbol{\Pi}_{\mathsf{BHH}}, \boldsymbol{R}) = I_{\mathcal{D}_{\mathsf{BHH}}}(\boldsymbol{X}; \boldsymbol{\Pi}_{\mathsf{BHH}}^R \mid \boldsymbol{R})$$

$$\text{(by chain rule of mutual information (Fact 2.2-(3)) and since } I(\boldsymbol{X}; \boldsymbol{R}) = 0 \text{ as } \boldsymbol{X} \perp \boldsymbol{R})$$

$$= I_{\mathcal{D}_{\mathsf{BHH}}}(\boldsymbol{X}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{\sigma}, \boldsymbol{J}, \boldsymbol{R}_{\mathsf{M}}) = \mathop{\mathbb{E}}_{j \in [t]} \left[I_{\mathcal{D}_{\mathsf{BHH}}}(\boldsymbol{X}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{\sigma}, \boldsymbol{R}_{\mathsf{M}}, \boldsymbol{J} = j) \right]$$

where the second last equality is because $\boldsymbol{R} = (\boldsymbol{\sigma}, \boldsymbol{J}, \boldsymbol{R}_{\mathsf{M}})$ ($\boldsymbol{R}_{\mathsf{M}}$ is the public randomness of Π_{Matching}), and the message of Π_{BHH} is the same as Π_{Matching} after fixing the index $j^\star$. For any $j \in [t]$, define $\boldsymbol{Y}_j := (\boldsymbol{X}_{1,j}, \boldsymbol{X}_{2,j}, \ldots, \boldsymbol{X}_{k,j})$ where $\boldsymbol{X}_{i,j}$ is a random variable for the vector $x^{(i,j)}$. With this notation, conditioned on $\boldsymbol{J} = j$, we have $\boldsymbol{X} = \boldsymbol{Y}_j$ and also the joint distribution of $(\boldsymbol{\Pi}_{\mathsf{Matching}}, \boldsymbol{\sigma}, \boldsymbol{Y}_j, \boldsymbol{R}_{\mathsf{M}})$ conditioned on $\boldsymbol{J} = j$, is the same under both $\mathcal{D}_{\mathsf{M}}$ and $\mathcal{D}_{\mathsf{BHH}}$. Hence,

$$\mathsf{ICost}_{\mathcal{D}_{\mathsf{BHH}}}(\Pi_{\mathsf{BHH}}) = \frac{1}{t} \sum_{j=1}^{t} I_{\mathcal{D}_{\mathsf{M}}}(\boldsymbol{Y}_j; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{\sigma}, \boldsymbol{R}_{\mathsf{M}}, \boldsymbol{J} = j)$$

$$= \frac{1}{t} \sum_{j=1}^{t} I_{\mathcal{D}_{\mathsf{M}}}(\boldsymbol{Y}_j; \boldsymbol{\Pi}_{\mathsf{Matching}}^{(1)}, \ldots, \boldsymbol{\Pi}_{\mathsf{Matching}}^{(k)} \mid \boldsymbol{\sigma}, \boldsymbol{R}_{\mathsf{M}}, \boldsymbol{J} = j)$$

$$\leq \frac{1}{t} \sum_{j=1}^{t} \sum_{i=1}^{k} I_{\mathcal{D}_{\mathsf{M}}}(\boldsymbol{Y}_j; \boldsymbol{\Pi}_{\mathsf{Matching}}^{(i)} \mid \boldsymbol{\sigma}, \boldsymbol{R}_{\mathsf{M}}, \boldsymbol{J} = j)$$

$$= \frac{1}{t} \sum_{j=1}^{t} \sum_{i=1}^{k} I_{\mathcal{D}_{\mathsf{M}}}(\boldsymbol{Y}_j; \boldsymbol{\Pi}_{\mathsf{Matching}}^{(i)} \mid \boldsymbol{\sigma}, \boldsymbol{R}_{\mathsf{M}})$$

where the last inequality is by conditional sub-additivity of mutual information (Fact 2.2-(4)) since $\boldsymbol{\Pi}_{\mathsf{Matching}}^{(i)} \perp \boldsymbol{\Pi}_{\mathsf{Matching}}^{<i} \mid \boldsymbol{\sigma}, \boldsymbol{Y}_j, \boldsymbol{R}_{\mathsf{M}}, \boldsymbol{J} = j$; this is because conditioned on the given random variables and $\boldsymbol{J} = j$, the message of each player $P^{(i)}$ (i.e., $\boldsymbol{\Pi}_{\mathsf{Matching}}^{(i)}$) is only a function of $x^{(i,j)}$ for $j \neq j^\star$ and since these vectors are chosen independently, the messages would be independent.

Moreover, the reason we can drop the conditioning on the event $\boldsymbol{J} = j$ (in the last equality above) is as follows: $\boldsymbol{\Pi}_{\mathsf{Matching}}^{(i)}$ is a function of $(\boldsymbol{X}_i, \boldsymbol{\sigma}_i)$ where $\boldsymbol{X}_i := (\boldsymbol{X}_{i,1}, \ldots, \boldsymbol{X}_{i,t})$ is a random variable for the vector $x^{(i)}$. $\boldsymbol{X}_i$ defines the graph G_i without the labels, i.e., over the set of vertices $V_i := [N]$ and $\boldsymbol{\sigma}_i$ is the random variable denoting how the vertices of the player $P^{(i)}$ are mapped to G, i.e., specifies the labels of vertices. Therefore, $(\boldsymbol{X}_i, \boldsymbol{\sigma}_i)$ is independent of $\boldsymbol{J} = j$ (given the input graph G_i, each matching has the same probability of being the chosen matching for $j^\star$); hence it is easy to see that all four random variables in above term are independent of the event $\boldsymbol{J} = j$.

Finally, since $\boldsymbol{Y}_j$ and $\boldsymbol{Y}^{-j}$ are independent of each other even conditioned on $\boldsymbol{\sigma}, \boldsymbol{R}_{\mathsf{M}}$, by conditional super-additivity of mutual information (Fact 2.2-(5)),

$$\mathsf{ICost}_{\mathcal{D}_{\mathsf{BHH}}}(\Pi_{\mathsf{BHH}}) \leq \frac{1}{t} \sum_{i=1}^{k} I_{\mathcal{D}_{\mathsf{M}}}(\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_t; \boldsymbol{\Pi}_{\mathsf{Matching}}^{(i)} \mid \boldsymbol{\sigma}, \boldsymbol{R}_{\mathsf{M}},)$$

$$= \frac{1}{t} \sum_{i=1}^{k} I_{\mathcal{D}_{\mathsf{M}}}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k; \boldsymbol{\Pi}_{\mathsf{Matching}}^{(i)} \mid \boldsymbol{\sigma}, \boldsymbol{R}_{\mathsf{M}},)$$

$$(\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_t \text{ uniquely define } \boldsymbol{X}_1, \ldots, \boldsymbol{X}_k \text{ and vice versa})$$

$$\leq \frac{1}{t} \sum_{i=1}^{k} I_{\mathcal{D}_{\mathsf{M}}}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k, \boldsymbol{\sigma}; \boldsymbol{\Pi}^{(i)}_{\mathsf{Matching}}, \boldsymbol{R}_{\mathsf{M}})$$

(by chain rule of mutual information (Fact 2.2-(3)))

$$= \frac{1}{t} \cdot \mathsf{ICost}_{\mathcal{D}_{\mathsf{M}}}(\Pi_{\mathsf{Matching}})$$

where the last equality is because $(\boldsymbol{X}_i, \boldsymbol{\sigma}_i)$ uniquely defines the input to player $P^{(i)}$. ∎

Theorem 10 now follows from Lemma 6.4, lower bound of $\Omega(r^{1-1/p}) = n^{1-O(\varepsilon)}$ for $\mathsf{BHH}^0_{r,p}$ in Corollary 6, and the choice of $t = \Theta(n)$ in the distribution.

7 Space Upper Bounds for α-Approximating Matching Size

In this section, we present our algorithms for achieving an α-approximation of the maximum matching size respectively in $\widetilde{O}(n/\alpha^2)$ space for insertion-only streams and in $\widetilde{O}(n^2/\alpha^4)$ space for dynamic streams, proving Theorem 1.

The main ingredient of both our algorithms is a simple *vertex sampling* procedure. In the rest of this section, we first define this sampling procedure and establish its connection to matching size estimation (Section 7.1). We then build on this connection to provide a *meta-algorithm* for matching size estimation (Section 7.2). Finally, we show how to implement this meta-algorithm in $\widetilde{O}(n/\alpha^2)$ space in insertion-only streams and $\widetilde{O}(n^2/\alpha^4)$ space in dynamic streams, which proves Theorem 1.

7.1 Vertex Sampling Procedure

Consider the following simple vertex sampling procedure.

> $\mathsf{Sample}_p(G)$: sample each vertex $v \in V$ in $G(V, E)$ w.p. p, using a *four-wise independent* hash function, and return the induced subgraph over the set of sampled vertices.

Note that since $O(\log n)$ bits suffices to store a four-wise independent hash function (see, e.g., [50]), the set of sampled vertices in $\mathsf{Sample}_p(G)$ can be also be stored (implicitly) in $O(\log n)$ bits.

The following lemma establishes that as long as $\mathrm{opt}(G)$ is not too small, the maximum matching size in the graph that $\mathsf{Sample}_p(G)$ outputs (for $p := \frac{\log n}{\alpha}$) can be directly used to obtain an α-approximation of $\mathrm{opt}(G)$.

Lemma 7.1. *Let $G(V, E)$ be any graph, $\alpha \geq \log n$, and $p := \frac{\log n}{\alpha}$; for $G_{\mathsf{smp}} := \mathsf{Sample}_p(G)$,*

1. *if $\mathrm{opt}(G) = \Omega(\alpha)$, then $\mathrm{opt}(G_{\mathsf{smp}}) \leq \frac{3 \log n}{\alpha} \cdot \mathrm{opt}(G)$ w.p. $1 - o(1)$.*

2. *if $\mathrm{opt}(G) = \Omega(\alpha^2)$, then $\mathrm{opt}(G_{\mathsf{smp}}) \geq \frac{\log^2 n}{2\alpha^2} \cdot \mathrm{opt}(G)$ w.p. $1 - o(\frac{1}{\log n})$.*

Note that for $\mathrm{opt}(G) = \Omega(\alpha^2)$, Lemma 7.1 immediately implies that w.p. $1 - o(1)$,

$$\frac{3 \log n}{\alpha} \cdot \mathrm{opt}(G) \leq \mathrm{opt}(G_{\mathsf{smp}}) \leq \frac{\log^2 n}{2\alpha^2} \cdot \mathrm{opt}(G).$$

Proof of Lemma 7.1. Fix a maximum matching $M^\star$ in G and denote the set of vertices matched in $M^\star$ by $V(M^\star)$. Moreover, let $V_{\mathsf{smp}}(M^\star)$ be the set of vertices in $V(M^\star)$ that are sampled by $\mathsf{Sample}_p(G)$.

We first prove Part (1) of the lemma. Since $M^\star$ is a maximum matching in G, every edge in G must be incident on at least one vertex in $V(M^\star)$. Consequently, in the sampled graph G_{smp}, every edge is incident on at least one vertex in $V_{\mathsf{smp}}(M^\star)$, and hence, $\mathrm{opt}(G_{\mathsf{smp}}) \leq |V_{\mathsf{smp}}(M^\star)|$; therefore, we only need to upper bound $|V_{\mathsf{smp}}(M^\star)|$. Let X be a random variable denoting $|V_{\mathsf{smp}}(M^\star)|$.

Now, $\mathbb{E}[X] = p \cdot |V(M^\star)| = p \cdot 2\mathrm{opt}(G) = \frac{2\log n}{\alpha} \cdot \mathrm{opt}(G)$ by the choice of p. Since $\mathrm{opt}(G) = \Omega(\alpha)$ by our assumption in Part (1), $\mathbb{E}[X] = \Omega(\log n)$. Moreover, because $\mathsf{Sample}_p(G)$ samples vertices using a four-wise independent hash function, $\mathrm{Var}[X] \leq \mathbb{E}[X]$ and hence by Chebyshev inequality,

$$\Pr\left(X \geq \frac{3\log n}{\alpha} \cdot \mathrm{opt}(G)\right) = \Pr\left(X \geq \frac{3}{2} \cdot \mathbb{E}[X]\right) \leq \Pr\left(|X - \mathbb{E}[X]| \geq \frac{\mathbb{E}[X]}{2}\right)$$
$$\leq \frac{\mathrm{Var}[X]}{(\mathbb{E}[X]/2)^2} \leq \frac{4}{\mathbb{E}[X]} = \frac{1}{\Omega(\log n)} = o(1)$$

This implies w.p. $1 - o(1)$, $|V_{\mathsf{smp}}(M^\star)| \leq \frac{3\log n}{\alpha} \cdot \mathrm{opt}(G)$, and since $\mathrm{opt}(G_{\mathsf{smp}}) \leq |V_{\mathsf{smp}}(M^\star)|$ we obtain the result in Part (1).

We now prove Part (2) of the lemma. Let $M^\star_{\mathsf{smp}}$ be the set of sampled edges from $M^\star$ that end up G_{smp}. Since $\mathrm{opt}(G_{\mathsf{smp}}) \geq |M^\star_{\mathsf{smp}}|$, it suffices to show that $|M^\star_{\mathsf{smp}}| \geq \frac{\log^2 n}{2\alpha^2} \cdot \mathrm{opt}(G)$. Let Y be a random variable denoting $|M^\star_{\mathsf{smp}}|$.

For each edge $e \in M^\star$, e appears in $M^\star_{\mathsf{smp}}$ iff both endpoints of e are sampled by $\mathsf{Sample}_p(G)$, which happens w.p. p^2 (due to four-wise independence in sampling vertices). Therefore, the expected number of edges in $M^\star_{\mathsf{smp}}$ is $\mathbb{E}[Y] = p^2 \cdot \mathrm{opt}(G) = \frac{\log^2 n}{\alpha^2} \cdot \mathrm{opt}(G)$. Since by assumption in Part (2), $\mathrm{opt}(G) = \Omega(\alpha^2)$, we have $\mathbb{E}[Y] = \Omega(\log^2 n)$. Moreover, since vertices are sampled in $\mathsf{Sample}_p(G)$ using a four-wise independent hash function, for any two edges in $M^\star$, the event that they appear in $M^\star_{\mathsf{smp}}$ is independent of each other; this implies $\mathrm{Var}[Y] \leq \mathbb{E}[Y]$, and hence by Chebyshev inequality,

$$\Pr\left(Y < \frac{\log^2 n}{2\alpha^2} \cdot \mathrm{opt}(G)\right) = \Pr\left(Y < \frac{\mathbb{E}[Y]}{2}\right) \leq \Pr\left(|Y - \mathbb{E}[Y]| \geq \frac{\mathbb{E}[Y]}{2}\right)$$
$$\leq \frac{\mathrm{Var}[Y]}{(\mathbb{E}[Y]/2)^2} \leq \frac{4}{\mathbb{E}[Y]} = \frac{1}{\Omega(\log^2 n)} = o(\frac{1}{\log n})$$

Therefore, w.p. $1 - o(\frac{1}{\log n})$, $|M^\star_{\mathsf{smp}}| \geq \frac{\log^2 n}{2\alpha^2} \cdot \mathrm{opt}(G)$, proving Part (2). ∎

7.2 The Meta Algorithm

In this section, we define our meta-algorithm for approximating the matching size in any graph G based on the vertex sampling procedure defined in the previous section. To continue, we need to define the notion of *matching size testers* that are used as subroutines in the meta-algorithm.

Definition 5 (γ-Matching Size Tester). *For any constant $0 < \gamma < 1$, a γ-matching size tester (denoted by Tester_γ) is an algorithm that given a graph G and a threshold k, outputs Yes if $\mathrm{opt}(G) \geq k$, outputs No if $\mathrm{opt}(G) \leq \gamma \cdot k$, and otherwise is allowed to output either Yes or No.*

Moreover, whenever $\mathsf{Tester}_\gamma(G, k)$ outputs No, it also outputs an estimate $\widetilde{\mathrm{opt}}$ such that $\gamma \cdot \mathrm{opt}(G) \leq \widetilde{\mathrm{opt}} \leq \mathrm{opt}(G)$.

Given any γ-matching size tester Tester_γ, consider the following algorithm (denoted by Algorithm 1) for achieving an $O(\alpha)$-approximation of maximum matching size.

1. For each value $\beta \in \{\log n, 2\log n, 2^2\log n, \ldots, \alpha\}$, let $G^\beta := \mathsf{Sample}_{\frac{\log n}{\beta}}(G)$. In parallel, run Tester_γ on each G^β with the parameter $\frac{\log^2 n}{2}$ (i.e., run $\mathsf{Tester}_\gamma(G^\beta, \frac{\log^2 n}{2})$).

2. In addition, for $\beta = \alpha$, also run $\mathsf{Tester}_\gamma(G^\alpha, \frac{n\log^2 n}{\alpha^2})$.

3. At the end of the stream, for each value β, we say β *passes* if $\mathsf{Tester}_\gamma(G^\beta, \frac{\log^2 n}{2})$ outputs Yes; otherwise, we say β *fails*.

   - If all β fail, output the estimate $\widetilde{\mathrm{opt}}_{\log n}$ returned by $\mathsf{Tester}_\gamma(G^{\log n}, \frac{\log^2 n}{2})$.
   - If all β pass, output $\max\left\{\alpha, \frac{\alpha}{\log^2 n} \cdot \widetilde{\mathrm{opt}}_\alpha\right\}$, where $\widetilde{\mathrm{opt}}_\alpha$ is defined as follows. if $\mathsf{Tester}_\gamma(G^\alpha, \frac{n\log^2 n}{\alpha^2})$ returns No, let $\widetilde{\mathrm{opt}}_\alpha$ be the estimate returned by $\mathsf{Tester}_\gamma(G^\alpha, \frac{n\log^2 n}{\alpha^2})$; otherwise, let $\widetilde{\mathrm{opt}}_\alpha := \frac{\gamma n\log^2 n}{\alpha^2}$.
   - Otherwise, output $\frac{\beta^*}{2}$ where β^* is the smallest β that fails.

We should remark right away that if $\mathrm{opt}(G) = \Omega(\alpha^2)$, running $\mathsf{Tester}_\gamma(G^\alpha, \frac{n\log^2 n}{\alpha^2})$ (step 2 in the algorithm) suffices to obtain an α-approximation (Lemma 7.1 essentially guarantees that $\mathrm{opt}(G^\alpha) \in [\frac{\mathrm{opt}(G)}{\alpha^2}, \frac{\mathrm{opt}(G)}{\alpha}]$). Therefore, running tester for $O(\log \alpha)$ different values (step 1 in the algorithm) is only for the case where $\mathrm{opt}(G) \le \alpha^2$.

Intuitively speaking, for the three cases that determine the output of the algorithm (step 3 in the algorithm):

- If all β fails, then all testers returns No, which means the maximum matching size in the sampled graphs are all small: this is for the case $\mathrm{opt}(G) = \widetilde{O}(1)$.

- If all β passes, then all testers return Yes, which means the maximum matching sizes are all large: this is for the case $\mathrm{opt}(G) > \alpha^2$.

- Finally, if some β pass and some β fail, then we are in the case $\mathrm{opt}(G) \in [\widetilde{O}(1), \alpha^2]$.

We now prove the correctness of Algorithm 1 through considering these three cases separately.

Lemma 7.2. *For any $\alpha \ge \log n$, Algorithm 1 outputs an $O(\alpha)$-approximation of $\mathrm{opt}(G)$ w.h.p.*

Proof. First notice that if all β fails, in particular, $\beta = \log n$ fails, and hence the estimate returned by $\mathsf{Tester}_\gamma(G^{\log n}, \log^2 n)$ (denoted by $\widetilde{\mathrm{opt}}_{\log n}$) is a γ-approximation of $\mathrm{opt}(G^{\log n})$. Furthermore, note that for $\beta = \log n$, the subsampling probability is $\frac{\log n}{\beta} = 1$, and hence, $G^{\log n} = G$. Therefore, $\widetilde{\mathrm{opt}}_{\log n}$ is a also γ-approximation of $\mathrm{opt}(G)$.

In the following, we analyze the other two cases: (i) all β pass (which would be the case where $\mathrm{opt}(G)$ is large) and (ii) some β pass and some β fail (which will be the case where $\mathrm{opt}(G)$ is small). The following two claims summarize the property of the β that passes and the property of the β that fails, which will be useful for the analysis.

Claim 7.3. *For any β where $\beta^2 \le \mathrm{opt}(G)$, β passes w.p. $1 - o(\frac{1}{\log n})$.*

Proof. By Lemma 7.1 Part (2), when $\beta^2 \leq \text{opt}(G)$, w.p. $1 - o(\frac{1}{\log n})$,

$$\text{opt}(G^\beta) \geq \frac{\log^2 n}{2\beta^2} \cdot \text{opt}(G) \geq \frac{\log^2 n}{2\beta^2} \cdot \beta^2 = \frac{\log^2 n}{2}.$$

Therefore, $\text{Tester}_\gamma(G^\beta, \frac{\log^2 n}{2})$ outputs Yes (and hence β passes). ■

Claim 7.4. *For the value β where $\frac{\beta}{2} \leq \text{opt}(G) \leq \beta$ (if one exists), β fails w.p. $1 - o(1)$.*

Proof. By Lemma 7.1 Part (1), when $\text{opt}(G) \geq \frac{\beta}{2}$, w.p. $1 - o(1)$,

$$\text{opt}(G^\beta) \leq \frac{3 \log n}{\beta} \cdot \text{opt}(G) \leq \frac{3 \log n}{\beta} \cdot \beta = 3 \log n < \frac{\gamma \log^2 n}{2} \quad \text{(for } n \text{ sufficiently large)}.$$

Therefore, $\text{Tester}_\gamma(G^\beta, \frac{\log^2 n}{2})$ outputs No (and hence β fails). ■

With Claim 7.3 and Claim 7.4, the correctness of case (ii) follows immediately.

Lemma 7.5. *If some β pass and some β fail, then $\frac{\beta^*}{2}$ is an $O(\alpha)$-approximation of $\text{opt}(G)$.*

Proof. We first show that $\frac{\beta^*}{2} \geq \frac{\text{opt}(G)}{2\alpha}$ and then show that $\frac{\beta^*}{2} \leq \text{opt}(G)$. To see $\frac{\beta^*}{2} \geq \frac{\text{opt}(G)}{2\alpha}$, by Claim 7.3, for each β where $\beta^2 \leq \text{opt}(G)$, w.p. $1 - o(\frac{1}{\log n})$, β passes. Therefore, we can apply a union bound over all $O(\log \alpha)(= O(\log n))$ choices of β, and claim that w.p. $1 - o(1)$, for all β where $\beta^2 \leq \text{opt}(G)$, β passes. Now, since β^* is the smallest β that fails, we have $\beta^{*2} \geq \text{opt}(G)$, which implies $\beta^* \geq \frac{\text{opt}(G)}{\beta^*} \geq \frac{\text{opt}(G)}{\alpha}$. Hence, $\frac{\beta^*}{2} \geq \frac{\text{opt}(G)}{2\alpha}$.

To see $\frac{\beta^*}{2} \leq \text{opt}(G)$, we consider two cases: $\alpha \leq \text{opt}(G)$ or $\alpha > \text{opt}(G)$. If $\alpha \leq \text{opt}(G)$, we trivially have $\frac{\beta^*}{2} \leq \frac{\alpha}{2} \leq \frac{\text{opt}(G)}{2} \leq \text{opt}(G)$. Now, if $\alpha > \text{opt}(G)$, there exists a unique $\beta' \in \{\log n, 2\log n, 2^2 \log n, \ldots, \alpha\}$ where $\frac{\beta'}{2} \leq \text{opt}(G) \leq \beta'$. Then by Claim 7.4, w.h.p. β' fails. Since β^* is the smallest that fails, $\beta^* \leq \beta'$. Hence $\frac{\beta^*}{2} \leq \frac{\beta'}{2} \leq \text{opt}(G)$. ■

It remains to analyze case (i).

Lemma 7.6. *If all β passes, $\max\left\{\alpha, \frac{\alpha}{\log^2 n} \cdot \widetilde{\text{opt}}_\alpha\right\}$ (denoted by* ALG*) is an $O(\alpha)$-approximation of $\text{opt}(G)$.*

Proof. Recall that if $\text{Tester}_\gamma(G^\alpha, \frac{n \log^2 n}{\alpha^2})$ returns No, $\widetilde{\text{opt}}_\alpha$ is the estimate returned by Tester_γ, and if Tester_γ returns Yes (i.e., $\text{opt}(G^\alpha) \geq \gamma \cdot \frac{n \log^2 n}{\alpha^2}$), $\widetilde{\text{opt}}_\alpha$ is defined to be $\gamma \cdot \frac{n \log^2 n}{\alpha^2}$.

Intuitively speaking, Tester_γ returning Yes is the special case where the sampled graph G^α has a matching of size (even) larger than $\frac{n}{\alpha^2}$, which implies that $\text{opt}(G)$ itself is very large ($\Omega(\frac{n}{\alpha})$ by Part (1) of Lemma 7.1). In this case, $\frac{n}{\alpha}$ is always an $O(\alpha)$-approximation (which is basically $\alpha \cdot \widetilde{\text{opt}}_\alpha$). We should remark that the expression we use for ALG is a unified expression that works for both Tester_γ outputs Yes and Tester_γ outputs No.

We now prove the lemma formally. First note that for either case, $\widetilde{\text{opt}}_\alpha \leq \text{opt}(G^\alpha)$. In the following, we first show that ALG $\leq \text{opt}(G)$, and then show that ALG $\geq \frac{\text{opt}(G)}{O(\alpha)}$.

To see that ALG $= \max\left\{\alpha, \frac{\alpha}{\log^2 n} \cdot \widetilde{\text{opt}}_\alpha\right\} \leq \text{opt}(G)$, firstly, if $\alpha > \text{opt}(G)$, then there exists β used by Algorithm 1 where $\frac{\beta}{2} \leq \text{opt}(G) \leq \beta$, and by Claim 7.4, this β fails w.p. $1 - o(1)$ (which

contradicts to the fact that all β pass). Therefore, $\alpha \leq \mathrm{opt}(G)$, and we only need to show that $\frac{\alpha}{\log^2 n} \cdot \widetilde{\mathrm{opt}}_\alpha \leq \mathrm{opt}(G)$. As pointed out above, $\mathrm{opt}(G^\alpha) \geq \widetilde{\mathrm{opt}}_\alpha$. Hence, w.h.p.,

$$
\begin{aligned}
\frac{\alpha}{\log^2 n} \cdot \widetilde{\mathrm{opt}}_\alpha &\leq \frac{\alpha}{\log^2 n} \cdot \mathrm{opt}(G^\alpha) \\
&\leq \frac{\alpha}{\log^2 n} \cdot \frac{3 \log n}{\alpha} \cdot \mathrm{opt}(G) \qquad \text{(By Lemma 7.1 Part (1))} \\
&\leq \mathrm{opt}(G)
\end{aligned}
$$

proving $\textsc{alg} \leq \mathrm{opt}(G)$.

To see that $\textsc{alg} = \max\left\{\alpha, \frac{\alpha}{\log^2 n} \cdot \widetilde{\mathrm{opt}}_\alpha\right\} = \frac{\mathrm{opt}(G)}{O(\alpha)}$, firstly, if $\mathrm{opt}(G) < \alpha^2$, trivially

$$
\textsc{alg} \geq \alpha > \frac{\mathrm{opt}(G)}{\alpha}.
$$

Therefore, we only need to consider $\mathrm{opt}(G) \geq \alpha^2$. There are two cases: $\mathsf{Tester}_\gamma(G^\alpha, \frac{n \log^2 n}{\alpha^2})$ returns Yes or returns No. If $\mathsf{Tester}_\gamma(G^\alpha, \frac{n \log^2 n}{\alpha^2})$ returns Yes, then $\widetilde{\mathrm{opt}}_\alpha := \frac{\gamma n \log^2 n}{\alpha^2}$, and hence

$$
\begin{aligned}
\textsc{alg} &\geq \frac{\alpha}{\log^2 n} \cdot \widetilde{\mathrm{opt}}_\alpha = \frac{\alpha}{\log^2 n} \cdot \frac{\gamma n \log^2 n}{\alpha^2} \\
&= \frac{\gamma n}{\alpha} \geq \frac{\gamma \cdot \mathrm{opt}(G)}{\alpha} = \frac{\mathrm{opt}(G)}{O(\alpha)}
\end{aligned}
$$

If $\mathsf{Tester}_\gamma(G^\alpha, \frac{n \log^2 n}{\alpha^2})$ returns No, then by the definition of Tester_γ, $\widetilde{\mathrm{opt}}_\alpha \geq \gamma \cdot \mathrm{opt}(G^\alpha)$. We have, w.h.p.,

$$
\begin{aligned}
\textsc{alg} &\geq \frac{\alpha}{\log^2 n} \cdot \widetilde{\mathrm{opt}}_\alpha \geq \frac{\alpha}{\log^2 n} \cdot \gamma \cdot \mathrm{opt}(G^\alpha) \\
&\geq \frac{\gamma \alpha}{\log^2 n} \cdot \frac{\log^2 n}{2\alpha^2} \cdot \mathrm{opt}(G) = \frac{\mathrm{opt}(G)}{O(\alpha)} \qquad \text{(Lemma 7.1 Part (2))}
\end{aligned}
$$

Therefore, $\textsc{alg} = \frac{\mathrm{opt}}{O(\alpha)}$ for all $\mathrm{opt}(G) \geq \alpha$, which completes the proof. $\blacksquare$

7.3 Implementing Matching Size Testers in Graph Streams

We now show how to implement matching size testers in insertion-only streams and dynamic streams.

Claim 7.7. *A 0.5-matching size tester* $\mathsf{Tester}_{0.5}(G, k)$ *can be implemented in* $\widetilde{O}(k)$ *space in insertion-only streams.*

Proof. Simply maintain a maximal matching M and stop when $k/2$ edges have been collected. If $|M| = k/2$, return Yes, and otherwise return No along with $|M|$ as the estimate. $\blacksquare$

Proof of Theorem 1, Part (1). Suppose Algorithm 1 returns a $c \cdot \alpha$-approximation (Lemma 7.2; c is a constant). First notice that if $\alpha < c \log n$, $\widetilde{O}(\frac{n}{\alpha^2})$ space is enough to store a maximal matching of the input graph G which is a 2-approximation of $\mathrm{opt}(G)$. Therefore, we only need to consider $\alpha \geq c \log n$. Define $\widehat{\alpha} = \alpha/c$; we have $\widehat{\alpha} \geq \log n$. Run Algorithm 1 for $\widehat{\alpha}$ using the tester by Claim 7.7.

By Lemma 7.2, Algorithm 1 returns a $c \cdot \widehat{\alpha}(= \alpha)$-approximation of $\mathrm{opt}(G)$ w.h.p. On the other hand, Algorithm 1 invokes $\mathsf{Tester}_\gamma(*, k)$ for $O(\log \alpha)$ times where the largest k used is $\widetilde{O}(\max\left\{\frac{n}{\alpha^2}, 1\right\}) = \widetilde{O}(\frac{n}{\alpha^2})$ (recall that $\alpha \leq \sqrt{n}$). Therefore, by Claim 7.7, the space requirement is $\widetilde{O}(\frac{n}{\alpha^2})$. $\blacksquare$

35

For implementing a matching size tester in dynamic streams, we use the following result from [9, 16].

Lemma 7.8 ([9, 16]). *There exists a constant γ such that a randomized γ-matching size tester* $\mathsf{Tester}_\gamma(G, k)$ *that succeeds w.p. $1 - o(\frac{1}{n})$ can be implemented in dynamic streams using $\widetilde{O}(k^2)$ space.*

One simple approach for implementing a tester for Lemma 7.8 is to randomly group the vertices into $\Theta(k)$ groups and compute a maximum matching between the groups. It is shown in [9] that this can be done in $\widetilde{O}(k^2)$ space, while w.h.p. the size of the maximum matching between the groups is either $\Omega(k)$ (hence tester outputs Yes) or $\Omega(\mathrm{opt})$ (hence tester outputs No, along with the matching size).

Proof of Theorem 1, Part (2). Suppose Algorithm 1 returns a $c \cdot \alpha$-approximation (Lemma 7.2; c is a constant). First notice that if $\alpha < c \log n$, $\widetilde{O}(\frac{n^2}{\alpha^4})$ bits of space is enough to maintain a counter for each edge slot in the input graph G, which can recover all edges in G. Therefore, we only need to consider $\alpha > c \log n$. Define $\widehat{\alpha} = \alpha/c$; we have $\widehat{\alpha} \geq \log n$. Run Algorithm 1 for $\widehat{\alpha}$ using the Tester_γ by Lemma 7.8. Since Algorithm 1 only invokes Tester_γ for $O(\log n)$ times, by Lemma 7.8, w.h.p., no Tester_γ fails.

Now by Lemma 7.2, Algorithm 1 outputs an $c \cdot \widehat{\alpha}(= \alpha)$-approximation of $\mathrm{opt}(G)$. On the other hand, Algorithm 1 invokes $\mathsf{Tester}_\gamma(*, k)$ for $O(\log \alpha)$ times where the largest k used is $\widetilde{O}(\max\left\{\frac{n}{\alpha^2}, 1\right\}) = \widetilde{O}(\frac{n}{\alpha^2})$ (recall that $\alpha \leq \sqrt{n}$). Therefore, by Lemma 7.8, the space requirement is $\widetilde{O}(\frac{n^2}{\alpha^4})$. ∎

Acknowledgements

References

[1] Ahn, K. J., and Guha, S. Linear programming in the semi-streaming model with application to the maximum matching problem. *Inf. Comput. 222* (2013), 59–79.

[2] Ahn, K. J., and Guha, S. Access to data and number of iterations: Dual primal algorithms for maximum matching under resource constraints. In *Proceedings of the 27th ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2015, Portland, OR, USA, June 13-15, 2015* (2015), pp. 202–211.

[3] Ahn, K. J., Guha, S., and McGregor, A. Graph sketches: sparsification, spanners, and subgraphs. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2012, Scottsdale, AZ, USA, May 20-24, 2012* (2012), pp. 5–14.

[4] Ai, Y., Hu, W., Li, Y., and Woodruff, D. P. New characterizations in turnstile streams with applications. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan* (2016), pp. 20:1–20:22.

[5] Alon, N. Testing subgraphs in large graphs. *Random Struct. Algorithms 21*, 3-4 (2002), 359–370.

[6] Alon, N., Matias, Y., and Szegedy, M. The space complexity of approximating the frequency moments. In *STOC* (1996), ACM, pp. 20–29.

[7] Alon, N., Moitra, A., and Sudakov, B. Nearly complete graphs decomposable into large induced matchings and their applications. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012* (2012), pp. 1079–1090.

[8] Alon, N., and Shapira, A. A characterization of easily testable induced subgraphs. *Combinatorics, Probability & Computing 15*, 6 (2006), 791–805.

[9] Assadi, S., Khanna, S., Li, Y., and Yaroslavtsev, G. Maximum matchings in dynamic graph streams and the simultaneous communication model. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016* (2016), pp. 1345–1364.

[10] Bar-Yossef, Z., Jayram, T. S., Kumar, R., and Sivakumar, D. An information statistics approach to data stream and communication complexity. In *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings* (2002), pp. 209–218.

[11] Bar-Yossef, Z., Jayram, T. S., Kumar, R., and Sivakumar, D. Information theory methods in communication complexity. In *Proceedings of the 17th Annual IEEE Conference on Computational Complexity, Montréal, Québec, Canada, May 21-24, 2002* (2002), pp. 93–102.

[12] Barak, B., Braverman, M., Chen, X., and Rao, A. How to compress interactive communication. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010* (2010), pp. 67–76.

[13] BIRK, Y., LINIAL, N., AND MESHULAM, R. On the uniform-traffic capacity of single-hop interconnections employing shared directional multichannels. *IEEE Transactions on Information Theory 39*, 1 (1993), 186–191.

[14] BURY, M., AND SCHWIEGELSHOHN, C. Sublinear estimation of weighted matchings in dynamic data streams. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings* (2015), pp. 263–274.

[15] CHAKRABARTI, A., SHI, Y., WIRTH, A., AND YAO, A. C. Informational complexity and the direct sum problem for simultaneous message complexity. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA* (2001), pp. 270–278.

[16] CHITNIS, R., CORMODE, G., ESFANDIARI, H., HAJIAGHAYI, M., McGREGOR, A., MONEMIZADEH, M., AND VOROTNIKOVA, S. Kernelization via sampling with applications to finding matchings and related problems in dynamic graph streams. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016* (2016), pp. 1326–1344.

[17] CHITNIS, R. H., CORMODE, G., HAJIAGHAYI, M. T., AND MONEMIZADEH, M. Parameterized streaming: Maximal matching and vertex cover. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015* (2015), pp. 1234–1251.

[18] CLARKSON, K. L., AND WOODRUFF, D. P. Numerical linear algebra in the streaming model. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009* (2009), pp. 205–214.

[19] COVER, T. M., AND THOMAS, J. A. *Elements of information theory (2. ed.).* Wiley, 2006.

[20] CROUCH, M., AND STUBBS, D. S. Improved streaming algorithms for weighted matching, via unweighted matching. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, September 4-6, 2014, Barcelona, Spain* (2014), pp. 96–104.

[21] EGGERT, S., KLIEMANN, L., AND SRIVASTAV, A. Bipartite graph matchings in the semi-streaming model. In *Algorithms - ESA 2009, 17th Annual European Symposium, Copenhagen, Denmark, September 7-9, 2009. Proceedings* (2009), pp. 492–503.

[22] EPSTEIN, L., LEVIN, A., MESTRE, J., AND SEGEV, D. Improved approximation guarantees for weighted matching in the semi-streaming model. *SIAM J. Discrete Math. 25*, 3 (2011), 1251–1265.

[23] ESFANDIARI, H., HAJIAGHAYI, M. T., LIAGHAT, V., MONEMIZADEH, M., AND ONAK, K. Streaming algorithms for estimating the matching size in planar graphs and beyond. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015* (2015), pp. 1217–1233.

[24] FEIGENBAUM, J., KANNAN, S., McGREGOR, A., SURI, S., AND ZHANG, J. On graph problems in a semi-streaming model. *Theor. Comput. Sci. 348*, 2-3 (2005), 207–216.

[25] FISCHER, E., LEHMAN, E., NEWMAN, I., RASKHODNIKOVA, S., RUBINFELD, R., AND SAMORODNITSKY, A. Monotonicity testing over general poset domains. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada* (2002), pp. 474–483.

[26] FOX, J. A new proof of the graph removal lemma. *Annals of Mathematics 174*, 1 (2011), 561–579.

[27] FOX, J., HUANG, H., AND SUDAKOV, B. On graphs decomposable into induced matchings of linear sizes. *arXiv preprint arXiv:1512.07852* (2015).

[28] GAVINSKY, D., KEMPE, J., KERENIDIS, I., RAZ, R., AND DE WOLF, R. Exponential separations for one-way quantum communication complexity, with applications to cryptography. *STOC* (2007), 516–525.

[29] GOEL, A., KAPRALOV, M., AND KHANNA, S. On the communication and streaming complexity of maximum bipartite matching. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms* (2012), SODA '12, SIAM, pp. 468–485.

[30] GOWERS, W. Some unsolved problems in additive/combinatorial number theory. *preprint* (2001).

[31] GURUSWAMI, V., AND ONAK, K. Superlinear lower bounds for multipass graph processing. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013* (2013), pp. 287–298.

[32] HÅSTAD, J., AND WIGDERSON, A. Simple analysis of graph tests for linearity and PCP. *Random Struct. Algorithms 22*, 2 (2003), 139–160.

[33] HUANG, Z., RADUNOVIC, B., VOJNOVIC, M., AND ZHANG, Q. Communication complexity of approximate matching in distributed graphs. In *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany* (2015), pp. 460–473.

[34] JAIN, R., RADHAKRISHNAN, J., AND SEN, P. A direct sum theorem in communication complexity via message compression. In *Automata, Languages and Programming, 30th International Colloquium, ICALP 2003, Eindhoven, The Netherlands, June 30 - July 4, 2003. Proceedings* (2003), pp. 300–315.

[35] KANE, D. M., NELSON, J., AND WOODRUFF, D. P. An optimal algorithm for the distinct elements problem. In *PODS* (2010), ACM, pp. 41–52.

[36] KAPRALOV, M. Better bounds for matchings in the streaming model. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013* (2013), pp. 1679–1697.

[37] KAPRALOV, M., KHANNA, S., AND SUDAN, M. Approximating matching size from random streams. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014* (2014), pp. 734–751.

[38] KONRAD, C. Maximum matching in turnstile streams. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings* (2015), pp. 840–852.

[39] KONRAD, C., MAGNIEZ, F., AND MATHIEU, C. Maximum matching in semi-streaming with few passes. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings* (2012), pp. 231–242.

[40] KUSHILEVITZ, E., AND NISAN, N. *Communication complexity.* Cambridge University Press, 1997.

[41] LI, Y., NGUYEN, H. L., AND WOODRUFF, D. P. On sketching matrix norms and the top singular vector. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014* (2014), pp. 1562–1581.

[42] LI, Y., NGUYEN, H. L., AND WOODRUFF, D. P. Turnstile streaming algorithms might as well be linear sketches. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014* (2014), pp. 174–183.

[43] LI, Y., SUN, X., WANG, C., AND WOODRUFF, D. P. On the communication complexity of linear algebraic problems in the message passing model. In *Distributed Computing - 28th International Symposium, DISC 2014, Austin, TX, USA, October 12-15, 2014. Proceedings* (2014), pp. 499–513.

[44] LI, Y., AND WOODRUFF, D. P. On approximating functions of the singular values in a stream. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016* (2016), pp. 726–739.

[45] LI, Y., AND WOODRUFF, D. P. Tight bounds for sketching the operator norm, schatten norms, and subspace embeddings. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, September 7-9, 2016, Paris, France* (2016), pp. 39:1–39:11.

[46] LOVÁSZ, L., AND PLUMMER, D. *Matching Theory.* AMS Chelsea Publishing Series. American Mathematical Soc., 2009.

[47] MCGREGOR, A. Finding graph matchings in data streams. In *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th InternationalWorkshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings* (2005), pp. 170–181.

[48] MCGREGOR, A. Graph stream algorithms: a survey. *SIGMOD Record 43*, 1 (2014), 9–20.

[49] MCGREGOR, A., AND VOROTNIKOVA, S. Planar matching in streams revisited. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, September 7-9, 2016, Paris, France* (2016), pp. 17:1–17:12.

[50] MOTWANI, R., AND RAGHAVAN, P. *Randomized Algorithms.* Cambridge University Press, 1995.

[51] PHILLIPS, J. M., VERBIN, E., AND ZHANG, Q. Lower bounds for number-in-hand multiparty communication complexity, made easy. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012* (2012), pp. 486–501.

[52] Ruzsa, I. Z., and Szemerédi, E. Triple systems with no six points carrying three triangles. *Combinatorics (Keszthely, 1976), Coll. Math. Soc. J. Bolyai 18* (1978), 939–945.

[53] Tao, T., and Vu, V. H. *Additive combinatorics*, vol. 105. Cambridge University Press, 2006.

[54] Tutte, W. T. The Factorization of Linear Graphs. *Journal of the London Mathematical Society s1-22*, 2 (1947), 107–111.

[55] Verbin, E., and Yu, W. The streaming complexity of cycle counting, sorting by reversals, and other problems. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011* (2011), pp. 11–25.

[56] Zelke, M. Weighted matching in the semi-streaming model. *Algorithmica 62*, 1-2 (2012), 1–20.

A An $O(\sqrt{n})$-Approximation Algorithm in $\mathrm{polylog}(n)$-Space

For completeness, we sketch the proof of a simple $O(\sqrt{n})$-approximation algorithm for matching size estimation in dynamic streams. We emphasize here that this algorithm is already known in the literature (see, e.g., [37]) and is provided here for the sake of completeness.

Theorem 11 (Folklore). *There exists a $\mathrm{polylog}(n)$-space algorithm that with high probability, outputs an $O(\sqrt{n})$-approximation of the maximum matching size in dynamic graph streams.*

Proof Sketch. Recall that $\mathrm{opt} := \mathrm{opt}(G)$ denotes the cardinality of a maximum matching in the graph $G(L, R, E)$. To achieve an $O(\sqrt{n})$-approximation to opt, we will establish a simple connection between the cardinality of a maximum matching and the number of neighbors of a set of $\sqrt{n}$ random vertices chosen from L. Let $S \subseteq L$ be a random set of size $\sqrt{n}$. We denote by $N(S)$ the set of neighbors of S in the final graph (at the end of the stream), and let $k = \min\{|N(S)|, \sqrt{n}\}$. We show that w.h.p $\Omega(k) \leq \mathrm{opt} \leq O(k\sqrt{n})$. Using the ℓ_0-estimation algorithm of Kane *et al.* [35], we can estimate $|N(S)|$ to within a constant factor using $\mathrm{polylog}(n)$ space with success probability .99 in dynamic steams. This, together with the aforementioned result, suffices to achieve an $O(\sqrt{n})$-approximation of the matching size. Note that the error can be made one-sided in a straightforward way, and the overall probability of success can be boosted to $(1 - 1/n)$ by running $O(\log n)$ parallel copies and taking the median value.

We now briefly explain how the aforementioned relation between k and opt is established. We show that there exist two constants $c_1, c_2 > 1$ such that for any $k \in [\sqrt{n}]$, if $\mathrm{opt} < k/c_1$, $|N(S)|$ is less than k, and if $\mathrm{opt} > c_2 k\sqrt{n}$, $|N(S)|$ is larger than k, each with probability at least 0.99. If $\mathrm{opt} < k/c_1$, then by the extended Hall's Theorem, there exists a set S' of $(n - k/c_1)$ vertices in L that has at most $k/c_1 (< k)$ neighbors. Since $k \leq \sqrt{n}$, the size of S' is large enough to ensure that with a constant probability (which can be made to 0.99 by choosing a large enough constant c_1), the $\sqrt{n}$ chosen vertices in S are a subset of S', and hence $|N(S)| < k$.

On the other hand, if $\mathrm{opt} > c_2 \cdot k\sqrt{n}$, then there exists two subset of vertices $A \subseteq L$ and $B \subseteq R$, with $|A| = |B| \geq c_2 \cdot k\sqrt{n}$ such that there exists a perfect matching between A and B in G. Consequently, with a constant probability (which can be made to 0.99 by choosing a large enough constant c_2), S contains more than k vertices of A, and the perfect matching ensures that the number of neighbors of S is more than k. ■