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 $O(n \cdot \text{poly}(\log n/\varepsilon))$-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).

*A full version of the paper is available on arXiv.

†Department of Computer and Information Science, University of Pennsylvania. Supported in part by National Science Foundation grants CCF-1116961, CCF-1552909, CCF-1617851, and IIS-1447470. Email: {sassadi,sanjeev,yangli2}@cis.upenn.edu.

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 [42] for a summary).

One of the most extensively studied problems in the streaming literature is the classical problem of finding a *maximum matching* [40]. Although significant advances have been made on understanding the space needed to compute a maximum matching in the streaming model [41, 21, 18, 19, 26, 34, 50, 3, 1, 28, 31, 32, 17, 14, 42, 2, 20, 33, 9, 13, 11, 43], 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 [31, 26] 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 [31, 26] 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 [32]). It was shown later in [20] that computing better than a 3/2-approximation requires $\Omega(\sqrt{n})$ bits of space. More recently, this lower bound was extended by [11] 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 (a proof of this result appears in the full version of the paper). 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* [32] or the input graph has *bounded arboricity* [20, 13, 11, 43].

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, 13], that it is also *sufficient* (see also [33]). 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 [20, 11], 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 than 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 [20, 43, 32]. 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 1.1. *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 1.1 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 1.2. *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 1.2 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.* [32], 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 1.2 make progress on this question in dynamic streams.

An interesting aspect of our lower bound in Theorem 1.2 is that it holds even for bounded arboricity

[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.

graphs. There is an active line of research on estimating matching size of bounded arboricity graphs in graph streams [13, 11, 20, 43], initiated by Esfandiari *et al.* [20]. The state-of-the-art is an $O(1)$-approximation in $\widetilde{O}(n^{4/5})$ space for dynamic streams in bounded-arboricity graphs [13, 11, 43].

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).

THEOREM 1.3. *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 $n^{\Omega(1/\log\log n)}$ [22], Theorem 1.3 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 [2, 1] (see also [42]).

Our lower bounds in Theorem 1.3 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 1.1), 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 [38] 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 [15, 11, 38, 35, 39, 37]. It is shown that exact computation of matrix rank in data streams requires $\Omega(n^2)$ space [15, 37] (even allowing multiple passes), and $(1+\varepsilon)$-approximation requires $n^{1-O(\varepsilon)}$ space [11]; the latter result was recently extended to all Schatten p-norms for *odd* values of p [38].

It is well-known that computing the maximum matching size is equivalent to computing the rank of the Tutte matrix [48, 40]. 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 [38].

1.3 Organization We start with preliminaries and notation in Section 2. A detailed technical overview of the paper is provided in Section 3. We present our lower bounds for α-approximation (first part of Theorem 1.2) in Section 4, our lower bounds for $(1+\varepsilon)$-approximation (Theorem 1.3) in Section 5, and our α-approximation algorithms (Theorem 1.1) in Section 6. Proof of the second part of Theorem 1.2 and omitted details are deferred to the full version of the paper.

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 $\boldsymbol{X}$, let $\textsc{supp}(\boldsymbol{X})$ denote its support set, and let $|\boldsymbol{X}| := \log|\textsc{supp}(\boldsymbol{X})|$. Given a k-dimensional tuple $X = (X_1, \ldots, X_k)$ and an $i \in [k]$, let $X^{<i} := (X_1, \ldots, X_{i-1})$, and $X^{-i} := (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_k)$.

2.1 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 [16].

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 know that $0 \le H(\boldsymbol{A}) \le |\boldsymbol{A}|$ and equality holds iff $\boldsymbol{A}$ is uniform on its support. Similarly, $I(\boldsymbol{A};\boldsymbol{B}) \ge 0$ 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 [16], Chapter 2).

FACT 2.1. *Let $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ be random variables.*

1. Conditioning reduces the entropy*: $H(\boldsymbol{A} \mid \boldsymbol{B},\boldsymbol{C}) \le H(\boldsymbol{A} \mid \boldsymbol{B})$; equality holds iff $\boldsymbol{A}$ and $\boldsymbol{C}$ are independent conditioned on $\boldsymbol{B}$.*
2. Chain rule for entropy*: $H(\boldsymbol{A},\boldsymbol{B}) = H(\boldsymbol{A}) + H(\boldsymbol{B} \mid \boldsymbol{A})$.*

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

4. Conditional sub-additivity of mutual information: *if* $\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_t$ *are conditionally independent given* $\boldsymbol{B}$, *then* $I(\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_t; \boldsymbol{B}) \leq \sum_{i=1}^{t} I(\boldsymbol{A}_i; \boldsymbol{B})$.

5. Conditional super-additivity of mutual information: *if* $\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_t$ *are conditionally independent given* $\boldsymbol{C}$, *then* $I(\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_t; \boldsymbol{B} \mid \boldsymbol{C}) \geq \sum_{i=1}^{t} I(\boldsymbol{A}_i; \boldsymbol{B} \mid \boldsymbol{C})$.

2.2 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 2.1. (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 [46], have been extensively studied as they arise naturally in property testing, PCP constructions, additive combinatorics, etc. (see, e.g., [47, 29, 22, 10, 7, 26, 5, 8, 24]). An important extensively studied case is $r = \Theta(n)$ [22, 23, 24], 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 $\mathsf{RS}(n)$ [23, 27, 24], and current best known constructions imply $\mathsf{RS}(n) = n^{\Omega(1/\log\log n)}$ [22] (see the full version of the paper for more details).

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 in the *two-player one-way communication* model and the *multi-party number-in-hand simultaneous message passing model* (SMP).

In the one-way model, the input is partitioned between two players Alice and Bob, Alice sends a single message to Bob, and Bob outputs the answer using his input and the message received from Alice. In the SMP model, the input is partitioned across k players and each player *simultaneously* sends a message to a central party called the *referee* who upon receiving the messages, outputs the answer. In both models, we use Π to denote a protocol used by the players and unless specified otherwise, we always assume that the protocol Π can be randomized (using public and private coin), *even against a prior distribution $\mathcal{D}$ of inputs.*

The *communication cost* of a one-way protocol Π for a problem P on a distribution $\mathcal{D}$, denoted $\|\Pi\|$, is the worst-case size of the message sent from Alice to Bob in the protocol Π, when the inputs are chosen from $\mathcal{D}$. Similarly, the *communication cost* of a SMP protocol Π is the sum of the worst-case size of the messages sent by players to the referee, i.e., $\|\Pi\| := \sum_{i \in [k]} |\Pi_i|$. *Communication complexity* $\mathsf{CC}^{\delta}_{\text{1-way},\mathcal{D}}(P)$ of a problem P with respect to $\mathcal{D}$ is the minimum communication cost of any one-way protocol Π that is required to solve P on *every instance* w.p. at least $1-\delta$. $\mathsf{CC}^{\delta}_{\text{SMP},\mathcal{D}}(P)$ in the SMP model is defined analogously.

Information Complexity. We also use the notion of (external) *information cost* of a protocol: 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 input to Alice and $\boldsymbol{\Pi} := \boldsymbol{\Pi}(\boldsymbol{X})$ is the random variable for the message sent from Alice to Bob in the protocol Π, *concatenated* with the *public* randomness $\boldsymbol{R}$. The *information complexity* $\mathsf{IC}^{\delta}_{\text{1-way},\mathcal{D}}(P)$ of P with respect to a distribution $\mathcal{D}$ is the minimum $\mathsf{ICost}_{\mathcal{D}}(\Pi)$ of any one-way protocol Π that solves P on *every instance* w.p. at least $1-\delta$.

Similarly, the information cost of a SMP protocol is defined as $\sum_{j=1}^{k} I_{\mathcal{D}}(\boldsymbol{\Pi}_j; \boldsymbol{X}_1, \ldots, \boldsymbol{X}_k)$, 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 $\boldsymbol{R}$. Information complexity $\mathsf{IC}^{\delta}_{\text{SMP},\mathcal{D}}(P)$ of a problem P in the SMP model is defined analogous to the one-way model.

The following well-known proposition (see, e.g., [12]) relates communication complexity and information complexity (a short proof is provided in the full version of the paper).

Proposition 2.1. *For every $0 < \delta < 1$ and every distribution $\mathcal{D}$:*

1. $\mathsf{CC}^{\delta}_{\text{1-way},\mathcal{D}}(P) \geq \mathsf{IC}^{\delta}_{\text{1-way},\mathcal{D}}(P)$,

2. $\mathsf{CC}^{\delta}_{\text{SMP},\mathcal{D}}(P) \geq \mathsf{IC}^{\delta}_{\text{SMP},\mathcal{D}}(P)$.

Connection to Streaming: It is a standard fact that lower bounds in the one-way communication model imply the same bounds on the space complexity of streaming algorithms in *insertion-only* streams. Recent results of [4, 36] prove a similar situation for the

SMP model and *dynamic* streams: 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 [49] in proving our lower bounds.

DEFINITION 2.2. ($\mathsf{BHH}_{n,t}$) *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 known as the *Boolean Hidden Matching* problem, BHM_n, and was originally introduced by Gavinsky *et al.* [25] 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 [49]. We further extend this result and establish a matching lower bound on the *information complexity* of $\mathsf{BHH}_{n,t}$ (see the full version of the paper).

For our purpose, it is more convenient to work with a special case of the $\mathsf{BHH}_{n,t}$ problem, namely $\mathsf{BHH}^0_{n,t}$ 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 $\mathsf{BHM}^0_n := \mathsf{BHH}^0_{n,2}$ (similar to BHM_n). It is known that (see, e.g. [49, 11, 38]) 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. This allows for extending the communication and information complexity lower bounds of $\mathsf{BHH}_{n,t}$ to $\mathsf{BHH}^0_{2n,t}$ problem; we summarize this in the following corollary whose proof is deferred to the full version of the paper.

COROLLARY 2.1. *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).*

$\mathsf{BHH}^0_{n,t}$ and Matching Size Estimation. The $\mathsf{BHH}^0_{n,t}$ problem has been used previously in [20, 11] to prove lower bounds for estimating matching size in data streams. We now briefly describe this connection.

The following reduction was first proposed by [11]. Given an instance $(x, \mathcal{M})$[3] of $\mathsf{BHH}^0_{n,t}$, 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 $\boldsymbol{e}$ in the t-hypermatching $\mathcal{M}$, Bob adds to E a clique between the vertices w_i where i is incident on $\boldsymbol{e}$.

The following claim, proven originally by [11], establishes the correctness of this reduction.

CLAIM 2.1. ([11]) *Suppose* $G(V \cup W, E)$ *is the graph obtained from an instance* $(x, \mathcal{M})$ *of* $\mathsf{BHH}^0_{n,t}$ *(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* $\mu(G) = \frac{3n}{4} - \frac{n}{2t}$.

3 Technical Overview

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 1.2) and dense graphs (Theorem 1.3 and second part of Theorem 1.2) using conceptually different techniques; we elaborate below on each case separately.

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

[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.

not known to the players). In the Yes case, the end-points of every edge are either both shared or both private such that $\text{opt}(G) = \Theta(V_P)$, and in the No case, every edge has one shared end-point and one private end-point, hence $\text{opt}(G) = \Theta(V_S)$.

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 [25]. 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* [45]; 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 [31, 33, 9, 30]. 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 1.3 is [11] (itself based on a prior result of [20]) 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 [49].

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 [26, 31, 9, 33] for proving lower bounds for *finding* approximate matchings. While it was possible to analyze the hard instances in [26, 31, 9, 33] 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.

Upper bounds. The main idea behind our algorithms in Theorem 1.1 is the following 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 $\text{opt}(G)$.

4 An $\Omega(\sqrt{n}/\alpha^{2.5})$ Lower Bound for α-Approximating Matching Size

In this section, we present our space lower bounds for α-approximation algorithms in dynamic streams. In particular, we consider the sparse graphs case (i.e., Part (1) of Theorem 1.2), 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 (the proof of Part (2) of Theorem 1.2 is deferred to the full version of the paper). As already remarked in Section 2, by the results of [4, 36], it suffices to prove the lower bound in the SMP model.

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[4] and the goal of the players is to approximate the maximum matching size of $G(V, \bigcup_{i\in[k]} M_i)$ 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 4.1. *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 $\delta < 1/2$:*

$$\mathsf{CC}^{\delta}_{\text{SMP},\mathcal{D}}(\mathsf{SMS}_{n,k}) = \Omega\left(\frac{\sqrt{n}}{k\sqrt{k}}\right)$$

Part (1) of Theorem 1.2 immediately follows from Theorem 4.1.

Proof. [Proof of Theorem 1.2, 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, 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 [36, 4]. Since $k = \Theta(\alpha)$, by Theorem 4.1, any single-pass streaming algorithm for matching size estimation in dynamic streams requires $\Omega(\sqrt{n}/\alpha^{2.5})$ bits of space.

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 4.1. This theorem is ultimately proved by a reduction from the BHM^0 problem defined in Section 2. 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 $\widetilde{O}(1)$

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

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.

4.0.1 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 2.1 (for $t = 2$). We also define $\mathcal{D}^{\mathsf{Y}}_{\mathsf{BHM}}$ and $\mathcal{D}^{\mathsf{N}}_{\mathsf{BHM}}$ as the distribution on Yes and No instances of $\mathcal{D}_{\mathsf{BHM}}$. We use $\mathcal{D}_{\mathsf{BHM}}$ to define the following input distribution $\mathcal{D}_{\mathsf{SMS}}$ for $\mathsf{SMS}_{n,k}$.

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 $(M_i^{\mathsf{B}}, x_i^{\mathsf{B}}) \sim \mathcal{D}_{\mathsf{BHM}}$.
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 $x_i^{\mathsf{B}}(j) = 1$, if $x_i^{\mathsf{B}}(j)$ is the ℓ-th smallest index with value 1, let $\sigma_i(j) := \sigma(\ell)$[a].
   - For each $j \in [\frac{2n}{k}]$ with $x_i^{\mathsf{B}}(j) = 0$, if $x_i^{\mathsf{B}}(j)$ 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 $M_i := \left\{(\sigma_i(u), \sigma_i(v)) \mid (u, v) \in M_i^{\mathsf{B}}\right\}$.

[a] Here, we use the fact that $\|x_i^{\mathsf{B}}\|_0 = \frac{n}{k}$ in $\mathcal{D}_{\mathsf{BHM}}$ by Corollary 2.1

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 4.1. (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 4.1. *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 $\mathcal{D}^{\mathsf{Y}}_{\mathsf{SMS}}$, 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 2.1), 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 $\frac{1}{2} \cdot \left|M_i^{\mathsf{B}}\right| = \frac{n}{2k}$ 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 $\mathcal{D}^{\mathsf{N}}_{\mathsf{SMS}}$, 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 $\left\{\sigma(1), \sigma(2), \ldots, \sigma(\frac{n}{k})\right\}$ (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 4.1, 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 $\text{opt}(G)$ 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 $\text{opt}(G)$ 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 4.2. (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 4.1. *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

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

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 $p_{f_{k+1}}^{\mathsf{N}}$ to both sides of the inequality)

$$p_{f_1}^{\mathsf{Y}} + p_{f_{k+1}}^{\mathsf{N}} < p_{f_{k+1}}^{\mathsf{Y}} + p_{f_{k+1}}^{\mathsf{N}} + 2\gamma \cdot k \tag{4.1}$$

$$= 1 + 2\gamma \cdot k = 2 \cdot (1 - \delta) \tag{4.2}$$

by our choice of γ. However, since Π_{SMS} is a δ-error protocol for $\mathsf{SMS}_{n,k}$, by Claim 4.1, 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, $\frac{1}{2} \cdot (p_{f_1}^{\mathsf{Y}} + p_{f_{k+1}}^{\mathsf{N}}) \geq 1 - \delta$, a contradiction to Eq (4.2).

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 $\frac{1}{2} + \gamma$.

4.0.2 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 4.1), and let input profiles $f_{i^\star}$ and $g_{i^\star}$ be the witness of $i^\star$. We design the following protocol Π_{BHM} using Π_{SMS} as a sub-routine.

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 4.2. *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 full version of the paper.

Proof. [Proof of Theorem 4.1]

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 4.2, 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 2.1, 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 $\mathsf{CC}^{\delta}_{\mathrm{SMP},\mathcal{D}}(\mathsf{SMS}_{n,k}) = \Omega\left(\frac{\sqrt{n}}{k\sqrt{k}}\right)$ for $\mathcal{D} := \frac{1}{2}\left(\mathcal{D}_{\mathsf{SMS}} \mid f_{i^\star}\right) + \frac{1}{2}\left(\mathcal{D}_{\mathsf{SMS}} \mid g_{i^\star}\right)$.

5 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 $M^{\mathsf{RS}}_1, \ldots, M^{\mathsf{RS}}_t$. For each matching M^{RS}_i, we assume an arbitrary ordering of the edges in M^{RS}_i, 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 $L(M^{\mathsf{RS}}_i) := \{u_{i,1}, \ldots, u_{i,r}\}$ and $R(M^{\mathsf{RS}}_i) := \{v_{i,1}, \ldots, v_{i,r}\}$. We emphasize that we do not require $G^{\mathsf{RS}}(V, E)$ to be necessarily a *bipartite* graph; each bipartition $L(M^{\mathsf{RS}}_i)$ and $R(M^{\mathsf{RS}}_i)$ (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[5] $\mathcal{M}$ 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 $\mathbf{e} \in \mathcal{M}$: $C_{\mathbf{e}} := \{v_{i,k} \mid k \in \mathbf{e}\}$.

5.1 Insertion-Only Streams We define the *two-player one-way communication* problem $\mathsf{Matching}_{n,\varepsilon}$ as the 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 5.1. (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 $\delta < \frac{1}{2}$:*

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

The lower bound of theorem 5.1, together with Proposition 2.1, 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 1.3.

In the following, we focus on proving Theorem 5.1. Suppose the maximum value for $\mathsf{RS}(n)$ is achieved by an (r,t)-RS graph with the parameter $r = \mathsf{c}_{\mathrm{rs}} \cdot n$. We propose the following (hard) input distribution $\mathcal{D}_{\mathsf{M}}$ for $\mathsf{Matching}_{n,\varepsilon}$.

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

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

- 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 $\mathcal{D}_{\mathrm{MM}}$. The following claim establishes the connection between I_{BHH} and maximum matching size in G.

CLAIM 5.1. *For $G \sim \mathcal{D}_{\mathsf{M}}$, let:*

$$\mathrm{opt}_{\mathsf{Yes}} := \min_G \Big(\mathrm{opt}(G) \mid I_{\mathsf{BHH}} \textit{ is a } \mathsf{Yes} \textit{ instance}\Big)$$
$$\mathrm{opt}_{\mathsf{No}} := \max_G \Big(\mathrm{opt}(G) \mid I_{\mathsf{BHH}} \textit{ is a } \mathsf{No} \textit{ instance}\Big)$$

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

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 2.1, if I_{BHH} is a Yes instance, then $\mathrm{opt}(H) = \frac{3r}{4}$. Hence, in this case,

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

[5] 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

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

$$\begin{aligned}\text{opt}(G) &= |V_2| + \text{opt}(H) \leq N - 2r + \frac{3r}{4} - \frac{r}{2p} \\ &= N - \frac{5\mathsf{c}_{\text{rs}}N}{4} - \frac{\mathsf{c}_{\text{rs}}}{2p}N = \frac{4-5\mathsf{c}_{\text{rs}}}{4} \cdot N - \frac{\mathsf{c}_{\text{rs}}}{2p}N\end{aligned}$$

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

Fix any δ-error protocol Π_{Matching} for $\mathsf{Matching}_{n,\varepsilon}$ on $\mathcal{D}_{\mathsf{M}}$; Claim 5.1 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 opt_{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 $V_1 \setminus V(M^{\mathsf{RS}}_{j^\star})$ and V_2. He also creates $E_{B,2}$ as the edges of the p-clique family of $\mathcal{M}$ on $R(M^{\mathsf{RS}}_{j^\star})$ (V_1, V_2, and $M^{\mathsf{RS}}_{j^\star}$ 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 opt_{No} and No otherwise.

The correctness of the protocol follows immediately from Claim 5.1. We now bound the information cost of this new protocol.

LEMMA 5.1. $\mathsf{ICost}_{\mathcal{D}_{\mathsf{BHH}}}(\Pi_{\mathsf{BHH}}) \leq \frac{1}{t} \cdot \mathsf{ICost}_{\mathcal{D}_{\mathsf{M}}}(\Pi_{\mathsf{Matching}})$.

Proof. We have,

$$\begin{aligned}&\mathsf{ICost}_{\mathcal{D}_{\mathsf{BHH}}}(\Pi_{\mathsf{BHH}}) = I_{\mathcal{D}_{\mathsf{BHH}}}(\boldsymbol{X}; \boldsymbol{\Pi}_{\mathsf{BHH}}, \boldsymbol{R}) \\ &\text{(by Fact 2.1-(3) and since } I(\boldsymbol{X}; \boldsymbol{R}) = 0 \text{ as } \boldsymbol{X} \perp \boldsymbol{R}) \\ &\qquad = I_{\mathcal{D}_{\mathsf{BHH}}}(\boldsymbol{X}; \boldsymbol{\Pi}^R_{\mathsf{BHH}} \mid \boldsymbol{R}) \\ &\text{(the message of } \Pi_{\mathsf{BHH}} \text{ is the same as } \Pi_{\mathsf{Matching}}) \\ &\qquad = I_{\mathcal{D}_{\mathsf{BHH}}}(\boldsymbol{X}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{J}) \\ &\qquad = \mathop{\mathbb{E}}_{j \in [t]}\left[I_{\mathcal{D}_{\mathsf{BHH}}}(\boldsymbol{X}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{J} = j)\right] \\ &\qquad = \frac{1}{t} \cdot \sum_{j=1}^{t} I_{\mathcal{D}_{\mathsf{BHH}}}(\boldsymbol{X}_j; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{J} = j) \\ &\qquad = \frac{1}{t} \cdot \sum_{j=1}^{t} I_{\mathcal{D}_{\mathsf{M}}}(\boldsymbol{X}_j; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{J} = j) \\ &\qquad = \frac{1}{t} \cdot \sum_{j=1}^{t} I_{\mathcal{D}_{\mathsf{M}}}(\boldsymbol{X}_j; \boldsymbol{\Pi}_{\mathsf{Matching}})\end{aligned}$$

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,

$$\begin{aligned}&\mathsf{ICost}_{\mathcal{D}_{\mathsf{BHH}}}(\Pi_{\mathsf{BHH}}) = \frac{1}{t} \cdot \sum_{j=1}^{t} I_{\mathcal{D}_{\mathsf{M}}}(\boldsymbol{X}_j; \boldsymbol{\Pi}_{\mathsf{Matching}}) \\ &\text{(by Fact 2.1-(5) since } \boldsymbol{X}_j \perp \boldsymbol{X}^{<j}) \\ &\qquad \leq \frac{1}{t} \cdot I_{\mathcal{D}_{\mathsf{M}}}(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_t; \boldsymbol{\Pi}_{\mathsf{Matching}}) \\ &\qquad = \frac{1}{t} \cdot I_{\mathcal{D}_{\mathsf{M}}}(\boldsymbol{E}_A; \boldsymbol{\Pi}_{\mathsf{Matching}}) \\ &\qquad = \frac{1}{t} \cdot \mathsf{ICost}_{\mathcal{D}_{\mathsf{M}}}(\Pi_{\mathsf{Matching}})\end{aligned}$$

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 5.1 now follows from Lemma 5.1, lower bound of $\Omega(r^{1-1/p}) = n^{1-O(\varepsilon)}$ for $\mathsf{BHH}^0_{r,p}$ in Corollary 2.1, and the choice of $t = \mathsf{RS}(n)$.

5.2 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. 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 5.2. (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* $\delta < \frac{1}{2}$*:*

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

Theorem 5.2, combined with Proposition 2.1, 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 [36, 4] (and $k = n^{o(1)}$), this immediately proves Part (2) of Theorem 1.3.

In the following, we focus on proving Theorem 5.2. 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 [11] on each induced matching in the hard instance of [9] for finding approximate matchings.

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

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 $p := \lfloor \frac{1}{8\varepsilon} \rfloor$.

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 $M^{\mathsf{RS}}_{j^\star}$ of G_i to $M_{j^\star} := M^{\mathsf{RS}}_{j^\star}|_{x^{(j^\star)}}$.
   (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^{RS}_j of G_i to $M_j := M^{\mathsf{RS}}_j|_{x^{(j)}}$.
   (d) Create the p-clique family of $\mathcal{M}$ on the vertices $R(M^{\mathsf{RS}}_{j^\star})$, 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].

Similar to the lower bound in Section 5.1, let I_{BHH} be the *embedded* $\mathsf{BHH}^0_{r,p}$ instance $(x^{(i)}, \mathcal{M})$. The following claim is analogous to Claim 5.1 in Section 5.1.

Claim 5.2. *For* $G \sim \mathcal{D}_{\mathsf{M}}$*, let:*

$$\mathrm{opt}_{\mathsf{Yes}} := \min_G \Big(\mathrm{opt}(G) \mid I_{\mathsf{BHH}} \textit{ is a } \mathsf{Yes} \textit{ instance} \Big)$$

$$\mathrm{opt}_{\mathsf{No}} := \max_G \Big(\mathrm{opt}(G) \mid I_{\mathsf{BHH}} \textit{ is a } \mathsf{No} \textit{ instance} \Big)$$

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 2.1. 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,

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

Note that, $\mathrm{opt}(G)$ 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 $\mathrm{opt}(H_i) = \frac{3r}{4} - \frac{r}{2p}$. Since the maximum matching size in G is at most the summation of the maximum matching size in each group, we have

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

and the gap between $\mathrm{opt}_{\mathsf{Yes}}$ and $\mathrm{opt}_{\mathsf{No}}$ follows.

Fix any δ-error protocol Π_{Matching} for $\mathsf{Matching}_{n,k,\varepsilon}$ on $\mathcal{D}_{\mathsf{M}}$; Claim 5.2 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 $\mathsf{BHH}^0_{r,p}$ on $\mathcal{D}_{\mathsf{BHH}}$, and then prove that the information cost of Π_{Matching} is t times the information cost of $\mathsf{BHH}^0_{r,p}$.

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 $\mathsf{BHH}^0_{r,p}$ to $\mathsf{Matching}_{n,k,\varepsilon}$:

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 $j^\star \in [t]$, 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 $\mathsf{BHH}^0_{r,p}$ 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 $R(M^{\mathsf{RS}}_{j^\star})$ 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 5.2. In the remainder of this section, we bound the information cost of this protocol.

LEMMA 5.2. $\mathsf{ICost}_{\mathcal{D}_{\mathsf{BHH}}}(\Pi_{\mathsf{BHH}}) \leq \frac{1}{t} \cdot \mathsf{ICost}_{\mathcal{D}_{\mathsf{M}}}(\Pi_{\mathsf{Matching}})$.

Proof. We have,

$$\begin{aligned}\mathsf{ICost}_{\mathcal{D}_{\mathsf{BHH}}}(\Pi_{\mathsf{BHH}}) &= I_{\mathcal{D}_{\mathsf{BHH}}}(\boldsymbol{X}; \boldsymbol{\Pi}_{\mathsf{BHH}}, \boldsymbol{R}) \\ &= I_{\mathcal{D}_{\mathsf{BHH}}}(\boldsymbol{X}; \boldsymbol{\Pi}^R_{\mathsf{BHH}} \mid \boldsymbol{R}) \\ &\text{(by Fact 2.1-(3) and since } I(\boldsymbol{X}; \boldsymbol{R}) = 0 \text{ as } \boldsymbol{X} \perp \boldsymbol{R}) \\ &= I_{\mathcal{D}_{\mathsf{BHH}}}(\boldsymbol{X}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{\sigma}, \boldsymbol{J}, \boldsymbol{R}_{\mathsf{M}}) \\ &= \mathop{\mathbb{E}}_{j}\left[I_{\mathcal{D}_{\mathsf{BHH}}}(\boldsymbol{X}; \boldsymbol{\Pi}_{\mathsf{Matching}} \mid \boldsymbol{\sigma}, \boldsymbol{R}_{\mathsf{M}}, \boldsymbol{J} = j)\right]\end{aligned}$$

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,

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

where the inequality is by conditional sub-additivity of mutual information (Fact 2.1-(4)) since $\boldsymbol{\Pi}^{(i)}_{\mathsf{Matching}} \perp \boldsymbol{\Pi}^{<i}_{\mathsf{Matching}} \mid \boldsymbol{\sigma}, \boldsymbol{Y}_j, \boldsymbol{R}_{\mathsf{M}}, \boldsymbol{J} = j$; this is because conditioned on the given random variables and $\boldsymbol{J} = j$, the message of each player $P^{(i)}$ (i.e., $\boldsymbol{\Pi}^{(i)}_{\mathsf{Matching}}$) 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: $\boldsymbol{\Pi}^{(i)}_{\mathsf{Matching}}$ 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 2.1-(5)),

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

(by chain rule of mutual information (Fact 2.1-(3)))

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

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

Theorem 5.2 now follows from Lemma 5.2, lower bound of $\Omega(r^{1-1/p}) = n^{1-O(\varepsilon)}$ for $\mathsf{BHH}^0_{r,p}$ in Corollary 2.1, and the choice of $t = \Theta(n)$ in the distribution.

6 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 1.1.

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 6.1). We then build on this connection to provide a *meta-algorithm* for matching size estimation (Section 6.2). 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 1.1.

6.1 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., [44]), 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 $\text{opt}(G)$ 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 $\text{opt}(G)$.

LEMMA 6.1. *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 $\text{opt}(G) = \Omega(\alpha)$, then $\text{opt}(G_{\mathsf{smp}}) \leq \frac{3\log n}{\alpha}\cdot\text{opt}(G)$ w.p. $1-o(1)$.*
2. *if $\text{opt}(G) = \Omega(\alpha^2)$, then $\text{opt}(G_{\mathsf{smp}}) \geq \frac{\log^2 n}{2\alpha^2}\cdot\text{opt}(G)$ w.p. $1-o(\frac{1}{\log n})$.*

Note that for $\text{opt}(G) = \Omega(\alpha^2)$, Lemma 6.1 immediately implies that w.p. $1-o(1)$,

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

Proof. [Proof of Lemma 6.1] 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, $\text{opt}(G_{\mathsf{smp}}) \leq |V_{\mathsf{smp}}(M^\star)|$; 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, $\mathbb{E}[X] = p\cdot|V(M^\star)| = p\cdot 2\text{opt}(G) = \frac{2\log n}{\alpha}\cdot\text{opt}(G)$ by the choice of p. Since $\text{opt}(G) = \Omega(\alpha)$ 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, $\text{Var}[X] \leq \mathbb{E}[X]$ and hence by Chebyshev inequality,

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

This implies w.p. $1-o(1)$, $|V_{\mathsf{smp}}(M^\star)| \leq \frac{3\log n}{\alpha}\cdot\text{opt}(G)$, and since $\text{opt}(G_{\mathsf{smp}}) \leq |V_{\mathsf{smp}}(M^\star)|$ 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 $\text{opt}(G_{\mathsf{smp}}) \geq \left|M^{\star}_{\mathsf{smp}}\right|$, it suffices to show that $\left|M^{\star}_{\mathsf{smp}}\right| \geq \frac{\log^2 n}{2\alpha^2} \cdot \text{opt}(G)$. Let Y be a random variable denoting $\left|M^{\star}_{\mathsf{smp}}\right|$.

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 $\mathbb{E}[Y] = p^2 \cdot \text{opt}(G) = \frac{\log^2 n}{\alpha^2} \cdot \text{opt}(G)$. Since by assumption in Part (2), $\text{opt}(G) = \Omega(\alpha^2)$, 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 $\text{Var}[Y] \leq \mathbb{E}[Y]$, and hence by Chebyshev inequality,

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

Therefore, w.p. $1 - o(\frac{1}{\log n})$, $\left|M^{\star}_{\mathsf{smp}}\right| \geq \frac{\log^2 n}{2\alpha^2} \cdot \text{opt}(G)$, proving Part (2).

6.2 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 6.1. (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* $\text{opt}(G) \geq k$, *outputs* No *if* $\text{opt}(G) \leq \gamma \cdot k$, *and otherwise is allowed to output either* Yes *or* No.

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

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

We should remark right away that if $\text{opt}(G) = \Omega(\alpha^2)$, 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 6.1 essentially guarantees that $\text{opt}(G^{\alpha}) \in [\frac{\text{opt}(G)}{\alpha^2}, \frac{\text{opt}(G)}{\alpha}]$). Therefore, running tester for $O(\log \alpha)$ different values (step 1 in the algorithm) is only for the case where $\text{opt}(G) \leq \alpha^2$.

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 $\text{opt}(G) = \widetilde{O}(1)$.
- If all β passes, then all testers return Yes, which means the maximum matching sizes are all large: this is for the case $\text{opt}(G) > \alpha^2$.
- Finally, if some β pass and some β fail, then we are in the case $\text{opt}(G) \in [\widetilde{O}(1), \alpha^2]$.

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

Lemma 6.2. *For any $\alpha \geq \log n$, Algorithm 1 outputs an $O(\alpha)$-approximation of* $\text{opt}(G)$ *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 $\widetilde{\text{opt}}_{\log n}$) is a γ-approximation of $\text{opt}(G^{\log n})$. Furthermore, note that

for $\beta = \log n$, the subsampling probability is $\frac{\log n}{\beta} = 1$, and hence, $G^{\log n} = G$. Therefore, $\widetilde{\text{opt}}_{\log n}$ is a also γ-approximation of $\text{opt}(G)$.

In the following, we analyze the other two cases: (i) all β pass (which would be the case where $\text{opt}(G)$ is large) and (ii) some β pass and some β fail (which will be the case where $\text{opt}(G)$ 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 6.1. *For any β where $\beta^2 \leq \text{opt}(G)$, β passes w.p. $1 - o(\frac{1}{\log n})$.*

Proof. By Lemma 6.1 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, $\mathsf{Tester}_\gamma(G^\beta, \frac{\log^2 n}{2})$ outputs Yes (and hence β passes).

CLAIM 6.2. *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 6.1 Part (1), when $\text{opt}(G) \geq \frac{\beta}{2}$, w.p. $1 - o(1)$,

$$\text{opt}(G^\beta) \leq \frac{3\log n}{\beta} \cdot \text{opt}(G) \leq \frac{3\log n}{\beta} \cdot \beta$$

(for sufficiently large n)

$$= 3\log n < \frac{\gamma \log^2 n}{2}$$

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

With Claim 6.1 and Claim 6.2, the correctness of case (ii) follows immediately.

LEMMA 6.3. *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 6.1, 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 6.2, 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 6.4. *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 $\mathsf{Tester}_\gamma(G^\alpha, \frac{n\log^2 n}{\alpha^2})$ 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 6.1). 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 $\text{ALG} \leq \text{opt}(G)$, and then show that $\text{ALG} \geq \frac{\text{opt}(G)}{O(\alpha)}$.

To see that $\text{ALG} = \max\left\{\alpha, \frac{\alpha}{\log^2 n} \cdot \widetilde{\text{opt}}_\alpha\right\} \leq \text{opt}(G)$, 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 6.2, this β fails w.p. $1 - o(1)$ (which contradicts to the fact that all β pass). Therefore, $\alpha \leq \text{opt}(G)$, and we only need to show that $\frac{\alpha}{\log^2 n} \cdot \widetilde{\text{opt}}_\alpha \leq \text{opt}(G)$. As pointed out above, $\text{opt}(G^\alpha) \geq \widetilde{\text{opt}}_\alpha$. Hence, w.h.p.,

$$\begin{aligned}
\frac{\alpha}{\log^2 n} \cdot \widetilde{\text{opt}}_\alpha &\leq \frac{\alpha}{\log^2 n} \cdot \text{opt}(G^\alpha) \\
\text{(By Lemma 6.1 Part (1))} \quad &\leq \frac{\alpha}{\log^2 n} \cdot \frac{3\log n}{\alpha} \cdot \text{opt}(G) \\
&\leq \text{opt}(G)
\end{aligned}$$

proving $\text{ALG} \leq \text{opt}(G)$.

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

$$\text{ALG} \geq \alpha > \frac{\text{opt}(G)}{\alpha}.$$

Therefore, we only need to consider $\text{opt}(G) \geq \alpha^2$. 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 $\widetilde{\text{opt}}_\alpha := \frac{\gamma n \log^2 n}{\alpha^2}$, and hence

$$\text{ALG} \geq \frac{\alpha}{\log^2 n} \cdot \widetilde{\text{opt}}_\alpha = \frac{\alpha}{\log^2 n} \cdot \frac{\gamma n \log^2 n}{\alpha^2}$$
$$= \frac{\gamma n}{\alpha} \geq \frac{\gamma \cdot \text{opt}(G)}{\alpha} = \frac{\text{opt}(G)}{O(\alpha)}$$

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

$$\text{ALG} \geq \frac{\alpha}{\log^2 n} \cdot \widetilde{\text{opt}}_\alpha \geq \frac{\alpha}{\log^2 n} \cdot \gamma \cdot \text{opt}(G^\alpha)$$

(Lemma 6.1 Part (2))

$$\geq \frac{\gamma\alpha}{\log^2 n} \cdot \frac{\log^2 n}{2\alpha^2} \cdot \text{opt}(G) = \frac{\text{opt}(G)}{O(\alpha)}$$

Therefore, $\text{ALG} = \frac{\text{opt}}{O(\alpha)}$ for all $\text{opt}(G) \geq \alpha$, which completes the proof.

6.3 Implementing Matching Size Testers in Graph Streams

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

Claim 6.3. *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.

Proof. [Proof of Theorem 1.1, Part (1)] Suppose Algorithm 1 returns a $c \cdot \alpha$-approximation (Lemma 6.2; 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 $\text{opt}(G)$. 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 6.3.

By Lemma 6.2, Algorithm 1 returns a $c \cdot \widehat{\alpha}(= \alpha)$-approximation of $\text{opt}(G)$ 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 $\widetilde{O}(\max\{\frac{n}{\alpha^2}, 1\}) = \widetilde{O}(\frac{n}{\alpha^2})$ (recall that $\alpha \leq \sqrt{n}$). Therefore, by Claim 6.3, the space requirement is $\widetilde{O}(\frac{n}{\alpha^2})$.

For implementing a matching size tester in dynamic streams, we use the following result from [9, 13].

Lemma 6.5. ([9, 13]) *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 6.5 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 $\Omega(\text{opt})$ (hence tester outputs No, along with the matching size).

Proof. [Proof of Theorem 1.1, Part (2)] Suppose Algorithm 1 returns a $c \cdot \alpha$-approximation (Lemma 6.2; 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 6.5. Since Algorithm 1 only invokes Tester_γ for $O(\log n)$ times, by Lemma 6.5, w.h.p., no Tester_γ fails.

Now by Lemma 6.2, Algorithm 1 outputs an $c \cdot \widehat{\alpha}(= \alpha)$-approximation of $\text{opt}(G)$. On the other hand, Algorithm 1 invokes $\mathsf{Tester}_\gamma(*, k)$ for $O(\log \alpha)$ times where the largest k used is $\widetilde{O}(\max\{\frac{n}{\alpha^2}, 1\}) = \widetilde{O}(\frac{n}{\alpha^2})$ (recall that $\alpha \leq \sqrt{n}$). Therefore, by Lemma 6.5, the space requirement is $\widetilde{O}(\frac{n^2}{\alpha^4})$.

Acknowledgements We thank Michael Kapralov for many helpful discussions. We are also thankful to the anonymous reviewers of SODA for many valuable comments.

References

[1] Kook Jin Ahn and Sudipto Guha. Linear programming in the semi-streaming model with application to the maximum matching problem. *Inf. Comput.*, 222:59–79, 2013.

[2] Kook Jin Ahn and Sudipto Guha. 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*, pages 202–211, 2015.

[3] Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. 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*, pages 5–14, 2012.

[4] Yuqing Ai, Wei Hu, Yi Li, and David P. Woodruff. New characterizations in turnstile streams with applications. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 20:1–20:22, 2016.

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

[6] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In *STOC*, pages 20–29. ACM, 1996.

[7] Noga Alon, Ankur Moitra, and Benny Sudakov. 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*, pages 1079–1090, 2012.

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

[9] Sepehr Assadi, Sanjeev Khanna, Yang Li, and Grigory Yaroslavtsev. 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*, pages 1345–1364, 2016.

[10] Yitzhak Birk, Nathan Linial, and Roy Meshulam. On the uniform-traffic capacity of single-hop interconnections employing shared directional multichannels. *IEEE Transactions on Information Theory*, 39(1):186–191, 1993.

[11] Marc Bury and Chris Schwiegelshohn. Sublinear estimation of weighted matchings in dynamic data streams. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages 263–274, 2015.

[12] Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, and Andrew Chi-Chih Yao. 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*, pages 270–278, 2001.

[13] Rajesh Chitnis, Graham Cormode, Hossein Esfandiari, MohammadTaghi Hajiaghayi, Andrew McGregor, Morteza Monemizadeh, and Sofya Vorotnikova. 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*, pages 1326–1344, 2016.

[14] Rajesh Hemant Chitnis, Graham Cormode, Mohammad Taghi Hajiaghayi, and Morteza Monemizadeh. 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*, pages 1234–1251, 2015.

[15] Kenneth L. Clarkson and David P. Woodruff. 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*, pages 205–214, 2009.

[16] Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.

[17] Michael Crouch and Daniel S. Stubbs. 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*, pages 96–104, 2014.

[18] Sebastian Eggert, Lasse Kliemann, and Anand Srivastav. Bipartite graph matchings in the semi-streaming model. In *Algorithms - ESA 2009, 17th Annual European Symposium, Copenhagen, Denmark, September 7-9, 2009. Proceedings*, pages 492–503, 2009.

[19] Leah Epstein, Asaf Levin, Julián Mestre, and Danny Segev. Improved approximation guarantees for weighted matching in the semi-streaming model. *SIAM J. Discrete Math.*, 25(3):1251–1265, 2011.

[20] Hossein Esfandiari, Mohammad Taghi Hajiaghayi, Vahid Liaghat, Morteza Monemizadeh, and Krzysztof Onak. 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*, pages 1217–1233, 2015.

[21] Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. On graph problems in a semi-streaming model. *Theor. Comput. Sci.*, 348(2-3):207–216, 2005.

[22] Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 474–483, 2002.

[23] Jacob Fox. A new proof of the graph removal lemma. *Annals of Mathematics*, 174(1):561–579, 2011.

[24] Jacob Fox, Hao Huang, and Benny Sudakov. On graphs decomposable into induced matchings of linear sizes. *arXiv preprint arXiv:1512.07852*, 2015.

[25] Dmitry Gavinsky, Julia Kempe, Iordanis Kerenidis, Ran Raz, and Ronald de Wolf. Exponential separations for one-way quantum communication complexity, with applications to cryptography. *STOC*, pages 516–525, 2007.

[26] Ashish Goel, Michael Kapralov, and Sanjeev Khanna. On the communication and streaming complexity of maximum bipartite matching. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 468–485. SIAM, 2012.

[27] WT Gowers. Some unsolved problems in additive/combinatorial number theory. *preprint*, 2001.

[28] Venkatesan Guruswami and Krzysztof Onak. Super-

linear 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*, pages 287–298, 2013.

[29] Johan Håstad and Avi Wigderson. Simple analysis of graph tests for linearity and PCP. *Random Struct. Algorithms*, 22(2):139–160, 2003.

[30] Zengfeng Huang, Bozidar Radunovic, Milan Vojnovic, and Qin Zhang. 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*, pages 460–473, 2015.

[31] Michael Kapralov. 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*, pages 1679–1697, 2013.

[32] Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. 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*, pages 734–751, 2014.

[33] Christian Konrad. Maximum matching in turnstile streams. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages 840–852, 2015.

[34] Christian Konrad, Frédéric Magniez, and Claire Mathieu. 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*, pages 231–242, 2012.

[35] Yi Li, Huy L. Nguyen, and David P. Woodruff. 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*, pages 1562–1581, 2014.

[36] Yi Li, Huy L. Nguyen, and David P. Woodruff. 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*, pages 174–183, 2014.

[37] Yi Li, Xiaoming Sun, Chengu Wang, and David P. Woodruff. 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*, pages 499–513, 2014.

[38] Yi Li and David P. Woodruff. 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*, pages 726–739, 2016.

[39] Yi Li and David P. Woodruff. 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*, pages 39:1–39:11, 2016.

[40] L. Lovász and D. Plummer. *Matching Theory*. AMS Chelsea Publishing Series. American Mathematical Soc., 2009.

[41] Andrew McGregor. 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*, pages 170–181, 2005.

[42] Andrew McGregor. Graph stream algorithms: a survey. *SIGMOD Record*, 43(1):9–20, 2014.

[43] Andrew McGregor and Sofya Vorotnikova. Planar matching in streams revisited. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, September 7-9, 2016, Paris, France*, pages 17:1–17:12, 2016.

[44] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[45] Jeff M. Phillips, Elad Verbin, and Qin Zhang. 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*, pages 486–501, 2012.

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

[47] Terence Tao and Van H Vu. *Additive combinatorics*, volume 105. Cambridge University Press, 2006.

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

[49] Elad Verbin and Wei Yu. 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*, pages 11–25, 2011.

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