Electronic Colloquium on Computational Complexity, Report No. 161 (2022)

Towards Multi-Pass Streaming Lower Bounds for Optimal Approximation of Max-Cut

Lijie Chen* Gillat Kol† Dmitry Paramonov‡
Raghuvansh R. Saxena§ Zhao Song¶ Huacheng Yu‖

Abstract

We consider the Max-Cut problem, asking how much space is needed by a *streaming* algorithm in order to estimate the value of the maximum cut in a graph. This problem has been extensively studied over the last decade, and we now have a near-optimal lower bound for *one-pass* streaming algorithms, showing that they require linear space to guarantee a better-than-2 approximation [KKS15, KK19]. This result relies on a lower bound for the *cycle-finding* problem, showing that it is hard for a *one-pass* streaming algorithm to find a cycle in a union of matchings.

The end-goal of our research is to prove a similar lower bound for *multi-pass* streaming algorithms that guarantee a better-than-2 approximation for Max-Cut, a highly challenging open problem. In this paper, we take a significant step in this direction, showing that even $o(\log n)$-pass streaming algorithms need $n^{\Omega(1)}$ space to solve the cycle-finding problem. Our proof is quite involved, dividing the cycles in the graph into "short" and "long" cycles, and using tailor-made lower bound techniques to handle each case.

*UC Berkeley.
†Princeton University.
‡Princeton University.
§Microsoft Research.
¶Adobe Research.
‖Princeton University.

ISSN 1433-8092

Contents

1 Introduction

How well can the value of the maximum cut (Max-Cut) in a graph be approximated with a polynomial time algorithm? This question was studied for decades, culminating in the celebrated Goemans-Williamson algorithm [GW95] that gives a 1.138 approximation, that was later shown to be optimal under the Unique Games Conjecture [KKMO07]. The Max-Cut question has also been of special interest to the streaming community [sub], and after extensive research efforts, the space complexity of *one-pass streaming algorithms* for Max-Cut is now well understood.

A recent effort by the streaming community is to devise lower bounds against *multi-pass* algorithms. This paper is a part of this effort, with the end goal of showing that streaming algorithms that compute a better-than-2 approximation of Max-Cut require at least $n^{\Omega(1)}$ space, *even if* $\omega(1)$ *passes are allowed*. Note that a 2-approximation is trivial, as a random cut contains at least half of the edges in the graph. However, such a lower bound is likely to be very challenging as it would subsume technically complex lower bounds in the streaming literature (surveyed below). In this paper, we take a significant step towards this goal and give a lower bound for an associated search problem.

$(1+\epsilon)$-approximation and the BHM problem. The *Boolean Hidden Matching* (BHM) is a popular two-party communication problem [BYJK04, GKK+07]. Here, Alice's input is a uniformly random cut over n vertices, and Bob's input is obtained by sampling a uniformly random matching and dropping all the edges that do not cross Alice's cut in the "yes" case[1], and dropping each edge independently with probability half in the "no" case. The goal of the parties is to determine which is the case. In their influential work, [GKK+07] showed a lower bound saying that any *one-way* protocol that solves the BHM problem must have Alice sending at least $\Omega(\sqrt{n})$ bits to Bob.

The seminal work of Verbin and Yu [VY11] uses this lower bound, together with novel "gadget-based" reductions, to prove lower bounds on the space required by graph streaming algorithms. More reductions were discovered by subsequent works [BS15, LW16, AKL17, BCK+18, EHL+18, HP19], including reductions from the Max-Cut problem [KK15, KKS15]. These lower bounds have also been extended to multi-pass streaming algorithms by recent works [AKSY20, AV21]. However, the gadget based reductions in this line of work only rule out small constant factor approximations for Max-Cut by streaming algorithms.

2-approximation and the distribution $\mathcal{G}$. Kapralov, Khanna, and Sudan [KKS15] (and subsequent works [KKSV17, KK19]) devised an improved reduction that can also rule out any streaming algorithm guaranteeing any approximation factor better-than-2. The best way to understand this result is to view their graph as a union of many matchings, with each

[1] As Alice's cut is uniformly random, this means that Bob drops half of his edges in expectation.

matching resembling an instance of the BHM problem[2].

In more detail, let $\mathcal{G} = \mathcal{G}_{n,T}$ be the distribution over n-vertex graphs whose edge set is a union of T matchings, selected independently and uniformly at random. Let $\mathcal{G}^Y$ be the "yes" distribution obtained by sampling a graph from $\mathcal{G}$, then sampling a uniformly random cut and deleting all the edges that are not in the cut. Let $\mathcal{G}^N$ be the "no" distribution obtained by sampling a graph from $\mathcal{G}$ and randomly deleting each edge with probability $\frac{1}{2}$. Observe that by construction, graphs in the support of $\mathcal{G}^Y$ have a cut that consists of all the edges, and that, as T increases, the maximum cut of graphs in the support of $\mathcal{G}^N$ has roughly half of the edges (with high probability). Therefore, a lower bound on algorithms distinguishing between these two distributions is also a lower bound on getting a better-than-2 approximation of Max-Cut.

As in other works in the streaming literature that give lower bounds for Max-Cut [AKSY20, AV21], this is done through the following perspective of a *cycle problem* (see also [CKKP21]): Graphs in the support of $\mathcal{G}^Y$ are bipartite, and therefore have no odd cycles, whereas graphs in the support of $\mathcal{G}^N$ have many (short) odd cycles with high probability. In this perspective, [KKS15] show that:

Theorem 1.1 ([KKS15], Informal)**.** *Any one-pass streaming algorithm that decides if an input graph has an odd cycle, under the promise that the graph was either sampled from $\mathcal{G}^Y$ or $\mathcal{G}^N$, must use $\Omega(\sqrt{n})$ space.*

At a very high level, to prove Theorem 1.1, [KKS15] use the BHM lower bound[3] to argue that even if the streaming algorithm knew the sampled cut, any given matching cannot help the algorithm distinguish between the two cases, and a hybrid argument over all matchings then yields the desired lower bound.

The cycle-finding problem. As Theorem 1.1 gives a lower bound for a *decision* problem, it also trivially implies a lower bound for the associated *search* problem of finding an odd cycle in a graph sampled from $\mathcal{G}^N$. Going back to the Max-Cut problem, this corresponds to finding a cycle-based *certificate* for proving that the graph has a small maximum cut. The proof of [KKS15] even shows the following slightly stronger search lower bound:

Theorem 1.2 ([KKS15], Informal)**.** *Any one-pass streaming algorithm that outputs a cycle[4] in a graph sampled from $\mathcal{G}$ with constant probability must use $\Omega(\sqrt{n})$ space.*

[2]In their original work, [KKS15] worked with sparse random graphs instead of matchings. This was done to show a lower bound for randomly-ordered streams but is not crucial for our purposes.

[3]The BHM lower bound is a one-way communication lower bound, but it is well known that such lower bounds imply streaming lower bounds.

[4]Ruling out algorithms outputting any cycle (not necessarily odd), as is done by [KKS15] and by our work (see Theorem 1.3), is not a huge overkill, at least if one wants to generalize the lower bound to only marginally more general constraint satisfaction problems, such as Max-2XOR. Recall that Max-2XOR is the same as Max-Cut except that each edge has a 0/1-label and the goal is to output a cut with as many 1 edges and as few 0 edges as possible.

1.1 Our Result

Theorems 1.1 and 1.2 above are restricted to one-pass streaming algorithms. This is because they crucially rely on the hardness of the BHM problem, and the BHM problem can be solved using only $\mathcal{O}(\log n)$ bits of communication if Bob is allowed to send one of his edges to Alice. Our main result in this paper is removing this restriction and showing a multi-pass analogue of Theorem 1.2. Getting a similar analogue of Theorem 1.1 would mean getting a lower bound against multi-pass streaming algorithms computing a better-than-2 approximation of Max-Cut, and is an outstanding problem that we hope to see resolved soon.

Theorem 1.3 (Main, see formal statement as Theorem 4.3)**.** *Any $o(\log n)$-pass streaming algorithm that outputs a cycle in a graph sampled from $\mathcal{G}$ with constant probability must use $n^{\Omega(1)}$ space.*

We mention that [AKSY20, AV21] implicitly show theorems akin to Theorem 1.3, proving that multi-pass streaming algorithms cannot find a cycle in the input graph, albeit with a different distribution $\mathcal{G}$. Specifically, [AKSY20, AV21] worked with a distribution over graphs that (roughly) are a union of vertex disjoint cycles of length k, for some constant length $k > 0$. However, requiring that the cycles are vertex disjoint implies that there is always a cut that contains all but one of the edges in every cycle, and therefore [AKSY20, AV21] only obtain a lower bound against algorithms that (roughly) guarantee a strong $\left(1 + \frac{1}{k}\right)$-approximation to Max-Cut.

In fact, the argument above applies to any distribution where the cycles are "more-or-less-disjoint", and the only way to get the optimal 2-approximation lower bound from a theorem like Theorem 1.3, is to work with a distribution where the cycles are unstructured and entangled with one another (like the distribution $\mathcal{G}$ used by [KKS15, KKSV17, KK19] and also used in this work). While proving Theorem 1.3 using such an entangled distribution is crucial, it is also the main source of hardness, as analyzing such distributions poses several challenges, as explained next.

1.2 Our Techniques

We now provide a very brief overview of our techniques. For a detailed exposition, see Section 2.

Recall that unlike previous lower bounds on multi-pass algorithms for cycle problems [AKSY20, AV21], our Theorem 1.3 imposes very little structure on the graph instances that it works with. This makes our proof very different from the proofs found in these works. Specifically, as [AKSY20, AV21] deal with graphs that are a union of vertex disjoint cycles of the same length, algorithms in their settings, roughly speaking, have only one way to output a cycle, which is to pick a start vertex and chase one of its edges till it loops back. This makes such algorithms amenable to "*pointer chasing* techniques", roughly saying that a small space algorithm can only advance by one edge in one pass, and implying that the number of passes must be comparable to the length of the cycles.

In contrast, our Theorem 1.3 shows a multi-pass lower bound for an extremely unstructured instance, with no guarantee on the length or the structure of the cycles it contains. In particular, our instances are likely to have extremely short cycles, even cycles of length 2, and an algorithm may just try to find one such short cycle in the graph and output it. As we allow the streaming algorithm to have up to $o(\log n)$ passes, it has enough passes to explore this short cycle and standard pointer chasing techniques will not apply.

To deal with such algorithms, we divide the cycles in the graph into *short cycles*, with length at most $\kappa \log n$, for some $\kappa > 0$, and *long cycles* that are longer than $\kappa \log n$. We then separately show that there is no low-space, $o(\log n)$-pass streaming algorithm that outputs a short cycle, and that there is no such algorithm that outputs a long cycle, and apply a union bound. Both of these proofs actually classify the respective cycles further to various *patterns*, where the pattern for a cycle says which of the T matchings each of its edges come from, and bound the probability of outputting a cycle following a given pattern (see Definition 3.2).

Short cycles. For a short cycle with a fixed pattern, we are able to show that finding such a cycle is equivalent to solving *set-intersection*, and use the set-intersection lower bounds from the literature [BFS86, Raz90, KS92]. As an example, consider algorithms that output cycles following the pattern $(1, 2)$, *i.e.*, cycles with two edges, where the first edge comes from the first matching and the second edge comes from the second matching. Observe that an algorithm can only output such a cycle if it finds an edge that is contained in the intersection of the first and second matchings, and thus, we can reduce to an instance of set-intersection. Of course, complications arise when dealing with other, more complicated patterns, but this underlying idea remains valid.

Long cycles. For a long cycle with a fixed pattern, we use the pointer-chasing techniques described above, carefully adapting them to our setting. The key difference is that in standard pointer chasing, the graph is a union of vertex disjoint paths and the goal is to chase one of these paths given its start vertex. For us, the various cycles that follow a pattern may not be vertex disjoint, and, moreover, it is okay to output any one of these cycles. For the former, we prove combinatorial lemmas showing that it is possible to carefully select a large set of vertex disjoint cycles with high probability, and *embed a pointer chasing instance* on these cycles. For the latter, we use a *direct product* result to show that outputting any specific such cycle is only possible with negligible probability, and then use a union bound over all cycles.

1.3 Additional Related Work

Boolean Hidden Matching. The BHM problem [BYJK04, GKK+07], was originally studied in order to get a separation between quantum and classical communication complexity. The communication complexity of BHM is $\Theta(\sqrt{n})$ in the one-way setting [GKK+07], and $\Theta(\log n)$ in the two-way and quantum settings. BHM is truly versatile and

has found surprising applications in various settings, such as distribution testing [AMN19], distributed computing [FGO17], property testing [BLWZ19], and sketching [KKP18].

Streaming algorithms. Streaming algorithms, first studied by [AMS99], is now one of the main algorithmic models used to study large graphs that arise in modern day applications [FKM+04, FKM+09]. Several graph problems are being actively pursued in this context, making it impossible to list all of them (see [McG14] for a survey). These include streaming algorithms for finding maximum matchings [McG05, GKK12, Kap13, AKLY16, AG18, ABB+19, GKMS19, AKSY20, AV21], shortest paths and reachability [FKM+09, GO16, BKKL17, AR20, CKP+21a], subgraph counting [BYKS02, BOV13, BKKL17, MVV16, CJ17, BC17, KMPV19], and random walks [SGP11, Jin19, CKP+21b].

Beyond Max-Cut. General *constraint satisfaction problems* (including and beyond Max-Cut) have also received a lot of attention in the streaming model. These include extending and generalizing the [KKS15] work to lower bounds for more problems [GT19, CGV20, CGSV21, CGS+21, BHP+21, SSV21] and also finding novel and interesting upper bounds [GVV17, BDV18].

1.4 Acknowledgments

The authors would like to thank Sepehr Assadi for useful discussions.

2 Overview of Techniques

2.1 Setup and high-level overview

As already discussed in Section 1.2, finding short cycles and long cycles is hard due to totally different reasons. Roughly speaking, finding short cycles is hard because we need to find an intersection between two matchings, and finding long cycles is hard because we have to chase many edges[5]. Below we discuss our approach in more detail. We begin with some notation and observations.

2.1.1 Two Cases: Short Simple Cycles and Long Simple Path

Let $n \in \mathbb{N}_{\geq 1}$ be the number of vertices and $T \in \mathbb{N}_{\geq 1}$ be a large constant. We will always assume that n is even. Let $\kappa \in (0, 1)$ be a small constant. We say a cycle is short, if it has at most $\kappa \cdot \log n$ many edges, and is long otherwise. We first make two simple but useful observations below:

[5]We mention that a combination of set intersection and pointer chasing lower bounds was also used by the (otherwise unrelated) works [BGGS19, GS20].

1. Any cycle contains a *simple cycle* (*i.e.*, a cycle that visits any vertex at most once), meaning that if an algorithm finds a cycle, it also finds a simple cycle. So it suffices to upper bound the probability of finding a simple cycle.

2. If an algorithm finds a long simple cycle with more than $\kappa \cdot \log n$ many edges, it also finds a *simple path* of length $\kappa \cdot \log n$ (*i.e.*, a path that visits any vertex at most once). This means that, to upper bound the probability of finding a long simple cycle, it suffices to upper bound the probability of finding a simple path of length $\kappa \cdot \log n$.

Based on the above observations, given a low-pass streaming algorithm $\mathbb{A}$, it suffices to upper bound the probability of the following two events:

1. $\mathbb{A}$ finds a simple cycle of length at most $\kappa \cdot \log n$.

2. $\mathbb{A}$ finds a simple path of length exactly $\kappa \cdot \log n$.

2.1.2 Patterns

Next, we introduce the concept of *patterns*, which helps us to find some structure in the graph distribution $\mathcal{G}_{n,T}$. Let $G = ([n], M_1 \circ M_2 \circ \cdots \circ M_T)$ be a graph[6] that is a union of T perfect matchings $M_1, \cdots, M_T$. A pattern $\vec{\tau} \in [T]^L$ for some integer $L \in \mathbb{N}$ tells you how to chase a path from a fixed starting point u: first traverse the edge incident on u in matching M_{τ_1} to reach vertex u_1, then traverse the edge incident on u_1 in matching M_{τ_2} to reach vertex u_2, and so on, until the last matching M_{τ_L}.

We use $\mathsf{Path}(G, u, \vec{\tau})$ to denote the resulting path (see Definition 3.2 for a formal definition). We first note that for the path to be simple, we must have $\tau_j \neq \tau_{j+1}$ for every $j \in [L-1]$, and for the cycle to be simple, we should additionally ensure that $\tau_1 \neq \tau_L$. We call such patterns *valid path patterns* and *valid cycle patterns*, respectively; see Section 3.2 for formal definitions.

Now, to upper bound the probability that the algorithm $\mathbb{A}$ finds a simple cycle of length at most $\kappa \cdot \log n$, we will instead upper bound the probability of $\mathbb{A}$ finding a simple cycle with a fixed pattern $\vec{\tau} \in [T]^L$ for some $L \leq \kappa \log n$, and apply a union bound over all $O(T^{\kappa \cdot \log n}) = O(n^{\kappa \log T})$ many such patterns. Similarly we will upper bound the probability of $\mathbb{A}$ finding a simple path with a fixed pattern $\vec{\tau} \in [T]^{\kappa \cdot \log n}$, and apply a union bound.

2.1.3 Our Strategy: Hiding a Hard Search Problem in Cycle/Path-Finding

For both short cycles and long paths, we show that finding a cycle/path is hard by embedding a hard communication problem $\mathcal{P}$ into an instance of the cycle/path finding problem. This means that any algorithm that outputs a cycle/path can also be used to solve the hard communication problem $\mathcal{P}$, giving us a lower bound.

[6] For two elements or vectors u, v, we use $u \circ v$ to denote the concatenation of u and v.

2.2 Lower Bounds for Short Cycles via Set-Intersection

2.2.1 Toy Case: $\vec{\tau} = (1, 2)$

Let us first consider the simple case when $\vec{\tau} = (1, 2)$. As explained in Section 1.2, in this case, we wish to find an edge common to two uniformly random perfect matchings M_1 and M_2. By a standard connection between streaming algorithms and communication protocols, it suffices to prove that any short two-party protocol where Alice's and Bob's inputs are uniformly random perfect matchings M_1 and M_2, cannot output an edge in the intersection of M_1 and M_2 with probability more than $n^{-\Omega(1)}$.

Starting point: set-intersection lower bounds with low success probability. Viewing the set of all potential edges as the universe $U = \binom{[n]}{2}$,[7] the aforementioned problem is exactly *set-intersection*, in which two players are given two sets $S, T \subseteq U$ with size $|S| = |T| = n/2$, and wish to find an element in $S \cap T$. There are however two complications: (1) standard lower bounds for distributional set-intersection start from the uniform distribution over all possible (S, T) such that $|S \cap T| = 1$ and $|S| = |T| = n/2$, while in our case, Alice and Bob are holding independently chosen subsets such that the edges in those subsets form a matching, and (2) we will need a lower bound showing that the success probability is at most $n^{-\Omega(1)}$, instead of "merely" a small constant.

The second difficulty is easier to resolve, and we do it by invoking the strong lower bound on set intersection in [AR20]. Specifically, [AR20] say that, for all $k, N \in \mathbb{N}$ such that $N \geq 4k$, if Alice and Bob's input are uniformly distributed over all possible pairs of sets $S, T \in \binom{[N]}{k}$ such that $|S \cap T| = 1$, any protocol with communication complexity at most $k^{1/3}$ can only find the unique element in $S \cap T$ with probability at most $O(k^{-1/3})$ (see Corollary 5.7). Thus, if we have $k = n^{\Omega(1)}$, then we will have the required success probability bound. Henceforth, we will use $\mathcal{D}_{N,k}$ to denote the distribution above and use $\mathsf{SI}_{N,k}$ to denote instances sampled from this distribution.

Embedding set-intersection into cycle finding. We still have to resolve the first difficulty. We set $N = \binom{n}{2}$ so that the universe corresponds to the set of all possible edges. Our idea is to *embed* an instance of $\mathsf{SI}_{N,k}$ into the problem of finding intersection of random matchings as follows: Alice and Bob get the input $(S, T) \leftarrow \mathcal{D}_{N,k}$, each of them first interprets their set S (resp. T) as a set of edges from U, and then extends this set into a matching uniformly at random[8]. Alice and Bob can then run the algorithm $\mathbb{A}$ that finds a cycle in the graph $G = ([n], M_1 \circ M_2)$ with the pattern $(1, 2)$ to find a collision between the generated matchings.

[7] For a set S and an integer $k \in \mathbb{N}_{\geq 1}$, we use $\binom{S}{k}$ to denote the collection of all k-size subsets of S.

[8] That is, if interpreting S gives Alice the edges $e_1, \cdots, e_k$ and V' is the set of vertices that are not touched by any of these edges, then, Alice adds a uniformly random perfect matching on V' to her input. Bob does the same.

However, the reduction above has a couple of problems. Recall that we want to show that an algorithm that finds a cycle with pattern $(1,2)$ over the distribution $\mathcal{G}_{n,T}$ can be used to solve set intersection over the distribution $\mathcal{D}_{N,k}$. The first problem is that starting from the distribution $\mathcal{D}_{N,k}$, the reduction above will not generate the distribution $\mathcal{G}_{n,T}$. One obvious issue is that with inputs (S,T) drawn from $\mathcal{D}_{N,k}$, the set S (resp. T) may not correspond to a set of vertex-disjoint edges, and then there is no way to extend them into perfect matchings. The solution is to notice that if we set $k = n^{1/3}$, then the probability of this bad event happening is low (in fact, $n^{-\Omega(1)}$), and we can condition on it not happening.

However, even with this fix, the distribution generated by the reduction is very far from the target distribution $\mathcal{G}_{n,T}$. In particular, in the above reduction, since $|S \cap T| = 1$, the resulting two matchings M_1 and M_2 always have at least one common edge, while in $\mathcal{G}_{n,T}$, the matchings M_1 and M_2 are disjoint (with constant probability). Nevertheless, these differences between $\mathcal{G}_{n,T}$ and the distribution generated are always in the "right" direction, in the sense that algorithms that find cycles over $\mathcal{G}_{n,T}$ will also find a cycle over the distribution generated by the reduction. For example, as the generated distribution does not have graphs where the matchings M_1 and M_2 are disjoint, a cycle finding algorithm would not fail because there are no $(1,2)$-pattern cycles in the graph. See Claim 5.10 for details.

The second problem in the reduction is that, given $\mathbb{A}$'s solution to the cycle finding problem, it is unclear if one can obtain a solution to the set intersection instance. This is because, if there are many $(1,2)$-pattern cycles in the generated graph $G = (n, M_1 \circ M_2)$, there is no guarantee that the cycle found by the cycle-finding algorithm $\mathbb{A}$ corresponds to the solution of the embedded $\mathsf{SI}_{N,k}$ instance. Rather, it could just be a cycle formed by edges that are added by Alice and Bob during in the reduction. Our key observation here is that the cycle-finding algorithm $\mathbb{A}$ does not know whether a $(1,2)$-pattern cycle in the generated graph G is "genuine" (*i.e.*, coming from the embedded $\mathsf{SI}_{N,k}$ instance) or "fake" (*i.e.*, involving edges that are added later by Alice and Bob during the reduction). So, intuitively, the worst thing $\mathbb{A}$ can do is output a random $(1,2)$-pattern cycle in the graph. We then prove a concentration inequality saying that for any short pattern $\vec{\tau}$ of length at most $\kappa \cdot \log n$, with probability $1 - n^{-\omega(1)}$, a graph $G \leftarrow \mathcal{G}_{n,T}$ has at most $\log^3 n$ cycles with pattern $\vec{\tau}$; see Lemma 5.9 for more details. This helps us show that the probability of $\mathbb{A}$ finding a $(1,2)$-pattern cycle is at most $k^{-1/3} \cdot \log^3 n \leq n^{-\Omega(1)}$, by our choice of k.

2.2.2 Generalization to Arbitrary Patterns $\vec{\tau}$

Now we discuss how to embed set-intersection into cycle finding with a fixed pattern $\vec{\tau}$, for a general $\vec{\tau}$ of length $L \leq \kappa \cdot \log n$, which is much more challenging. For simplicity, we assume that $\vec{\tau}$ has at least one occurrence of 1 (*i.e.*, the matching M_1 is involved in the cycle).[9]

First, since we wish that the found cycle with length L corresponds directly to the common element in the starting $\mathsf{SI}_{N,k}$ problem, we should set $N = n^L$ so that the universe $[N]$

[9]See the proof of Lemma 4.1 for the general case.

corresponds to all possible length-L cycles[10].

Second, we still wish to use the standard connection between streaming algorithms and communication protocols, and give all matchings $M_1, \cdots, M_T$ in the graph $G = ([n], M_1 \circ \cdots \circ M_T)$ to two players Alice and Bob. We will simply give M_1 to Alice, and the rest $M_{\geq 2}$ to Bob.

Our key idea is that, given a sequence $\vec{u} \in [n]^L$, if for every $\ell \in [L]$ such that $\tau_\ell = 1$, Alice adds $(u_\ell, u_{\ell+1})$ to her set of edges (we use u_{L+1} to denote u_1, for notational convenience), and for every $\ell \in [L]$ such that $\tau_\ell \in \{2, \cdots, T\}$, Bob adds $(u_\ell, u_{\ell+1})$ to his set of edges. Then, in the combined graph of Alice and Bob, $\vec{u}$ is a cycle with pattern $\vec{\tau}$, and thus can potentially be detected by the cycle-finding algorithm $\mathbb{A}$.

Our reduction from $\mathsf{SI}_{N,k}$ over distribution $\mathcal{D}_{N,k}$ to finding a pattern-$\vec{\tau}$ cycle then works as follows:

1. Alice and Bob get $S, T \in \binom{[N]}{k}$ distributed according to $\mathcal{D}_{N,k}$. Alice (resp. Bob) interprets S (resp. T) as k vectors $\vec{s}^{(1)}, \vec{s}^{(2)}, \cdots, \vec{s}^{(k)}$ (resp. $\vec{t}^{(1)}, \vec{t}^{(2)}, \cdots, \vec{t}^{(k)}$) from $[n]^L$.
2. Initially, Alice lets M_1 be the empty set, and Bob lets $M_2, \cdots, M_T$ be empty sets too.
3. For every $\vec{s}^{(i)}$, for every $\ell \in [L]$ such that $\tau_\ell = 1$, Alice adds $(s_\ell^{(i)}, s_{\ell+1}^{(i)})$ to M_1.
4. For every $\vec{t}^{(i)}$, for every $\ell \in [L]$ such that $\tau_\ell \neq 1$, Bob adds $(t_\ell^{(i)}, t_{\ell+1}^{(i)})$ to M_{τ_ℓ}.
5. At the end, Alice extends M_1 to a perfect matching uniformly at random, and Bob extends $M_2, \cdots, M_T$ to perfect matchings uniformly at random as well.

Crucially, by previous discussions, the common element $S \cap T$ is going to be a cycle with pattern $\vec{\tau}$ in the joint graph $G = ([n], M_1 \circ M_{\geq 2})$. So this reduction makes sense. Still, the three issues in the toy example occur here as well. First, it is possible that for some S, T, Step (2) and (3) above do not generate valid partial matchings. However, by setting k small enough (say $k = n^{1/3}$), we can show that the probability of this event happening is small, and we can condition on this event not happening.

Second, the resulting graph G may contain more than one cycle with pattern $\vec{\tau}$. Similarly to the toy case, we make use of the observation that $\mathbb{A}$ does not know which $\vec{\tau}$-pattern cycle is genuine or fake, and derive the lower bound using the concentration inequality we proved regarding the number of $\vec{\tau}$-pattern cycles in a graph $G \leftarrow \mathcal{G}_{n,T}$. See Section 5.1.2 for more details of the proof and how the third issue is addressed in a way similar to the toy example.

2.3 Lower Bounds for Paths via Pointer-Chasing

In the long simple path case, for a fixed pattern $\vec{\tau} \in [T]^L$ where $L = \kappa \cdot \log n$, we will prove that an $o(\log n)$-pass streaming algorithm $\mathbb{A}$ cannot find a *simple path* of pattern $\vec{\tau}$ with probability at least $n^{-\omega(1)}$.

[10]Here we interpret a length-L cycle as a sequence of L vertices from $[n]$, and we allow non-simple cycles and even self-loops.

As already discussed in Section 1.2, the reason finding a long simple path in $G \leftarrow \mathcal{G}_{n,T}$ is difficult is that this requires the streaming algorithm $\mathbb{A}$ to *chase* from a vertex $u \in [n]$ for L steps, following the pattern $\vec{\tau}$, and pointer chasing is well-known to be hard for low-pass streaming algorithms. Hence, our strategy here is to reduce a certain pointer chasing instance into the problem of finding a simple path with pattern $\vec{\tau}$.

To simplify the discussions, we will focus on the case that $T = 3$ and $\vec{\tau}$ is a repetition of $(1, 2, 3)$. Again, we wish to study a related communication problem, in which there are three players P_1, P_2, P_3 such that P_i holds the matching M_i, and their goal is to output a simple path with pattern $\vec{\tau}$ in the joint graph $G = ([n], M_1 \circ M_2 \circ M_3)$.

For simplicity, we assume that L is a multiple of 3. Our starting point is the following search version of the pointer chasing problem that is defined over a graph with $L + 1$ layers $V_1, \cdots, V_{L+1}$ each consisting of m vertices, and L matchings $W_1, \cdots, W_L$ such that for every $i \in [L]$, W_i is a perfect bipartite matching between the layers V_i and V_{i+1}: Player P_i gets matchings $W_i, W_{i+3}, W_{i+6}, \cdots$ as input, and their goal is to output a length-L path from *any vertex* in V_1 to any vertex in V_{L+1}. Since $L = \Omega(\log n)$, using direct product theorem for communication protocols, we are able to prove that communication protocols with $o(\log n)$ round complexity and m^{ε} communication complexity for some constant $\varepsilon \in (0, 1)$ can solve this problem with probability at most $m^{-\omega(1)}$. We will set $m = n^{\gamma}$ for some small constant $\gamma \in (0, 1)$, so that a success probability upper bounded by $m^{-\omega(1)} = n^{-\omega(1)}$ is good enough. For simplicity, we let $V_i = \{(i-1) \cdot m + 1, \cdots, i \cdot m\}$. Then the whole vertex set V is $[(L + 1) \cdot m]$.

We can then embed the pointer-chasing instance above into a path-finding problem as follows:

1. Using public randomness, P_1, P_2, P_3 jointly sample a random injective function $\phi \colon [(L+1) \cdot m] \to [n]$. For $i \in [3]$, player P_i also initializes M_i be the empty set.

2. For each player P_i, for every edge (u, v) from its input $W_i, W_{i+3}, W_{i+6}, \cdots$, P_i adds $(\phi(u), \phi(v))$ into M_i.

3. Finally, each player P_i extends M_i into a perfect matching uniformly at random.

First, we observe that the above procedure gives a valid partial matching for each player P_i after Step (2), so that they can always extend their inputs into perfect matchings at Step (3). Second, one can see that for every $u \in V_1 = [m]$, the generated graph contains a simple $\vec{\tau}$-pattern path starting from $\phi(u)$. Hence, our hope is to show that if the streaming algorithm $\mathbb{A}$ finds a simple $\vec{\tau}$-pattern path in the resulting graph G, then, with a reasonable probability, the path starts from vertices in the set $\{\phi(u) : u \in [m]\}$. This means that any streaming algorithm $\mathbb{A}$ that finds a simple $\vec{\tau}$-pattern path in $G \leftarrow \mathcal{G}_{n,T}$ with probability $n^{-O(1)}$ contradicts the hardness of the pointer-chasing problem, as required for our lower bound.

The key observation, again, is that $\mathbb{A}$ does not know which $\vec{\tau}$-pattern path in G is genuine (*i.e.*, coming from the pointer-chasing problem via the mapping ϕ) or fake (*i.e.*, involving

vertices added by P_i's in the Step (3) of the reduction). The actual analysis, however, is much trickier than the short cycle case, and we have to prove a sophisticated concentration inequality regarding the number of possible embeddings of a pointer chasing instance in a graph $G \in \mathcal{G}_{n,T}$. This involves a lot of additional technical work, that we defer to Section 6.

3 Preliminaries

3.1 Notation

We use $\mathbb{N}$ to denote all non-negative integers, and $\mathbb{N}_{\geq 1}$ to denote all positive integers. We also use $2\mathbb{N}$ (resp. $2\mathbb{N}_{\geq 1}$) to denote all non-negative (resp. positive) even integers. For two elements or vectors u, v, we use $u \circ v$ to denote the concatenation of u and v.

We often use bold font letters (*e.g.*, $\boldsymbol{X}$) to denote random variables, and calligraphic font letters (*e.g.*, $\mathcal{X}$) to denote distributions. For two random variables $\boldsymbol{X}$ and $\boldsymbol{Y}$, and for $Y \in \mathrm{supp}(\boldsymbol{Y})$, we use $(X|\boldsymbol{Y} = Y)$ to denote $\boldsymbol{X}$ conditioned on $\boldsymbol{Y} = Y$. For two lists a and b, we use $a \circ b$ to denote their concatenation.

For two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ on set $\mathcal{X}$ and $\mathcal{Y}$ respectively, we use $\mathcal{D}_1 \otimes \mathcal{D}_2$ to denote their product distribution over $\mathcal{X} \times \mathcal{Y}$, and $\|\mathcal{D}_1 - \mathcal{D}_2\|_{\mathrm{TV}}$ to denote the total variation distance between them.

Let $n \in \mathbb{N}_{\geq 1}$. We use $[n]$ to denote the set $\{1, \cdots, n\}$. We often use symbols such as $\vec{x}$ to emphasize that $\vec{x}$ is a *vector*, and we often use x_i to denote its i-th entry and $|\vec{x}|$ to denote the length of $\vec{x}$. For a set S and $m \in \mathbb{N}$, we use $\binom{S}{m}$ to denote all size-m subsets of S.

3.2 Graphs

Formally, a labeled undirected graph G is a tuple $(V, \vec{E}, \vec{\mu})$, where V is the set of vertices, $\vec{E} = ((u_i, v_i))_{i \in [m]}$ is a list of edges such that $u_i, v_i \in V$, and $\vec{\mu} = (\mu_1, \cdots, \mu_m)$ is a list of labels. In the streaming model, it is presented as a stream of tuples (u_i, v_i, μ_i), from $i = 1$ to $i = m$. Similarly, an undirected graph G is a pair $(V, \vec{E})$.

An ordered matching $\vec{M}$ on a set of vertices V is a list of vertex-disjoint undirected edges. The size of a matching $\vec{M}$ is simply the number of edges in it. For a vertex set V of even size, we use $\mathcal{M}_V$ to denote the uniform distribution over all ordered matchings on V with size $|V|/2$.

Definition 3.1. *Let $n \in 2\mathbb{N}_{\geq 1}$ and $T \in \mathbb{N}_{\geq 1}$. We define $\mathcal{G}_{n,T}$ as the following distribution on undirected graphs:*

- *We set $V = [n]$.*
- *For each $i \in [T]$, we draw $\vec{M}^i \leftarrow \mathcal{M}_V$, independently across all i. Then we set $\vec{E} = \vec{M}^1 \circ \vec{M}^2 \circ \cdots \circ \vec{M}^T$.*

A (undirected) path $\vec{w}$ is a list of edges $e_1, \cdots, e_k$ such that $e_i = (u_i, v_i)$ and for all $i \in [k-1]$, $v_i = u_{i+1}$. (Note that since we are working with undirected graphs, we can swap u_i and v_i if necessary.) Similarly, a (undirected) cycle is a path $\vec{w}$ that additionally satisfies $v_k = u_1$. We say a path or a cycle is simple, if no vertices except for the starting vertex u_1 is visited twice.

Definition 3.2. *Let $n \in 2\mathbb{N}_{\geq 1}$ and $T \in \mathbb{N}_{\geq 1}$ and $G = ([n], \vec{E}) \in \mathrm{supp}(\mathcal{G}_{n,T})$. Let $\vec{E} = \vec{M}^1 \circ \vec{M}^2 \circ \cdots \circ \vec{M}^T$ where $\vec{M}^i$ is the i-th matching according to Definition 3.1. Let $v_{\mathsf{s}} \in [n]$, $L \in \mathbb{N}$, and $\vec{\tau} \in [T]^L$. We define $\mathsf{Path}(G, v_{\mathsf{s}}, \vec{\tau})$ as the output of the following algorithm:*

1. *Let $v_0 = v_{\mathsf{s}}$ and $\vec{w}$ be an empty list.*
2. *For i from 1 to L:*
   (a) *Let e be the unique edge in the matching $\vec{M}^{\tau_i}$ that is adjacent to the vertex v_{i-1}. If no such e exists, return $\perp$.*
   (b) *Add e to the end of $\vec{w}$. Let v_i be the endpoint of e other than v_{i-1}.*
3. *Return $\vec{w}$.*

In other words, $\mathsf{Path}(G, v_{\mathsf{s}}, \vec{\tau})$ (if exists) is the unique path in G that starts from v_{s} and follows the *pattern* $\vec{\tau}$. We say $\vec{\tau}$ is a *valid path pattern*, if for every $j \in [|\vec{\tau}| - 1]$, it holds that $\tau_j \neq \tau_{j+1}$. We also say $\vec{\tau}$ is a *valid cycle pattern*, if it is a valid path pattern and also $\tau_{|\vec{\tau}|} \neq \tau_1$.

4 Lower Bounds for Finding Cycles

Our lower bound for finding cycle will follow from the following two lemmas.

Notation. Fix a graph $G \in \mathrm{supp}(\mathcal{G}_{n,T})$ and $\vec{\tau} \in [T]^L$. We let $\mathbb{C}_{\vec{\tau}}(G)$ be the set of simple cycles in G with pattern $\vec{\tau}$, and $\mathbb{C}(G)$ be the set of all simple cycles in G. We also let $\mathbb{L}_{\vec{\tau}}(G)$ be the set of simple paths in G with pattern $\vec{\tau}$.

Lemma 4.1 (Lower bound for finding a short cycle with a fixed pattern $\vec{\tau}$)**.** *There exist $\varepsilon, \delta \in (0,1)$ such that for all $T \in \mathbb{N}_{\geq 1}$ and for all sufficiently large $n \in 2\mathbb{N}_{\geq 1}$ the following holds: For all $L \in [\log n]$, valid cycle pattern $\vec{\tau} \in [T]^L$, and n^{ε}-pass n^{ε}-space streaming algorithms $\mathbb{A}$, we have*

$$\Pr_{G \leftarrow \mathcal{G}_{n,T}} \left[\mathbb{A}(G) \in \mathbb{C}_{\vec{\tau}}(G) \right] \leq n^{-\delta}. \tag{1}$$

Lemma 4.2 (Lower bound for finding a long path with a fixed pattern $\vec{\tau}$)**.** *There exist $\varepsilon, \delta, \gamma_0 \in (0,1)$ such that for all $T \in \mathbb{N}_{\geq 1}$ and for all sufficiently large $n \in 2\mathbb{N}_{\geq 1}$ the following*

holds: For all $L \in [\gamma_0 \cdot \log n]$, $p \leq (L-15)/4T$, valid path pattern $\vec{\tau} \in [T]^L$, and p-pass n^{ε}-space streaming algorithms $\mathbb{A}$, we have

$$\Pr_{G \leftarrow \mathcal{G}_{n,T}}\left[\mathbb{A}(G) \in \mathbb{L}_{\vec{\tau}}(G)\right] \leq n^{3-\delta L/p}. \tag{2}$$

Theorem 4.3. *There exist $\varepsilon, \delta \in (0,1)$ such that for all $T \in \mathbb{N}_{\geq 1}$ and for all sufficiently large $n \in 2\mathbb{N}_{\geq 1}$ the following holds: for all $o(\log n)$-pass n^{ε}-space streaming algorithms $\mathbb{A}$, we have*

$$\Pr_{G \leftarrow \mathcal{G}_{n,T}}\left[\mathbb{A}(G) \in \mathbb{C}(G)\right] \leq n^{-\delta}. \tag{3}$$

Proof. Let ε be the minimum of the ε constants from Lemma 4.1 and Lemma 4.2, and δ_1 be the minimum of the δ constants from Lemma 4.1 and Lemma 4.2.

Fix an $o(\log n)$-pass n^{ε}-space streaming algorithm $\mathbb{A}$. Let $L = \kappa \log n$ for a constant $\kappa \in (0,1)$ to be chosen later. For notational convenience, we also use $\mathbb{C}_{\leq L}(G)$ and $\mathbb{C}_{>L}(G)$ to denote the set of simple cycles in G with length at most L and greater than L, respectively. Then we have

$$\Pr_{G \leftarrow \mathcal{G}_{n,T}}\left[\mathbb{A}(G) \in \mathbb{C}(G)\right] \leq \Pr_{G \leftarrow \mathcal{G}_{n,T}}\left[\mathbb{A}(G) \in \mathbb{C}_{\leq L}(G)\right] + \Pr_{G \leftarrow \mathcal{G}_{n,T}}\left[\mathbb{A}(G) \in \mathbb{C}_{>L}(G)\right].$$

First, by Lemma 4.1, we have

$$\begin{aligned}
\Pr_{G \leftarrow \mathcal{G}_{n,T}}\left[\mathbb{A}(G) \in \mathbb{C}_{\leq L}(G)\right] &\leq \sum_{\substack{\vec{\tau} \in [T]^L \\ \vec{\tau} \text{ is a valid cycle pattern}}} \Pr_{G \leftarrow \mathcal{G}_{n,T}}\left[\mathbb{A}(G) \in \mathbb{C}_{\vec{\tau}}(G)\right] \\
&\leq T^L \cdot n^{-\delta_1} \\
&\leq 2^{\log T \cdot \kappa \cdot \log n} \cdot n^{-\delta_1} \\
&\leq n^{\kappa \cdot \log T - \delta_1}.
\end{aligned}$$

We now set $\kappa = \min(\frac{\delta_1}{2 \log T}, \gamma_0)$ so that we have

$$\Pr_{G \leftarrow \mathcal{G}_{n,T}}\left[\mathbb{A}(G) \in \mathbb{C}_{\leq L}(G)\right] \leq n^{-\delta_1/2}. \tag{4}$$

Now, we let $\mathbb{L}_{=L}(G)$ denote the set of simple paths in G with length exactly L. Given the algorithm $\mathbb{A}$, we construct another algorithm $\widetilde{\mathbb{A}}$ who outputs the first L edges in the cycle found by $\mathbb{A}$ (if $\mathbb{A}$ does not output a valid cycle, $\widetilde{\mathbb{A}}$ just outputs $\perp$). Now, we note that $\widetilde{\mathbb{A}}$ has the same pass and space complexity as $\mathbb{A}$, and whenever $\mathbb{A}$ finds a cycle in $\mathbb{C}_{>L}(G)$, $\widetilde{\mathbb{A}}$ outputs a path in $\mathbb{L}_{=L}(G)$.

Hence, by Lemma 4.2, we have

$$\Pr_{G \leftarrow \mathcal{G}_{n,T}}\left[\mathbb{A}(G) \in \mathbb{C}_{>L}(G)\right] \leq \Pr_{G \leftarrow \mathcal{G}_{n,T}}\left[\widetilde{\mathbb{A}}(G) \in \mathbb{L}_{=L}(G)\right]$$

$$\leq \sum_{\substack{\vec{\tau}\in[T]^L \\ \vec{\tau} \text{ is a valid path pattern}}} \Pr_{G\leftarrow\mathcal{G}_{n,T}}\left[\widetilde{\mathbb{A}}(G)\in\mathbb{L}_{\vec{\tau}}(G)\right]$$
$$\leq T^L\cdot n^{3-\delta_1\cdot L/o(\log n)}$$
$$\leq n^{-\omega(1)}. \tag{5}$$

Putting (4) and (5) together and set $\delta=\delta_1/3$ completes the proof. □

5 Lower Bounds for Finding a Short Cycle

Recall that $\mathbb{C}_{\vec{\tau}}(G)$ is the set of simple cycles in G with the pattern $\vec{\tau}$. In this section we prove Lemma 4.1, which is restated below.

Reminder of Lemma 4.1. *There exist $\varepsilon,\delta\in(0,1)$ such that for all $T\in\mathbb{N}_{\geq 1}$ and for all sufficiently large $n\in 2\mathbb{N}_{\geq 1}$ the following holds: For all $L\in[\log n]$, valid cycle pattern $\vec{\tau}\in[T]^L$, and n^ε-pass n^ε-space streaming algorithms $\mathbb{A}$, we have*

$$\Pr_{G\leftarrow\mathcal{G}_{n,T}}\left[\mathbb{A}(G)\in\mathbb{C}_{\vec{\tau}}(G)\right]\leq n^{-\delta}. \tag{6}$$

To prove Lemma 4.1, we will indeed prove a stronger communication complexity lower bound first, and then show Lemma 4.1 as an easy corollary. We first define the following communication problem.

Definition 5.1 (The $\mathsf{Cycle\text{-}Search}_{n,T,\vec{\tau}}$ problem)**.** *Let $n\in 2\mathbb{N}_{\geq 1}$, $T,L\in\mathbb{N}_{\geq 1}$ and $\vec{\tau}\in[T]^L$ such that $\vec{\tau}$ is a valid cycle pattern. In the $\mathsf{Cycle\text{-}Search}_{n,T,\vec{\tau}}$ problem, Alice holds a perfect matching M^1 on $[n]$ and Bob holds $T-1$ perfect matchings $M^2,\ldots,M^T$ on $[n]$, their goal is to output a simple cycle in $G=([n],M^1\circ M^2\circ\ldots\circ M^T)$ with pattern $\vec{\tau}$.*

Slightly abusing notation, we can also view $\mathcal{G}_{n,T}$ (a distribution over graphs that is the union of T uniform random perfect matchings) as an input distribution to $\mathsf{Cycle\text{-}Search}_{n,T,\vec{\tau}}$. Given $G=([n],\vec{M}^1,\vec{M}^2,\ldots,\vec{M}^T)\leftarrow\mathcal{G}_{n,T}$, we first convert these ordered matchings $\vec{M}^i$ into their unordered counterparts M^i, and then give M^1 to Alice, and $M^2,\ldots,M^T$ to Bob. We will write $(M^1,M^{\geq 2})\leftarrow\mathcal{G}_{n,T}$ to denote that Alice's input M^1 and Bob's input $M^{\geq 2}=(M^2,\ldots,M^T)$ are generated as above.

We will prove the following lower bound for $\mathsf{Cycle\text{-}Search}_{n,T,\vec{\tau}}$.

Lemma 5.2. *There exists $\varepsilon,\delta\in(0,1)$, such that for all $T\in\mathbb{N}_{\geq 1}$, for all sufficiently large $n\in 2\mathbb{N}_{\geq 1}$, $L\in[\log n]$, valid cycle pattern $\vec{\tau}\in[T]^L$ such that $\vec{\tau}$ contains at least one occurrence of 1, and for all two-party communication protocols Π with communication*

complexity at most n^{ε},

$$\Pr_{\substack{(M^1,M^{\geq 2})\leftarrow\mathcal{G}_{n,T}\\ G=([n],M^1\circ M^{\geq 2})}}\left[\Pi(M^1,M^{\geq 2})\in\mathbb{C}_{\vec{\tau}}(G)\right]\leq n^{-\delta}. \tag{7}$$

Before proving Lemma 5.2, we show that Lemma 4.1 follows immediately from Lemma 5.2.

Proof of Lemma 4.1. Let ε,δ be the constants guaranteed by Lemma 5.2. Let $\mu\in[T]$ be an index that occurs at least once in $\vec{\tau}$. We consider the following communication problem:

- A list of unordered matchings $M^1,\ldots,M^T$ are drawn from $\mathcal{G}_{n,T}$.
- Alice is given the matching M^{μ}, and Bob is given the rest of the matchings, $M^1,\ldots,M^{\mu-1},M^{\mu+1},\ldots,M^T$, denoted by $M^{-\mu}$.
- The goal is output a cycle from $\mathbb{C}_{\vec{\tau}}(G)$, where $G=([n],M^1,M^2,\ldots,M^T)$.

Since all matchings in $\mathcal{G}_{n,T}$ are independently and identically distributed (*i.e.*, they are distributed uniformly over all perfect matchings on $[n]$), Lemma 5.2 implies that[11] for all two-party communication protocols Π with communication complexity at most n^{ε},

$$\Pr_{\substack{(M^{\mu},M^{-\mu})\leftarrow\mathcal{G}_{n,T}\\ G=([n],M^{\mu}\circ M^{-\mu})}}\left[\Pi(M^{\mu},M^{-\mu})\in\mathbb{C}_{\vec{\tau}}(G)\right]\leq n^{-\delta}. \tag{8}$$

Since Alice and Bob can simulate a p-pass, s-space complexity streaming algorithm $\mathbb{A}$ over the input stream $(M^1,M^2,\ldots,M^T)$ by a two-party protocol with $ps\cdot T$ communication complexity[12], it follows that no $n^{\varepsilon/3}$-pass, $n^{\varepsilon/3}$-space algorithm $\mathbb{A}$ violates (6), since otherwise there is a communication protocol Π with $n^{2\varepsilon/3}\cdot T<n^{\varepsilon}$ communication complexity that violates (8), contradicting Lemma 5.2.

□

5.1 Proof of Lemma 5.2

In the rest of this section we will prove Lemma 5.2 by a reduction from a sparse version of the well-known set-intersection problem. We first introduce this problem together with some notation.

[11] An algorithm for this new communication problem where $\mu\neq 1$ can be used to solve the special case that $\mu=1$ (corresponding to Lemma 5.2) simply by swapping matchings M^1 with M^{μ}. We note that here we crucially used the fact that Lemma 5.2 applies to *communication protocols* instead of streaming algorithms over the input stream $(M^1,\ldots,M^T)$.

[12] The factor of T comes from the fact that in each pass, we may alternate at most T times between matchings from Alice and from Bob.

5.1.1 A Sparse Variant of the Set-Intersection Problem

Given a matrix $M \in \Sigma^{n\times m}$ and a row index $i \in [n]$, we use $\mathsf{row}(M, i)$ to denote its i-th row vector (*i.e.*, $\mathsf{row}(M, i) = (M_{i,1}, M_{i,2}, \ldots, M_{i,m})$). We will need the following communication problem.

Definition 5.3 (The $\mathsf{Sparse\text{-}SI}_{n,k,L}$ problem)**.** *Let $n, k, L \in \mathbb{N}_{\geq 1}$. In the $\mathit{Sparse\text{-}SI}_{n,k,L}$ problem, Alice and Bob get matrices $M^{\mathsf{A}}, M^{\mathsf{B}} \in [n]^{k\times L}$, respectively. The goal for them is to find a common row of M^{A} and M^{B} (i.e., a vector $X \in [n]^L$ such that $\mathsf{row}(M^{\mathsf{A}}, i) = \mathsf{row}(M^{\mathsf{B}}, j) = X$ for some $i, j \in [k]$).*

We will consider the following hard distribution for $\mathsf{Sparse\text{-}SI}_{n,k,L}$.

Definition 5.4. *Let $n, k, L \in \mathbb{N}_{\geq 1}$. We define the following distribution $\mathcal{D}^{\mathsf{S\text{-}SI}}_{n,k,L}$ for the problem $\mathit{Sparse\text{-}SI}_{n,k,L}$: Alice and Bob's inputs are uniformly distributed over all $(M^{\mathsf{A}}, M^{\mathsf{B}}) \in [n]^{k\times L} \times [n]^{k\times L}$ satisfying the following two conditions:*

1. *There exist two indices $i, j \in [k]$ such that $\mathsf{row}(M^{\mathsf{A}}, i) = \mathsf{row}(M^{\mathsf{B}}, j)$.*
2. *Let M be the $(2k-1)\times L$ matrix obtained by first removing the j-th row from M^{B} and then concatenating M^{A} and M^{B} (i.e., putting M^{A} on the top of M^{B}). All entries in M are distinct.*

We will need the following lower bound for $\mathsf{Sparse\text{-}SI}_{n,k,L}$ over $\mathcal{D}^{\mathsf{S\text{-}SI}}_{n,k,L}$.

Lemma 5.5. *Let $n, k, L \in \mathbb{N}_{\geq 1}$ such that $k = n^{1/3}$ and $L \in [\log n]$. No two-party communication protocol with complexity at most $n^{0.1}$ solves $\mathit{Sparse\text{-}SI}_{n,k,L}$ over $\mathcal{D}^{\mathsf{S\text{-}SI}}_{n,k,L}$ with probability more than $1/n^{0.1}$.*

To prove Lemma 5.5, we will use a reduction from the standard set-intersection problem $\mathsf{SI}_{n,k}$. In $\mathsf{SI}_{n,k}$, Alice and Bob get sets $A, B \subseteq [n]$, respectively, such that $|A| = |B| = k$ and their goal is to output an element from $A \cap B$.

Let $\mathcal{D}^{\mathsf{SI}}_{n,k}$ be the following distribution over inputs to $\mathsf{SI}_{n,k}$: Alice and Bob's inputs are drawn at uniformly random from all pairs $A, B \subseteq [n]$ such that $|A| = |B| = k$ and $|A \cap B| = 1$.

We need the following theorem well known result.

Theorem 5.6 ([Raz90, KS92, BM13, AR20])**.** *For every $\varepsilon \in (0, 1)$ and $k \in \mathbb{N}_{\geq 1}$, any protocol solving $\mathsf{SI}_{4k,k}$ with probability ε over $\mathcal{D}^{\mathsf{SI}}_{4k,k}$ requires communication complexity at least $\Omega(\varepsilon^2 \cdot k)$.*

The lower bound of Theorem 5.6 only applies to solving $\mathsf{SI}_{4k,k}$, it can be easily generalized to the case of solving $\mathsf{SI}_{n,k}$ for any $n \geq 4k$.

Corollary 5.7. *For every $\varepsilon \in (0, 1)$, $n, k \in \mathbb{N}$ such that $n \geq 4k$, any protocol solving $\mathsf{SI}_{n,k}$ with probability ε over $\mathcal{D}^{\mathsf{SI}}_{n,k}$ requires communication complexity at least $\Omega(\varepsilon^2 \cdot k)$.*

Proof. We will show how to reduce solving $\mathsf{SI}_{4k,k}$ over $\mathcal{D}^{\mathsf{SI}}_{4k,k}$ to solving $\mathsf{SI}_{n,k}$ over $\mathcal{D}^{\mathsf{SI}}_{n,k}$, while preserving the success probability.

Suppose Alice and Bob get sets $A, B \subseteq [4k]$, they use public randomness to sample an injective mapping $\pi\colon [4k] \to [n]$, and construct their new inputs

$$A' = \{\pi(u) : u \in A\} \quad \text{and} \quad B' = \{\pi(u) : u \in B\}.$$

One can see that when (A, B) are drawn from $\mathcal{D}^{\mathsf{SI}}_{4k,k}$, (A', B') are distributed according to $\mathcal{D}^{\mathsf{SI}}_{n,k}$, and given the intersection $u \in A' \cap B'$, we know that $\pi^{-1}(u)$ is the intersection of A and B, which completes the proof. □

Now we are ready to prove Lemma 5.5.

Proof of Lemma 5.5. Let $N = n^L$ and ϕ be a bijection from $[N]$ to $[n]^L$.

Let $\widetilde{\mathcal{D}}$ be the uniform distribution over all $(M^{\mathsf{A}}, M^{\mathsf{B}}) \in [n]^{k\times L} \times [n]^{k\times L}$ satisfying the following two conditions:

1. There exist two indices $i, j \in [k]$ such that $\mathsf{row}(M^{\mathsf{A}}, i) = \mathsf{row}(M^{\mathsf{B}}, j)$.
2. Let M be the $(2k-1) \times L$ matrix obtained by first removing the j-th row from M^{B} and then concatenating M^{A} and M^{B} (*i.e.*, putting M^{A} on the top of M^{B}). All rows in M are distinct (*i.e.*, for all $1 \le a < b \le 2k-1$, $\mathsf{row}(M, a) \neq \mathsf{row}(M, b)$).

Let $\mathcal{D} = \mathcal{D}^{\mathsf{S\text{-}SI}}_{n,k,L}$. We note that $\mathcal{D}$ is indeed $\widetilde{\mathcal{D}}$ conditioning on the event that all entries of M are distinct, which happens with probability at least $1 - (2kL)^2/n$ by a union bound. Hence, we have that $\|\mathcal{D} - \widetilde{\mathcal{D}}\|_{\mathrm{TV}} \le (2kL)^2/n \le n^{-0.2}$.

Note that $\mathsf{Sparse\text{-}SI}_{n,k,L}$ over the distribution $\widetilde{\mathcal{D}}$ is indeed $\mathsf{SI}_{N,k}$ in disguise: Alice and Bob can both apply ϕ to each row of their matrices M^{A} and M^{B} to get two sets A' and B', and $A' \cap B'$ corresponds to the common row of M^{A} and M^{B}. By Corollary 5.7, we know that communication protocol with complexity $n^{0.1}$ cannot solve $\mathsf{Sparse\text{-}SI}_{n,k,L}$ with probability more than $n^{-0.11}$ over $\widetilde{\mathcal{D}}$. Hence, since $\|\widetilde{\mathcal{D}} - \mathcal{D}\|_{\mathrm{TV}} \le n^{-0.2}$, it follows that communication protocol with complexity $n^{0.1}$ cannot solve $\mathsf{Sparse\text{-}SI}_{n,k,L}$ with probability more than $n^{-0.11} - n^{-0.2} \le n^{-0.1}$ over $\mathcal{D}$, which completes the proof. □

5.1.2 A Reduction from Sparse-SI$_{n,k,L}$ to Cycle-Search$_{n,T,\vec{\tau}}$

We will use the following reduction from $\mathsf{Sparse\text{-}SI}_{n,k,L}$ to $\mathsf{Cycle\text{-}Search}_{n,T,\vec{\tau}}$. We will assume $\vec{\tau}$ contains at least one occurrence of 1.

Reduction from Sparse-SI$_{n,k,L}$ to Cycle-Search$_{n,T,\vec{\tau}}$: Red-Cyc$(M^{\mathsf{A}}, M^{\mathsf{B}})$

- Alice gets $M^{\mathsf{A}} \in [n]^{k\times L}$ and Bob gets $M^{\mathsf{B}} \in [n]^{k\times L}$.
- Return $\perp$ if $(M^{\mathsf{A}}, M^{\mathsf{B}}) \notin \mathrm{supp}(\mathcal{D}^{\mathsf{S\text{-}SI}}_{n,k,L})$.

- Alice generates M^1 as follows:
  - For every $i \in [k]$ and every $j \in [L]$ such that $\tau_j = 1$, Alice adds the edge $(M^{\mathsf{A}}_{i,j}, M^{\mathsf{A}}_{i,(j \bmod L)+1})$ to M^1.[a]
  - Alice extends M^1 into a perfect matching uniformly at random.
- Similarly, Bob generates $M^2, \ldots, M^T$ as follows:
  - For every $i \in [k]$, every $\mu \in \{2, \ldots, T\}$, and every $j \in [L]$ such that $\tau_j = \mu$, Bob adds the edge $(M^{\mathsf{B}}_{i,j}, M^{\mathsf{B}}_{i,(j \bmod L)+1})$ to M^μ.
  - For every $\mu \in \{2, \ldots, T\}$, Bob extends M^μ into a perfect matching uniformly at random.

[a] We write $(j \bmod L) + 1$ in the subscript as we index starting from 1 instead of 0.

Notation. For $n, L \in \mathbb{N}_{\geq 1}$, we let $\mathcal{S}_{n,L}$ be the set of all the vectors from $[n]^L$ whose entries are all distinct.

Let $n, k, L \in \mathbb{N}_{\geq 1}$, $i, j \in [k]$, and $X \in \mathcal{S}_{n,L}$. We define $\mathcal{D}^{\mathsf{S\text{-}SI}}_{n,k,L;i,j,X}$ to be the distribution $\mathcal{D}^{\mathsf{S\text{-}SI}}_{n,k,L}$ conditioning on the event that $\mathsf{row}(M^{\mathsf{A}}, i) = \mathsf{row}(M^{\mathsf{B}}, j) = X$.

We then define

$$\mathcal{R}_{n,T,\vec{\tau};i,j,X} := \mathsf{Red\text{-}Cyc}(\mathcal{D}^{\mathsf{S\text{-}SI}}_{n,k,L;i,j,X}),$$

which is the outputted distribution of the reduction $\mathsf{Red\text{-}Cyc}$ where Alice and Bob draw their inputs jointly from $\mathcal{D}^{\mathsf{S\text{-}SI}}_{n,k,L;i,j,X}$.

Let $\mathcal{G}_{n,T,\vec{\tau};X}$ to be the distribution $\mathcal{G}_{n,T}$ conditioning on the event that the graph contains X as a $\vec{\tau}$ pattern cycle. Slightly abusing notation, we also identify a graph $G \in \mathrm{supp}(\mathcal{G}_{n,T})$ by a list of T perfect matchings $M^1, M^2, \ldots, M^T$.[13]

Given $G \in \mathrm{supp}(\mathcal{G}_{n,T})$ and a pattern $\vec{\tau} \in [T]^L$, we define $\#_{\vec{\tau}}(G)$ as the number of simple cycles in G with pattern $\vec{\tau}$ (*i.e.*, $\#_{\vec{\tau}}(G) = |\mathbb{C}_{\vec{\tau}}(G)|$).

We will need the following two lemmas.

Lemma 5.8. *For all $T \in \mathbb{N}_{\geq 1}$, for all sufficiently large $n \in 2\mathbb{N}_{\geq 1}$ the following holds: letting $k = n^{1/3}$, for every $L \in [\log n]$, $\vec{\tau} \in [T]^L$, $X \in \mathcal{S}_{n,L}$ and $i, j \in [k]$, it holds that*

$$\|\mathcal{R}_{n,k,T,\vec{\tau};i,j,X} - \mathcal{G}_{n,T,\vec{\tau};X}\|_{\mathrm{TV}} \leq 1/n^{0.1}.$$

Lemma 5.9. *For all $T \in \mathbb{N}_{\geq 1}$, for all sufficiently large $n \in 2\mathbb{N}_{\geq 1}$ the following holds: for*

[13] We note that since now we are aiming to prove the communication complexity lower bound, the orderings of the edges within individual matchings do not matter, so we (Alice and Bob, indeed) will simply "forget" their orderings.

every $L \in [\log n]$ *and valid cycle pattern* $\vec{\tau} \in [T]^L$*, it holds that*

$$\Pr_{G \leftarrow \mathcal{G}_{n,T}} [\#_{\vec{\tau}}(G) > \log^3 n] \leq n^{-100},$$

and

$$1/2 \leq \mathop{\mathbb{E}}_{G \leftarrow \mathcal{G}_{n,T}} [\#_{\vec{\tau}}(G)] \leq 2.$$

Now we are ready to prove Lemma 5.2.

Proof of Lemma 5.2. Let $\varepsilon, \delta \in (0,1)$ to be specified later. For the sake of contradiction, we will first assume the existence of a communication protocol Π_{cyc} with complexity n^{ε} such that

$$\Pr_{\substack{G \leftarrow \mathcal{G}_{n,T} \\ G=([n], M^1 \circ M^{\geq 2})}} \left[\Pi_{\mathsf{cyc}}(G) \in \mathbb{C}_{\vec{\tau}}(G)\right] > n^{-\delta}, \text{[14]} \tag{9}$$

and then construct another protocol Π_{SI} that contradicts Lemma 5.5. Recall that $k = n^{1/3}$ in Lemma 5.5. Now we specify the protocol Π_{SI}.

The protocol Π_{SI} for $\mathsf{Sparse\text{-}SI}_{n,k,L}$

1. Alice gets $M^{\mathsf{A}} \in [n]^{k \times L}$ and Bob gets $M^{\mathsf{B}} \in [n]^{k \times L}$.
2. Alice and Bob simulate $\mathsf{Red\text{-}Cyc}(M^{\mathsf{A}}, M^{\mathsf{B}})$ to get their new inputs M^1 and $M^{\geq 2}$, respectively. (Note that this step does not require communication, according to $\mathsf{Red\text{-}Cyc}$.)
3. Alice and Bob run Π_{cyc} with inputs being M^1 and $M^{\geq 2}$, respectively.
4. If Π_{cyc} returns a cycle C, Alice and Bob then outputs the vertices in C, in the same order they appear in C.

In the rest of the proof, for simplicity we will use $\mathcal{G}_X$ to denote $\mathcal{G}_{n,T,\vec{\tau},X}$, $\mathcal{R}_X$ to denote $\mathcal{R}_{n,k,T,\vec{\tau},i,j,X}$, and $\mathcal{D}^{\mathsf{S\text{-}SI}}_X$ to denote $\mathcal{D}^{\mathsf{S\text{-}SI}}_{i,j,X}$. Their other parameters in the subscripts $(n, K, L, T, \vec{\tau}, i, j)$ will always be clear from the context.

The success probability p_{suc} of Π_{SI} over $\mathcal{D}^{\mathsf{S\text{-}SI}}$ can be calculated as follows:

$$p_{\mathsf{suc}} = \Pr_{X \leftarrow \mathcal{S}_{n,L}} \Pr_{i,j \leftarrow [n]} \Pr_{(M^{\mathsf{A}}, M^{\mathsf{B}}) \leftarrow \mathcal{D}^{\mathsf{S\text{-}SI}}_{i,j,X}} [\Pi_{\mathsf{SI}}(M^{\mathsf{A}}, M^{\mathsf{B}}) = X].$$

From now on, we will slightly abuse the notation by identify an ordered cycle C with the list of its vertices. (Since we only care about cycles with pattern $\vec{\tau}$, the latter uniquely determines the former.)

[14] For notation convenience, given a graph $G = ([n], M^1 \circ M^{\geq 2})$, we use $\Pi_{\mathsf{cyc}}(G)$ to denote $\Pi_{\mathsf{cyc}}(M^1, M^{\geq 2})$.

We wish to lower bound

$$\begin{aligned}
&\Pr_{X\leftarrow\mathcal{S}_{n,L}}\ \Pr_{(M^{\mathsf{A}},M^{\mathsf{B}})\leftarrow\mathcal{D}^{\mathsf{S\text{-}SI}}_X}[\Pi_{\mathsf{SI}}(M^{\mathsf{A}},M^{\mathsf{B}})=X]\\
=&\Pr_{X\leftarrow\mathcal{S}_{n,L}}\ \Pr_{(M^{1},M^{\geq 2})\leftarrow\mathcal{R}_X}[\Pi_{\mathsf{cyc}}(M^{1},M^{\geq 2})=X]\\
\geq&\Pr_{X\leftarrow\mathcal{S}_{n,L}}\ \Pr_{G\leftarrow\mathcal{G}_X}[\Pi_{\mathsf{cyc}}(G)=X]-n^{-0.1}. && \text{(Lemma 5.8)}
\end{aligned}$$

Next we define $\widetilde{\mathcal{G}}$ as the following distribution: draw $X\leftarrow\mathcal{S}_{n,L}$, $G\leftarrow\mathcal{G}_X$, and then output G. We have

$$\Pr_{X\leftarrow\mathcal{S}_{n,L}}\ \Pr_{G\leftarrow\mathcal{G}_X}[\Pi_{\mathsf{cyc}}(G)=X]=\Pr_{G\leftarrow\widetilde{\mathcal{G}}}\ \Pr_{X\leftarrow\mathbb{C}_{\vec{\tau}}(G)}[\Pi_{\mathsf{cyc}}(G)=X].$$

We need the following claim that helps us to analyze the above quantity.

Claim 5.10. *The following two statements hold:*

1. $\Pr_{G\leftarrow\widetilde{\mathcal{G}}}[\Pi_{\mathsf{cyc}}(G)\in\mathbb{C}_{\vec{\tau}}(G)]\geq n^{-\delta}/2$.
2. $\Pr_{G\leftarrow\widetilde{\mathcal{G}}}[\#_{\vec{\tau}}(G)>\log^3 n]\leq 1/n^{50}$.

Before proving Claim 5.10, we first show it implies our lemma. We have

$$\begin{aligned}
&\Pr_{G\leftarrow\widetilde{\mathcal{G}}}\ \Pr_{X\leftarrow\mathbb{C}_{\vec{\tau}}(G)}[\Pi_{\mathsf{cyc}}(G)=X]\\
\geq&\mathop{\mathbb{E}}_{G\leftarrow\widetilde{\mathcal{G}}}\frac{1}{\#_{\vec{\tau}}(G)}\cdot\mathbb{1}_{\{\Pi_{\mathsf{cyc}}(G)\in\mathbb{C}_{\vec{\tau}}(G)\}}\\
\geq&\mathop{\mathbb{E}}_{G\leftarrow\widetilde{\mathcal{G}}}\frac{1}{\log^3 n}\cdot\mathbb{1}_{\{\Pi_{\mathsf{cyc}}(G)\in\mathbb{C}_{\vec{\tau}}(G)\ \wedge\ \#_{\vec{\tau}}(G)\leq\log^3 n\}}\\
\geq&\frac{1}{\log^3 n}\cdot(n^{-\delta}/2-n^{-50}). && \text{(Claim 5.10)}
\end{aligned}$$

Putting everything together and setting $\delta=0.05$ and $\varepsilon=0.1$, we have

$$p_{\mathsf{suc}}\geq\frac{1}{\log^3 n}\cdot(n^{-\delta}/2-n^{-50})-n^{-0.1}\geq n^{-0.1}.$$

Noting that Π_{SI} has the same communication complexity as Π_{cyc}, we have established that Π_{SI} solves $\mathcal{D}^{\mathsf{S\text{-}SI}}_{n,k,L}$ over $\mathcal{D}^{\mathsf{S\text{-}SI}}_{n,k,L}$ with probability at least $n^{-0.1}$ with communication complexity $n^{0.1}$, contradiction to Lemma 5.5. This completes the proof for the lemma.

Finally, we prove Claim 5.10.

Proof of Claim 5.10. Let $S_{\mathcal{G}}=\mathrm{supp}(\mathcal{G})$. Note that $\widetilde{\mathcal{G}}$'s support is a subset of $S_{\mathcal{G}}$. Fix $G\in S_{\mathcal{G}}$, we note that the probability of G is drawn from $\widetilde{\mathcal{G}}$ is proportional to $\#_{\vec{\tau}}(G)$, so we

have

$$\widetilde{\mathcal{G}}(G) = \frac{\#_{\vec{\tau}}(G)}{\sum_{H \in S_{\mathcal{G}}} \#_{\vec{\tau}}(H)}.$$

Therefore

$$\frac{\widetilde{\mathcal{G}}(G)}{\mathcal{G}(G)} = \frac{\#_{\vec{\tau}}(G)}{\mathbb{E}_{H \in S_{\mathcal{G}}} \#_{\vec{\tau}}(H)}.$$

Applying Lemma 5.9, we have

$$\#_{\vec{\tau}}(G)/2 \leq \frac{\widetilde{\mathcal{G}}(G)}{\mathcal{G}(G)} \leq 2\#_{\vec{\tau}}(G). \tag{10}$$

Now we are ready to prove Item (1).

$$\begin{aligned}
\Pr_{G \leftarrow \widetilde{\mathcal{G}}}[\Pi_{\mathsf{cyc}}(G) \in \mathbb{C}_{\vec{\tau}}(G)] &= \mathop{\mathbb{E}}_{G \leftarrow \mathcal{G}} \frac{\widetilde{\mathcal{G}}(G)}{\mathcal{G}(G)} \cdot \mathbb{1}_{\{\Pi_{\mathsf{cyc}}(G) \in \mathbb{C}_{\vec{\tau}}(G)\}} \\
&\geq \mathop{\mathbb{E}}_{G \leftarrow \mathcal{G}} \#_{\vec{\tau}}(G)/2 \cdot \mathbb{1}_{\{\Pi_{\mathsf{cyc}}(G) \in \mathbb{C}_{\vec{\tau}}(G)\}} && \text{(By (10))} \\
&\geq \frac{1}{2} \cdot \mathop{\mathbb{E}}_{G \leftarrow \mathcal{G}} \mathbb{1}_{\{\Pi(G) \in \mathbb{C}_{\vec{\tau}}(G)\}} \\
&\geq n^{-\delta}/2. && \square
\end{aligned}$$

Next, we prove Item (2).

$$\begin{aligned}
&\Pr_{G \leftarrow \widetilde{\mathcal{G}}}[\#_{\vec{\tau}}(G) > \log^3 n] \\
&= \mathop{\mathbb{E}}_{G \leftarrow \mathcal{G}} \frac{\widetilde{\mathcal{G}}(G)}{\mathcal{G}(G)} \cdot \mathbb{1}_{\{\#_{\vec{\tau}}(G) > \log^3 n\}} \\
&\leq \mathop{\mathbb{E}}_{G \leftarrow \mathcal{G}} 2 \cdot \#_{\vec{\tau}}(G) \cdot \mathbb{1}_{\{\#_{\vec{\tau}}(G) > \log^3 n\}} && \text{(By (10))} \\
&\leq 2n \cdot \mathop{\mathbb{E}}_{G \leftarrow \mathcal{G}} \mathbb{1}_{\{\#_{\vec{\tau}}(G) > \log^3 n\}} && (\#_{\vec{\tau}}(G) \leq n) \\
&\leq n^{-50}. && \text{(Lemma 5.9)}
\end{aligned}$$

$\square$

5.2 Proof of Lemma 5.8

In this section we prove Lemma 5.8, which is restated below.

Reminder of Lemma 5.8. *For all $T \in \mathbb{N}_{\geq 1}$, for all sufficiently large $n \in 2\mathbb{N}_{\geq 1}$ the following holds: letting $k = n^{1/3}$, for every $L \in [\log n]$, $\vec{\tau} \in [T]^L$, $X \in \mathcal{S}_{n,L}$ and $i, j \in [k]$, it holds that*

$$\|\mathcal{R}_{n,k,T,\vec{\tau};i,j,X} - \mathcal{G}_{n,T,\vec{\tau};X}\|_{\mathrm{TV}} \leq 1/n^{0.1}.$$

Let C_n be the number of perfect matchings on an n vertex set (assuming that $n \in 2\mathbb{N}_{\geq 1}$). We need the following fact regarding C_n.

Fact 5.11. *Let $n \in 2\mathbb{N}_{\geq 1}$ be sufficiently large. For every $k \in \mathbb{N}$ such that $k < n/2$, we have*

$$\frac{C_{n-2k}}{C_n} = \prod_{i \in [k]} \frac{1}{(n-2i+1)}.$$

In particular, for every $k \in \mathbb{N}$ such that $k < n^{0.34}$, it holds that

$$n^{-k} \leq \frac{C_{n-2k}}{C_n} \leq n^{-k} \cdot (1 + n^{-0.2}).$$

Proof of Lemma 5.8. For notational convenience, throughout the proof we will use $\mathcal{R}_X$ to denote $\mathcal{R}_{n,k,T,\vec{\tau},i,j,X}$ and $\mathcal{G}_X$ to denote $\mathcal{G}_{n,T,\vec{\tau},X}$.

Let $X_{[i]}$ be the edges in X that belongs to M^i if treating X as a $\vec{\tau}$-pattern cycle[15], we can see that $\mathcal{G}_X$ is the uniform distribution over lists of T perfect matchings $(M^1, M^2, \ldots, M^T)$ such that $X_{[i]} \subseteq M^i$.

We first observe that $\mathcal{R}_X$ can be alternatively described as below. We will also define an auxiliary distribution $\widetilde{\mathcal{R}}_X$ to help the analysis.

Alternative sampling procedures Samp_X and $\widetilde{\mathsf{Samp}}_X$ for $\mathcal{R}_X$ and $\widetilde{\mathcal{R}}_X$, respectively

- Let $\ell = |\{\tau_j = 1 : j \in [L]\}|$.

Sampler Samp_X for $\mathcal{R}_X$ Alice gets $M^{\mathsf{A}} \in [n]^{(k-1)\times 2\ell}$ and Bob gets $M^{\mathsf{B}} \in [n]^{(k-1)\times L}$ from the uniform distribution over all pairs $(M^{\mathsf{A}}, M^{\mathsf{B}})$ such that the union of $M^{\mathsf{A}}, M^{\mathsf{B}}, X$ has distinct entries.

Sampler $\widetilde{\mathsf{Samp}}_X$ for $\widetilde{\mathcal{R}}_X$ Alice gets $M^{\mathsf{A}} \in [n]^{(k-1)\times 2\ell}$ and Bob gets $M^{\mathsf{B}} \in [n]^{(k-1)\times L}$ from the uniform distribution over all such pairs $(M^{\mathsf{A}}, M^{\mathsf{B}})$.

- If the union of $M^{\mathsf{A}}, M^{\mathsf{B}}, X$ does not have distinct entries, then return $\perp$ and terminate. (This is only relevant for $\widetilde{\mathcal{R}}_X$.)
- Alice generates M^1 as follows:
  - Alice first sets $M^1 = X_{[1]}$.
  - For every $i \in [k-1]$ and $j \in [\ell]$, Alice adds the edge $(M^{\mathsf{A}}_{i,2j-1}, M^{\mathsf{A}}_{i,2j})$ to M^1.
  - Alice extends M^1 into a perfect matching uniformly at random.

[15] That is, $X_{[i]} = \{(X_\ell, X_{\ell \bmod L+1}) : \ell \in [L] \wedge \tau_\ell = i\}$.

- Similarly, Bob generates $M^2, \ldots, M^T$ as follows:
  - For every $\mu \in \{2, \ldots, T\}$, Bob first sets $M^\mu = X_{[\mu]}$.
  - For every $i \in [k]$, every $\mu \in \{2, \ldots, T\}$, and every $j \in [L]$ such that $\tau_j = \mu$, Bob adds the edge $(M^{\mathsf{B}}_{i,j}, M^{\mathsf{B}}_{i,(j \bmod L)+1})$ to M^μ.
  - For every $\mu \in \{2, \ldots, T\}$, Bob extends M^μ into a perfect matching uniformly at random.

We first prove the following claim.

Claim 5.12. *It holds that*

$$\|\widetilde{\mathcal{R}}_X - \mathcal{R}_X\|_{\mathrm{TV}} \leq n^{-0.2},$$

and

$$\widetilde{\mathcal{R}}_X(\perp) \leq n^{-0.2}.$$

Proof. Let $\mathcal{D}_X$ and $\widetilde{\mathcal{D}}_X$ be the distribution of the pairs $(M^{\mathsf{A}}, M^{\mathsf{B}})$ in Samp_X and $\widetilde{\mathsf{Samp}}_X$, respectively. It suffices to show that $\|\mathcal{D}_X - \widetilde{\mathcal{D}}_X\|_{\mathrm{TV}} \leq n^{-0.2}$. Let $\mathcal{E}$ be the probability that the union of $M^{\mathsf{A}}, M^{\mathsf{B}}, X$ has distinct entries. We note that $\mathcal{D}_X$ is simply $\widetilde{\mathcal{D}}_X$ conditioning on the event $\mathcal{E}$.

By a simple union bound, we have $\Pr_{\widetilde{\mathcal{D}}_X}[\mathcal{E}] \geq 1-(2kL)^2/n$, which implies $\widetilde{\mathcal{R}}_X(\perp) \leq n^{-0.2}$ and $\|\mathcal{D}_X - \widetilde{\mathcal{D}}_X\|_{\mathrm{TV}} \leq n^{-0.2}$, and therefore completes the proof. □

From now on we are going to show $\mathcal{G}_X$ and $\widetilde{\mathcal{R}}_X$ are close. We will use the following claim.

Claim 5.13. *For all $G \in \mathrm{supp}(\mathcal{G}_X)$,*

$$\frac{\widetilde{\mathcal{R}}_X(G)}{\mathcal{G}_X(G)} \leq (1 + n^{-0.15}).$$

Proof. Fix $G \in \mathcal{G}_X$. We note that if G is generated by the procedure for generating $\widetilde{\mathcal{R}}_X$, then $(M^{\mathsf{A}}, M^{\mathsf{B}})$ are indeed completely determined by $(k-1) \cdot 2\ell$ entries. Hence we have

$$\widetilde{\mathcal{R}}_X(G) \leq \frac{n^{(k-1)\cdot 2\ell}}{n^{(k-1)\cdot(2\ell+L)}} \cdot \left[\prod_{i\in[T]} C_{n-2k|X_{[i]}|}\right]^{-1}.$$

Also, note that

$$\mathcal{G}_X(G) = \left[\prod_{i\in[T]} C_{n-2|X_{[i]}|}\right]^{-1},$$

we have

$$\frac{\widetilde{\mathcal{R}}_X(G)}{\mathcal{G}_X(G)} \leq n^{-(k-1)\cdot L} \prod_{i\in[T]} \left[\frac{C_{n-2|X_{[i]}|}}{C_{n-2k|X_{[i]}|}}\right]$$

By Fact 5.11 and noting $k|X_{[i]}| \le n^{1/3} \cdot \log n \le n^{0.34}$, the above can be bounded by

$$n^{-(k-1)\cdot L} \cdot n^{(k-1)\cdot L} \cdot (1+n^{-0.2})^{2T} \le (1+n^{-0.15}). \qquad \square$$

Now, note that

$$\|\widetilde{\mathcal{R}}_X - \mathcal{G}_X\|_{\mathrm{TV}} = \widetilde{\mathcal{R}}_X(\perp) + \sum_{G \in \mathrm{supp}(\mathcal{G}_X)} \max(0, \widetilde{\mathcal{R}}_X(G) - \mathcal{G}_X(G)).$$

By Claim 5.13 and Claim 5.12, the above can be bounded

$$n^{-0.2} + \sum_{G \in \mathrm{supp}(\mathcal{G}_X)} \mathcal{G}_X(G) \cdot n^{-0.15} \le n^{-0.2} + n^{-0.15} \le n^{-0.1},$$

which completes the proof. $\square$

5.3 Proof of Lemma 5.9

In this section we prove Lemma 5.9, which is restated below.

Reminder of Lemma 5.9. *For all $T \in \mathbb{N}_{\ge 1}$, for all sufficiently large $n \in 2\mathbb{N}_{\ge 1}$ the following holds: for every $L \in [\log n]$ and valid cycle pattern $\vec{\tau} \in [T]^L$, it holds that*

$$\Pr_{G \leftarrow \mathcal{G}_{n,T}}[\#_{\vec{\tau}}(G) > \log^3 n] \le n^{-100},$$

and

$$1/2 \le \mathop{\mathbb{E}}_{G \leftarrow \mathcal{G}_{n,T}}[\#_{\vec{\tau}}(G)] \le 2.$$

Proof. We first bound $\mathbb{E}_{G \leftarrow \mathcal{G}_{n,T}}[\#_{\vec{\tau}}(G)]$. By linearity of expectation, we have that

$$\mathop{\mathbb{E}}_{G \leftarrow \mathcal{G}_{n,T}}[\#_{\vec{\tau}}(G)] = \sum_{v_{\mathsf{s}} \in [n]} \Pr_{G \leftarrow \mathcal{G}_{n,T}}[\mathsf{Path}(G, v_{\mathsf{s}}, \vec{\tau}) \in \mathbb{C}_{\vec{\tau}}(G)].$$

So it suffices to bound $\Pr_{G \leftarrow \mathcal{G}_{n,T}}[\mathsf{Path}(G, v_{\mathsf{s}}, \vec{\tau}) \in \mathbb{C}_{\vec{\tau}}(G)]$ for a fixed $v_{\mathsf{s}} \in [n]$. We will analyze the following "lazy procedure" when determining if $\mathsf{Path}(G, v_{\mathsf{s}}, \vec{\tau}) \in \mathbb{C}_{\vec{\tau}}(G)$:

1. Let $v_0 = v_{\mathsf{s}}$ and $\vec{w}^{(0)}$ be an empty list.
2. For i from 1 to L:
   (a) Let e be the unique edge in the matching $\vec{M}^{\tau_i}$ that is adjacent to the vertex v_{i-1}.
   (b) Let v_i be the endpoint of e other than v_{i-1}. If $i < L$ and v_i is already visited in $\vec{w}^{(i-1)}$ (*i.e.*, v_i is the endpoints of some edges in $\vec{w}^{(i-1)}$), then return NO.

(c) $\vec{w}^{(i)} = \vec{w}^{(i-1)} \circ e$.

3. If $v_L = v_0$, return YES. Otherwise return NO.

Intuitively, in Step (2) we check whether we get a simple path, and in Step (3) we check whether we get a cycle. Now we analyze the probability that the above procedure returns YES. Let $\mathcal{E}_i$ be the event that the procedure does not return NO before the end of i-th loop at Step (2). We first calculate $\Pr[\mathcal{E}_i|\mathcal{E}_{i-1}]$.

Note that conditioning on $\mathcal{E}_{i-1}$, the path has visited i vertices $v_0, \ldots, v_{i-1}$. Let t_i be the number of edges from M^{τ_i} that is contained in $\vec{w}^{(i-1)}$. We note that $t_i \leq i/2$. We can see that the endpoint of e other than v_{i-1} has $n-1-2t_i$ many equally likely choices, and only $i-2t_i$ many of them causes the procedure to return NO. Hence, we have

$$\Pr[\mathcal{E}_i|\mathcal{E}_{i-1}] = 1 - \frac{i-2t_i}{n-1-2t_i}.$$

Now we analyze the probability of the procedure outputting YES conditioning on $\mathcal{E}_L$. Again, we note that the other endpoint of the last edge e has $n-1-2t_L$ many equally likely choices, but only 1 of them (v_{s}) causes the procedure to return YES. Hence the probability is $\frac{1}{n-1-2t_L}$.

Hence, we have

$$\begin{aligned} p_{\mathsf{single}} &= \Pr_{G \leftarrow \mathcal{G}_{n,T}}[\mathsf{Path}(G, v_{\mathsf{s}}, \vec{\tau}) \in \mathbb{C}_{\vec{\tau}}(G)] \\ &= \frac{1}{n-1-2t_L} \cdot \prod_{i \in [L-1]} \left(1 - \frac{i-2t_i}{n-1-2t_i}\right). \end{aligned}$$

Note that $2t_i \leq i$ for all $i \in [L]$, we have

$$p_{\mathsf{single}} \leq \frac{1}{n-1-L} \leq 2/n,$$

and

$$p_{\mathsf{single}} \geq \frac{1}{n} \cdot \prod_{i \in [L-1]} \left(1 - \frac{i}{n-1}\right) \geq 1/2n,$$

the last inequality follows from $L \in [\log n]$.

The desired bound on $\mathbb{E}_{G \leftarrow \mathcal{G}_{n,T}}[\#_{\vec{\tau}}(G)]$ then follows from the fact that it equals $n \cdot p_{\mathsf{single}}$.

Upper bounding $\Pr_{G \leftarrow \mathcal{G}_{n,T}}[\#_{\vec{\tau}}(G) > \log^3 n]$**.** We first note that a vertex $u \in [n]$ in $G \in \mathrm{supp}(\mathcal{G}_{n,T})$ can only be contained in at most L many cycles with pattern $\vec{\tau}$, since fixing its position in the pattern $\vec{\tau}$ completely determines the cycle. Hence, a cycle $C \in \mathbb{C}_{\vec{\tau}}(G)$ can share vertices with at most L^2 many other cycles in $\mathbb{C}_{\vec{\tau}}(G)$.

Assuming now that $\#_{\vec{\tau}}(G) = |\mathbb{C}_{\vec{\tau}}(G)| > \log^3 n$. We consider a dependence graph $V_{\mathbb{C}}$ with vertices being $\mathbb{C}_{\vec{\tau}}(G)$, and we add an edge between two cycles in $V_{\mathbb{C}}$ if they share a

vertex. By previous discussions, we know that $V_{\mathbb{C}}$ has maximum degree $\Delta \le L^2 \le \log^2 n$. By a standard coloring argument, it follows that $V_{\mathbb{C}}$ has an independent set of size at least $(\log^3 n)/(\Delta+1) \ge \log n - 1$.

Let $\ell = \log n - 1$. From the above discussion, we know that $\#_{\vec{\tau}}(G) > \log^3 n$ implies the existence of ℓ many vertex-disjoint pattern-$\vec{\tau}$ cycles in G. We denote the latter event as $\mathcal{E}_{\mathsf{nice}}$ and will upper bound $\Pr[\mathcal{E}_{\mathsf{nice}}]$ instead.

Let $S = \{s_1, s_2, \ldots, s_\ell\}$ be a subset of $[n]$. For every possible length-L paths $\vec{W} = (\vec{w}^1, \vec{w}^2, \ldots, \vec{w}^\ell)$, we will show that conditioning on the event

$$\mathcal{E}_{\mathsf{path}}^{S,\vec{W}} = \bigwedge_{i\in[\ell]} \left[\mathsf{Path}(G, s_i, \vec{\tau}) = \vec{w}^i\right],$$

the probability that all of $\mathsf{Path}(G, s_i, \vec{\tau})$ are vertex-disjoint simple cycles with pattern $\vec{\tau}$, denoted as event $\mathcal{E}_{\mathsf{nice}}^S$ are at most

$$(n - \ell \cdot L)^{-\ell}.$$

Now, conditioning on $\mathcal{E}_{\mathsf{path}}$, if for any $i \neq j$, $\vec{w}^i$ and $\vec{w}^j$ share at least one vertex, then by definition $\mathcal{E}_{\mathsf{nice}}^S$ happens with probability 0. So we can assume all of $\vec{w}^i$ are pair-wise vertex-disjoint. In this case, we note that $\mathcal{E}_{\mathsf{nice}}^S$ happens if and only if the following event happens: for every $i \in [\ell]$, the unique edge from M^{τ_L} that is adjacent to w_L^i, connects to w_1^i.

Since all $\vec{w}^i$'s are vertex disjoint, the above happens with probability at most $(n-\ell\cdot L)^{-\ell}$.

Putting the above together, it follows that

$$\Pr[\mathcal{E}_{\mathsf{nice}}^S] \le (n - \ell \cdot L)^{-\ell} \le (n - \log^2 n)^{-\ell}.$$

By a union bound, we have

$$\begin{aligned}
\Pr[\mathcal{E}_{\mathsf{nice}}] &\le \sum_{S\subset[n],|S|=\ell} \Pr[\mathcal{E}_{\mathsf{nice}}^S] \\
&\le \binom{n}{\ell} \cdot (n - \log^2 n)^{-\ell} \\
&\le \frac{n^\ell}{\ell!} \cdot (n - \log^2 n)^{-\ell} \\
&\le \left(\frac{n}{n - \log^2 n}\right)^\ell \cdot \frac{1}{\ell!} \\
&\le \left(\frac{1}{1 - \log^2 n/n}\right)^\ell \cdot \frac{1}{\ell!} \\
&\le n^{-100}. && (n \text{ is sufficiently large and } \ell = \log n - 1)
\end{aligned}$$

Finally, recall that $\#_{\vec{\tau}}(G) > \log^3 n$ implies $\mathcal{E}_{\mathsf{nice}}$, it follows that $\Pr[\#_{\vec{\tau}}(G) > \log^3 n] \le n^{-100}$ as well, which completes the proof. □

6 Lower Bounds for Finding a Long Path

Recall that $\mathbb{L}_{\vec{\tau}}(G)$ is the set of simple paths in G with pattern $\vec{\tau}$. In this section we prove Lemma 4.2, which is restated below.

Reminder of Lemma 4.2. *There exist $\varepsilon, \delta, \gamma_0 \in (0,1)$ such that for all $T \in \mathbb{N}_{\geq 1}$ and for all sufficiently large $n \in 2\mathbb{N}_{\geq 1}$ the following holds: For all $L \in [\gamma_0 \cdot \log n]$, $p \leq (L-15)/4T$, valid path pattern $\vec{\tau} \in [T]^L$, and p-pass n^{ε}-space streaming algorithms $\mathbb{A}$, we have*

$$\Pr_{G \leftarrow \mathcal{G}_{n,T}} \left[\mathbb{A}(G) \in \mathbb{L}_{\vec{\tau}}(G) \right] \leq n^{3-\delta L/p}. \tag{11}$$

We will also call the problem of finding an element in $\mathbb{L}_{\vec{\tau}}(G)$ as the $\mathsf{Path\text{-}Search\text{-}Streaming}_{n,T,\vec{\tau}}$ problem.

6.1 Lower Bounds for a Specific Pointer Chasing Problem

We will prove Lemma 4.2 by a reduction from a specific pointer chasing problem.

Definition 6.1 (The $\mathsf{ASPC}_{n,d}$ problem)**.** *Let $n \in \mathbb{N}_{\geq 1}$ and $d \in 2\mathbb{N}_{\geq 1}$. In the $\mathsf{ASPC}_{n,d}$ problem, there are two players Alice and Bob, and d permutations $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_d)$ on $[n]$. Alice gets all the odd-indexed permutations $\pi_1, \pi_3, \ldots, \pi_{d-1}$, and Bob gets all the even-indexed permutations $\pi_2, \pi_4, \ldots, \pi_d$. Let $\pi_{\leq i} = \pi_i \circ \pi_{i-1} \circ \ldots \circ \pi_1$ for every $i \in [d]$. Their goal is to output the path $\mathsf{path}_{\vec{\pi}}(s) = (s, \pi_{\leq 1}(s), \pi_{\leq 2}(s), \ldots, \pi_{\leq d}(s))$ for <u>some</u> $s \in [n]$.*

For notational convenience, let $\mathbb{P}(\vec{\pi}) = \{\mathsf{path}_{\vec{\pi}}(s) : s \in [n]\}$. The goal of Alice and Bob can then be restated as outputting an element from $\mathbb{P}(\vec{\pi})$. We also let $\mathcal{P}_{n,d}$ denote the uniform distribution over all possible $\vec{\pi}$ consisting of d permutations on $[n]$.

We need the following lower bound for $\mathsf{ASPC}_{n,d}$; see Appendix A for a proof.

Lemma 6.2 (Lower Bounds for $\mathsf{ASPC}_{n,d}$)**.** *There exist $\varepsilon, \delta \in (0,1)$ such that for all sufficiently large $n \in \mathbb{N}$ the following holds: for all $d \in [\log n]$, $p \leq (d-6)/2$, and all p-round communication protocols Π with at most n^{ε} communication complexity, it holds that*

$$\Pr_{\vec{\pi} \leftarrow \mathcal{P}_{n,d}} [\Pi(\vec{\pi}) \in \mathbb{P}(\vec{\pi})] \leq n^{1-\delta d/p},$$

where $\Pi(\vec{\pi})$ denotes the output of Π when Alice gets the input $\pi_1, \pi_3, \ldots, \pi_{d-1}$ and Bob gets the input $\pi_2, \pi_4, \ldots, \pi_d$.

To make use of Lemma 6.2, we first reduce ASPC to another auxiliary problem, which is closer to the $\mathsf{Path\text{-}Search\text{-}Streaming}$ problem considered in Lemma 4.2.

Definition 6.3 (The $\mathsf{Path\text{-}Finding}_{n,T,\vec{\tau}}$ problem)**.** *Let $n, T, L \in \mathbb{N}_{\geq 1}$ and $\vec{\tau} \in [T]^L$ be a valid path pattern. In the $\mathsf{Path\text{-}Finding}_{n,\vec{\tau}}$ problem, there is a graph H consisting of $L+1$ layers of*

vertices $\vec{V} = (V_1, V_2, \ldots, V_{L+1})$*, each with size* n*, and* L *set of edges* $\vec{W} = (W_1, W_2, \ldots, W_L)$ *such that* W_i *is a perfect bipartite matching between layers* V_i *and* V_{i+1}*. There are* T *players* $P_1, \ldots, P_T$*, such that the* i*-th players gets all* W_ℓ *such that* $\tau_\ell = i$ *as input. Their goal is to output a directed path from the first layer* V_1 *to the last layer* V_{L+1}*.*

For simplicity, we will always assume $V_i = \{(i-1) \cdot n + 1, (i-1) \cdot n + 2, \ldots, i \cdot n\}$ for each $i \in [L]$. We also let $\mathcal{W}_{n,L}$ be the uniform distribution over all possible $\vec{W}$ consisting of L perfect bipartite matchings, where the i-th matching is between V_i and V_{i+1}. We denote $\mathbb{P}(\vec{W})$ as the set of all directed paths from the first layer V_1 to the last layer V_{L+1}, going through the graph defined by $\vec{W}$. The goal of $\mathsf{Path\text{-}Finding}_{n,\vec{\tau}}$ can then be restated as output an element of $\mathbb{P}(\vec{W})$.

Using a reduction from the ASPC problem, we have the following lower bound for Path-Finding.

Lemma 6.4 (Lower bounds for $\mathsf{Path\text{-}Finding}_{n,T,\vec{\tau}}$)**.** *There exist* $\varepsilon, \delta \in (0,1)$ *such that for all* $T \in \mathbb{N}$ *and for all sufficiently large* $n \in \mathbb{N}$ *the following holds: for all* $L \in [\log n]$*, valid path pattern* $\vec{\tau} \in [L]^T$*,* $p \leq (L-15)/4T$*, and all* p*-round communication protocol* Π *with at most* n^ε *communication complexity in the blackboard model,*[16] *it holds that*

$$\Pr_{\vec{W} \leftarrow \mathcal{W}_{n,L}} [\Pi(\vec{W}) \in \mathbb{P}(\vec{W})] \leq n^{1-\delta L/p},$$

where $\Pi(\vec{W})$ *denotes the output of the protocol* Π *when the* T *players get their inputs from* $\vec{W}$ *according to the pattern* $\vec{\tau}$*.*

Proof. We first partition the T players into two disjoint sets $T_1, T_2 \subseteq \{P_1, \ldots, P_T\}$ such that there are at least $(L-1)/2$ indices ℓ such that τ_ℓ and $\tau_{\ell+1}$ are not in the same set. Such partition always exists by a probabilistic argument, since a random partition gives $(L-1)/2$ such indices in expectation.

This allows us to view $\vec{\tau}$ as $d \geq \lceil (L-1)/2 \rceil + 1$ segments that alternate between players in T_1 and players in T_2. We will view blackboard communication protocols for $\mathsf{Path\text{-}Finding}_{n,T,\vec{\tau}}$ as a two-player communication protocol between "player" T_1 and "player" T_2.

Formally, let $\vec{\tau}_1, \ldots, \vec{\tau}_d$ be the segments of τ such that each odd $\vec{\tau}_i$ has all its coordinates in T_1, and each even $\vec{\tau}_i$ has all its coordinates in T_2. Fix a protocol Π for $\mathsf{Path\text{-}Finding}_{n,T,\vec{\tau}}$. We will use it to solve $\mathsf{ASPC}_{n,d}$, then apply the lower bound in Lemma 6.2.

Consider the following protocol for $\mathsf{ASPC}_{n,d}$.

[16] That is, in each round, from the first player to the T-th player, each player writes some bits to a blackboard that can be seen by everyone. And the final output of the protocol is only determined by the content of the blackboard at the end of the protocol. The communication complexity of the protocol is the maximum total number of bits written on the blackboard.

Protocol for $\mathsf{ASPC}_{n,d}$

Inputs $\pi_1, \ldots, \pi_d$

Communication

1. for each odd i, Alice samples uniformly random matchings in segment $\vec{\tau}_i$ *conditioned on* their composition equal to π_i
2. for each even i, Bob samples uniformly random matchings in segment $\vec{\tau}_i$ *conditioned on* their composition equal to π_i
3. denote the graph they generated by H, Alice and Bob run Π on H, where Alice simulates all players in T_1 and Bob simulates all players in T_2, and obtain a path Q in H
4. output the path for $\vec{\pi}$ obtained by composing all segments $\vec{\tau}_1, \ldots, \vec{\tau}_d$ of Q

When $\pi_1, \ldots, \pi_d$ are uniform, H is a uniformly random graph. Alice knows the inputs for all players in T_1, and Bob knows the inputs for all players in T_2. Hence, the players can simulate Π. Moreover, the number of rounds in our protocol for $\mathsf{ASPC}_{n,d}$ is at most T times the number of rounds in Π. When Π outputs a correct path Q in H, the output of the protocol for $\mathsf{ASPC}_{n,d}$ is correct.

Hence, by Lemma 6.2, the probability that Π outputs a correct path is at most $n^{1-\delta d/p} \leq n^{1-\delta L/(2p)}$. By reparametrizing, we prove the lemma. □

6.2 Proof of Lemma 4.2

Now we are ready to prove Lemma 4.2 by a reduction from the Path-Finding problem.

Reduction Red-Path from $\mathsf{Path\text{-}Finding}_{n^\gamma,T,\vec{\tau}}$ to $\mathsf{Path\text{-}Search\text{-}Streaming}_{n,T,\vec{\tau}}$

Parameters $\gamma = 10^{-3}$. $n, T, \vec{\tau}$ are parameters for the desired Path-Search-Streaming problem instance. Let $m = n^\gamma$.

Input for $\mathsf{Path\text{-}Finding}_{m,T,\vec{\tau}}$ There is a graph H consisting of $L+1$ layers of vertices $\vec{V} = (V_1, V_2, \ldots, V_{L+1})$, each with size m, and L matchings $\vec{W} = (W_1, W_2, \ldots, W_L)$ such that W_i is a perfect bipartite matching between layers V_i and V_{i+1}. There are T players $P_1, \ldots, P_T$, such that the P_i gets all W_ℓ such that $\tau_\ell = i$ as input.

We also have $V_i = \{(i-1)\cdot m + 1, (i-1)\cdot m + 2, \ldots, i \cdot m\}$ for each $i \in [L]$, and $\bigcup_{i\in[L+1]} V_i = [m \cdot (L+1)]$.

- All T players first use public randomness to sample an injective function

$\varphi\colon [m\cdot(L+1)] \to [n]$.

- For each $i \in [T]$:
  1. Let E_i be the set of all edges from $\{W_\ell : \tau_\ell = i\}$. Player P_i first constructs a partial matching $M_i = \{(\varphi(u), \varphi(v)) : (u,v) \in E_i\}$.[a]
  2. P_i then extends M_i into a perfect matching over $[n]$ uniformly at random.

[a] M_i is indeed a partial matching since $\vec{\tau}$ is a valid path pattern, as required by the definition of Path-Search-Streaming$_{n,T,\vec{\tau}}$.

Notation. We call a subset $X \subseteq [n]$ a *valid starting subset* of a graph $G = ([n], M^1 \circ \ldots \circ M^T) \in \mathrm{supp}(\mathcal{G}_{n,T})$ with respect to the pattern $\vec{\tau}$, if for every $u \in X$, $\mathsf{Path}(G, u, \vec{\tau})$ is simple, and for every two distinct $u, v \in X$, $\mathsf{Path}(G, u, \vec{\tau})$ and $\mathsf{Path}(G, v, \vec{\tau})$ are vertex-disjoint. We also use $\mathbb{X}_{m,\vec{\tau}}(G)$ to denote the set of all valid starting subset of G of size m with respect to $\vec{\tau}$. For a subset $S \subseteq [n]$, we use $\mathbb{X}^S_{m,\vec{\tau}}(G)$ to denote the subset of $\mathbb{X}_{m,\vec{\tau}}(G)$ that contains S as a subset.

For a subset $X \subseteq [n]$ with size n^γ, we use $\mathcal{G}_{n,T,\vec{\tau};X}$ to denote the uniform distribution over all possible graphs $G \in \mathrm{supp}(\mathcal{G}_{n,T})$ such that $X \in \mathbb{X}_{|X|,\vec{\tau}}(G)$. Also, let $\mathcal{R}_{n,T,\vec{\tau};X}$ be the distribution outputted by Red-Path given inputs drawn from $\mathcal{W}_{n,L}$ and conditioning on the event that $\{\varphi(i) : i \in [n^\gamma]\} = X$. We have the following observation.

Observation 6.5. *Let $n, T, \vec{\tau}$ and γ be as in the reduction* Red-Path*. For every $X \subseteq [n]$ with size n^γ, it holds that the distributions $\mathcal{R}_{n,T,\vec{\tau};X}$ and $\mathcal{G}_{n,T,\vec{\tau};X}$ are identical.*

We also need the following lemma.

Lemma 6.6. *Let $\gamma = 10^{-3}$. For all $T \in \mathbb{N}_{\geq 1}$, for all sufficiently large $n \in 2\mathbb{N}_{\geq 1}$ the following holds: letting $m = n^\gamma$, for every $L \in [\log n]$ and valid path pattern $\vec{\tau} \in [T]^L$, it holds that*

1. $$\Pr_{G \leftarrow \mathcal{G}_{n,T}}\left[|\mathbb{X}_{m,\vec{\tau}}(G)| \leq \frac{1}{2}\cdot\binom{n}{m}\right] \leq n^{-\log n},$$
2. $$\Pr_{G \leftarrow \mathcal{G}_{n,T}}\left[|\mathbb{X}^{\{1\}}_{m,\vec{\tau}}(G)| \leq \frac{1}{2}\cdot\binom{n-1}{m-1} \wedge \mathsf{Path}(G, 1, \vec{\tau}) \in \mathbb{L}_{\vec{\tau}}(G)\right] \leq n^{-\log n}.$$

Now we are ready to prove Lemma 4.2.

Proof of Lemma 4.2. Let $\varepsilon, \delta \in (0,1)$ be two constants to be specified later. Let $\widetilde{\varepsilon}, \widetilde{\delta}$ be the constants in Lemma 6.4, and $\gamma_0 = \gamma = 10^{-3}$.

Given a p-pass n^ε-space streaming algorithm $\mathbb{A}$ such that

$$\Pr_{G \leftarrow \mathcal{G}_{n,T}}[\mathbb{A}(G) \in \mathbb{L}_{\vec{\tau}}(G)] > n^{3-\delta L/p}, \tag{12}$$

we will construct a communication protocol for $\mathsf{Path\text{-}Finding}_{m,T,\vec{\tau}}$ that violates Lemma 6.4.

First, we note that (12) implies that there exists a vertex $s^* \in [n]$ such that

$$\Pr_{G \leftarrow \mathcal{G}_{n,T}} [\mathbb{A}(G) = \mathsf{Path}(G, s^*, \vec{\tau}) \wedge \mathbb{A}(G) \in \mathbb{L}_{\vec{\tau}}(G)] > n^{2-\delta L/p}. \tag{13}$$

We set $p_{\mathsf{suc}} = n^{2-\delta L/p}$ for notational convenience. By symmetry, we can assume that $s^* = 1$.

Our protocol Π for $\mathsf{Path\text{-}Finding}_{m,T,\vec{\tau}}$ works by first running $\mathsf{Red\text{-}Path}$ to obtain a $\mathsf{Path\text{-}Search\text{-}Streaming}_{n,T,\vec{\tau}}$ instance, and then simulating the streaming algorithm $\mathbb{A}$ using p rounds and $n^{\varepsilon} \cdot (p \cdot T)$ bits of communication to obtain $\mathbb{A}$'s output, a length-L path $\vec{v} = (v_1, v_2, \ldots, v_{L+1}) \in [n]$. Finally, it constructs a new length-L path $\vec{u}$ in the $\mathsf{Path\text{-}Finding}_{m,T,\vec{\tau}}$ by setting $u_i = \varphi^{-1}(v_i)$ for every $i \in [L+1]$, and outputs $\vec{u}$ (if some v_i is not in the range of φ, or $\mathbb{A}$ does not output a valid length-L path $\vec{u}$, Π simply outputs $\perp$).

Now we analyze the success probability of Π over the distribution $\mathcal{W}_{n,L}$. We first note that conditioning on the event that $\{\varphi(i) : i \in [m]\} = X$, the output distribution of $\mathsf{Red\text{-}Path}$ is $\mathcal{R}_{n,T,\vec{\tau};X}$, which is identical to $\mathcal{G}_{n,T,\vec{\tau};X}$ by Observation 6.5. From now on, we will denote $\mathcal{G}_{n,T,\vec{\tau};X}$ by $\mathcal{G}_X$ for simplicity.

The success probability can then be lower bounded by

$$\Pr_{X \leftarrow \binom{[n]}{m}} \Pr_{G \leftarrow \mathcal{G}_X} [\mathbb{A}(G) = \mathsf{Path}(G, 1, \vec{\tau}) \wedge \mathbb{A}(G) \in \mathbb{L}_{\vec{\tau}}(G) \wedge 1 \in X]. \tag{14}$$

Now, let $\widetilde{\mathcal{G}}$ be the distribution generated as follows: first draw $X \leftarrow \binom{[n]}{m}$, then draw $G \leftarrow \mathcal{G}_X$ and output G. (14) can then be alternatively written as

$$\begin{aligned} &\Pr_{G \leftarrow \widetilde{\mathcal{G}}} \Pr_{X \in \mathbb{X}_{m,\vec{\tau}}(G)} [\mathbb{A}(G) = \mathsf{Path}(G, 1, \vec{\tau}) \wedge \mathbb{A}(G) \in \mathbb{L}_{\vec{\tau}}(G) \wedge 1 \in X] \\ = &\Pr_{G \leftarrow \widetilde{\mathcal{G}}} \mathbb{1}_{\{\mathbb{A}(G) = \mathsf{Path}(G,1,\vec{\tau}) \wedge \mathbb{A}(G) \in \mathbb{L}_{\vec{\tau}}(G)\}} \cdot \Pr_{X \in \mathbb{X}_{m,\vec{\tau}}(G)} [1 \in X]. \end{aligned} \tag{15}$$

To lower bound (15), we need the following claim.

Claim 6.7. *For every event $\mathcal{E}$, it holds that*

$$\Pr_{G \leftarrow \widetilde{\mathcal{G}}} [\mathcal{E}(G)] \geq \frac{1}{2} \cdot \left[\Pr_{G \leftarrow \mathcal{G}} [\mathcal{E}(G)] - n^{-\log n} \right].$$

Proof. We first note that $\mathrm{supp}(\widetilde{\mathcal{G}}) \subseteq \mathrm{supp}(\mathcal{G})$, and for $G \in \mathrm{supp}(\mathcal{G})$, we have

$$\widetilde{\mathcal{G}}(G) = \frac{\mathbb{X}_{m,\vec{\tau}}(G)}{\sum_{H \in \mathrm{supp}(\mathcal{G})} \mathbb{X}_{m,\vec{\tau}}(H)},$$

which implies that

$$\frac{\widetilde{\mathcal{G}}(G)}{\mathcal{G}(G)} = \frac{\mathbb{X}_{m,\vec{\tau}}(G)}{\mathbb{E}_{H\in\mathrm{supp}(\mathcal{G})}\,\mathbb{X}_{m,\vec{\tau}}(H)}.$$

Now, we have

$$\begin{aligned}
\Pr_{G\leftarrow\widetilde{\mathcal{G}}}[\mathcal{E}(G)] &= \Pr_{G\leftarrow\mathcal{G}} \frac{\widetilde{\mathcal{G}}(G)}{\mathcal{G}(G)} \cdot [\mathcal{E}(G)] \\
&= \Pr_{G\leftarrow\mathcal{G}} \frac{\mathbb{X}_{m,\vec{\tau}}(G)}{\mathbb{E}_{H\in\mathrm{supp}(\mathcal{G})}\,\mathbb{X}_{m,\vec{\tau}}(H)} \cdot [\mathcal{E}(G)] \\
&\geq \binom{n}{m}^{-1} \cdot \binom{n}{m} \cdot 1/2 \cdot \Pr_{G\leftarrow\mathcal{G}}\left[\mathbb{X}_{m,\vec{\tau}}(G) > \frac{1}{2}\cdot\binom{n}{m} \wedge \mathcal{E}(G)\right] \\
&\qquad\qquad (\mathbb{E}_{H\in\mathrm{supp}(\mathcal{G})}\,\mathbb{X}_{m,\vec{\tau}}(H) \leq \tbinom{n}{m}) \\
&\geq \frac{1}{2}\cdot(\Pr_{G\leftarrow\mathcal{G}}[\mathcal{E}(G)] - n^{-\log n}). \qquad \text{(Lemma 6.6)}
\end{aligned}$$

□

Now we are ready to lower bound (15). We have

$$\begin{aligned}
&\Pr_{G\leftarrow\widetilde{\mathcal{G}}} \mathbb{1}_{\{\mathbb{A}(G)=\mathsf{Path}(G,1,\vec{\tau})\wedge\mathbb{A}(G)\in\mathbb{L}_{\vec{\tau}}(G)\}} \cdot \Pr_{X\in\mathbb{X}_{m,\vec{\tau}}(G)}[1\in X] \\
&\geq \frac{\binom{n-1}{m-1}/2}{\binom{n}{m}} \cdot \Pr_{G\leftarrow\widetilde{\mathcal{G}}}\left[\mathbb{A}(G)=\mathsf{Path}(G,1,\vec{\tau})\wedge\mathbb{A}(G)\in\mathbb{L}_{\vec{\tau}}(G)\wedge\mathbb{X}^{\{1\}}_{m,\vec{\tau}}(G)\geq\binom{n-1}{m-1}\cdot\frac{1}{2}\right] \\
&\geq n^{-1}\cdot\left(\Pr_{G\leftarrow\mathcal{G}}\left[\mathbb{A}(G)=\mathsf{Path}(G,1,\vec{\tau})\wedge\mathbb{A}(G)\in\mathbb{L}_{\vec{\tau}}(G)\wedge\mathbb{X}^{\{1\}}_{m,\vec{\tau}}(G)\geq\binom{n-1}{m-1}\cdot\frac{1}{2}\right] - n^{-\log n}\right) \\
&\qquad\qquad \text{(Claim 6.7)} \\
&\geq n^{-1}\cdot(p_{\mathsf{suc}} - 2\cdot n^{-\log n}). \\
&\qquad \text{(Lemma 6.6 and } \mathbb{A}(G)=\mathsf{Path}(G,1,\vec{\tau})\wedge\mathbb{A}(G)\in\mathbb{L}_{\vec{\tau}}(G) \text{ implies } \mathsf{Path}(G,1,\vec{\tau})\in\mathbb{L}_{\vec{\tau}}(G))
\end{aligned}$$

Now, we set $\varepsilon = \widetilde{\varepsilon}/2$, which means Π has communication complexity $n^{\varepsilon}\cdot(p\cdot T)\leq n^{\widetilde{\varepsilon}}$. We also set $\delta = \frac{1}{2}\cdot\gamma\cdot\widetilde{\delta}$. Then the success probability of Π over $\mathcal{W}_{n,L}$ is at least

$$n^{-1}\cdot(n^{-\delta L/p+2} - 2\cdot n^{-\log n}) \geq n^{-\delta L/p+0.5} = n^{-\frac{1}{2}\widetilde{\delta}\gamma L/p+1/2} > m^{-\widetilde{\delta}L/p+1}, \tag{16}$$

contradicting Lemma 6.4. This completes the proof. □

6.3 Proof of Item (1) of Lemma 6.6

Reminder of Item (1) of Lemma 6.6. *Let $\gamma = 10^{-3}$. For all $T\in\mathbb{N}_{\geq 1}$, for all sufficiently large $n\in 2\mathbb{N}_{\geq 1}$ the following holds: letting $m = n^{\gamma}$, for every $L\in[\log n]$ and*

valid path pattern $\vec{\tau} \in [T]^L$, it holds that

$$\Pr_{G \leftarrow \mathcal{G}_{n,T}} \left[|\mathbb{X}_{m,\vec{\tau}}(G)| \leq \frac{1}{2} \cdot \binom{n}{m} \right] \leq n^{-\log n}.$$

Proof.

Notation and setup. Throughout the proof, we use $\boldsymbol{G}$ to denote a random variable drawn from $\mathcal{G}_{n,T}$. For every $S \in \binom{[n]}{m}$, we use $\boldsymbol{Y}_S$ to denote the random variable $\mathbb{1}_{\{S \in \mathbb{X}_{m,\vec{\tau}}(\boldsymbol{G})\}}$ (*i.e.*, $\boldsymbol{Y}_S$ equals 1 if $S \in \mathbb{X}_{m,\vec{\tau}}(G)$ and 0 otherwise). We also let $M = \binom{n}{m}$ and

$$\boldsymbol{Y} = \sum_{S \in \binom{[n]}{m}} \boldsymbol{Y}_S = |\mathbb{X}_{m,\vec{\tau}}(\boldsymbol{G})|. \tag{17}$$

Item (1) can then be restated as

$$\Pr[\boldsymbol{Y}/M \leq 1/2] \leq n^{-\log n}. \tag{18}$$

For each $S \in \binom{[n]}{m}$, we also define $\boldsymbol{Z}_S = (1 - \boldsymbol{Y}_S)$ and $\boldsymbol{Z} = \sum_{S \in \binom{[n]}{m}} \boldsymbol{Z}_S$. Let $\ell \leq n^{1/3}$ be an even integer to be chosen later. We will prove (18) by upper bounding

$$\mathbb{E}[(\boldsymbol{Z}/M)^\ell]. \tag{19}$$

Expanding (19). Now, for each $1 \leq u < v \leq n$, we define $\boldsymbol{W}_{u,v}$ to be the indicator random variable that the following three conditions all hold: (1) $\mathsf{Path}(\boldsymbol{G}, u, \vec{\tau})$ is simple, (2) $\mathsf{Path}(\boldsymbol{G}, v, \vec{\tau})$ is simple, and (3) $\mathsf{Path}(\boldsymbol{G}, u, \vec{\tau})$ and $\mathsf{Path}(\boldsymbol{G}, v, \vec{\tau})$ share at least one vertex. Also, for each $u \in [n]$, we define $\boldsymbol{B}_u$ to be the indicator random variable that $\mathsf{Path}(\boldsymbol{G}, u, \vec{\tau})$ is not simple. By the definition of $\boldsymbol{Z}_S$, we can see that for every $S \in \binom{[n]}{m}$,

$$\boldsymbol{Z}_S \leq \sum_{u,v \in S,\ u<v} \boldsymbol{W}_{u,v} + \sum_{u \in S} \boldsymbol{B}_u. \tag{20}$$

Plugging (20) in the definition of $\boldsymbol{Z}$, we have

$$\boldsymbol{Z} \leq \sum_{S \in \binom{[n]}{m}} \left[\sum_{u,v \in S,\ u<v} \boldsymbol{W}_{u,v} + \sum_{u \in S} \boldsymbol{B}_u \right] \tag{21}$$

$$= \sum_{1 \leq u < v \leq n} \boldsymbol{W}_{u,v} \cdot \binom{n-2}{m-2} + \sum_{u \in [n]} \boldsymbol{B}_u \cdot \binom{n-1}{m-1}, \tag{22}$$

which further implies that

$$\boldsymbol{Z}/M \leq \sum_{1 \leq u < v \leq n} \boldsymbol{W}_{u,v} \cdot \frac{m(m-1)}{n(n-1)} + \sum_{u \in [n]} \boldsymbol{B}_u \cdot \frac{m}{n}.$$

The inequality above can be further simplified to

$$\boldsymbol{Z}/M \leq \mathop{\mathbb{E}}_{1\leq u<v\leq n}[\boldsymbol{W}_{u,v}] \cdot \frac{m(m-1)}{2} + \mathop{\mathbb{E}}_{u\in[n]}[\boldsymbol{B}_u] \cdot m. \tag{23}$$

Raising both sides of (23) to the ℓ-th power, we have

$$\begin{aligned}
(\boldsymbol{Z}/M)^\ell &\leq \sum_{k=0}^{\ell} \binom{\ell}{k} \cdot \left(\frac{m(m-1)}{2}\right)^k \cdot m^{\ell-k} \left[\mathop{\mathbb{E}}_{\substack{1\leq u_1<v_1\leq n \\ \vdots \\ 1\leq u_k<v_k\leq n}} \prod_{i\in[k]} \boldsymbol{W}_{u_i,v_i} \cdot \mathop{\mathbb{E}}_{w_1,\ldots,w_{\ell-k}\in[n]} \prod_{i\in[\ell-k]} \boldsymbol{B}_{w_i} \right] \\
&\leq \sum_{k=0}^{\ell} 2^\ell \cdot \left(m^2/2\right)^k \cdot m^{\ell-k} \left[\mathop{\mathbb{E}}_{\substack{1\leq u_1<v_1\leq n \\ \vdots \\ 1\leq u_k<v_k\leq n}} \prod_{i\in[k]} \boldsymbol{W}_{u_i,v_i} \cdot \mathop{\mathbb{E}}_{w_1,\ldots,w_{\ell-k}\in[n]} \prod_{i\in[\ell-k]} \boldsymbol{B}_{w_i} \right] \\
&\leq m^{2\ell} \cdot \sum_{k=0}^{\ell} \left[\mathop{\mathbb{E}}_{\substack{1\leq u_1<v_1\leq n \\ \vdots \\ 1\leq u_k<v_k\leq n}} \prod_{i\in[k]} \boldsymbol{W}_{u_i,v_i} \cdot \mathop{\mathbb{E}}_{w_1,\ldots,w_{\ell-k}\in[n]} \prod_{i\in[\ell-k]} \boldsymbol{B}_{w_i} \right].
\end{aligned}$$

Taking the expectation of both sides, we have

$$\mathbb{E}[(\boldsymbol{Z}/M)^\ell] \leq m^{2\ell} \cdot \sum_{k=0}^{\ell} \left[\mathop{\mathbb{E}}_{\substack{1\leq u_1<v_1\leq n \\ \vdots \\ 1\leq u_k<v_k\leq n}} \mathop{\mathbb{E}}_{w_1,\ldots,w_{\ell-k}\in[n]} \mathbb{E} \left[\prod_{i\in[k]} \boldsymbol{W}_{u_i,v_i} \cdot \prod_{i\in[\ell-k]} \boldsymbol{B}_{w_i} \right] \right]. \tag{24}$$

In the rest of the proof, we will focus on upper bounding the right side of (24). We will upper bound each summand above separately depending on whether $k \geq \ell/2$ or $k < \ell/2$.

The case when $k < \ell/2$. We first focus on the case that $k < \ell/2$. We set $t = \ell - k$ and note that $t \geq \ell/2$.

Now, first note that we have

$$\mathop{\mathbb{E}}_{\substack{1\leq u_1<v_1\leq n \\ \vdots \\ 1\leq u_k<v_k\leq n}} \mathop{\mathbb{E}}_{w_1,\ldots,w_{\ell-k}\in[n]} \mathbb{E} \left[\prod_{i\in[k]} \boldsymbol{W}_{u_i,v_i} \cdot \prod_{i\in[\ell-k]} \boldsymbol{B}_{w_i} \right] \leq \mathop{\mathbb{E}}_{w_1,\ldots,w_t\in[n]} \mathbb{E} \left[\prod_{i\in[t]} \boldsymbol{B}_{w_i} \right]. \tag{25}$$

So in the following we will upper bound

$$\mathop{\mathbb{E}}_{w_1,\ldots,w_t\in[n]} \mathbb{E}\left[\prod_{i\in[t]} \boldsymbol{B}_{w_i}\right].$$

We will first condition on the event that the number of distinct elements in $w_1,\ldots,w_t$ is more than $t/2$. We first show the probability that this event does not happen is small, in particular

$$\Pr_{w_1,\ldots,w_t\in[n]}\left[\left|\{w_i\}_{i\in[t]}\right| \le t/2\right] \le \binom{n}{t/2}\cdot\left(\frac{t/2}{n}\right)^t \le n^{-t/4}. \tag{26}$$

So now we assume that $\left|\{w_i\}_{i\in[t]}\right| = r > t/2$, and we will upper bound

$$\mathbb{E}\left[\prod_{i\in S} \boldsymbol{B}_i\right]$$

for any $S \in \binom{[n]}{r}$.

Claim 6.8. *For every $r \le n^{1/3}$ and $S \in \binom{[n]}{r}$, it holds that*

$$\mathbb{E}\left[\prod_{i\in S} \boldsymbol{B}_i\right] \le n^{-r/2}.$$

Combining (26) and Claim 6.8, we have

$$\mathop{\mathbb{E}}_{w_1,\ldots,w_t\in[n]} \mathbb{E}\left[\prod_{i\in[t]} \boldsymbol{B}_{w_i}\right] \le n^{-t/4} + n^{-t/4} \le n^{-\ell/8} + n^{-\ell/8} \le 2n^{-\ell/8}. \tag{27}$$

Putting (27) and (25) together, and recall that we assumed $k < \ell/2$, we have

$$\sum_{k=0}^{\ell/2-1}\left[\mathop{\mathbb{E}}_{\substack{1\le u_1<v_1\le n\\ \vdots\\ 1\le u_k<v_k\le n}} \mathop{\mathbb{E}}_{w_1,\ldots,w_{\ell-k}\in[n]} \mathbb{E}\left[\prod_{i\in[k]} \boldsymbol{W}_{u_i,v_i}\cdot\prod_{i\in[\ell-k]} \boldsymbol{B}_{w_i}\right]\right] \le (\ell/2)\cdot 2n^{-\ell/8} \le \ell\cdot n^{-\ell/8}. \tag{28}$$

The case when $k \ge \ell/2$. Next we consider the case when $k \ge \ell/2$. We have that

$$\mathop{\mathbb{E}}_{\substack{1\le u_1<v_1\le n\\ \vdots\\ 1\le u_k<v_k\le n}} \mathop{\mathbb{E}}_{w_1,\ldots,w_{\ell-k}\in[n]} \mathbb{E}\left[\prod_{i\in[k]} \boldsymbol{W}_{u_i,v_i}\cdot\prod_{i\in[\ell-k]} \boldsymbol{B}_{w_i}\right] \le \mathop{\mathbb{E}}_{\substack{1\le u_1<v_1\le n\\ \vdots\\ 1\le u_k<v_k\le n}} \mathbb{E}\left[\prod_{i\in[k]} \boldsymbol{W}_{u_i,v_i}\right]. \tag{29}$$

In the following we will upper bound

$$\mathop{\mathbb{E}}_{\substack{1\le u_1<v_1\le n\\ \vdots\\ 1\le u_k<v_k\le n}} \mathbb{E}\left[\prod_{i\in[k]} \boldsymbol{W}_{u_i,v_i}\right].$$

Let $S = \{(u_i, v_i)\}_{i\in[r]}$ be a set of pairs. We say that S is *valid*, if the following two conditions hold: (1) all of $u_1, \ldots, u_r, v_1, \ldots, v_r$ are distinct elements of $[n]$ and (2) $u_i < v_i$ for every $i \in [r]$.

We need the following two claims.

Claim 6.9. *For every $k \le n^{1/3}$, it holds that*

$$\Pr_{\substack{1\le u_1<v_1\le n\\ \vdots\\ 1\le u_k<v_k\le n}} \left[\exists \text{ a valid set } S \text{ s.t. } |S| \ge k/2 \text{ and } S \subseteq \{(u_i, v_i)\}_{i\in[k]}\right] \ge 1 - n^{-k/4}.$$

Claim 6.10. *For every $r \le n^{1/3}$. Let S be a valid set of pairs such that $|S| = r$. It holds that*

$$\mathbb{E}\left[\prod_{(u,v)\in S} \boldsymbol{W}_{u,v}\right] \le n^{-r/2}.$$

Combining Claim 6.9 and Claim 6.10, we immediately have

$$\mathop{\mathbb{E}}_{\substack{1\le u_1<v_1\le n\\ \vdots\\ 1\le u_k<v_k\le n}} \mathbb{E}\left[\prod_{i\in[k]} \boldsymbol{W}_{u_i,v_i}\right] \le n^{-k/4} + n^{-k/4} \le 2n^{-\ell/8}. \tag{30}$$

Putting (30) and (29) together and recall that we assumed $k \ge \ell/2$, we have

$$\sum_{k=\ell/2}^{\ell} \left[\mathop{\mathbb{E}}_{\substack{1\le u_1<v_1\le n\\ \vdots\\ 1\le u_k<v_k\le n}} \mathop{\mathbb{E}}_{w_1,\ldots,w_{\ell-k}\in[n]} \mathbb{E}\left[\prod_{i\in[k]} \boldsymbol{W}_{u_i,v_i} \cdot \prod_{i\in[\ell-k]} \boldsymbol{B}_{w_i}\right]\right] \le (\ell/2+1)\cdot 2n^{-\ell/8} \le 2\ell \cdot n^{-\ell/8}. \tag{31}$$

Proving Item (1). Now, plugging (27) and (30) into (24), we have

$$\mathbb{E}[(\boldsymbol{Z}/M)^\ell] \le m^{2\ell} \cdot 3\ell \cdot n^{-\ell/8}. \tag{32}$$

Recall that $\boldsymbol{Z}/M = 1 - \boldsymbol{Y}/M$, and hence Item (1) is equivalent to

$$\Pr[\boldsymbol{Z}/M > 1/2] \le n^{-\log n}. \tag{33}$$

To prove (33), we now set $\ell = \log^2 n$. By Markov's inequality, we have

$$\begin{aligned}
\Pr[\boldsymbol{Z}/M > 1/2] &= \Pr[(\boldsymbol{Z}/M)^\ell > 2^{-\ell}] \\
&\le 2^\ell \cdot \mathbb{E}[(\boldsymbol{Z}/M)^\ell] \\
&\le 2^\ell \cdot m^{2\ell} \cdot 3\ell \cdot n^{-\ell/8} && \text{(By (32))} \\
&\le 2^\ell \cdot 3\ell \cdot n^{-\ell/8+2\ell\cdot\gamma} && (m = n^\gamma) \\
&\le 2^\ell \cdot 3\ell \cdot n^{-\ell/10} && (1/8 - 2\gamma > 1/10) \\
&\le n^{-\log n}. && (\ell = \log^2 n)
\end{aligned}$$

□

6.4 Proof of Item (2) of Lemma 6.6

Reminder of Item (2) of Lemma 6.6. *Let $\gamma = 10^{-3}$. For all $T \in \mathbb{N}_{\ge 1}$, for all sufficiently large $n \in 2\mathbb{N}_{\ge 1}$ the following holds: letting $m = n^\gamma$, for every $L \in [\log n]$ and valid path pattern $\vec{\tau} \in [T]^L$, it holds that*

$$\Pr_{G \leftarrow \mathcal{G}_{n,T}}\left[|\mathbb{X}^{\{1\}}_{m,\vec{\tau}}(G)| \le \frac{1}{2} \cdot \binom{n-1}{m-1} \wedge \mathsf{Path}(G, 1, \vec{\tau}) \in \mathbb{L}_{\vec{\tau}}(G)\right] \le n^{-\log n}.$$

Proof.

Notation and setup. We define random variables $\boldsymbol{G}$, $\boldsymbol{W}_{u,v}$, $\boldsymbol{B}_u$, $\boldsymbol{Y}_S$, and $\boldsymbol{Z}_S$ in the same way as in the proof of Item (1) of Lemma 6.6. We will however define $\boldsymbol{Z}$, $\boldsymbol{Y}$, and M differently as below.

Let $\mathcal{S}^{\{1\}} = \{S : S \in \binom{[n]}{m} \wedge 1 \in S\}$. We define

$$\boldsymbol{Y} = \sum_{S \in \mathcal{S}^{\{1\}}} \boldsymbol{Y}_S, \quad \boldsymbol{Z} = \sum_{S \in \mathcal{S}^{\{1\}}} \boldsymbol{Z}_S, \quad \text{and} \quad M = \binom{n-1}{m-1}.$$

By definition, we have

$$\boldsymbol{Y} = |\mathbb{X}^{\{1\}}_{m,\vec{\tau}}(\boldsymbol{G})|.$$

Recall that $\boldsymbol{B}_u$ is the indicator that $\mathsf{Path}(\boldsymbol{G}, u, \vec{\tau})$ is not simple. Our goal can then be restated as proving

$$\Pr[\boldsymbol{Z}/M > 1/2 \wedge \boldsymbol{B}_1 = 0] \le n^{-\log n},$$

which is equivalent to

$$\Pr[\boldsymbol{Z}/M \cdot (1 - \boldsymbol{B}_1) > 1/2] \le n^{-\log n}.$$

We will prove the above by upper bounding

$$\mathbb{E}\left[\left(\boldsymbol{Z}/M \cdot (1 - \boldsymbol{B}_1)\right)^{\ell}\right], \tag{34}$$

for some parameter $\ell < n^{1/3}$ to be specified later.

Expanding (34). Recall that

$$\boldsymbol{Z}_S \le \sum_{u,v \in S,\ u<v} \boldsymbol{W}_{u,v} + \sum_{u \in S} \boldsymbol{B}_u.$$

We have

$$\begin{aligned}
\boldsymbol{Z} &\le \sum_{S \in \mathcal{S}^{\{1\}}} \left[\sum_{u,v \in S,\ u<v} \boldsymbol{W}_{u,v} + \sum_{u \in S} \boldsymbol{B}_u \right] \\
&= \sum_{2 \le u < v \le n} \boldsymbol{W}_{u,v} \cdot \binom{n-3}{m-3} + \sum_{2 \le u \le n} \boldsymbol{B}_u \cdot \binom{n-2}{m-2} + \sum_{2 \le v \le n} \boldsymbol{W}_{1,v} \cdot \binom{n-2}{m-2} + \boldsymbol{B}_1 \cdot \binom{n-1}{m-1}.
\end{aligned}$$

Consequently,

$$\boldsymbol{Z}(1 - \boldsymbol{B}_1) \le \sum_{2 \le u < v \le n} \boldsymbol{W}_{u,v} \cdot \binom{n-3}{m-3} + \sum_{2 \le u \le n} \boldsymbol{B}_u \cdot \binom{n-2}{m-2} + \sum_{2 \le v \le n} \boldsymbol{W}_{1,v} \cdot \binom{n-2}{m-2}.$$

Recall that $M = \binom{n-1}{m-1}$, dividing both sides by M, we further have

$$\begin{aligned}
\boldsymbol{Z}(1 - \boldsymbol{B}_1)/M &\le \sum_{2 \le u < v \le n} \boldsymbol{W}_{u,v} \cdot \frac{(m-1)(m-2)}{(n-1)(n-2)} + \sum_{2 \le u \le n} \boldsymbol{B}_u \cdot \frac{m-1}{n-1} + \sum_{2 \le v \le n} \boldsymbol{W}_{1,v} \cdot \frac{m-1}{n-1}. \\
&\le \mathop{\mathbb{E}}_{2 \le u < v \le n} \boldsymbol{W}_{u,v} \cdot m^2 + \mathop{\mathbb{E}}_{2 \le u \le n} \boldsymbol{B}_u \cdot m + \mathop{\mathbb{E}}_{2 \le v \le n} \boldsymbol{W}_{1,v} \cdot m.
\end{aligned}$$

Taking the ℓ-th power and then the expectation of both sides, we have

$$\begin{aligned}
&\mathbb{E}\left[\left(\boldsymbol{Z}/M \cdot (1 - \boldsymbol{B}_u)\right)^{\ell}\right] \\
&\le 3^{\ell} \cdot m^{2\ell} \sum_{\alpha,\beta,\theta : \alpha+\beta+\theta=\ell} \mathop{\mathbb{E}}_{\substack{2 \le u_1 < v_1 \le n \\ \vdots \\ 2 \le u_\alpha < v_\alpha \le n}} \mathop{\mathbb{E}}_{w_1,\ldots,w_\beta \in \{2,\ldots,n\}} \mathop{\mathbb{E}}_{z_1,\ldots,z_\theta \in \{2,\ldots,n\}} \mathbb{E}\left[\prod_{i \in [\alpha]} \boldsymbol{W}_{u_i,v_i} \cdot \prod_{i \in [\beta]} \boldsymbol{B}_{w_i} \cdot \prod_{i \in [\theta]} \boldsymbol{W}_{1,z_i} \right].
\end{aligned}$$

We divide the triples (α, β, θ) into three categories: (1) $\alpha \ge \ell/3$, (2) $\alpha < \ell/3$ and $\beta \ge \ell/3$, and (3) $\alpha, \beta < \ell/3$ and $\theta \ge \ell/3$, and bound them separately. Let $\mathcal{I}_1, \mathcal{I}_2$, and $\mathcal{I}_3$ be the set of

triples (α,β,θ) that satisfies (1), (2), and (3), respectively.

The case when $(\alpha,\beta,\theta)\in\mathcal{I}_1$. First, we have

$$\sum_{(\alpha,\beta,\theta)\in\mathcal{I}_1}\mathop{\mathbb{E}}_{\substack{2\le u_1<v_1\le n\\ \vdots\\ 2\le u_\alpha<v_\alpha\le n}}\mathop{\mathbb{E}}_{w_1,\dots,w_\beta\in\{2,\dots,n\}}\mathop{\mathbb{E}}_{z_1,\dots,z_\theta\in\{2,\dots,n\}}\mathbb{E}\left[\prod_{i\in[\alpha]}\boldsymbol{W}_{u_i,v_i}\cdot\prod_{i\in[\beta]}\boldsymbol{B}_{w_i}\cdot\prod_{i\in[\theta]}\boldsymbol{W}_{1,z_i}\right]$$
$$\le\sum_{(\alpha,\beta,\theta)\in\mathcal{I}_1}\mathop{\mathbb{E}}_{\substack{2\le u_1<v_1\le n\\ \vdots\\ 2\le u_\alpha<v_\alpha\le n}}\mathbb{E}\left[\prod_{i\in[\alpha]}\boldsymbol{W}_{u_i,v_i}\right].$$

By Claim 6.9 and Claim 6.10, we have

$$\mathop{\mathbb{E}}_{\substack{2\le u_1<v_1\le n\\ \vdots\\ 2\le u_\alpha<v_\alpha\le n}}\mathbb{E}\left[\prod_{i\in[\alpha]}\boldsymbol{W}_{u_i,v_i}\right]\le 2(n-1)^{-\alpha/4}\le 2(n-1)^{-\ell/12}.$$

Therefore

$$\sum_{(\alpha,\beta,\theta)\in\mathcal{I}_1}\mathop{\mathbb{E}}_{\substack{2\le u_1<v_1\le n\\ \vdots\\ 2\le u_\alpha<v_\alpha\le n}}\mathop{\mathbb{E}}_{w_1,\dots,w_\beta\in\{2,\dots,n\}}\mathop{\mathbb{E}}_{z_1,\dots,z_\theta\in\{2,\dots,n\}}\mathbb{E}\left[\prod_{i\in[\alpha]}\boldsymbol{W}_{u_i,v_i}\cdot\prod_{i\in[\beta]}\boldsymbol{B}_{w_i}\cdot\prod_{i\in[\theta]}\boldsymbol{W}_{1,z_i}\right]\le\ell^2\cdot 2(n-1)^{-\ell/12}. \tag{35}$$

The case when $(\alpha,\beta,\theta)\in\mathcal{I}_2$. Next, we have

$$\sum_{(\alpha,\beta,\theta)\in\mathcal{I}_2}\mathop{\mathbb{E}}_{\substack{2\le u_1<v_1\le n\\ \vdots\\ 2\le u_\alpha<v_\alpha\le n}}\mathop{\mathbb{E}}_{w_1,\dots,w_\beta\in\{2,\dots,n\}}\mathop{\mathbb{E}}_{z_1,\dots,z_\theta\in\{2,\dots,n\}}\mathbb{E}\left[\prod_{i\in[\alpha]}\boldsymbol{W}_{u_i,v_i}\cdot\prod_{i\in[\beta]}\boldsymbol{B}_{w_i}\cdot\prod_{i\in[\theta]}\boldsymbol{W}_{1,z_i}\right]$$
$$\le\sum_{(\alpha,\beta,\theta)\in\mathcal{I}_2}\mathop{\mathbb{E}}_{w_1,\dots,w_\beta\in\{2,\dots,n\}}\mathbb{E}\left[\prod_{i\in[\beta]}\boldsymbol{B}_{w_i}\right].$$

By (26) and Claim 6.8, we have

$$\mathop{\mathbb{E}}_{w_1,\dots,w_\beta\in[n]}\mathbb{E}\left[\prod_{i\in[\beta]}\boldsymbol{B}_{w_i}\right]\le 2(n-1)^{-\beta/4}\le 2(n-1)^{-\ell/12}.$$

Therefore

$$\sum_{(\alpha,\beta,\theta)\in\mathcal{I}_2} \mathop{\mathbb{E}}_{\substack{2\le u_1<v_1\le n\\ \vdots\\ 2\le u_\alpha<v_\alpha\le n}} \mathop{\mathbb{E}}_{w_1,\dots,w_\beta\in\{2,\dots,n\}} \mathop{\mathbb{E}}_{z_1,\dots,z_\theta\in\{2,\dots,n\}} \mathbb{E}\left[\prod_{i\in[\alpha]} \boldsymbol{W}_{u_i,v_i}\cdot\prod_{i\in[\beta]}\boldsymbol{B}_{w_i}\cdot\prod_{i\in[\theta]}\boldsymbol{W}_{1,z_i}\right] \le \ell^2\cdot 2(n-1)^{-\ell/12}. \tag{36}$$

The case when $(\alpha,\beta,\theta)\in\mathcal{I}_3$. Finally, we have

$$\begin{aligned}&\sum_{(\alpha,\beta,\theta)\in\mathcal{I}_3} \mathop{\mathbb{E}}_{\substack{2\le u_1<v_1\le n\\ \vdots\\ 2\le u_\alpha<v_\alpha\le n}} \mathop{\mathbb{E}}_{w_1,\dots,w_\beta\in\{2,\dots,n\}} \mathop{\mathbb{E}}_{z_1,\dots,z_\theta\in\{2,\dots,n\}} \mathbb{E}\left[\prod_{i\in[\alpha]} \boldsymbol{W}_{u_i,v_i}\cdot\prod_{i\in[\beta]}\boldsymbol{B}_{w_i}\cdot\prod_{i\in[\theta]}\boldsymbol{W}_{1,z_i}\right]\\ \le &\sum_{(\alpha,\beta,\theta)\in\mathcal{I}_3} \mathop{\mathbb{E}}_{z_1,\dots,z_\theta\in\{2,\dots,n\}} \mathbb{E}\left[\prod_{i\in[\theta]}\boldsymbol{W}_{1,z_i}\right].\end{aligned}$$

By (26), we have

$$\Pr_{z_1,\dots,z_\theta\in\{2,\dots,n\}}[|\{z_i\}_{i\in[\theta]}|\ge\theta/2]\ge 1-(n-1)^{-\theta/4}. \tag{37}$$

We also need the following claim.

Claim 6.11. *For every $r>(\log n+1)^2$ and $S\in\binom{\{2,\dots,n\}}{r}$, it holds that*

$$\mathbb{E}\left[\prod_{u\in S}\boldsymbol{W}_{1,u}\right]=0.$$

Proof. Recall that $\boldsymbol{W}_{1,u}$ means that both $\mathsf{Path}(\boldsymbol{G},1,\vec{\tau})$ and $\mathsf{Path}(\boldsymbol{G},u,\vec{\tau})$ are simple and they share at least one vertex. Since one vertex μ can only be on $\mathsf{Path}(\boldsymbol{G},u,\vec{\tau})$ for at most $L+1$ many vertices u (since μ's position in the pattern $\vec{\tau}$ completely determines the path $\mathsf{Path}(\boldsymbol{G},u,\vec{\tau})$), we know that $\mathsf{Path}(\boldsymbol{G},1,\vec{\tau})$ can intersect with $\mathsf{Path}(\boldsymbol{G},u,\vec{\tau})$ for at most $(L+1)^2\le(\log n+1)^2$ many distinct u. Hence, if $|S|>(\log n+1)^2$, it must hold that $\prod_{u\in S}\boldsymbol{W}_{1,u}=0$, which completes the proof. □

Now, we set $\ell=\log^3 n$. Combining (37) and Claim 6.11, we have that

$$\mathop{\mathbb{E}}_{z_1,\dots,z_\theta\in\{2,\dots,n\}}\mathbb{E}\left[\prod_{i\in[\theta]}\boldsymbol{W}_{1,z_i}\right]\le(n-1)^{-\ell/12}.$$

Therefore

$$\sum_{(\alpha,\beta,\theta)\in\mathcal{I}_3} \mathop{\mathbb{E}}_{\substack{2\le u_1<v_1\le n\\ \vdots \\ 2\le u_\alpha<v_\alpha\le n}} \mathop{\mathbb{E}}_{w_1,\ldots,w_\beta\in\{2,\ldots,n\}} \mathop{\mathbb{E}}_{z_1,\ldots,z_\theta\in\{2,\ldots,n\}} \mathbb{E}\left[\prod_{i\in[\alpha]} \boldsymbol{W}_{u_i,v_i}\cdot\prod_{i\in[\beta]}\boldsymbol{B}_{w_i}\cdot\prod_{i\in[\theta]}\boldsymbol{W}_{1,z_i}\right] \le \ell^2\cdot(n-1)^{-\ell/12}. \tag{38}$$

Putting (35), (36), and (38) together, we have

$$\mathbb{E}\left[\left(\boldsymbol{Z}/M\cdot(1-\boldsymbol{B}_u)\right)^\ell\right] \le 3^\ell\cdot m^{2\ell}\cdot 5\cdot\ell^2\cdot(n-1)^{-\ell/12}.$$

Finally, by Markov's inequality, we have that

$$\begin{aligned}\Pr[(\boldsymbol{Z}/M\cdot(1-\boldsymbol{B}_u))^\ell > 2^{-\ell}] &\le 2^\ell\cdot 3^\ell\cdot m^{2\ell}\cdot 5\cdot\ell^2\cdot(n-1)^{-\ell/12}\\ &\le 7^\ell\cdot m^{2\ell}\cdot n^{-\ell/13}\\ &\le 7^\ell\cdot n^{-\ell/13+2\ell\cdot\gamma}\\ &\le n^{-\log n}. \qquad (-1/13+2\gamma<0 \text{ and } \ell=\log^3 n)\end{aligned}$$

This completes the proof.

□

6.5 Omitted Proofs

We first prove Claim 6.9 (restated below).

Reminder of Claim 6.9. *For every $k\le n^{1/3}$, it holds that*

$$\Pr_{\substack{1\le u_1<v_1\le n\\ \vdots\\ 1\le u_k<v_k\le n}}\left[\exists \text{ a valid set } S \text{ s.t. } |S|\ge k/2 \text{ and } S\subseteq\{(u_i,v_i)\}_{i\in[k]}\right]\ge 1-n^{-k/4}.$$

Proof. We consider the following greedy algorithm that constructs a valid subset $\widetilde{S}$ of $\{(u_i,v_i)\}_{i\in[k]}$:

- $\widetilde{S}=\emptyset$ initially.
- For every $i\in[k]$, if

$$\{u_i,v_i\}\cap\left(\{u_\ell\}_{\ell\in[i-1]}\cup\{v_\ell\}_{\ell\in[i-1]}\right)=\emptyset,$$

  we add (u_i,v_i) to $\widetilde{S}$ (*i.e.*, if (u_i,v_i) does not share any element with all previous $i-1$ pairs).

It is straightforward to verify that $\widetilde{S}$ is always a valid subset of $\{(u_i, v_i)\}_{i\in[k]}$.

To prove the claim, it suffices to prove that

$$\Pr_{\substack{1\le u_1<v_1\le n\\ \vdots \\ 1\le u_k<v_k\le n}}\left[|\widetilde{S}|\le k/2\right]\le n^{-k/4}.$$

We note that $|\widetilde{S}|\le k/2$ happens only if there exists a subset $W\in\binom{[k]}{k/2}$ such that in the greedy algorithm for constructing $\widetilde{S}$, for every $i\in W$, (u_i, v_i) is not added to $\widetilde{S}$. We say such a subset W is bad.

We then have

$$\Pr_{\substack{1\le u_1<v_1\le n\\ \vdots \\ 1\le u_k<v_k\le n}}\left[|\widetilde{S}|\le k/2\right]\le \sum_{W\in\binom{[k]}{k/2}}\Pr_{\substack{1\le u_1<v_1\le n\\ \vdots \\ 1\le u_k<v_k\le n}}\left[W \text{ is bad}\right].$$

Now, we note that conditioning on the values of $((u_\ell, v_\ell))_{\ell\in[i-1]}$, the probability that (u_i, v_i) is not added to $\widetilde{S}$ is at most

$$\begin{aligned}
1-\binom{(n-2(i-1))}{2}\Big/\binom{n}{2} &\le 1-\frac{n-2(i-1)}{n}\cdot\frac{n-2(i-1)-1}{n-1}\\
&\le 1-\left(1-\frac{2(i-1)}{n}\right)\cdot\left(1-\frac{2(i-1)}{n-1}\right)\\
&\le \frac{2(i-1)}{n}+\frac{2(i-1)}{n-1}\le 6k/n.
\end{aligned}$$

So the probability that $W\in\binom{[k]}{[k/2]}$ is bad can be bounded by $(6k/n)^{k/2}$, and we have

$$\begin{aligned}
\Pr_{\substack{1\le u_1<v_1\le n\\ \vdots \\ 1\le u_k<v_k\le n}}\left[|\widetilde{S}|\le k/2\right] &\le \binom{k}{k/2}\cdot(6k/n)^{k/2}\\
&\le 2^k\cdot(6k/n)^{k/2}\\
&\le (24k/n)^{k/2}\\
&\le n^{-k/4}. && (k\le n^{1/3})
\end{aligned}$$

This completes the proof. □

Before proving Claim 6.8, we give a template for constructing a sampling procedure for the distribution $\mathcal{G}_{n,T}$.

Sampler Samp_F for $\mathcal{G}_{n,T}$

1. Initially $M^1, M^2, \ldots, M^T$ are all empty sets, and $R^1, R^2, \ldots, R^T$ are all $[n]$.
2. While not all of R^i's are empty
   (a) Pick a vertex $u \in [n]$ and an index $i \in [T]$ such that $u \in R^i$.
   (b) Sample uniformly at random a vertex v from $R^i \setminus \{u\}$.
   (c) Add (u, v) to M^i, and remove u and v from R^i.
3. Output $G = ([n], M^1 \circ M^2 \circ \ldots M^T)$.

Picking rule Formally, in Step (2.a), the pair (u, i) is determined by a (potentially probabilistic) function F that maps the current partial matchings $(M^1, \ldots, M^T)$ to an element $(u, i) \in [n] \times [T]$ such that $u \in R^i$.[a] We call such a function F a *valid picking rule*.

[a] Note that for each $i \in [T]$, R_i is determined by M^i.

We have the following observation.

Observation 6.12. *For any valid picking rule F, the output distribution of Samp_F is identical to $\mathcal{G}_{n,T}$.*

Now we are ready to prove Claim 6.8.

Reminder of Claim 6.8. *For every $r \leq n^{1/3}$ and $S \in \binom{[n]}{r}$, it holds that*

$$\mathbb{E}\left[\prod_{u \in S} \boldsymbol{B}_u\right] \leq n^{-r/2}.$$

Proof. Let $s_1 < s_2 < s_3 < \cdots < s_r$ be the elements of S. In the following we analyze a particular sampler SampB^S that instantiates the sampling template Samp by fixing a particular picking rule.

Sampler SampB^S for $\mathcal{G}_{n,T}$

1. Initially $M^1, M^2, \ldots, M^T$ are all empty sets, and $R^1, R^2, \ldots, R^T$ are all $[n]$. Let $K = \emptyset$.
2. For each $i \in [r]$:
   (a) If $s_i \in K$, go to label **END**.
   (b) Set $s_{i,0} = s_i$ and add s_i to K.

(c) For each $j \in [L]$:

i. If $s_{i,j-1} \notin R^{\tau_j}$, go to label **END**.

ii. Sample uniformly at random a vertex v from $R^{\tau_j} \setminus \{s_{i,j-1}\}$.

iii. Add $(s_{i,j-1}, v)$ to M^{τ_j}, and remove $s_{i,j-1}$ and v from R^{τ_j}.

iv. Set $s_{i,j} = v$. If $v \in K$, go to label **END**.

v. Add v to K.

(d) Go to label **STOP**.[a]

(e) **END**

3. **STOP:** While not all of R^i's are empty

(a) Pick a vertex $u \in [n]$ and an index $i \in [T]$ such that $u \in R^i$.

(b) Sample uniformly at random a vertex v from $R^i \setminus \{u\}$.

(c) Add (u, v) to M^i, and remove u and v from R^i.

4. Output $G = ([n], M^1 \circ M^2 \circ \dots M^T)$.

[a] At this point, we know that $\mathsf{Path}(G, s_i, \vec{\tau})$ is simple.

Intuitively speaking, in the above sampler, for each s_i, we try to first sample the walk $\mathsf{Path}(G, s_i, \vec{\tau})$, and maintain K as the set of visited vertices. Whenever we encounter a vertex that is already visited before (by the current s_i or earlier s_j for $j < i$), we simply stop sampling the current walk. Also, when we have successfully sampled the whole walk $\mathsf{Path}(G, s_i, \vec{\tau})$ without encountering any already visited vertices, we know that $\mathsf{Path}(G, s_i, \vec{\tau})$ is simple, meaning that $\boldsymbol{B}_{s_i} = 0$, and we can already go to **STOP** to sample the rest of the graphs since we already know that $\prod_{u \in S} \boldsymbol{B}_u = 0$.

Now we formally analyze SampB^S. Let $\mathcal{E}_i$ be the event that it reaches **END** at the i-th iteration of Step (2) (*i.e.*, it does not reach Step (2.d) and go directly to **STOP**). By previous discussions, we note that $\neg\mathcal{E}_r$ implies that we had sampled a simple path with pattern $\vec{\tau}$ starting from some s_i, and therefore $\prod_{u \in S} \boldsymbol{B}_u = 0$. Hence, we have

$$\mathbb{E}\left[\prod_{u \in S} \boldsymbol{B}_u\right] \leq \Pr[\mathcal{E}_r].$$

Hence it suffices to upper bound $\Pr[\mathcal{E}_r]$, we will indeed prove

Claim 6.13. *For every $i \in [r]$, it holds that*

$$\Pr[\mathcal{E}_i | \mathcal{E}_{i-1}] \leq 1/\sqrt{n}.$$

Claim 6.13 immediately implies that

$$\mathbb{E}\left[\prod_{i\in S}\boldsymbol{B}_i\right] \le n^{-r/2},$$

which completes the proof of Claim 6.8.

Finally, we prove Claim 6.13.

Proof of Claim 6.13. Fix $i \in [r]$, conditioning on the event $\mathcal{E}_{i-1}$, let $\boldsymbol{K}_{i-1}$ be the set K at the end of $(i-1)$-th iteration of Step (2). We can see that $\{s_1, s_2, \ldots, s_{i-1}\} \subseteq \boldsymbol{K}$ and $|\boldsymbol{K}| \le (i-1)\cdot(L+1)$. Now we further conditioning on the size n_K of $\boldsymbol{K}_{i-1}$. We can see that $\boldsymbol{K}_{i-1} \setminus \{s_1, s_2, \ldots, s_{i-1}\}$ is a uniformly random subset of $[n] \setminus \{s_1, s_2 \ldots, s_{i-1}\}$ with size $n_K - (i-1) \le i \cdot L$.

Now we lower bound $\Pr[\neg\mathcal{E}_i|\mathcal{E}_{i-1}]$. We note that this happens if (1) $s_i \notin \boldsymbol{K}_{i-1}$ and (2) $\mathsf{Path}(G, s_i, \vec{\tau})$ is simple and does not visited any vertices in $\boldsymbol{K}_{i-1}$. By our previous discussion and a direct calculation, $s_i \notin \boldsymbol{K}_{i-1}$ happens with probability at least $1 - \frac{i\cdot L}{n/2}$.

We then conditioning on the event $s_i \notin \boldsymbol{K}_{i-1}$, and also the value of $\boldsymbol{K}_{i-1}$ to be K_{i-1}. We can then calculate the probability of $\mathsf{Path}(G, s_i, \vec{\tau})$ is simple and does not visited any vertices in $\boldsymbol{K}_{i-1}$ is at least

$$\left(1 - \frac{i\cdot L}{n/2}\right)^L.$$

Putting everything together and recall that $L \le \log n$ and $i \le r \le n^{1/3}$, we have

$$\Pr[\neg\mathcal{E}_i|\mathcal{E}_{i-1}] \ge \left(1 - \frac{i\cdot L}{n/2}\right)^{L+1} \ge 1 - 1/\sqrt{n},$$

which completes the proof. □

□

Reminder of Claim 6.10. *For every $r \le n^{1/3}$. Let S be a valid set of pairs such that $|S| = r$. It holds that*

$$\mathbb{E}\left[\prod_{(u,v)\in S}\boldsymbol{W}_{u,v}\right] \le n^{-r/2}.$$

Proof. Let $(u_1, v_1), (u_2, v_2), \ldots, (u_r, v_r)$ be the elements of S. We will analyze the following sampler SampW.

Sampler SampW^S for $\mathcal{G}_{n,T}$

1. Initially $M^1, M^2, \ldots, M^T$ are all empty sets, and $R^1, R^2, \ldots, R^T$ are all $[n]$. Let $K = \emptyset$.

2. For each $i \in [r]$:

   (a) For each $\mu \in \{u_i, v_i\}$

      i. If $\mu \in K$, go to label **END**.

      ii. Set $\mu_{\mathsf{cur}} = \mu$.

      iii. For each $j \in [L]$:

         A. If $\mu_{\mathsf{cur}} \notin R^{\tau_j}$, go to label **END**.

         B. Sample uniformly at random a vertex ν from $R^{\tau_j} \setminus \{\mu_{\mathsf{cur}}\}$.

         C. Add $(\mu_{\mathsf{cur}}, \nu)$ to M^{τ_j}, and remove μ_{cur} and ν from R^{τ_j}.

         D. Set $\mu_{\mathsf{cur}} = \nu$. If $\nu \in K$, go to label **END**.

         E. Add ν to K.

   (b) go to label **STOP**.[a]

   (c) **END**

   (d) Add u_i and v_i to K.

3. **STOP:** While not all of R^i's are empty

   (a) Pick a vertex $u \in [n]$ and an index $i \in [T]$ such that $u \in R^i$.

   (b) Sample uniformly at random a vertex v from $R^i \setminus \{u\}$.

   (c) Add (u, v) to M^i, and remove u and v from R^i.

4. Output $G = ([n], M^1 \circ M^2 \circ \dots M^T)$.

[a] At this point, we know that both $\mathsf{Path}(G, u_i, \vec{\tau})$ and $\mathsf{Path}(G, v_i, \vec{\tau})$ are simple, and they do not share any vertices.

Intuitively speaking, in the above sampler, we maintain K as the set of visited vertices. For each $i \in [r]$, we try to first sample the walk $\mathsf{Path}(G, u_i, \vec{\tau})$. Whenever we encounter a vertex that is already visited before (by the current u_i or earlier u_j, v_j for $j < i$), we simply stop the sampling the induced walk. When we successfully sampled the whole walk $\mathsf{Path}(G, u_i, \vec{\tau})$ without encountering any already visited vertices, we know that $\mathsf{Path}(G, u_i, \vec{\tau})$ is simple. We then similarly try to sample the walk $\mathsf{Path}(G, v_i, \vec{\tau})$. If we successfully reach Step (2.b), it means that both $\mathsf{Path}(G, u_i, \vec{\tau})$ and $\mathsf{Path}(G, v_i, \vec{\tau})$ are simple, and they do not share any vertices, meaning that $W_{u_i, v_i} = 0$. We can then go to **STOP** to sample the rest of the graphs since we already know that $\prod_{(u,v)\in S} \boldsymbol{W}_{u,v} = 0$.

Our proof below follows the same structure of that of Claim 6.8. Now we formally analyze SampW^S. Let $\mathcal{E}_i$ be the event that it reaches **END** at the i-th iteration of Step (2) (*i.e.*, it does not reach Step (2.b) and then go directly to **STOP**). By previous discussions, we note

that $\neg\mathcal{E}_r$ implies that $\prod_{(u,v)\in S} \boldsymbol{W}_{u,v} = 0$. Hence, we have

$$\mathbb{E}\left[\prod_{(u,v)\in S} \boldsymbol{W}_{u,v}\right] \leq \Pr[\mathcal{E}_r].$$

Hence it suffices to upper bound $\Pr[\mathcal{E}_r]$, we will indeed prove

Claim 6.14. *For every $i \in [r]$, it holds that*

$$\Pr[\mathcal{E}_i|\mathcal{E}_{i-1}] \leq 1/\sqrt{n}.$$

Claim 6.14 immediately implies that

$$\mathbb{E}\left[\prod_{(u,v)\in S} \boldsymbol{W}_{u,v}\right] \leq n^{-r/2},$$

which completes the proof of Claim 6.10.

Finally, we prove Claim 6.14.

Proof of Claim 6.14. Fix $i \in [r]$, conditioning on the event $\mathcal{E}_{i-1}$, let $\boldsymbol{K}_{i-1}$ be the set K at the end of $(i-1)$-th iteration of Step (2). Let $V_{i-1} = \{u_\ell\}_{\ell\in[i-1]} \cup \{v_\ell\}_{\ell\in[i-1]}$. We can see that $V_{i-1} \subseteq \boldsymbol{K}$ and $|\boldsymbol{K}| \leq 2\cdot(i-1)\cdot(L+1)$. Now we further conditioning on the size n_K of $\boldsymbol{K}_{i-1}$. We can see that $\boldsymbol{K}_{i-1} \setminus V_{i-1}$ is a uniformly random subset of $[n] \setminus V_{i-1}$ with size $n_K - 2\cdot(i-1) \leq 2\cdot i\cdot L$.

Now we lower bound $\Pr[\neg\mathcal{E}_i|\mathcal{E}_{i-1}]$. We note that this happens if (1) $u_i \notin \boldsymbol{K}_{i-1}$ and (2) $\mathsf{Path}(G, u_i, \vec{\tau})$ is simple and does not visited any vertices in $\boldsymbol{K}_{i-1}$, (3) $v_i \notin \boldsymbol{K}_{i-1} \cup \mathsf{Path}(G, u_i, \vec{\tau})$ and (4) $\mathsf{Path}(G, v_i, \vec{\tau})$ is simple and does not visited any vertices in $\boldsymbol{K}_{i-1} \cup \mathsf{Path}(G, u_i, \vec{\tau})$.

Note that conditioning on both of (1) and (2) hold, $\mathsf{Path}(G, u_i, \vec{\tau}) \setminus \{u_i\}$ distributes as a uniform size-L subset of $[n] \setminus (\boldsymbol{K}_{i-1} \cup \{u_i\})$.

Hence, by a similar calculation as in Claim 6.13, we have

$$\Pr[\neg\mathcal{E}_i|\mathcal{E}_{i-1}] \geq \left(1 - \frac{O(i\cdot L)}{n}\right)^{2(L+1)} \geq 1 - 1/\sqrt{n},$$

which completes the proof. □

□

A Proof of Lemma 6.2

Reminder of Lemma 6.2. *There exist $\varepsilon, \delta \in (0,1)$ such that for all sufficiently large $n \in \mathbb{N}$ the following holds: for all $d \in [\log n]$, $p \le (d-6)/2$, and all p-round communication protocols Π with at most n^{ε} communication complexity, it holds that*

$$\Pr_{\vec{\pi} \leftarrow \mathcal{P}_{n,d}} [\Pi(\vec{\pi}) \in \mathbb{P}(\vec{\pi})] \le n^{1-\delta d/p},$$

where $\Pi(\vec{\pi})$ denotes the output of Π when Alice gets the input $\pi_1, \pi_3, \ldots, \pi_{d-1}$ and Bob gets the input $\pi_2, \pi_4, \ldots, \pi_d$.

To prove the lemma, we will exploit the direct product structure in the problem: We can view the length-d output as $\Theta(d/p)$ segments of length $\Theta(p)$, and argue that each segment is hard to compute with n^{ε} communication in p rounds, then apply a direct product theorem. We first formally prove this reduction.

We define the pointer chasing problem with a fixed starting vertex s as follows.

Definition A.1. *Let $n, t \in \mathbb{N}$ such that t is even. In the $\mathsf{PC}_{n,t}$ problem, there are two players Alice and Bob, a start vertex $s \in [n]$ and t permutations $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_t)$ on $[n]$. Both Alice and Bob know s. Alice also gets all the odd permutations $\pi_1, \pi_3, \ldots, \pi_{t-1}$, and Bob also gets all the even permutations $\pi_2, \pi_4, \ldots, \pi_t$. Let $\pi_{\le i} = \pi_i \circ \pi_{i-1} \circ \ldots \circ \pi_1$. Their goal is to output the path $\mathsf{path}_{\vec{\pi}}(s) = (s, \pi_{\le 1}(s), \pi_{\le 2}(s), \ldots, \pi_{\le t}(s))$.*

Lemma A.2. *Suppose there is a p-round protocol with at most S bits of communication that solves $\mathsf{ASPC}_{n,d}$ with probability greater than $n^{1-\delta d/p}$ when $\vec{\pi}$ is sampled from $\mathcal{P}_{n,d}$. Then for $k, t \in \mathbb{N}$ such that t is even and $k(2t+2) \le d$, there is a p-round protocol with at most S bits of communication such that given $\vec{\pi}_1, \vec{\pi}_2, \ldots, \vec{\pi}_k \leftarrow \mathcal{P}_{n,t}$ and $s_1, \ldots, s_k \in_{unif} [n]$ independently, the protocol outputs $(\mathsf{path}_{\vec{\pi}_1}(s_1), \ldots, \mathsf{path}_{\vec{\pi}_k}(s_k))$ with probability greater than $n^{-\delta d/p}$.*

Proof. Fix a protocol Π that solves $\mathsf{ASPC}_{n,d}$ with the claimed properties, and fix k and t.

Consider the following protocol Π' that solves k independent instances of $\mathsf{PC}_{n,t}$.

Protocol Π' for k instances of $\mathsf{PC}_{n,t}$

Inputs: $s_1, \ldots, s_k$ and $\vec{\pi}_1, \ldots, \vec{\pi}_k$, where each $\vec{\pi}_i = (\pi_{i,1}, \ldots, \pi_{i,t})$
Construct d permutations $\vec{\pi}' = (\pi'_1, \ldots, \pi'_d)$

1. For $i \in [k]$ and $j \in [t]$, set $\pi'_{(i-1)(2t+2)+j}$ to $\pi_{i,j}$, and set $\pi'_{i(2t+2)-j}$ to $\pi_{i,j}^{-1}$
2. For $i \in [k]$, set $\pi'_{(i-1)(2t+2)+t+1}$ to the identify matching, and set $\pi'_{i(2t+2)}$ to any fixed matching such that $\pi'_{i(2t+2)}(s_i) = s_{i+1}$ (s_{k+1} is assumed to be 1)
3. For $i > k(2t+2)$, set π'_i to the identity matching

Construct d random permutations $\vec{\pi} = (\pi_1, \ldots, \pi_d)$

4. Alice knows all π'_i for odd $i \in [d]$ and Bob knows all π'_i for even $i \in [d]$
5. They use public random bits to sample random permutations $\tau_0, \ldots, \tau_d$
6. They set π_i to $\tau_i^{-1} \circ \pi'_i \circ \tau_{i-1}$

Simulate Π and compute outputs

7. Run Π on $\vec{\pi}$ and obtain a path $\mathsf{path}_{\vec{\pi}}(s) = (s, \pi_{\le 1}(s), \ldots, \pi_{\le d}(s))$ for some $s \in [n]$
8. If $\tau_0(s) \neq s_1$, then output FAIL
9. Otherwise, for $i \in [k]$ and $j \in [t]$, compute and output $\pi_{i,\le j}(s_i) = \tau_{(i-1)(2t+2)+j}(\pi_{\le (i-1)(2t+2)+j}(s))$

Since $\tau_0, \ldots, \tau_d$ are random independent permutations over $[n]$, all π_i generated in step 6 must be uniform and independent, following the same distribution as generated by $\mathcal{P}_{n,d}$. Thus, by our assumption on Π, it successfully outputs a path $\mathsf{path}_{\vec{\pi}}(s)$ with probability greater than $n^{1-\delta d/p}$, and the communication cost is as claimed.

Moreover, since we applied random permutations τ_i, s_1 becomes independent of $\vec{\pi}$. In particular, Π only takes $\vec{\pi}$ as input, which implies that s_1 is independent of s. Hence, we output FAIL in step 8 with probability $1/n$.

Finally, by construction, for $i \in [d]$, we always have $\tau_i(\pi_{\le i}(s)) = \pi'_i(\tau_{i-1}(\pi_{\le i-1}(s))) = \pi'_{\le i}(\tau_0(s))$. Thus, if $\tau_0(s) = s_1$, then $(\tau_0(s), \tau_1(\pi_{\le 1}(s)), \ldots)$ is the path on $\vec{\pi}'$ starting from s_1. By the construction of π', we have $\pi'_{\le (i-1)(2t+2)}(s_1) = s_i$. Thus, $\pi_{i,\le j}(s_i) = \pi'_{(i-1)(2t+2)+j}(s_1) = \tau_{(i-1)(2t+2)+j}(\pi_{(i-1)(2t+2)+j}(s))$. The protocol successfully computes $\mathsf{path}_{\vec{\pi}_i(s_i)}$ for all $i \in [k]$ with probability greater than $n^{-\delta d/p}$. □

A.1 Direct product

Next, we apply a generic direct product theorem to derive a protocol for $\mathsf{PC}_{n,t}$. The following argument is a simplification of [BRWY13].

Definition A.3. *A generalized protocol* Π *is a* distribution *over triples*

$$(X, Y, \boldsymbol{M}),$$

where $\boldsymbol{M} = (M_0, \ldots, M_r)$ *such that each* M_i *is chosen from a prefix-free set* $\mathcal{M}_i(M_{<i})$ *which depends only on* $M_{<i}$*. The last message* M_r *is the output of the protocol. The* $\theta^{\log}$*-cost of* Π *with respect to an input distribution* μ *is*

$$\theta^{\log}_{\mu}(\Pi) := 2\boldsymbol{D}_{\mathrm{KL}}(\Pi_{X,Y} \,\|\, \mu) + \frac{1}{2} \cdot (I_{\Pi}(X; M_0 \mid Y) + I_{\Pi}(Y; M_0 \mid X))$$

$$+ \sum_{\text{odd } i \in [1,r]} I_\Pi(Y; M_i \mid X, M_{<i}) + \sum_{\text{even } i \in [1,r]} I_\Pi(X; M_i \mid Y, M_{<i}).$$

The communication cost is

$$\max_{\boldsymbol{M}:\Pi(\boldsymbol{M})>0} \sum_{i=1}^{r} |M_i| .$$

The following lemma relates the $\theta^{\log}$-cost to the success probability of a standard protocol.

Lemma A.4. *If there is a generalized protocol Π with $\theta^{\log}$-cost at most θ with respect to μ computing a function f, then there is a standard protocol Π' computing f with probability at least*

$$2^{-6(\theta+1)}.$$

Moreover, Π and Π' have the same communication cost.

Proof. Consider the following standard protocol Π':

Protocol Π' for inputs $(X, Y) \sim \mu$

1. view the public random bits as a sequence of $|\mathcal{M}_0|$ independent samples $(M_0^{(i)}, t^{(i)})_{i \in [|\mathcal{M}_0|]}$ for uniform $M_0^{(i)} \in \mathcal{M}_0$ and $t^{(i)} \in [0,1]$
2. if there is a unique $M_0^{(i)}$ such that $t^{(i)} \le \Pi(M_0^{(i)} \mid X)$, Alice sets M_0^A to $M_0^{(i)}$
3. if there is a unique $M_0^{(i)}$ such that $t^{(i)} \le \Pi(M_0^{(i)} \mid Y)$, Bob sets M_0^B to $M_0^{(i)}$
4. otherwise they set M_0^A or M_0^B arbitrarily
5. for $i = 1, \ldots, r$
6. if i is odd, Alice samples $M_i \sim \Pi_{M_i|X,M_{<i}}$ for $M_0 = M_0^A$, and sends M_i
7. if i is even, Bob samples $M_i \sim \Pi_{M_i|Y,M_{<i}}$ for $M_0 = M_0^B$, and sends M_i

Clearly, the communication cost of Π' is the same as that of Π.
For each fixed i, the probability that $t^{(i)} \le \Pi(M_0^{(i)} \mid X)$ is equal to

$$\sum_{m_0 \in \mathcal{M}_0} \frac{1}{|\mathcal{M}_0|} \cdot \Pi(M_0^{(i)} = m_0 \mid X) = \frac{1}{|\mathcal{M}_0|}.$$

By union bound, the probability that either $t^{(i)} \le \Pi(M_0^{(i)} \mid X)$ or $t^{(i)} \le \Pi(M_0^{(i)} \mid Y)$ is at most $2/|\mathcal{M}_0|$. Thus, the probability that both Alice and Bob set M_0^A and M_0^B to M_0 is at least

$$\min\{\Pi(M_0 \mid X), \Pi(M_0 \mid Y)\} \cdot (1 - 2/|\mathcal{M}_0|)^{|\mathcal{M}_0|-1} \ge \frac{1}{8} \cdot \min\{\Pi(M_0 \mid X), \Pi(M_0 \mid Y)\},$$

where we assumed without loss of generality that $|\mathcal{M}_0| \geq 4$. Thus, when (X, Y) is sampled from μ, we have

$$\Pi'(X, Y, \boldsymbol{M}) \geq \frac{1}{8} \cdot \mu(X, Y) \cdot \min\left\{\Pi(M_0 \mid X), \Pi(M_0 \mid Y)\right\} \cdot \prod_{\text{odd } i \in [1,r]} \Pi(M_i \mid X, M_{<i}) \cdot \prod_{\text{even } i \in [1,r]} \Pi(M_i \mid Y, M_{<i}).$$

Let Π'_A be the distribution such that

$$\Pi'_A(X, Y, \boldsymbol{M}) = \mu(X, Y) \cdot \Pi(M_0 \mid X) \cdot \prod_{\text{odd } i \in [1,r]} \Pi(M_i \mid X, M_{<i}) \cdot \prod_{\text{even } i \in [1,r]} \Pi(M_i \mid Y, M_{<i}),$$

and Π'_B be the distribution such that

$$\Pi'_B(X, Y, \boldsymbol{M}) = \mu(X, Y) \cdot \Pi(M_0 \mid Y) \cdot \prod_{\text{odd } i \in [1,r]} \Pi(M_i \mid X, M_{<i}) \cdot \prod_{\text{even } i \in [1,r]} \Pi(M_i \mid Y, M_{<i}).$$

Thus, we have

$$\Pi'(X, Y, \boldsymbol{M}) \geq \frac{1}{8} \cdot \min\left\{\Pi'_A(X, Y, \boldsymbol{M}), \Pi'_B(X, Y, \boldsymbol{M})\right\}.$$

Now, observe that

$$\begin{aligned}
&\boldsymbol{D}_{\mathrm{KL}}(\Pi \,\|\, \Pi'_A) \\
&= \mathop{\mathbb{E}}_{(X,Y,\boldsymbol{M}) \sim \Pi} \left[\log\left(\frac{\Pi(X, Y, \boldsymbol{M})}{\Pi'_A(X, Y, \boldsymbol{M})}\right)\right] \\
&= \mathop{\mathbb{E}}_{(X,Y,\boldsymbol{M}) \sim \Pi} \left[\log\left(\frac{\Pi(X, Y) \cdot \Pi(M_0 \mid X, Y) \cdot \prod_{\text{odd } i \in [1,r]} \Pi(M_i \mid X, Y, M_{<i}) \cdot \prod_{\text{even } i \in [1,r]} \Pi(M_i \mid X, Y, M_{<i})}{\mu(X, Y) \cdot \Pi(M_0 \mid X) \cdot \prod_{\text{odd } i \in [1,r]} \Pi(M_i \mid X, M_{<i}) \cdot \prod_{\text{even } i \in [1,r]} \Pi(M_i \mid Y, M_{<i})}\right)\right] \\
&= \boldsymbol{D}_{\mathrm{KL}}(\Pi_{X,Y} \,\|\, \mu) + I_\Pi(Y; M_0 \mid X) + \sum_{\text{odd } i \in [1,r]} I_\Pi(M_i; Y \mid X, M_{<i}) + \sum_{\text{even } i \in [1,r]} I_\Pi(M_i; X \mid Y, M_{<i}).
\end{aligned}$$

Similarly,

$$\boldsymbol{D}_{\mathrm{KL}}(\Pi \,\|\, \Pi'_B) = \boldsymbol{D}_{\mathrm{KL}}(\Pi_{X,Y} \,\|\, \mu) + I_\Pi(X; M_0 \mid Y) + \sum_{\text{odd } i \in [1,r]} I_\Pi(M_i; Y \mid X, M_{<i}) + \sum_{\text{even } i \in [1,r]} I_\Pi(M_i; X \mid Y, M_{<i}).$$

Therefore,

$$\boldsymbol{D}_{\mathrm{KL}}(\Pi \,\|\, \Pi'_A) + \boldsymbol{D}_{\mathrm{KL}}(\Pi \,\|\, \Pi'_B) = 2\theta^{\log}_\mu(\Pi) - 2\boldsymbol{D}_{\mathrm{KL}}(\Pi_{X,Y} \,\|\, \mu) \leq 2\theta.$$

Claim A.5. *Let P, Q_1, Q_2 be three distributions. We must have*

$$\sum_x \min\{P(x), Q_1(x), Q_2(x)\} \geq 2^{-3(\max\{\boldsymbol{D}_{\mathrm{KL}}(P \,\|\, Q_1), \boldsymbol{D}_{\mathrm{KL}}(P \,\|\, Q_2)\}+1)}.$$

We will prove the claim later. The claim implies that

$$\sum_{X,Y,\boldsymbol{M}} \min\{\Pi(X,Y,\boldsymbol{M}), \Pi'_A(X,Y,\boldsymbol{M}), \Pi'_B(X,Y,\boldsymbol{M})\} \geq 2^{-6\theta-3},$$

which in turn, implies that

$$\sum_{X,Y,\boldsymbol{M}} \min\{\Pi(X,Y,\boldsymbol{M}), \Pi'(X,Y,\boldsymbol{M})\} \geq \frac{1}{8} \cdot 2^{-6\theta-3} \geq 2^{-6(\theta+1)}.$$

Since Π computes f, Π' must compute f with probability at least $2^{-6(\theta+1)}$. This proves the lemma. □

Proof of Claim A.5. Let

$$E_0 := \{x : P(x) \leq Q_1(x) \wedge P(x) \leq Q_2(x)\},$$

$$E_1 := \{x : Q_1(x) \leq P(x) \wedge Q_1(x) \leq Q_2(x)\},$$

and

$$E_2 := \{x : Q_2(x) \leq P(x) \wedge Q_2(x) \leq Q_1(x)\}.$$

Then at least one of E_0, E_1, E_2 has probability at least $1/3$ under distribution P.

If $P(E_0) \geq 1/3$, then

$$\sum_x \min\{P(x), Q_1(x), Q_2(x)\} \geq 1/3.$$

The lemma holds.

Suppose $P(E_1) \geq 1/3$. We have

$$\sum_x \min\{P(x), Q_1(x), Q_2(x)\} \geq Q_1(E_1).$$

On the other hand,

$$\begin{aligned}\boldsymbol{D}_{\mathrm{KL}}(P \,\|\, Q_1) &= \mathop{\mathbb{E}}_{x\sim P}[\log(P(x)/Q_1(x))] \\ &= P(E_1) \cdot \mathop{\mathbb{E}}_{x\sim P|E_1}[\log(P(x)/Q_1(x))] + P(\overline{E_1}) \mathop{\mathbb{E}}_{x\sim P|\overline{E_1}}[\log(P(x)/Q_1(x))]\end{aligned}$$

which by the convexity of $f(t) = \log(1/t)$, is

$$\begin{aligned}
&\geq P(E_1) \log \left(\mathop{\mathbb{E}}_{x \sim P|E_1} [Q_1(x)/P(x)]^{-1} \right) + P(\overline{E_1}) \log \left(\mathop{\mathbb{E}}_{x \sim P|\overline{E_1}} [Q_1(x)/P(x)]^{-1} \right) \\
&= P(E_1) \log \left(P(E_1)/Q_1(E_1) \right) + (1 - P(E_1)) \log \left((1 - P(E_1))/(1 - Q_1(E_1)) \right) \\
&\geq P(E_1) \log(1/Q_1(E_1)) - 1 \\
&\geq \frac{1}{3} \log(1/Q_1(E_1)) - 1.
\end{aligned}$$

That is, $Q_1(E_1) \geq 2^{-3(\boldsymbol{D}_{\mathrm{KL}}(P \,\|\, Q_1)+1)}$. The lemma holds. The case where $P(E_2) \geq 1/3$ is similar. □

The following lemma is implicitly proved in [BRWY13].

Lemma A.6. *Let Π be a standard protocol with input distribution μ and W be an event, let Π^W be the distribution of Π conditioned on W, then*

$$\theta_\mu^{\log}(\Pi^W) \leq 5 \log(1/\Pi(W)).$$

Proof. Consider $\theta_\mu^{\log}(\Pi^W)$. For the first term, since $\Pi(X, Y) = \mu(X, Y)$, we have

$$\begin{aligned}
\boldsymbol{D}_{\mathrm{KL}}(\Pi^W_{X,Y} \,\|\, \mu) &= \mathop{\mathbb{E}}_{(X,Y) \sim \Pi|W} \left[\log \left(\frac{\Pi(X, Y \mid W)}{\mu(X, Y)} \right) \right] \\
&\leq \mathop{\mathbb{E}}_{(X,Y) \sim \Pi|W} [\log (1/\Pi(W))] \\
&= \log(1/\Pi(W)).
\end{aligned}$$

For the second term, since (X, Y) and M_0 are independent in Π, we have

$$\begin{aligned}
I_{\Pi^W}(X; M_0 \mid Y) &= \sum_{x,y,m_0} \Pi(X = x, M_0 = m_0 \mid Y = y, W) \cdot \log \left(\frac{\Pi(X = x \mid M_0 = m_0, Y = y, W)}{\Pi(X = x \mid Y = y, W)} \right) \\
&= \sum_{x,y,m_0} \Pi(X = x, M_0 = m_0 \mid Y = y, W) \cdot \log \left(\frac{\Pi(X = x \mid M_0 = m_0, Y = y, W)}{\Pi(X = x \mid Y = y)} \right) \\
&\quad - \sum_{x,y,m_0} \Pi(X = x, M_0 = m_0 \mid Y = y, W) \cdot \log \left(\frac{\Pi(X = x \mid Y = y, W)}{\Pi(X = x \mid Y = y)} \right) \\
&= \sum_{x,y,m_0} \Pi(X = x, M_0 = m_0 \mid Y = y, W) \cdot \log \left(\frac{\Pi(X = x \mid M_0 = m_0, Y = y, W)}{\Pi(X = x \mid M_0 = m_0, Y = y)} \right) \\
&\quad - \boldsymbol{D}_{\mathrm{KL}}(\Pi_{X|W,Y} \,\|\, \Pi_{X|Y}) \\
&\leq \sum_{y,m_0} \Pi(Y = y, M_0 = m_0 \mid W) \cdot \log(1/\Pi(W \mid Y = y, M_0 = m_0))
\end{aligned}$$

which by the concavity of log, is

$$\begin{aligned}
&\leq \log\left(\sum_{y,m_0} \frac{\Pi(Y=y, M_0=m_0 \mid W)}{\Pi(W \mid Y=y, M_0=m_0)}\right) \\
&= \log\left(\sum_{y,m_0} \frac{\Pi(Y=y, M_0=m_0)}{\Pi(W)}\right) \\
&= \log(1/\Pi(W)).
\end{aligned}$$

Similarly, $I_{\Pi^W}(Y; M_0 \mid X) \leq \log(1/\Pi(W))$.

For the third term, fix an odd i, we have

$$\begin{aligned}
&I_{\Pi^W}(Y; M_i \mid X, M_{<i}) \\
&= \sum_{x,y,m_{\leq i}} \Pi(X=x, Y=y, M_{\leq i}=m_{\leq i} \mid W) \cdot \log\left(\frac{\Pi(M_i=m_i \mid X=x, Y=y, M_{<i}=m_{<i}, W)}{\Pi(M_i=m_i \mid X=x, M_{<i}=m_{<i}, W)}\right) \\
&= \sum_{x,y,m_{\leq i}} \Pi(X=x, Y=y, M_{\leq i}=m_{\leq i} \mid W) \cdot \log\left(\frac{\Pi(M_i=m_i \mid X=x, Y=y, M_{<i}=m_{<i}, W)}{\Pi(M_i=m_i \mid X=x, M_{<i}=m_{<i})}\right) \\
&\quad + \sum_{x,y,m_{\leq i}} \Pi(X=x, Y=y, M_{\leq i}=m_{\leq i} \mid W) \cdot \log\left(\frac{\Pi(M_i=m_i \mid X=x, M_{<i}=m_{<i})}{\Pi(M_i=m_i \mid X=x, M_{<i}=m_{<i}, W)}\right).
\end{aligned}$$

Note that the second term is at most 0, since its negation is an expected KL-divergence. For the first term, we have $\Pi(M_i=m_i \mid X=x, M_{<i}=m_{<i}) = \Pi(M_i=m_i \mid X=x, Y=y, M_{<i}=m_{<i})$, since Π is a standard protocol and i is odd. Thus, we have

$$\begin{aligned}
&I_{\Pi^W}(Y; M_i \mid X, M_{<i}) \\
&= \sum_{x,y,m_{\leq i}} \Pi(X=x, Y=y, M_{\leq i}=m_{\leq i} \mid W) \cdot \log\left(\frac{\Pi(M_i=m_i \mid X=x, Y=y, M_{<i}=m_{<i}, W)}{\Pi(M_i=m_i \mid X=x, Y=y, M_{<i}=m_{<i})}\right) \\
&= \mathop{\mathbb{E}}_{x,y,m_{<i}\sim\Pi_{X,Y,M_{<i}}} \boldsymbol{D}_{\mathrm{KL}}(\Pi_{M_i|X=x,Y=y,M_{<i}=m_{<i},W} \,\|\, \Pi_{M_i|X=x,Y=y,M_{<i}=m_{<i}}).
\end{aligned}$$

Thus, the third term in $\theta^{\log}_{\mu}(\Pi^W)$ is

$$\begin{aligned}
&\sum_{\text{odd } i\in[1,r]} I_{\Pi^W}(Y; M_i \mid X, M_{<i}) \\
&\quad \leq \sum_{\text{odd } i\in[1,r]} \mathop{\mathbb{E}}_{x,y,m_{<i}\sim\Pi_{X,Y,M_{<i}}} \boldsymbol{D}_{\mathrm{KL}}(\Pi_{M_i|X=x,Y=y,M_{<i}=m_{<i},W} \,\|\, \Pi_{M_i|X=x,Y=y,M_{<i}=m_{<i}}) \\
&\quad \leq \sum_{i\in[1,r]} \mathop{\mathbb{E}}_{x,y,m_{<i}\sim\Pi_{X,Y,M_{<i}}} \boldsymbol{D}_{\mathrm{KL}}(\Pi_{M_i|X=x,Y=y,M_{<i}=m_{<i},W} \,\|\, \Pi_{M_i|X=x,Y=y,M_{<i}=m_{<i}})
\end{aligned}$$

which by the chain rule of KL-divergence, is

$$\begin{aligned}
&= \mathop{\mathbb{E}}_{x,y,m_0\sim\Pi_{X,Y,M_0}} \boldsymbol{D}_{\mathrm{KL}}(\Pi_{\boldsymbol{M}|X=x,Y=y,M_0=m_0,W} \,\|\, \Pi_{\boldsymbol{M}|X=x,Y=y,M_0=m_0}) \\
&\leq \mathop{\mathbb{E}}_{x,y,m_0\sim\Pi_{X,Y,M_0}} \log(1/\Pi(W \mid X=x, Y=y, M_0=m_0)) \\
&\leq \log(1/\Pi(W)).
\end{aligned}$$

The same argument proves that the last term is also at most $\log(1/\Pi(W))$. Thus, the lemma holds. □

The following lemma decomposes a protocol for k instances into one protocol for one instance and one protocol for $k-1$ instances.

Lemma A.7. *Let μ be a distribution over input pairs (X,Y). Let Π be a generalized protocol on k input pairs $(X_1,\ldots,X_k,Y_1,\ldots,Y_k)$ that uses C bits of communication and computes f^k. There is a generalized protocol $\Pi^{(<k)}$ and a generalized protocol $\Pi^{(k)}$ such that*

- *both $\Pi^{(<k)}$ and $\Pi^{(k)}$ use at most C bits of communication;*
- *$\theta^{\log}_{\mu^{k-1}}(\Pi^{(<k)}) + \theta^{\log}_{\mu}(\Pi^{(k)}) \leq \theta^{\log}_{\mu^k}(\Pi)$;*
- *$\Pi^{(<k)}$ computes f^{k-1} and $\Pi^{(k)}$ computes f.*

By repeatedly applying the above lemma, we prove the following lemma as a corollary.

Lemma A.8. *If there is an r-message protocol with $\theta^{\log}$-cost θ with respect to μ^n and C bits of communication that computes f^n. Then there is an r-message protocol with $\theta^{\log}$-cost θ/n with respect to μ and C bits of communication that computes f.*

Proof of Lemma A.7. Let $(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{M})$ be random variables distributed according to Π. Let $S \subsetneq [k]$ be a nonempty proper subset of the instances (think of $S=[k-1]$), and denote its complement by $\overline{S}$. Consider the following protocol η^X_S for $f^{|S|}$ with respect to $\mu^{|S|}$, which defines a distribution over triples

$$(\boldsymbol{X}^\eta, \boldsymbol{Y}^\eta, \boldsymbol{M}^\eta).$$

Protocol η^X_S:
1. sample $(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{M}) \sim \Pi$
2. **set** $\boldsymbol{X}^\eta := \boldsymbol{X}_S$ and $\boldsymbol{Y}^\eta := \boldsymbol{Y}_S$
3. **set** $M^\eta_0 := \boldsymbol{X}_{\overline{S}} \circ M_0$
4. for $i=1,\ldots,r-1$, **set** $M^\eta_i := M_i$
5. **set** M^η_r to M_r restricted to coordinates in S

Compared to Π, η^X_S restricts the input pair (X,Y) to coordinates only in S, prepends $\boldsymbol{X}_{\overline{S}}$ to the public random bits, and restricts the output to coordinates only in S. Since $M_r = f^n(\boldsymbol{X},\boldsymbol{Y})$, $M^\eta_r = f^{|S|}(\boldsymbol{X}_S,\boldsymbol{Y}_S)$. Hence, η^X_S is an r-message protocol that computes $f^{|S|}$.

Similarly, we define $\eta^Y_{\overline{S}}$ as follows.

Protocol $\eta^Y_{\overline{S}}$:
1. sample $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{M}) \sim \Pi$
2. **set** $\boldsymbol{X}^\eta := \boldsymbol{X}_{\overline{S}}$ and $\boldsymbol{Y}^\eta := \boldsymbol{Y}_{\overline{S}}$
3. **set** $M_0^\eta := \boldsymbol{Y}_S \circ M_0$
4. for $i = 1, \ldots, r-1$, **set** $M_i^\eta := M_i$
5. **set** M_r^η to M_r restricted to coordinates in $\overline{S}$

We prepend $\boldsymbol{Y}_S$ to M_0, and restrict the output to coordinates in $\overline{S}$. Similarly, $\eta^Y_{\overline{S}}$ is an r-message protocol that computes $f^{|\overline{S}|}$.

To prove the lemma, we will set $\Pi^{(<k)}$ to η^X_S and set $\Pi^{(k)}$ to $\eta^Y_{\overline{S}}$ for $S = [k-1]$. Clearly, both protocols use at most C bits of communication. It remains to show that their $\theta^{\log}$-costs sum up to that of Π.

Analysis of the $\theta^{\log}$-cost. Next, we analyze their $\theta^{\log}$-costs. We first focus on η^X_S. The mutual information between the input and the public random bits is

$$I_\eta(\boldsymbol{X}^\eta; M_0^\eta \mid \boldsymbol{Y}^\eta) + I_\eta(\boldsymbol{Y}^\eta; M_0^\eta \mid \boldsymbol{X}^\eta) = I_\Pi(\boldsymbol{X}_S; \boldsymbol{X}_{\overline{S}}, M_0 \mid \boldsymbol{Y}_S) + I_\Pi(\boldsymbol{Y}_S; \boldsymbol{X}_{\overline{S}}, M_0 \mid \boldsymbol{X}_S).$$

The mutual information between $\boldsymbol{Y}^\eta$ and the odd messages is

$$\sum_{\text{odd } i \in [1,r]} I_\eta(\boldsymbol{Y}^\eta; M_i^\eta \mid \boldsymbol{X}^\eta, M_{<i}^\eta) \le \sum_{\text{odd } i \in [1,r]} I_\Pi(\boldsymbol{Y}_S; M_i \mid \boldsymbol{X}, M_{<i}).$$

The mutual information between $\boldsymbol{X}^\eta$ and the even messages is

$$\begin{aligned}
&\sum_{\text{even } i \in [1,r]} I_\eta(\boldsymbol{X}^\eta; M_i^\eta \mid \boldsymbol{Y}^\eta, M_{<i}^\eta) \\
&\quad\le \sum_{\text{even } i \in [1,r]} I_\Pi(\boldsymbol{X}_S; M_i \mid \boldsymbol{Y}_S, \boldsymbol{X}_{\overline{S}}, M_{<i}) \\
&\quad= \sum_{\text{even } i \in [1,r]} \left(I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}}, M_i \mid \boldsymbol{Y}_S, \boldsymbol{X}_{\overline{S}}, M_{<i}) - I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}} \mid \boldsymbol{Y}_S, \boldsymbol{X}_{\overline{S}}, M_{<i}, M_i)\right) \\
&\quad= \sum_{\text{even } i \in [1,r]} \big(I_\Pi(\boldsymbol{X}_S; M_i \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}, M_{<i}) \\
&\qquad\quad + I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}} \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, M_{<i}) - I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}} \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, M_{<i+1})\big),
\end{aligned}$$

Summing up all terms, we have

$$\begin{aligned}
&\theta^{\log}_{\mu^{|S|}}(\eta^X_S) && (39)\\
&\quad\le 2\boldsymbol{D}_{\mathrm{KL}}(\Pi_{\boldsymbol{X}_S, \boldsymbol{Y}_S} \,\|\, \mu^{|S|}) + \frac{1}{2} \cdot I_\Pi(\boldsymbol{X}_S; \boldsymbol{X}_{\overline{S}}, M_0 \mid \boldsymbol{Y}_S) + \frac{1}{2} \cdot I_\Pi(\boldsymbol{Y}_S; \boldsymbol{X}_{\overline{S}}, M_0 \mid \boldsymbol{X}_S) && (40)\\
&\qquad + \sum_{\text{odd } i \in [1,r]} I_\Pi(\boldsymbol{Y}_S; M_i \mid \boldsymbol{X}, M_{<i}) + \sum_{\text{even } i \in [1,r]} I_\Pi(\boldsymbol{X}_S; M_i \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}, M_{<i}) && (41)
\end{aligned}$$

$$+ \sum_{\text{even } i\in[1,r]} \left(I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}} \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, M_{<i}) - I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}} \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, M_{<i+1})\right). \tag{42}$$

Similarly, for $\eta^Y_{\overline{S}}$, the mutual information between the input and the public random bits is

$$I_\Pi(\boldsymbol{X}_{\overline{S}}; \boldsymbol{Y}_S, M_0 \mid \boldsymbol{Y}_{\overline{S}}) + I_\Pi(\boldsymbol{Y}_{\overline{S}}; \boldsymbol{Y}_S, M_0 \mid \boldsymbol{X}_{\overline{S}}).$$

The mutual information between $\boldsymbol{Y}^\eta$ and the odd messages is at most

$$\begin{aligned}
&\sum_{\text{odd } i\in[1,r]} I_\Pi(\boldsymbol{Y}_{\overline{S}}; M_i \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, M_{<i}) \\
&= \sum_{\text{odd } i\in[1,r]} \left(I_\Pi(\boldsymbol{Y}_{\overline{S}}; \boldsymbol{X}_S, M_i \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, M_{<i}) - I_\Pi(\boldsymbol{Y}_{\overline{S}}; \boldsymbol{X}_S \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, M_{<i+1})\right) \\
&= \sum_{\text{odd } i\in[1,r]} \Big(I_\Pi(\boldsymbol{Y}_{\overline{S}}; M_i \mid \boldsymbol{X}, \boldsymbol{Y}_S, M_{<i}) \\
&\qquad + I_\Pi(\boldsymbol{Y}_{\overline{S}}; \boldsymbol{X}_S \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, M_{<i}) - I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}} \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, M_{<i+1})\Big).
\end{aligned}$$

The mutual information between $\boldsymbol{X}^\eta$ and the even messages is at most

$$\sum_{\text{even } i\in[1,r]} I_\Pi(\boldsymbol{X}_{\overline{S}}; M_i \mid \boldsymbol{Y}, M_{<i}).$$

Summing up all terms, we have

$$\theta^{\log}_{\mu^{k-|S|}}(\eta^Y_{\overline{S}}) \tag{43}$$

$$\leq 2\boldsymbol{D}_{\mathrm{KL}}(\Pi_{\boldsymbol{X}_{\overline{S}},\boldsymbol{Y}_{\overline{S}}} \,\|\, \mu^{k-|S|}) + \frac{1}{2}\cdot I_\Pi(\boldsymbol{X}_{\overline{S}}; \boldsymbol{Y}_S, M_0 \mid \boldsymbol{Y}_{\overline{S}}) + \frac{1}{2}\cdot I_\Pi(\boldsymbol{Y}_{\overline{S}}; \boldsymbol{Y}_S, M_0 \mid \boldsymbol{X}_{\overline{S}}) \tag{44}$$

$$+ \sum_{\text{odd } i\in[1,r]} I_\Pi(\boldsymbol{Y}_{\overline{S}}; M_i \mid \boldsymbol{X}, \boldsymbol{Y}_S, M_{<i}) + \sum_{\text{even } i\in[1,r]} I_\Pi(\boldsymbol{X}_{\overline{S}}; M_i \mid \boldsymbol{Y}, M_{<i}) \tag{45}$$

$$+ \sum_{\text{odd } i\in[1,r]} \left(I_\Pi(\boldsymbol{Y}_{\overline{S}}; \boldsymbol{X}_S \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, M_{<i}) - I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}} \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, M_{<i+1})\right). \tag{46}$$

Next, we sum up Equation (39) and (43). The first lines (40) and (44) sum up to

$$\begin{aligned}
&2\boldsymbol{D}_{\mathrm{KL}}(\Pi_{\boldsymbol{X}_S,\boldsymbol{Y}_S} \,\|\, \mu^{|S|}) + 2\boldsymbol{D}_{\mathrm{KL}}(\Pi_{\boldsymbol{X}_{\overline{S}},\boldsymbol{Y}_{\overline{S}}} \,\|\, \mu^{k-|S|}) \\
&\quad + \frac{1}{2}\cdot I_\Pi(\boldsymbol{X}_S; \boldsymbol{X}_{\overline{S}}, M_0 \mid \boldsymbol{Y}_S) + \frac{1}{2}\cdot I_\Pi(\boldsymbol{Y}_S; \boldsymbol{X}_{\overline{S}}, M_0 \mid \boldsymbol{X}_S) \\
&\quad + \frac{1}{2}\cdot I_\Pi(\boldsymbol{X}_{\overline{S}}; \boldsymbol{Y}_S, M_0 \mid \boldsymbol{Y}_{\overline{S}}) + \frac{1}{2}\cdot I_\Pi(\boldsymbol{Y}_{\overline{S}}; \boldsymbol{Y}_S, M_0 \mid \boldsymbol{X}_{\overline{S}}) \\
&= 2\boldsymbol{D}_{\mathrm{KL}}(\Pi_{\boldsymbol{X},\boldsymbol{Y}} \,\|\, \mu^k) - 2I_\Pi(\boldsymbol{X}_S, \boldsymbol{Y}_S; \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_{\overline{S}}) \\
&\quad + \frac{1}{2}\cdot \left(I_\Pi(\boldsymbol{X}_S; \boldsymbol{X}_{\overline{S}} \mid \boldsymbol{Y}_S) + I_\Pi(\boldsymbol{X}_S; M_0 \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S) + I_\Pi(\boldsymbol{Y}_S; \boldsymbol{X}_{\overline{S}} \mid \boldsymbol{X}_S) + I_\Pi(\boldsymbol{Y}_S; M_0 \mid \boldsymbol{X})\right)
\end{aligned}$$

$$
\begin{aligned}
&\quad + \frac{1}{2} \cdot (I_\Pi(\boldsymbol{X}_{\overline{S}}; \boldsymbol{Y}_S \mid \boldsymbol{Y}_{\overline{S}}) + I_\Pi(\boldsymbol{X}_{\overline{S}}; M_0 \mid \boldsymbol{Y}) + I_\Pi(\boldsymbol{Y}_{\overline{S}}; \boldsymbol{Y}_S \mid \boldsymbol{X}_{\overline{S}}) + I_\Pi(\boldsymbol{Y}_{\overline{S}}; M_0 \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S)) \\
&\le 2\boldsymbol{D}_{\mathrm{KL}}(\Pi_{\boldsymbol{X},\boldsymbol{Y}} \,\|\, \mu^k) - 2I_\Pi(\boldsymbol{X}_S, \boldsymbol{Y}_S; \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_{\overline{S}}) \\
&\quad + \frac{1}{2} \cdot (I_\Pi(\boldsymbol{X}_S; \boldsymbol{X}_{\overline{S}} \mid \boldsymbol{Y}_S) + I_\Pi(\boldsymbol{X}_S; M_0 \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S) + I_\Pi(\boldsymbol{X}_S, \boldsymbol{Y}_S; \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_{\overline{S}}) + I_\Pi(\boldsymbol{Y}_S; M_0 \mid \boldsymbol{X})) \\
&\quad + \frac{1}{2} \cdot (I_\Pi(\boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_{\overline{S}}; \boldsymbol{X}_S, \boldsymbol{Y}_S) + I_\Pi(\boldsymbol{X}_{\overline{S}}; M_0 \mid \boldsymbol{Y}) + I_\Pi(\boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_{\overline{S}}; \boldsymbol{Y}_S) + I_\Pi(\boldsymbol{Y}_{\overline{S}}; M_0 \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S)) \\
&= 2\boldsymbol{D}_{\mathrm{KL}}(\Pi_{\boldsymbol{X},\boldsymbol{Y}} \,\|\, \mu^k) - I_\Pi(\boldsymbol{X}_S, \boldsymbol{Y}_S; \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_{\overline{S}}) + \frac{1}{2} \cdot I_\Pi(\boldsymbol{X}_S; \boldsymbol{X}_{\overline{S}} \mid \boldsymbol{Y}_S) + \frac{1}{2} \cdot I_\Pi(\boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_{\overline{S}}; \boldsymbol{Y}_S) \\
&\quad + \frac{1}{2} \cdot (I_\Pi(\boldsymbol{X}_S; M_0 \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S) + I_\Pi(\boldsymbol{Y}_S; M_0 \mid \boldsymbol{X}) + I_\Pi(\boldsymbol{X}_{\overline{S}}; M_0 \mid \boldsymbol{Y}) + I_\Pi(\boldsymbol{Y}_{\overline{S}}; M_0 \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S)) \\
&\le 2\boldsymbol{D}_{\mathrm{KL}}(\Pi_{\boldsymbol{X},\boldsymbol{Y}} \,\|\, \mu^k) - I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}} \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S) \\
&\quad + \frac{1}{2} \cdot (I_\Pi(\boldsymbol{X}_S; M_0 \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S) + I_\Pi(\boldsymbol{Y}_S; M_0 \mid \boldsymbol{X}) + I_\Pi(\boldsymbol{X}_{\overline{S}}; M_0 \mid \boldsymbol{Y}) + I_\Pi(\boldsymbol{Y}_{\overline{S}}; M_0 \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S)) \\
&= 2\boldsymbol{D}_{\mathrm{KL}}(\Pi_{\boldsymbol{X},\boldsymbol{Y}} \,\|\, \mu^k) \\
&\quad + \frac{1}{2} \cdot (-I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}} \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S) + I_\Pi(\boldsymbol{X}_S; M_0 \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S) + I_\Pi(\boldsymbol{X}_{\overline{S}}; M_0 \mid \boldsymbol{Y})) \\
&\quad + \frac{1}{2} \cdot (-I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}} \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S) + I_\Pi(\boldsymbol{Y}_S; M_0 \mid \boldsymbol{X}) + I_\Pi(\boldsymbol{Y}_{\overline{S}}; M_0 \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S)) \\
&= 2\boldsymbol{D}_{\mathrm{KL}}(\Pi_{\boldsymbol{X},\boldsymbol{Y}} \,\|\, \mu^k) \\
&\quad + \frac{1}{2} \cdot (-I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}} \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, M_0) + I_\Pi(\boldsymbol{X}; M_0 \mid \boldsymbol{Y})) \\
&\quad + \frac{1}{2} \cdot (-I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}} \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, M_0) + I_\Pi(\boldsymbol{Y}; M_0 \mid \boldsymbol{X})) \\
&= 2\boldsymbol{D}_{\mathrm{KL}}(\Pi_{\boldsymbol{X},\boldsymbol{Y}} \,\|\, \mu^k) + \frac{1}{2} \cdot (I_\Pi(\boldsymbol{X}; M_0 \mid \boldsymbol{Y}) + I_\Pi(\boldsymbol{Y}; M_0 \mid \boldsymbol{X})) - I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}} \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, M_0).
\end{aligned}
$$

The second lines (41) and (45) sum up to

$$
\begin{aligned}
&\sum_{\text{odd } i \in [1,r]} I_\Pi(\boldsymbol{Y}_S; M_i \mid \boldsymbol{X}, M_{<i}) + \sum_{\text{odd } i \in [1,r]} I_\Pi(\boldsymbol{Y}_{\overline{S}}; M_i \mid \boldsymbol{X}, \boldsymbol{Y}_S, M_{<i}) \\
&\quad + \sum_{\text{even } i \in [1,r]} I_\Pi(\boldsymbol{X}_S; M_i \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}, M_{<i}) + \sum_{\text{even } i \in [1,r]} I_\Pi(\boldsymbol{X}_{\overline{S}}; M_i \mid \boldsymbol{Y}, M_{<i}) \\
&= \sum_{\text{odd } i \in [1,r]} I_\Pi(\boldsymbol{Y}; M_i \mid \boldsymbol{X}, M_{<i}) + \sum_{\text{even } i \in [1,r]} I_\Pi(\boldsymbol{X}; M_i \mid \boldsymbol{Y}, M_{<i}).
\end{aligned}
$$

Finally, the third lines (42) and (46) sum up to

$$
\begin{aligned}
&\sum_{i \in [1,r]} (I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}} \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, M_{<i}) - I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}} \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, M_{<i+1})) \\
&= I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}} \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, M_0) - I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}} \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, \boldsymbol{M}) \\
&\le I_\Pi(\boldsymbol{X}_S; \boldsymbol{Y}_{\overline{S}} \mid \boldsymbol{X}_{\overline{S}}, \boldsymbol{Y}_S, M_0).
\end{aligned}
$$

Summing up all lines gives us

$$\begin{aligned}
&\theta^{\log}_{\mu^{|S|}}(\eta^X_S) + \theta^{\log}_{\mu^{k-|S|}}(\eta^Y_{\overline{S}}) \\
&\le 2\boldsymbol{D}_{\mathrm{KL}}(\Pi_{\boldsymbol{X},\boldsymbol{Y}} \,\|\, \mu^k) + \frac{1}{2}\cdot(I_\Pi(\boldsymbol{X}; M_0 \mid \boldsymbol{Y}) + I_\Pi(\boldsymbol{Y}; M_0 \mid \boldsymbol{X})) \\
&\quad + \sum_{\text{odd } i\in[1,r]} I_\Pi(\boldsymbol{Y}; M_i \mid \boldsymbol{X}, M_{<i}) + \sum_{\text{even } i\in[1,r]} I_\Pi(\boldsymbol{X}; M_i \mid \boldsymbol{Y}, M_{<i}) \\
&= \theta^{\log}_{\mu^k}(\Pi).
\end{aligned}$$

This completes the proof of the lemma. □

Finally, by combining Lemma A.6, Lemma A.8 and Lemma A.4, we prove the following direct product result.

Lemma A.9. *If there is a r-message protocol Π that computes f^n with probability q under input distribution μ^n using C bits of communication, then there is a r-message protocol Π' that computes f with probability $2^{-6}\cdot q^{30/n}$ under μ using C bits of communication.*

Proof. Consider the distribution induced by running Π on input distribution μ^n. Let W be the event that Π succeeds. Then $\Pi(W) \ge q$. Lemma A.6 implies that $\theta^{\log}(\Pi^W) \le 5\log(1/q)$. Next, Lemma A.8 implies that there is a protocol that computes f with $\theta^{\log}$-cost with respect to μ at most $5\cdot n^{-1}\log(1/q)$. Finally, by Lemma A.4, it implies a protocol Π' that computes f under μ with probability at least $2^{-6(5\cdot n^{-1}\log(1/q)+1)} \ge 2^{-6}\cdot q^{30/n}$. □

A.2 Lower bound for $\mathsf{PC}_{n,t}$

The following lemma is a direct corollary of Lemma A.9.

Lemma A.10. *Suppose for $k, t \in \mathbb{N}$ such that t is even and $k(2t+2) \le d$, there is a p-round protocol with at most S bits of communication such that given $\vec{\pi}_1, \vec{\pi}_2, \ldots, \vec{\pi}_k \leftarrow \mathcal{P}_{n,t}$ and $s_1, \ldots, s_k \in_{unif} [n]$ independently, the protocol outputs $(\mathsf{path}_{\vec{\pi}_1}(s_1), \ldots, \mathsf{path}_{\vec{\pi}_k}(s_k))$ with probability greater than $n^{-\delta d/p}$. Then there is a p-round protocol with at most S bits of communication computing $\mathsf{PC}_{n,t}$ with probability at least $2^{-6}\cdot n^{-30\delta d/(pk)}$ for $\vec{\pi} \leftarrow \mathcal{P}_{n,t}$ and $s \in_{unif} [n]$.*

Finally, we use the following lower bound for $\mathsf{PC}_{n,t}$, whose proof is similar to that of Lemma 4.11 in [AV21] and the standard pointer chasing lower bound [NW91].

Lemma A.11. *Any $(t-2)$-message protocol Π with at most $n^{1/4}$ bits of communication cannot solve $\mathsf{PC}_{n,t}$ with probability greater than $2t\cdot n^{-1/8}$.*

Thus, Lemma 6.2 is a direct corollary of Lemma A.2, Lemma A.10 and Lemma A.11 for $\varepsilon = 1/4$ and $\delta = 0.001$ by setting $t = p+2$ and $k = \lfloor d/(2t+2)\rfloor$, as we have

$$2^{-6}\cdot n^{-30\delta d/(pk)} \ge 2^{-6}\cdot n^{-0.03d/p(\lfloor d/(2p+6)\rfloor)} > 2t\cdot n^{-1/8},$$

since $t \le \log n$.

Proof. Let X be Alice's matchings $(\pi_1, \dots, \pi_{t-1})$, and let Y be Bob's matchings $(\pi_2, \dots, \pi_t)$. We will inductively prove the following: For $i \in [0, t-2]$, the distribution of

$$\pi_{\le i+2}(s) \mid \pi_1, \dots, \pi_{i+1}, s, M_{\le i}$$

is $i \cdot n^{-1/8}$-close to uniform in total variation distance in expectation. In particular for $i = t-2$, the ℓ_∞-norm is at most $(t-2) \cdot n^{-1/8} + 1/n$ in expectation. That is, in expectation, one cannot predict $\pi_{\le t}(s)$ with probability better than $(t-2) \cdot n^{-1/8} + 1/n$ given $\pi_1, \dots, \pi_{t-1}, s, M_{\le t-2}$. Since the output of the protocol is determined by $M_{\le t-2}$, it implies that the overall success probability is at most $(t-2) \cdot n^{-1/8} + 1/n \le 2t \cdot n^{-1/8}$.

Base case: $i = 0$. We first prove the base case when $i = 0$. The distribution of

$$\pi_{\le 2}(s) \mid \pi_1, s, M_0$$

is the uniform distribution over $[n]$, since π_2 is still uniform conditioned on (π_1, s, M_0) (which determines $\pi_1(s)$). Hence, the total variation distance is 0 in expectation.

Induction: $i-1$ to i. By symmetry, assume that i is odd.

Consider the matching π_{i+2}. Since all permutations are independent in the input distribution, we have

$$H(\pi_{i+2} \mid \pi_1, \dots, \pi_i, s) = \log(n!),$$

which implies that

$$H(\pi_{i+2} \mid \pi_1, \dots, \pi_i, s, M_{\le i}) \ge \log(n!) - |M_{\le i}| \ge \log(n!) - n^{1/4}.$$

The following lemma from [AKSY20] relates the entropy of a permutation to the entropy of its random coordinate.

Lemma A.12 (Lemma A.13 in [AKSY20])**.** *Let π be a random permutation over $[n]$. If $H(\pi) \ge \log n! - n/8$, then*

$$n \log n - \sum_{x \in [n]} H(\pi(x)) \le 4\sqrt{(\log n! - H(\pi))n} + 3.$$

It implies that for a uniform $x \in [n]$ (independent of π_{i+2} conditioned on $(\pi_1, \dots, \pi_i, s, M_{\le i})$), we have

$$\mathop{\mathbb{E}}_{x \in [n]} [H(\pi_{i+2}(x) \mid \pi_1, \dots, \pi_i, s, M_{\le i}, x)] \ge \log n - \frac{4\sqrt{n^{1/4} \cdot n} + 3}{n} \ge \log n - n^{-1/4}.$$

In particular, by Pinsker's inequality and the concavity of square-root, we obtain that for a uniform $x \in [n]$, the distribution of

$$\pi_{i+2}(x) \mid \pi_1, \ldots, \pi_i, s, M_{\leq i}, x$$

is $n^{-1/8}$-close to the uniform distribution over $[n]$ in expectation.

Now suppose the claim holds for $i-1$, i.e.,

$$\pi_{\leq i+1}(s) \mid \pi_1, \ldots, \pi_i, s, M_{\leq i-1}$$

is $(i-1)n^{-1/8}$-close to uniform. We observe that conditioned on $(\pi_1, \ldots, \pi_i, s, M_{\leq i-1})$, $\pi_{i+1}(s)$ is determined by π_{i+1}, which is part of Bob's input. By the Markov property of communication protocols, π_{i+1} is independent of Alice's inputs conditioned on $(\pi_1, \ldots, \pi_i, s, M_{\leq i-1})$. Thus, π_{i+1} is also independent of M_i conditioned on $(\pi_1, \ldots, \pi_i, s, M_{\leq i-1})$. Hence, the distribution of

$$\pi_{\leq i+1}(s) \mid \pi_1, \ldots, \pi_i, s, M_{\leq i}$$

is $(i-1)n^{-1/8}$-close to uniform.

By the Markov property again, $\pi_{\leq i+1}(s)$ and π_{i+2} are independent conditioned on $(\pi_1, \ldots, \pi_i, s, M_{\leq i})$. Therefore, by the triangle inequality, the distribution of

$$\pi_{i+2}(\pi_{\leq i+1}(s)) \mid \pi_1, \ldots, \pi_i, s, M_{\leq i}, \pi_{\leq i+1}(s)$$

is $i \cdot n^{-1/8}$-close to uniform in expectation.

Finally, since π_{i+1} and π_{i+2} are independent conditioned on $(\pi_1, \ldots, \pi_i, s, M_{\leq i}, \pi_{\leq i+1}(s))$. The distribution of

$$\pi_{i+2}(\pi_{\leq i+1}(s)) \mid \pi_1, \ldots, \pi_{i+1}, s, M_{\leq i}, \pi_{\leq i+1}(s)$$

is $i \cdot n^{-1/8}$-close to uniform in expectation. Observing that $\pi_{\leq i+1}(s)$ is determined by other variables in the conditioned, we complete the induction. □

References

[ABB+19] Sepehr Assadi, MohammadHossein Bateni, Aaron Bernstein, Vahab Mirrokni, and Cliff Stein. Coresets meet edcs: algorithms for matching and vertex cover on massive graphs. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1616–1635. SIAM, 2019.

[AG18] Kook Jin Ahn and Sudipto Guha. Access to data and number of iterations: Dual primal algorithms for maximum matching under resource constraints. *ACM Transactions on Parallel Computing (TOPC)*, 4(4):1–40, 2018.

[AKL17] Sepehr Assadi, Sanjeev Khanna, and Yang Li. On estimating maximum matching size in graph streams. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1723–1742. SIAM, 2017.

[AKLY16] 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*, pages 1345–1364. SIAM, 2016.

[AKSY20] Sepehr Assadi, Gillat Kol, Raghuvansh R Saxena, and Huacheng Yu. Multi-pass graph streaming lower bounds for cycle counting, max-cut, matching size, and other problems. In *FOCS*. https://arxiv.org/pdf/2009.03038.pdf, 2020.

[AMN19] Alexandr Andoni, Tal Malkin, and Negev Shekel Nosatzki. Two party distribution testing: Communication and security. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[AMS99] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and system sciences*, 58(1):137–147, 1999.

[AR20] Sepehr Assadi and Ran Raz. Near-quadratic lower bounds for two-pass graph streaming algorithms. In *FOCS*. https://arxiv.org/pdf/2009.01161.pdf, 2020.

[AV21] Sepehr Assadi and N Vishvajeet. Graph streaming lower bounds for parameter estimation and property testing via a streaming xor lemma. In *53rd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2021*, pages 612–625. Association for Computing Machinery, 2021.

[BC17] Suman K Bera and Amit Chakrabarti. Towards tighter space bounds for counting triangles and other substructures in graph streams. In *34th Symposium on Theoretical Aspects of Computer Science*, 2017.

[BCK+18] Vladimir Braverman, Stephen Chestnut, Robert Krauthgamer, Yi Li, David Woodruff, and Lin Yang. Matrix norms in data streams: Faster, multi-pass and row-order. In *International Conference on Machine Learning*, pages 649–658. PMLR, 2018.

[BDV18] Aditya Bhaskara, Samira Daruki, and Suresh Venkatasubramanian. Sublinear algorithms for maxcut and correlation clustering. In *45th International Colloquium on Automata, Languages, and Programming*, 2018.

[BFS86] László Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 337–347. IEEE, 1986.

[BGGS19] Mitali Bafna, Badih Ghazi, Noah Golowich, and Madhu Sudan. Communication-rounds tradeoffs for common randomness and secret key generation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1861–1871, 2019.

[BHP+21] Joanna Boyland, Michael Hwang, Tarun Prasad, Noah Singer, and Santhoshini Velusamy. Closed-form expressions for the sketching approximability of (some) symmetric boolean csps. *arXiv preprint arXiv:2112.06319*, 2021.

[BKKL17] Ruben Becker, Andreas Karrenbauer, Sebastian Krinninger, and Christoph Lenzen. Near-optimal approximate shortest paths and transshipment in distributed and streaming models. In *31 International Symposium on Distributed Computing*, 2017.

[BLWZ19] Maria-Florina Balcan, Yi Li, David P Woodruff, and Hongyang Zhang. Testing matrix rank, optimally. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 727–746. SIAM, 2019.

[BM13] Mark Braverman and Ankur Moitra. An information complexity approach to extended formulations. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 161–170, 2013.

[BOV13] Vladimir Braverman, Rafail Ostrovsky, and Dan Vilenchik. How hard is counting triangles in the streaming model? In *International Colloquium on Automata, Languages, and Programming*, pages 244–254. Springer, 2013.

[BRWY13] Mark Braverman, Anup Rao, Omri Weinstein, and Amir Yehudayoff. Direct products in communication complexity. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 746–755. IEEE Computer Society, 2013.

[BS15] Marc Bury and Chris Schwiegelshohn. Sublinear estimation of weighted matchings in dynamic data streams. In *ESA*, pages 263–274. 2015.

[BYJK04] Ziv Bar-Yossef, Thathachar S Jayram, and Iordanis Kerenidis. Exponential separation of quantum and classical one-way communication complexity. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 128–137, 2004.

[BYKS02] Ziv Bar-Yossef, Ravi Kumar, and D Sivakumar. Reductions in streaming algorithms, with an application to counting triangles in graphs. In *SODA*, volume 2, pages 623–632, 2002.

[CGS+21] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, Ameya Velingker, and Santhoshini Velusamy. Linear space streaming lower bounds for approximating csps. *arXiv preprint arXiv:2106.13078*, 2021.

[CGSV21] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, and Santhoshini Velusamy. Approximability of all finite csps with linear sketches. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1197–1208. IEEE, 2021.

[CGV20] Chi-Ning Chou, Alexander Golovnev, and Santhoshini Velusamy. Optimal streaming approximations for all boolean max-2csps and max-ksat. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 330–341. IEEE, 2020.

[CJ17] Graham Cormode and Hossein Jowhari. A second look at counting triangles in graph streams (corrected). *Theoretical Computer Science*, 683:22–30, 2017.

[CKKP21] Ashish Chiplunkar, John Kallaugher, Michael Kapralov, and Eric Price. Factorial lower bounds for (almost) random order streams. *arXiv preprint arXiv:2110.10091*, 2021.

[CKP+21a] Lijie Chen, Gillat Kol, Dmitry Paramonov, Raghuvansh R Saxena, Zhao Song, and Huacheng Yu. Almost optimal super-constant-pass streaming lower bounds for reachability. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 570–583, 2021.

[CKP+21b] Lijie Chen, Gillat Kol, Dmitry Paramonov, Raghuvansh R Saxena, Zhao Song, and Huacheng Yu. Near-optimal two-pass streaming algorithm for sampling random walks over directed graphs. In *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[EHL+18] Hossein Esfandiari, Mohammadtaghi Hajiaghayi, Vahid Liaghat, Morteza Monemizadeh, and Krzysztof Onak. Streaming algorithms for estimating the matching size in planar graphs and beyond. *ACM Transactions on Algorithms (TALG)*, 14(4):1–23, 2018.

[FGO17] Orr Fischer, Shay Gershtein, and Rotem Oshman. On the multiparty communication complexity of testing triangle-freeness. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, pages 111–120, 2017.

[FKM+04] Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. On graph problems in a semi-streaming model. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 531–543. Springer, 2004.

[FKM+09] Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. Graph distances in the data-stream model. *SIAM Journal on Computing*, 38(5):1709–1727, 2009.

[GKK+07] Dmitry Gavinsky, Julia Kempe, Iordanis Kerenidis, Ran Raz, and Ronald De Wolf. Exponential separations for one-way quantum communication complexity, with applications to cryptography. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 516–525, 2007.

[GKK12] 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)*, pages 468–485. SIAM, 2012.

[GKMS19] Buddhima Gamlath, Sagar Kale, Slobodan Mitrovic, and Ola Svensson. Weighted matchings via unweighted augmentations. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 491–500, 2019.

[GO16] Venkatesan Guruswami and Krzysztof Onak. Superlinear lower bounds for multipass graph processing. *Algorithmica*, 76(3):654–683, 2016.

[GS20] Noah Golowich and Madhu Sudan. Round complexity of common randomness generation: The amortized setting. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1076–1095. SIAM, 2020.

[GT19] Venkatesan Guruswami and Runzhou Tao. Streaming hardness of unique games. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 2019.

[GVV17] Venkatesan Guruswami, Ameya Velingker, and Santhoshini Velusamy. Streaming complexity of approximating max 2csp and max acyclic subgraph. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, nov 1995.

[HP19] Zengfeng Huang and Pan Peng. Dynamic graph stream algorithms in o (n) space. *Algorithmica*, 81(5):1965–1987, 2019.

[Jin19] Ce Jin. Simulating random walks on graphs in the streaming model. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, volume 124, pages 46:1–46:15, 2019.

[Kap13] Michael Kapralov. Better bounds for matchings in the streaming model. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 1679–1697. SIAM, 2013.

[KK15] Dmitry Kogan and Robert Krauthgamer. Sketching cuts in graphs and hypergraphs. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 367–376, 2015.

[KK19] Michael Kapralov and Dmitry Krachun. An optimal space lower bound for approximating max-cut. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 277–288, 2019.

[KKMO07] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable csps? *SIAM J. Comput.*, 37(1):319–357, 2007.

[KKP18] John Kallaugher, Michael Kapralov, and Eric Price. The sketching complexity of graph and hypergraph counting. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 556–567. IEEE, 2018.

[KKS15] Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Streaming lower bounds for approximating MAX-CUT. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1263–1282, 2015.

[KKSV17] Michael Kapralov, Sanjeev Khanna, Madhu Sudan, and Ameya Velingker. (1+ ω (1))-approximation to max-cut requires linear space. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1703–1722, 2017.

[KMPV19] John Kallaugher, Andrew McGregor, Eric Price, and Sofya Vorotnikova. The complexity of counting cycles in the adjacency list streaming model. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 119–133, 2019.

[KS92] Bala Kalyanasundaram and Georg Schintger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, 1992.

[LW16] Yi Li and David P Woodruff. On approximating functions of the singular values in a stream. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing (STOC)*, pages 726–739, 2016.

[McG05] Andrew McGregor. Finding graph matchings in data streams. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 170–181. Springer, 2005.

[McG14] Andrew McGregor. Graph stream algorithms: a survey. *ACM SIGMOD Record*, 43(1):9–20, 2014.

[MVV16] Andrew McGregor, Sofya Vorotnikova, and Hoa T Vu. Better algorithms for counting triangles in data streams. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 401–411, 2016.

[NW91] Noam Nisan and Avi Wigderson. Rounds in communication complexity revisited. In Cris Koutsougeras and Jeffrey Scott Vitter, editors, *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 419–429. ACM, 1991.

[Raz90] Alexander A Razborov. On the distributional complexity of disjointness. In *International Colloquium on Automata, Languages, and Programming*, pages 249–253. Springer, 1990.

[SGP11] Atish Das Sarma, Sreenivas Gollapudi, and Rina Panigrahy. Estimating pagerank on graph streams. *Journal of the ACM (JACM)*, 58(3):1–19, 2011.

[SSV21] Noah Singer, Madhu Sudan, and Santhoshini Velusamy. Streaming approximation resistance of every ordering csp. *arXiv preprint arXiv:2105.01782*, 2021.

[sub] List of open problems in sublinear algorithms: Problem 45. https://sublinear.info/45.

[VY11] 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*, pages 11–25. SIAM, 2011.

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il