Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 64 (2021)

# Streaming approximation resistance of every ordering CSP

Noah Singer* Madhu Sudan† Santhoshini Velusamy‡

July 7, 2021

**Abstract**

An ordering constraint satisfaction problem (OCSP) is given by a positive integer $k$ and a constraint predicate $\Pi$ mapping permutations on $\{1,\ldots,k\}$ to $\{0,1\}$. Given an instance of OCSP($\Pi$) on $n$ variables and $m$ constraints, the goal is to find an ordering of the $n$ variables that maximizes the number of constraints that are satisfied, where a constraint specifies a sequence of $k$ distinct variables and the constraint is satisfied by an ordering on the $n$ variables if the ordering induced on the $k$ variables in the constraint satisfies $\Pi$. Ordering constraint satisfaction problems capture natural problems including "Maximum acyclic subgraph (MAS)" and "Betweenness".

In this work we consider the task of approximating the maximum number of satisfiable constraints in the (single-pass) streaming setting, where an instance is presented as a stream of constraints. We show that for every $\Pi$, OCSP($\Pi$) is approximation-resistant to $o(n)$-space streaming algorithms, i.e., algorithms using $o(n)$ space cannot distinguish streams where almost every constraint is satisfiable from streams where no ordering beats the random ordering by a noticeable amount. This space bound is tight up to polylogarithmic factors. In the case of MAS our result shows that for every $\varepsilon > 0$, MAS is not $1/2+\varepsilon$-approximable in $o(n)$ space. The previous best inapproximability result only ruled out a $3/4$-approximation in $o(\sqrt{n})$ space.

Our results build on recent works of Chou, Golovnev, Sudan, Velingker, and Velusamy who show tight, linear-space inapproximability results for a broad class of (non-ordering) constraint satisfaction problems (CSPs) over arbitrary (finite) alphabets. Our results are obtained by building a family of appropriate CSPs (one for every $q$) from any given OCSP, and applying their work to this family of CSPs. To convert the resulting hardness results for CSPs back to our OCSP, we show that the hard instances from this earlier work have the following "small-set expansion" property: If the CSP instance is viewed as a hypergraph in the natural way, then for every partition of the hypergraph into small blocks most of the hyperedges are incident on vertices from distinct blocks. By exploiting this combinatorial property, in combination with the hardness results of the resulting families of CSPs, we give optimal inapproximability results for all OCSPs.

*Harvard College. Email: noahsinger@college.harvard.edu.

†School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts, USA. Supported in part by a Simons Investigator Award and NSF Award CCF 1715187. Email: madhu@cs.harvard.edu.

‡School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts, USA. Supported in part by a Simons Investigator Award and NSF Award CCF 1715187. Email: svelusamy@g.harvard.edu.



ISSN 1433-8092

# 1 Introduction

In this work we consider the complexity of "approximating" "ordering constraint satisfaction problems (OCSPs)" in the "streaming setting". We introduce these notions below before describing our results.

## 1.1 Orderings and Constraint Satisfaction Problems

In this work we consider optimization problems where the solution space is all possible orderings of $n$ variables. The Travelling Salesperson Problem and most forms of scheduling fit this framework, though our work considers a more restricted class of problems, namely *ordering constraint satisfaction problems (OCSPs)*. OCSPs as a class were first defined by Guruswami, Håstad, Manokaran, Raghavendra, and Charikar [GHM+11]. To describe them here, we first set up some notation and terminology, and then give some examples.

We let $[n]$ denote the set $\{0, \ldots, n-1\}$ and $\mathsf{S}_n$ denote the set of permutations on $[n]$, i.e., the set of bijections $\boldsymbol{\sigma} : [n] \to [n]$. We sometimes use $[\boldsymbol{\sigma}(0)\, \boldsymbol{\sigma}(1) \cdots \boldsymbol{\sigma}(n-1)]$ to denote $\boldsymbol{\sigma} : [n] \to [n]$. The solution space of ordering problems is $\mathsf{S}_n$, i.e., an *assignment* to $n$ variables is given by $\boldsymbol{\sigma} \in \mathsf{S}_n$. Given $k$ distinct integers $a_0, \ldots, a_{k-1}$ we define $\mathsf{ord}(a_0, \ldots, a_{k-1})$ to be the unique permutation in $\mathsf{S}_k$ which sorts $a_0, \ldots, a_{k-1}$. In other words, $\mathsf{ord}(a_0, \ldots, a_{k-1})$ is the unique permutation $\boldsymbol{\pi} \in \mathsf{S}_k$ such that $a_{\boldsymbol{\pi}(0)} < \cdots < a_{\boldsymbol{\pi}(k-1)}$. A *$k$-ary ordering constraint function* is given by a predicate $\Pi : \mathsf{S}_k \to \{0, 1\}$. An *ordering constraint application* on $n$ variables is given by a constraint function $\Pi$ and a $k$-tuple $\mathbf{j} = (j_0, j_1, \ldots, j_{k-1}) \in [n]^k$ where the $j_i$'s are distinct. In the interest of brevity we will often skip the term "ordering" below and further refer to constraint functions as "functions" and constraint applications as "constraints". A constraint $(\Pi, \mathbf{j})$ is *satisfied* by an assignment $\boldsymbol{\sigma} \in \mathsf{S}_n$ if $\Pi(\mathsf{ord}(\boldsymbol{\sigma}|_{\mathbf{j}})) = 1$, where $\boldsymbol{\sigma}|_{\mathbf{j}}$ is the $k$-tuple $(\boldsymbol{\sigma}(j_0), \ldots, \boldsymbol{\sigma}(j_{k-1})) \in [n]^k$.

A *maximum ordering constraint satisfaction problem*, $\mathsf{Max\text{-}OCSP}(\Pi)$, is specified by a single ordering constraint function $\Pi : \mathsf{S}_k \to \{0, 1\}$, for some positive integer arity $k$. An *instance* of $\mathsf{Max\text{-}OCSP}(\Pi)$ on $n$ variables is given by $m$ constraints $C_0, \ldots, C_{m-1}$ where $C_i = (\Pi, \mathbf{j}(i))$, i.e., the application of the function $\Pi$ to the variables $\mathbf{j}(i) = (j(i)_0, \ldots, j(i)_{k-1})$. (We omit $\Pi$ from the description of a constraint $C_i$ when clear from context.) The *value* of an ordering $\boldsymbol{\sigma} \in \mathsf{S}_n$ on an instance $\Psi = (C_0, \ldots, C_{m-1})$, denoted $\mathsf{val}_\Psi(\boldsymbol{\sigma})$, is the fraction of constraints satisfied by $\boldsymbol{\sigma}$, i.e., $\mathsf{val}_\Psi(\boldsymbol{\sigma}) = \frac{1}{m} \sum_{i \in [m]} \Pi(\mathsf{ord}(\boldsymbol{\sigma}|_{\mathbf{j}(i)}))$. The optimal value of $\Psi$ is defined as $\mathsf{val}_\Psi = \max_{\boldsymbol{\sigma} \in \mathsf{S}_n} \{\mathsf{val}_\Psi(\boldsymbol{\sigma})\}$.

The simplest, and arguably most interesting, problem which fits the $\mathsf{Max\text{-}OCSP}$ framework is the *maximum acyclic subgraph* ($\mathsf{MAS}$) problem. In this problem, the input is a directed graph on $n$ vertices, and the goal is to find an ordering of the vertices which maximize the number of forward edges. A simple depth-first search algorithm can decide whether a given graph $G$ has a *perfect* ordering (i.e., one which has *no* back edges); however, Karp [Kar72], in his famous list of 21 $\mathsf{NP}$-complete problems, proved the $\mathsf{NP}$-completeness of deciding whether, given a graph $G$ and a parameter $k$, there exists an ordering of the vertices such that at least $k$ edges are forward. For our purposes, $\mathsf{MAS}$ can be viewed as a 2-ary $\mathsf{Max\text{-}OCSP}$ problem, by defining the ordering constraint predicate $\Pi_{\mathsf{MAS}} : \mathsf{S}_2 \to \{0, 1\}$ given by $\Pi_{\mathsf{MAS}}([0\ 1]) = 1$ and $\Pi_{\mathsf{MAS}}([1; 0]) = 0$, and associating vertices with variables and edges with constraints. Indeed, an edge/constraint $(u, v)$ (where $u, v \in [n]$ are distinct variables/vertices) will be satisfied by an assignment/ordering $\boldsymbol{\sigma} \in \mathsf{S}_n$ iff $\Pi_{\mathsf{MAS}}(\mathsf{ord}(\boldsymbol{\sigma}|_{(u,v)})) = 1$, or equivalently, iff $\boldsymbol{\sigma}(u) < \boldsymbol{\sigma}(v)$.

A second natural $\mathsf{Max\text{-}OCSP}$ problem is the *maximum betweenness* ($\mathsf{MaxBtwn}$) problem. This is a 3-ary OCSP in which an ordering $\boldsymbol{\sigma}$ satisfies a constraint $(u, v, w)$ iff $\boldsymbol{\sigma}(v)$ is between $\boldsymbol{\sigma}(u)$

and $\boldsymbol{\sigma}(w)$, i.e., iff $\boldsymbol{\sigma}(u) < \boldsymbol{\sigma}(v) < \boldsymbol{\sigma}(w)$ *or* $\boldsymbol{\sigma}(u) > \boldsymbol{\sigma}(v) > \boldsymbol{\sigma}(w)$, and the goal is again to find the maximum number of satisfiable constraints. This is given by the constraint satisfaction function $\Pi_{\mathsf{Btwn}} : \mathsf{S}_3 \to \{0,1\}$ given by $\Pi_{\mathsf{Btwn}}([0\ 1\ 2]) = 1, \Pi_{\mathsf{Btwn}}([2\ 1\ 0]) = 1$, and $\Pi_{\mathsf{Btwn}}(\boldsymbol{\pi}) = 0$ for all other $\boldsymbol{\pi} \in \mathsf{S}_3$. The complexity of maximizing betweenness was originally studied by Opatrny [Opa79], who proved that even deciding whether a set of betweenness constraints is perfectly satisfiable is NP-complete.

### 1.2 Approximability

In this work we consider the *approximability* of ordering constraint satisfaction problems. We say that a (randomized) algorithm $A$ is an *$\alpha$-approximation algorithm* for Max-OCSP($\Pi$) if for every instance $\Psi$, $\alpha \cdot \mathsf{val}_\Psi \leq A(\Psi) \leq \mathsf{val}_\Psi$ with probability at least $2/3$ over the internal coin tosses of $A$. Thus our approximation factors $\alpha$ are numbers in the interval $[0,1]$.

Given $\Pi : \mathsf{S}_k \to \{0,1\}$ let $\rho(\Pi) = \frac{|\{\boldsymbol{\pi} \in \mathsf{S}_k | \Pi(\boldsymbol{\pi})=1\}|}{k!}$ denote the probability that $\Pi$ is satisfied by a random ordering. Every instance $\Psi$ of Max-OCSP($\Pi$) satisfies $\mathsf{val}_\Psi \geq \rho(\Pi)$ and thus the trivial algorithm that always outputs $\rho(\Pi)$ is a $\rho(\Pi)$-approximation algorithm for Max-OCSP($\Pi$). Under what conditions it is possible to beat this trivial approximation is a major open question.

For MaxBtwn, the trivial algorithm is a $\frac{1}{3}$-approximation. Chor and Sudan [CS98] showed that $(\frac{47}{48} + \varepsilon)$-approximating MaxBtwn is NP-hard, for every $\varepsilon > 0$. The $\frac{47}{48}$ factor was improved to $\frac{1}{2}$ by Austrin, Manokaran, and Wenner [AMW15]. For MAS, the trivial algorithm is a $\frac{1}{2}$-approximation. Newman [New00] showed that $(\frac{65}{66} + \varepsilon)$-approximating MAS is NP-hard, for every $\varepsilon > 0$. [AMW15] improved the $\frac{65}{66}$ to $\frac{14}{15}$, and Bhangale and Khot [BK19] further improved the factor to $\frac{2}{3}$.

We could hope that for every nontrivial nontrivial Max-OCSP($\Pi$), it is NP-hard to even $(\rho(\Pi) + \varepsilon)$-approximate Max-OCSP($\Pi$) for any constant factor $\varepsilon > 0$. This property is called *approximation resistance* (and we define it more carefully in the setting of streaming algorithms below). Approximation resistance based on NP-hardness is known for certain constraint satisfaction problems which do not fall under the Max-OCSP framework; this includes the seminal result of Håstad [Hås01] that it is NP-hard to $(\frac{7}{8} + \varepsilon)$-approximate Max3AND for any $\varepsilon > 0$. But as far as we know, such results are lacking for *any* Max-OCSP problem.

Given this state of affairs, Guruswami, Håstad, Manokaran, Raghavendra, and Charikar [GHM+11] proved the "next best thing": assuming the unique games conjecture (UGC) of Khot [Kho02], every Max-OCSP($\Pi$) is approximation-resistant. But the question of proving approximation resistance for polynomial-time algorithms without relying on unproven assumptions such as UGC and P $\neq$ NP remains unsolved. Towards this goal, in this work, we consider the approximability of Max-OCSP's in the *(single-pass) streaming model*, which we define below.

### 1.3 Streaming algorithms

A (single-pass) streaming algorithm is defined as follows. An instance $\Psi = (C_0, \ldots, C_{m-1})$ of Max-OCSP($\Pi$) is presented as a stream of constraints with the $i$th element of the stream being $\mathbf{j}(i)$ where $C_i = (\Pi, \mathbf{j}(i))$. A streaming algorithm $A$ updates its state with each element of the stream and at the end produces the output $A(\Psi) \in [0,1]$ (which is supposed to estimate $\mathsf{val}_\Psi$). The measure of complexity of interest to us is the space used by $A$ and in particular we distinguish between algorithms that use space polylogarithmic in the input length and space that grows polynomially ($\Omega(n^\delta)$ for $\delta > 0$) in the input length.

We say that a problem Max-OCSP($\Pi$) is *approximable (in the streaming setting)* if we can beat the trivial $\rho(\Pi)$-approximation algorithm by a positive constant factor. Specifically Max-OCSP($\Pi$) is said to be *approximable* if for every $\delta > 0$ there exists $\varepsilon > 0$ and a space $O(n^\delta)$ algorithm $A$ that is a $(\rho(\Pi) + \varepsilon)$-approximation algorithm for Max-OCSP($\Pi$), We say Max-OCSP($\Pi$) is *approximation-resistant (in the streaming setting)* otherwise.

In recent years, investigations into CSP approximability in the streaming model have been strikingly successful, resulting in tight characterizations of streaming approximability for many problems [KK15, KKS15, KKSV17, GVV17, GT19, KK19, CGV20, CGSV21a, CGSV21b, CGS+21]. Most of these papers studied approximability, not of *ordering* CSPs, but of "non-ordering CSPs" where the variables can take values in a finite alphabet. ([GVV17] and [GT19] are the exceptions, and we will discuss them below.) While single-pass streaming algorithms are a weaker model than general polynomial-time algorithms, we do remark that nontrivial approximations for many problems are possible in the streaming setting. In particular, the Max2AND problem is (roughly) $\frac{4}{9}$-approximable in the streaming setting (whereas the trivial approximation is a $\frac{1}{4}$-approximation) [CGV20].

### 1.4 Main result and comparison to prior and related works

**Theorem 1.1** (Main theorem)**.** *For every $k \in \mathbb{N}$ and every $\Pi : \mathsf{S}_k \to \{0, 1\}$, Max-OCSP($\Pi$) is approximation resistant in the (single-pass) streaming setting. In particular for every $\varepsilon > 0$, every $(\rho(\Pi) + \varepsilon)$-approximation algorithm $A$ for Max-OCSP($\Pi$) requires $\Omega(n)$ space.*

In particular our theorem implies that MAS is not $1/2 + \varepsilon$-approximable in $o(n)$ space for every $\varepsilon > 0$, and MaxBtwn is not $1/3 + \varepsilon$-approximable. Theorem 1.1 is restated in Section 3 and proved there.

Theorem 1.1 parallels the classical result of [GHM+11], who prove that Max-OCSP($\Pi$) is approximation resistant with respect to *polynomial-time* algorithms, for every $\Pi$, assuming the unique games conjecture. In our setting of streaming algorithms, the only problem that seems to have been previously explored in the literature was MAS, and even in this case a tight approximability result was not known.

In the case of MAS, Guruswami, Velingker, and Velusamy [GVV17] proved that for every $\varepsilon > 0$, MAS is not $(\frac{7}{8} + \varepsilon)$-approximable in $o(\sqrt{n})$ space using a gadget reduction from the Boolean hidden matching problem [GKK+08]. A stronger $o(\sqrt{n})$-space, 3/4-approximation hardness for MAS is indicated in the work of Guruswami and Tao [GT19], who prove streaming bounds for unique games, an "non-ordering" CSP problem, and suggest a reduction from unique games to MAS.

As far as we know, our result is the first tight approximability result for Max-OCSP($\Pi$) for *any* non-constant $\Pi$ in $\Omega(n^\delta)$ space for any $\delta > 0$, and it yields tight approximability results for *every* $\Pi$ in *linear* space. We remark that this linear space bound is also optimal (up to logarithmic factors); similarly to the observation in [CGS+21] for non-ordering CSPs, Max-OCSP($\Pi$) values can be approximated arbitrarily well in $\widetilde{O}(n)$ space by subsampling $O(n)$ constraints from the input instance and then solving the Max-OCSP($\Pi$) problem on this subinstance exactly.[1]

Chakrabarti, Ghosh, McGregor, and Vorotnikova [CGMV20] recently also studied directed graph ordering problems (e.g., acyclicity testing, $(s, t)$-connectivity, topological sorting) in the streaming setting. For the problems that considered in [CGMV20], their work gives *super-linear*

[1]This assumes a definition of streaming complexity which makes no restriction on time complexity. Of course, if we restrict to polynomial time, then assuming the unique games conjecture, no nontrivial approximation will be possible.

space lower bounds even for multi-pass streaming algorithms. Note that for our problems an $\widetilde{O}(n)$ upper bound holds, suggesting that their problems are not OCSPs. Indeed this is true, but one of the problems considered is close enough to MAS to allow a more detailed comparison. The specific problem is the *minimum feedback arc set* (MFAS) problem, the goal of which is to output the fractional size of the smallest set of edges whose removal produces an acyclic subgraph. In other words, the sum of MFAS value of a graph and the MAS value of the graph is exactly one. [CGMV20] proved that for every $\kappa > 1$, $\kappa$-approximating[2] the MFAS value requires $\Omega(n^2)$ space in the streaming setting (for a single pass, and more generally $\Omega(n^{1+\Omega(1/p)}/p^{O(1)})$ space for $p$ passes). Note that such lower bounds are obtained using instances with optimum MFAS values that are $o(1)$. Thus the MAS values in the same graph are $1 - o(1)$ (even in the **NO** instances) and thus these results usually do not imply any hardness of approximation for MAS.

### 1.5 Techniques

Our general approach is to start with a hardness result for CSPs over alphabets of size $q$ (i.e., constraint satisfaction problems where the variables take values in $[q]$), and then to reduce these CSPs to the OCSP at hand. While this general approach is not new, the optimality of our results seems to come from the fact that we choose the CSP problem carefully, and are able to get optimal hardness results for problems of our choice thanks to a general result of Chou, Golovnev, Sudan, Velingker and Velusamy [CGS+21]. Thus whereas previous approaches towards proving hardness of MAS, for example, were unable to get optimal hardness results for MAS despite starting with optimal hardness results of the source (unique games), by choosing our source problem more carefully we manage to get optimal hardness results. In the remainder of this section, we describe and motivate this approach towards proving the approximation-resistance of Max-OCSP's.

#### 1.5.1 Special case: The intuition for MAS

We start by describing our proof technique for the special case of the MAS problem. In this section, for readability, we (mostly) use the language of graphs, edges, and vertices instead of instances, constraints, and variables.

Similarly to earlier work in the setting of streaming approximability (e.g., [KKS15]), we prove inapproximability of MAS by exhibiting a pair of distributions, which we denote $\mathcal{G}^Y$ and $\mathcal{G}^N$, satisfying the following two properties:

1. $\mathcal{G}^Y$ and $\mathcal{G}^N$ are "indistinguishable" to streaming algorithms (to be defined formally below)
2. (With high probability) $\mathcal{G}^Y$ has high MAS values ($\approx 1$) and $\mathcal{G}^N$ has low MAS values ($\approx \frac{1}{2}$)

The existence of such distributions would suffice to establish the theorem: there cannot be any streaming approximation for MAS, since any such algorithm would be able to distinguish these distributions. But how are we to actually construct distributions $\mathcal{G}^Y$ and $\mathcal{G}^N$ satisfying these properties?

The strategy which has proved successful in past work for proving streaming approximation resistance of other varieties of CSPs was roughly to let the $\mathcal{G}^N$ graphs be completely random, while $\mathcal{G}^Y$ graphs are sampled with "hidden structure", which is essentially a very good assignment. Then,

[2] For minimization problems a $\kappa$ approximation is one whose value is at least the minimum value and at most $\kappa$ times larger than the minimum. Thus approximation factors are larger than 1.

one would show that streaming algorithms cannot detect the existence of such hidden structure, via a reduction to a communication game (typically a variant of Boolean hidden matching [GKK+08, VY11]). In our setting, we might hope that the hidden structure could simply be an ordering; that is, we could hope to define $\mathcal{G}^Y$ by first sampling a random ordering of the vertices, then sampling edges which go forward with respect to this ordering, and then perhaps adding some noise. But unfortunately, we lack the techniques to prove communication lower bounds when orderings are the hidden structure.

Hence, instead of seeking a direct proof of an indistinguishability result, in this paper, we turn back to earlier indistinguishability results proven in the context of *non-ordering CSPs*. In this setting, variables take on values in an alphabet $[q]$, and constraints specify allowed values of subsets of the variables. In particular, two distinct variables may take on the same value in $[q]$, whereas in the ordering setting, every variable in $[n]$ must get a distinct value in $[n]$. (See Section 4.1 for a formal definition.) We will set $q$ to be a large constant, carefully design a non-ordering CSP function, employ past results (i.e., [CGS+21]) to characterize its streaming inapproximability, examine the $\mathcal{G}^Y$ and $\mathcal{G}^N$ graphs created in the reduction, and then show that $\mathcal{G}^N$ graphs have low $\mathsf{MAS}$ values while the hidden structure in the $\mathcal{G}^Y$ graphs — even if it isn't an ordering per se — guarantees high $\mathsf{MAS}$ values.

Why would we expect such an idea to work out, and how do we properly choose the non-ordering CSP constraint function? To begin, this constraint function will be a 2-ary function $f : [q]^2 \to \{0,1\}$. Let $\mathsf{Max\text{-}CSP}(f)$ denote the non-ordering CSP problem of maximizing the number of $f$ constraints satisfied by an assignment $\mathbf{b} \in [q]^n$. We will view an input graph $G$ simultaneously as an instance of $\mathsf{MAS}$ and as an instance of $\mathsf{Max\text{-}CSP}(f)$, with the same underlying set of edges/constraints. For a graph $G$, let $\mathsf{val}_G$ denote its $\mathsf{MAS}$ value and $\overline{\mathsf{val}}_G$ its value in $\mathsf{Max\text{-}CSP}(f)$. We will choose $f$ so that the indistinguishable hard distributions $\mathcal{G}^Y$ and $\mathcal{G}^N$ (originating from the reduction of [CGS+21]) have the following four properties:

1. With high probability over $G \sim \mathcal{G}^Y$, $\overline{\mathsf{val}}_G \approx 1$.
2. With high probability over $G \sim \mathcal{G}^N$, $\overline{\mathsf{val}}_G \approx \frac{1}{2}$.
3. For all $G$, $\mathsf{val}_G \geq \overline{\mathsf{val}}_G$.
4. With high probability over $G \sim \mathcal{G}^N$, $\mathsf{val}_G$ is not much larger than $\overline{\mathsf{val}}_G$.

Together, these items will suffice to prove the theorem since Item 2 and Item 4 together imply that with high probability over $G \sim \mathcal{G}^N$, $\mathsf{val}_G \approx \frac{1}{2}$, while Item 1 and Item 3 together imply that with high probability over $G \sim \mathcal{G}^Y$, $\mathsf{val}_G \approx 1$.

Concretely, we setup the non-ordering CSP function as follows. Recall that $\Pi_{\mathsf{MAS}}([0\ 1]) = 1$ while $\Pi_{\mathsf{MAS}}([1\ 0]) = 0$. We define the constraint function $f^q_{\mathsf{MAS}} : [q]^2 \to \{0,1\}$ by $f^q_{\mathsf{MAS}}(x,y) = 1$ iff $x < y$. Note that $f^q_{\mathsf{MAS}}$ is supported on $\frac{q(q-1)}{2} \approx \frac{1}{2}$ pairs in $[q]^2$. We first show that [CGS+21]'s results imply that $\mathsf{Max\text{-}CSP}(f^q_{\mathsf{MAS}})$ is approximation-resistant, and pick $\mathcal{G}^Y$ and $\mathcal{G}^N$ as the **YES** and **NO** distributions witnessing this result. This immediately yields Item 1 and Item 2 above. It remains to prove Item 4 and Item 3. In the remainder of this subsection, we sketch the proofs; see Figure 1 for a visual depiction, and Section 4 for the formal proofs.

Towards Item 3, we take advantage of the fact that $\mathsf{Max\text{-}CSP}(f^q_{\mathsf{MAS}})$ captures a "$q$-coarsening" of $\mathsf{MAS}$. We consider an arbitrary $\mathsf{Max\text{-}CSP}(f^q_{\mathsf{MAS}})$-assignment $\mathbf{b} \in [q]^n$ for a graph $G$, which assigns to the $i$-th vertex a value $b_i \in [q]$. We construct an ordering of $G$'s vertices by first placing the "block" of vertices assigned value 0, then the block of vertices assigned 1, etc., finally placing the vertices assigned value $q-1$. (Within any particular block, the vertices may be ordered arbitrarily.) Now whenever an edge $(u, v)$ is satisfied by $\mathbf{b}$ when viewing $G$ as an instance of $\mathsf{Max\text{-}CSP}(f^q_{\mathsf{MAS}})$ — that is, whenever $b_v > b_u$ — the same edge will be satisfied by our constructed ordering when viewing $G$ as an instance of $\mathsf{MAS}$. Hence $\mathsf{val}_G \geq \overline{\mathsf{val}}_G$.

Towards Item 4, we can no longer use the results of [CGS+21] as a black box. Instead, we show that the graphs $\mathcal{G}^N$ are "small partition expanders" in a specific sense: any partition of the constraint graph into $q$ roughly equal sized blocks has very few edges, specifically a $o(1)$ fraction, which lie *within* the blocks. Now, we think of an ordering $\boldsymbol{\sigma} \in \mathsf{S}_n$ variables as dividing the $n$ variables into $q$ blocks with variables $\boldsymbol{\sigma}(0), \ldots, \boldsymbol{\sigma}(n/q-1)$ being in the first block, $\boldsymbol{\sigma}(n/q), \ldots, \boldsymbol{\sigma}(2n/q-1)$ being in the second block and so on. Whenever an edge $(u, v)$ is satisfied by $\boldsymbol{\sigma}$ when viewing $G$ as an instance of $\mathsf{MAS}$, it will also be satisfied by our constructed ordering when viewing $G$ as an instance of $\mathsf{Max\text{-}CSP}(f^q_{\mathsf{MAS}})$, *unless* $u$ and $v$ end up in the same block; but by the small partition expansion condition, this happens only for $o(1)$ fraction of the edges. Hence $\mathsf{val}_G \leq \overline{\mathsf{val}}_G + o(1)$.

We remark in passing that our notion of coarsening is somewhat similar to, but not the same as, that used in previous works, notably [GHM+11]. In particular the techniques used to compare the OCSP value (before coarsening) with the non-ordering CSP value (after coarsening) are somewhat different: Their analysis involves more sophisticated tools such as influence of variables and Gaussian noise stability. The proof of Item 4 in our setting, in contrast, uses a more elementary analysis of the type common with random graphs. Finally, we remark that in the rest of the paper, in the interest of self-containedness, our construction will "forget" about $\mathsf{Max\text{-}CSP}(f^q_{\mathsf{MAS}})$, define the distributions $\mathcal{G}^Y$ and $\mathcal{G}^N$ explicitly, and treat $\overline{\mathsf{val}}_G$ simply as an artifact of the analysis which calculates the $\mathsf{MAS}$ values of $\mathcal{G}^Y$ and $\mathcal{G}^N$, but we hope that this discussion has motivated the construction.

### 1.5.2 Extending to general ordering CSPs

Extending the idea to other OCSPs involves two additional steps. Given the constraint function $\Pi$ (of arity $k$) and positive integer $q$, we define $f^q_\Pi$ analogously to $f^q_{\mathsf{MAS}}$. We then explicitly describe the **YES** and **NO** distributions of $\mathsf{Max\text{-}CSP}(f^q_\Pi)$ which the general theorem of [CGS+21] shows are indistinguishable to $o(n)$ space algorithms. Crucial to this application is the observation that $f^q_\Pi$ is an "$1-k-1/q$-wide" function, where $f^q_\Pi$ is $\omega$-wide if there exists a vector $\mathbf{v} = (v_0, \ldots, v_{k-1}) \in [q]^k$ such that for an $\omega$-fraction of $a \in [q]$, we have $f^q_\Pi(v_0 + a, \ldots, v_{k-1} + a) = 1$. This would allow us to conclude that $\mathsf{Max\text{-}CSP}(f^q_\Pi)$ is hard to approximate to within factor of roughly $\rho/\omega$, though as in the special case of $\mathsf{MAS}$ we do not use this result explicitly.[3] Instead, the second step of our proof replicates Item 4 above. We give an analysis of the partition expansion in the **NO** instances arising from the construction in [CGS+21]. Specifically we show that the constraint hypergraph is now a "small partition hypergraph expander", in the sense that any partition into $q$ roughly equal sized blocks would have very few hyperedges that contain even two vertices from the same block. With these two additional ingredients in place, and following the same template as in the hardness for $\mathsf{MAS}$, we immediately get the approximation resistance of $\mathsf{Max\text{-}OCSP}(\Pi)$ for general $\Pi$.

**This version.** The current version of this paper improves on a previous version of this paper [SSV21] that gave only $\Omega(\sqrt{n})$ space lower bounds for all OCSPs. Our improvement to $\Omega(n)$

[3] Indeed, the "width" observation is involved in the proof of Item 1 and Item 2 even in the $\mathsf{MAS}$ case (with $k = 2$).

space lower bounds comes by invoking the more recent results of [CGS+21], whereas our previous version used the strongest lower bounds for CSPs that were available at the time from an earlier work of Chou, Golovnev, Sudan, and Velusamy [CGSV21b]. The results of [CGSV21b] are quantitatively weaker for the problems considered in [CGS+21], though their results apply to a broader collection of problems. Interestingly for our application, which covers *all* OCSPs, the narrower set of problems considered in [CGS+21] suffices. We also note that the proof in this version of our paper is more streamlined thanks to the notion of "wide" constraints introduced and used in [CGS+21].

**Organization of the rest of the paper.** In Section 2 we introduce some notation we use and background material. In Section 3 we prove our main theorem, Theorem 1.1. In this section we also introduce two distributions on $\mathsf{Max\text{-}OCSP}(\Pi)$ instances, the **YES** distribution and the **NO** distribution, and state lemmas asserting that these distributions are concentrated on instances with high, and respectively low, OCSP value; and that these distributions are indistinguishable to single-pass small space streaming algorithms. We prove the lemmas on the OCSP values in Section 4, and prove the indistinguishability lemma in Section 5.

# 2 Preliminaries and definitions

## 2.1 Basic notation

Some of the notation we use is already introduced in Section 1.1. Here we introduce some more notation we use.

The *support* of an ordering constraint function $\Pi : \mathsf{S}_k \to \{0,1\}$ is the set $\mathsf{supp}(\Pi) = \{\boldsymbol{\pi} \in \mathsf{S}_k | \Pi(\boldsymbol{\pi}) = 1\}$.

Addition of elements in $[q]$ is implicitly taken modulo $q$.

Throughout this paper we will be working with $k$-uniform *ordered* hypergraphs, or simply $k$-hypergraphs, defined in the sequel. Given a finite set $V$, an (ordered, self-loop-free) *$k$-hyperedge* $e = (v_1, \ldots, v_k)$ is a sequence of $k$ distinct elements $v_1, \ldots, v_k \in V$. We stress that the ordering of vertices within an edge is important to us. An (ordered, self-loop-free, multi-) *$k$-hypergraph* $G = (V, E)$ is given by a set of vertices $V$ and a multiset $E = E(G) \subseteq V^k$ of *$k$-hyperedges* A $k$-hyperedge $\mathbf{e}$ is *incident* on a vertex $v$ if $v$ appears in $\mathbf{e}$. Let $\Gamma(\mathbf{e}) \subseteq V$ denote the set of vertices to which a $k$-hyperedge $\mathbf{e}$ is incident, and let $m = m(G)$ denote the number of $k$-hyperedges in $G$.

A $k$-hypergraph is a *$k$-hypermatching* if it has the property that no pair of (distinct) $k$-hyperedges is incident on the same vertex. For $\alpha \leq \frac{1}{k}$, an *$\alpha$-partial $k$-hypermatching* is a $k$-hypermatching which contains $\alpha n$ $k$-hyperedges. We let $\mathcal{H}_{k,n,\alpha}$ denote the uniform distribution over all $\alpha$-partial $k$-hypermatchings on $[n]$.

A vector $\mathbf{b} = (b_0, \ldots, b_{n-1}) \in [q]^n$ may be viewed as a *$q$-partition* of $[n]$ into *blocks* $\mathbf{b}^{-1}(0), \ldots, \mathbf{b}^{-1}(q-1)$, where the $i$-th block $\mathbf{b}^{-1}(i)$ is defined as the set of indices $\{j \in [n] : b_j = i\}$. Given $\mathbf{b} = (b_0, \ldots, b_{n-1}) \in [q]^n$ and an indexing vector $\mathbf{j} = (j_0, \ldots, j_{k-1}) \in [n]^k$, we define $\mathbf{b}|_{\mathbf{j}} = (b_{j_0}, \ldots, b_{j_{k-1}})$.

Given an instance $\Psi$ of $\mathsf{Max\text{-}OCSP}(\Pi)$ on $n$ variables, we define the *constraint hypergraph* $G(\Psi)$ to be the $k$-hypergraph on $[n]$, where each $k$-hyperedge corresponds to a constraint (given by the exact same $k$-tuple). We also let $m(\Psi)$ denote the number of constraints in $\Psi$ (equiv., the number of $k$-hyperedges in $G(\Psi)$).

## 2.2 Concentration bound

We also require the following form of *Azuma's inequality*, a concentration inequality for submartingales. For us the following form, for Boolean-valued random variables with bounded conditional expectations taken from Kapralov and Krachun [KK19], is particularly convenient.

**Lemma 2.1** ([KK19, Lemma 2.5])**.** *Let $X_0, \ldots, X_{m-1}$ be (not necessarily independent) $\{0,1\}$-valued random variables, such that for some $p \in (0,1)$, $\mathsf{E}[X_i \mid X_0, \ldots, X_{i-1}] \le p$ for every $i \in [m]$. Then if $\mu := pm$, for every $\nu > 0$,*

$$\Pr[X_0 + \cdots + X_{m-1} \ge \mu + \nu] \le \exp\left(-\frac{1}{2} \cdot \frac{\nu^2}{\mu + \nu}\right).$$

# 3 The streaming space lower bound

In this section we prove our main theorem, modulo some lemmas that we prove in later sections. We restate the theorem below for convenience.

**Theorem 1.1** (Main theorem)**.** *For every $k \in \mathbb{N}$ and every $\Pi : \mathsf{S}_k \to \{0,1\}$, Max-OCSP($\Pi$) is approximation resistant in the (single-pass) streaming setting. In particular for every $\varepsilon > 0$, every $(\rho(\Pi) + \varepsilon)$-approximation algorithm $A$ for Max-OCSP($\Pi$) requires $\Omega(n)$ space.*

Our lower bound is proved, as is usual for such statements, by showing that no small space algorithm can "distinguish" **YES** instances with OCSP value at least $1 - \varepsilon/2$, from **NO** instances with OCSP value at most $\rho(\Pi) + \varepsilon/2$. Such a statement is in turn proved by exhibiting two families of distributions, the **YES** distributions and the **NO** distributions, and showing these are indistinguishable. Specifically we choose some parameters $q, T, \alpha$ and a permutation $\boldsymbol{\pi} \in \mathsf{S}_k$ carefully and define two distributions $\mathcal{G}^Y = \mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}(\Pi)$ and $\mathcal{G}^N = \mathcal{G}^{N}_{q,n,\alpha,T}(\Pi)$. We claim that for our choice of parameters $\mathcal{G}^Y$ is supported on instances with value at least $1-\varepsilon/2$ — this is asserted in Lemma 3.3. Similarly we claim that $\mathcal{G}^N$ is mostly supported (with probability $1 - o(1)$) on instances with value at most $\rho(\Pi)+\varepsilon/2$ (see Lemma 3.4). Finally we assert in Lemma 3.5 that any algorithm that distinguishes $\mathcal{G}^Y$ from $\mathcal{G}^N$ with "advantage" at least 1/8 (i.e., accepts $\Psi \sim \mathcal{G}^Y$ with probability 1/8 more than $\Psi \sim \mathcal{G}^N$) requires $\Omega(n)$ space.

Assuming Lemma 3.3, Lemma 3.4, and Lemma 3.5 the proof of Theorem 1.1 is straightforward and proved at the end of this section. Proofs of Lemma 3.3 and Lemma 3.4 are in Section 4 and of Lemma 3.5 in Section 5.

## 3.1 Distribution of hard instances

For $\ell, k \in [q]$, define the $k$-tuple of "contiguous" values $\mathbf{v}_q^{(\ell)} = (\ell, \ldots, \ell + k - 1) \in [q]^k$. Crucially, since the addition here is taken modulo $q$, we may have $\ell + k - 1 < \ell$ and in particular $\mathsf{ord}(\mathbf{v}_q^{(\ell)})$ may not be the identity.

For a $k$-tuple $\mathbf{a} = (a_0, \ldots, a_{k-1})$ and a permutation $\boldsymbol{\pi} \in \mathsf{S}_k$, define the *permuted* $k$-tuple $\mathbf{a}_{\boldsymbol{\pi}}$ as $(a_{\boldsymbol{\pi}^{-1}(0)}, \ldots, a_{\boldsymbol{\pi}^{-1}(k-1)})$. In particular, we have $(\mathbf{v}_q^{(\ell)})_{\boldsymbol{\pi}} = (\boldsymbol{\pi}^{-1}(0) + \ell, \ldots, \boldsymbol{\pi}^{-1}(k-1) + \ell)$. We define $\mathbf{a}_{\boldsymbol{\pi}}$ in this way because:

**Proposition 3.1.** *If $\mathbf{a}$ is a $k$-tuple of distinct integers, then $\mathsf{ord}(\mathbf{a}_{\boldsymbol{\pi}}) = \mathsf{ord}(\mathbf{a}) \circ \boldsymbol{\pi}$ (where $\circ$ denotes composition of permutations).*

*Proof.* Recall that $\mathsf{ord}(\mathbf{a})$ is the unique permutation $\boldsymbol{\tau}$ such that $a_{\boldsymbol{\tau}(0)} < \cdots < a_{\boldsymbol{\tau}(k-1)}$. Let $\boldsymbol{\tau} = \mathsf{ord}(\mathbf{a})$, and let $\boldsymbol{\sigma} = \mathsf{ord}(\mathbf{a}_{\boldsymbol{\pi}})$, so that $\boldsymbol{\sigma}$ is the unique permutation such that $a_{\boldsymbol{\sigma}(\boldsymbol{\pi}^{-1}(0))} < \cdots < a_{\boldsymbol{\sigma}(\boldsymbol{\pi}^{-1}(k-1))}$. Then $\boldsymbol{\tau} = \boldsymbol{\sigma} \circ \boldsymbol{\pi}^{-1}$. Hence $\boldsymbol{\tau} \circ \boldsymbol{\pi} = \boldsymbol{\sigma}$, as desired. □

We now formally define our **YES** and **NO** distributions for $\mathsf{Max}\text{-}\mathsf{OCSP}(\Pi)$.

**Definition 3.2** **($\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}(\Pi)$ and $\mathcal{G}^{N}_{q,n,\alpha,T}(\Pi)$).** *For $k \in \mathbb{N}$ and $\Pi : \mathsf{S}_k \to \{0,1\}$, let $q, n, T \in \mathbb{N}$, $\alpha > 0$, and let $B = N$ or $B = (Y, \boldsymbol{\pi})$ for some $\boldsymbol{\pi} \in \mathsf{supp}(\Pi)$. We define the distribution $\mathcal{G}^{B}_{q,n,\alpha,T}$, over $n$-variable $\mathsf{Max}\text{-}\mathsf{OCSP}(\Pi)$ instances, as follows:*

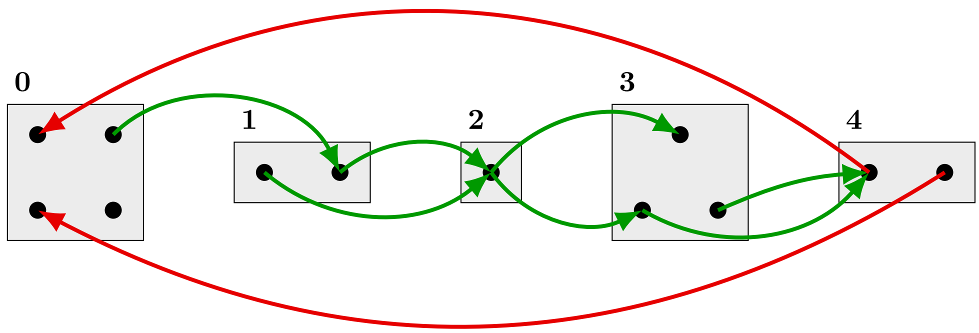
1. *Sample a uniformly random $q$-partition $\mathbf{b} = (b_0, \ldots, b_{n-1}) \in [q]^n$.*
2. *Sample $T$ hypermatchings independently $\widetilde{G}_0, \ldots, \widetilde{G}_{T-1} \sim \mathcal{H}_{k,n,\alpha}$.*
3. *For each $t \in [T]$, do the following:*
   - *Let $G_t$ be an empty $k$-hypergraph on $[n]$.*
   - *For each $k$-hyperedge $\widetilde{\mathbf{e}} = (j_0, \ldots, j_{k-1}) \in E(\widetilde{G}_t)$:*
     - ***(YES)** If $B = (Y, \boldsymbol{\pi})$, and there exists $\ell \in [q]$ such that $\mathbf{b}|_{\mathbf{j}} = (\mathbf{v}^{(\ell)}_q)_{\boldsymbol{\pi}}$, add $\widetilde{\mathbf{e}}$ to $G_t$ with probability $\frac{1}{q}$.*
     - ***(NO)** If $B = N$, add $\widetilde{\mathbf{e}}$ to $G_t$ with probability $\frac{1}{q^k}$.*
4. *Let $G := G_0 \cup \cdots \cup G_{T-1}$.*
5. *Return the $\mathsf{Max}\text{-}\mathsf{OCSP}(\Pi)$ instance $\Psi$ on $n$ variables given by the constraint hypergraph $G$.*

*We say that an algorithm* **ALG** *achieves advantage $\delta$ in distinguishing $\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}(\Pi)$ from $\mathcal{G}^{N}_{q,n,\alpha,T}(\Pi)$ if there exists an $n_0$ such that for all $n \geq n_0$, we have*

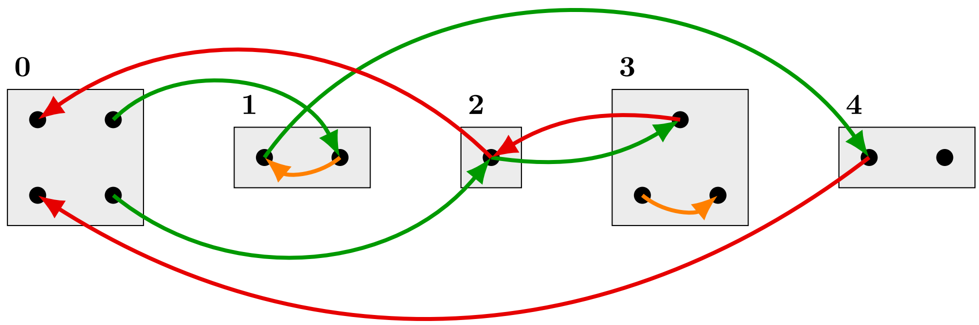
$$\left| \Pr_{\Psi \sim \mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}(\Pi)}[\mathbf{ALG}(\Psi) = 1] - \Pr_{\Psi \sim \mathcal{G}^{N}_{q,n,\alpha,T}(\Pi)}[\mathbf{ALG}(\Psi) = 1] \right| \geq \delta.$$

We make several remarks on this definition. Firstly, note that the constraints within $\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}(\Pi)$ and $\mathcal{G}^{N}_{q,n,\alpha,T}(\Pi)$ do not directly depend on $\Pi$. We still parameterize the distributions by $\Pi$, since they are formally distributions over $\mathsf{Max}\text{-}\mathsf{OCSP}(\Pi)$ instances; $\Pi$ also determines the set of allowed permutations $\boldsymbol{\pi}$ in the **YES** case as well as the underlying arity $k$. However, we will omit the parameterization $(\Pi)$ when clear from context. Secondly, we note that when sampling an instance from $\mathcal{G}^{N}_{q,n,\alpha,T}$, the partition $\mathbf{b}$ has no effect, and so $\mathcal{G}^{N}_{q,n,\alpha,T}$ is completely random. Hence these instances fit into the standard paradigm for streaming lower bounds of "random graphs vs. random graphs with hidden structure". Finally, we observe that the number of constraints in both distributions is distributed as a sum of $m = n\alpha T$ independent Bernoulli$(\frac{1}{q^k})$ random variables.

In the following section we state lemmas which highlight the main properties of the distributions above. See Figure 1 for a visual interpretation of the distributions in the case of $\mathsf{MAS}$.




(a) Constraint graph of a sample MAS instance drawn from $\mathcal{G}^Y$




(b) Constraint graph of a sample MAS instance drawn from $\mathcal{G}^N$

Figure 1: The constraint graphs of MAS instances which could plausibly be drawn from $\mathcal{G}^Y$ and $\mathcal{G}^N$, respectively, for $q = 5$ and $n = 12$. Recall that MAS is a binary Max-OCSP with ordering constraint function $\Pi$ supported only on $[0\ 1]$. According to the definition of $\mathcal{G}^Y$ (see Definition 3.2, with $\boldsymbol{\pi} = [0\ 1]$), instances are sampled by first sampling a $q$-partition $\mathbf{b} = (b_0, \ldots, b_{n-1}) \in [q]^n$, and then sampling some edges; every sampled edge $(u, v)$ must satisfy $b_v = b_u + 1 \pmod q$. On the other hand, there are no requirements on $(b_u, b_v)$ for instances sampled from $\mathcal{G}^N$. Above, the blocks of the partition $\mathbf{b}$ are labelled $0, \ldots, 4$, and the reader can verify that the edges satisfy the appropriate requirements. We also color the edges in a specific way: We color an edge $(u, v)$ green, orange, or red if $b_v > b_u$, $b_v = b_u$, or $b_v < b_u$, respectively. This visually suggests important elements of our proofs that $\mathcal{G}^Y$ has MAS values close to 1 and $\mathcal{G}^N$ has MAS values close to $\frac{1}{2}$ (for formal statements, see Lemma 3.3 and Lemma 3.4, respectively). Specifically, in the case of $\mathcal{G}^Y$, if we arbitrarily arrange the vertices in each block, we will get an ordering in which every green edge is satisfied, and we expect all but $\frac{1}{q}$ fraction of the edges to be satisfied (i.e., all but those which go from block $q - 1$ to block 0). On the other hand, if we executed a similar process in $\mathcal{G}^N$, the resulting ordering would satisfy all green edges and some subset of the orange edges; together, in expectation, these account only for $\frac{q(q+1)}{2q^2} = \frac{q+1}{2q} \approx \frac{1}{2}$ fraction of the edges.

### 3.2 Statement of key lemmas

Our first lemma shows that $\mathcal{G}^Y$ is supported on instances of high value.

**Lemma 3.3** ($\mathcal{G}^Y$ has high $\textsf{Max-OCSP}(\Pi)$ values)**.** *For every ordering constraint satisfaction function $\Pi$, every $\boldsymbol{\pi} \in \textsf{supp}(\Pi)$ and $\Psi \sim \mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}$, we have $\textsf{val}_\Psi \geq 1 - \frac{k-1}{q}$ (i.e., this occurs with probability 1).*

We prove Lemma 3.3 in Section 4.2. Next we assert that $\mathcal{G}^N$ is supported mostly on instances of low value.

**Lemma 3.4** ($\mathcal{G}^N$ has low $\textsf{Max-OCSP}(\Pi)$ values)**.** *For every $k$-ary ordering constraint function $\Pi : \mathsf{S}_k \to \{0,1\}$, and every $\varepsilon > 0$, there exists $q_0 \in \mathbb{N}$ and $\alpha_0 \geq 0$ such that for all $q \geq q_0$ and $\alpha \leq \alpha_0$, there exists $T_0 \in \mathbb{N}$ such that for all $T \geq T_0$, for sufficiently large $n$, we have*

$$\Pr_{\Psi \sim \mathcal{G}^N_{q,n,\alpha,T}}\left[\textsf{val}_\Psi \geq \rho(\Pi) + \frac{\varepsilon}{2}\right] \leq 0.01.$$

We prove Lemma 3.4 in Section 4.3. We note that this lemma is more technically involved than Lemma 3.3 and this is the proof that needs the notion of "small partition expanders". Finally the following lemma asserts the indistinguishability of $\mathcal{G}^Y$ and $\mathcal{G}^N$ to small space streaming algorithms. We remark that this lemma follows directly from the work of [CGS+21].

**Lemma 3.5.** *For every $q, k \in \mathbb{N}$ there exists $\alpha_0(k) > 0$ such that for every $T \in \mathbb{N}$, $\alpha \in (0, \alpha_0(k)]$ the following holds: For every $\Pi : \mathsf{S}_k \to \{0,1\}$ and $\boldsymbol{\pi} \in \textsf{supp}(\Pi)$, every streaming algorithm* **ALG** *distinguishing $\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}$ from $\mathcal{G}^N_{q,n,\alpha,T}$ with advantage $1/8$ for all lengths $n$ uses space $\Omega(n)$.*

### 3.3 Proof of Theorem 1.1

We now prove Theorem 1.1.

*Proof of Theorem 1.1.* Let $A$ be a $\rho(\Pi) + \varepsilon$ approximation algorithm for $\textsf{Max-OCSP}(\Pi)$ that uses space $s$. Fix $\boldsymbol{\pi} \in \textsf{supp}(\Pi)$. Consider the algorithm **ALG** defined as follows: on input $\Psi$, an instance of $\textsf{Max-OCSP}(\Pi)$, if $A(\Psi) \geq \rho(\Pi) + \frac{\varepsilon}{2}$, then **ALG** outputs 1, else, it outputs 0. Observe that **ALG** uses $O(s)$ space. Set $q_0 \geq \frac{2(k-1)}{\varepsilon}$ such that the condition of Lemma 3.4 holds. Set $\alpha_0 \in (0, \alpha_0(k)]$ such that the conditions of Lemma 3.4 holds. Consider any $q \geq q_0$ and $\alpha \leq \alpha_0$: let $T_0$ be set as in Lemma 3.4. Consider any $T \geq T_0$: since $q \geq \frac{2(k-1)}{\varepsilon}$, it follows from Lemma 3.3 that for $\Psi \sim \mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}$, we have $\textsf{val}_\Psi \geq 1 - \frac{\varepsilon}{2}$, and hence with probability at least $2/3$, $A(\Psi) \geq \rho(\Pi) + \frac{\varepsilon}{2}$. Therefore, $\mathbb{E}_{\Psi \sim \mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}}[\mathbf{ALG}(\Psi) = 1] \geq 2/3$. Similarly, by the choice of $q_0, \alpha_0, T_0$, it follows from Lemma 3.4 that

$$\Pr_{\Psi \sim \mathcal{G}^N_{q,n,\alpha,T}}\left[\textsf{val}_\Psi \geq \rho(\Pi) + \frac{\varepsilon}{2}\right] \leq 0.01,$$

and hence, $\mathbb{E}_{\Psi \sim \mathcal{G}^N_{q,n,\alpha,T}}[\mathbf{ALG}(\Psi) = 1] \leq \frac{1}{3} + 0.01$. Therefore, **ALG** distinguishes $\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}$ from $\mathcal{G}^N_{q,n,\alpha,T}$ with advantage $1/8$. By applying Lemma 3.5, we conclude that the space complexity of $A$ is at least $\Omega(n)$. ☐

# 4 Bounds on $\textsf{Max-OCSP}(\Pi)$ values of $\mathcal{G}^Y$ and $\mathcal{G}^N$

The goal of this section is to prove our technical lemmas which lower bound the $\textsf{Max-OCSP}(\Pi)$ values of $\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}$ (Lemma 3.3) and upper bound the $\textsf{Max-OCSP}(\Pi)$ values of $\mathcal{G}^{N}_{q,n,\alpha,T}$ (Lemma 3.4).

## 4.1 CSPs and coarsening

In preparation for proving the lemmas, we recall the definition of (non-ordering) *constraint satisfaction problems (CSPs)*, whose solution spaces are $[q]^n$ (as opposed to $\mathsf{S}_n$), and define an operation called *$q$-coarsening* on $\textsf{Max-OCSP}$'s, which restricts the solution space from $\mathsf{S}_n$ to $[q]^n$.

A *maximum constraint satisfaction problem*, $\textsf{Max-CSP}(f)$, is specified by a single constraint function $f : [q]^k \to \{0,1\}$, for some positive integer $k$. An *instance* of $\textsf{Max-CSP}(f)$ on $n$ variables is given by $m$ constraints $C_0, \ldots, C_{m-1}$ where $C_i = (f, \mathbf{j}(i))$, i.e., the application of the function $f$ to the variables $\mathbf{j}(i) = (j(i)_0, \ldots, j(i)_{k-1})$. (Again, $f$ is omitted when clear from context.) The *value* of an assignment $\mathbf{b} \in [q]^n$ on an instance $\Phi = (C_0, \ldots, C_{m-1})$, denoted $\overline{\mathsf{val}}^q_\Phi(\mathbf{b})$, is the fraction of constraints satisfied by $\mathbf{b}$, i.e., $\overline{\mathsf{val}}^q_\Phi(\mathbf{b}) = \frac{1}{m}\sum_{i\in[m]} f(\mathbf{b}|_{\mathbf{j}(i)})$, where (recall) $\mathbf{b}|_{\mathbf{j}} = (b_{j_0}, \ldots, b_{j_{k-1}})$ for $\mathbf{b} = (b_0, \ldots, b_{n-1}), \mathbf{j} = (j_0, \ldots, j_{k-1})$. The optimal value of $\Phi$ is defined as $\overline{\mathsf{val}}^q_\Phi = \max_{\mathbf{b}\in[q]^n}\{\overline{\mathsf{val}}^q_\Phi(\mathbf{b})\}$.

**Definition 4.1** ($q$-coarsening)**.** *Let $\Pi$ be a $k$-ary $\textsf{Max-OCSP}$ and let $q \in \mathbb{N}$. The* $q$-coarsening *of $\Pi$ is the $k$-ary $\textsf{Max-CSP}$ problem $\textsf{Max-CSP}(f^q_\Pi)$ where we define $f^q_\Pi : [q]^k \to \{0,1\}$ as follows: For $\mathbf{a} \in [q]^k$, $f^q_\Pi(\mathbf{a}) = 1$ iff the entries in $\mathbf{a}$ are all distinct and $\Pi(\mathsf{ord}(\mathbf{a})) = 1$. The* $q$-coarsening of an instance *$\Psi$ of $\textsf{Max-OCSP}(\Pi)$ is the instance $\Phi$ of $\textsf{Max-CSP}(f^q_\Pi)$ given by the identical collection of constraints.*

The following lemma captures the idea that coarsening restricts the space of possible solutions; compare to Lemma 4.8 below.

**Lemma 4.2.** *If $q \in \mathbb{N}$, $\Psi$ is an instance of $\textsf{Max-OCSP}(\Pi)$, and $\Phi$ is the $q$-coarsening of $\Psi$, then $\mathsf{val}_\Psi \geq \overline{\mathsf{val}}^q_\Phi$.*

*Proof.* We will show that for every assignment $\mathbf{b} \in [q]^n$ to $\Phi$, we can construct an assignment $\boldsymbol{\sigma} \in \mathsf{S}_n$ to $\Psi$ such that $\mathsf{val}_\Psi(\boldsymbol{\sigma}) \geq \overline{\mathsf{val}}^q_\Phi(\mathbf{b})$. Consider an assignment $\mathbf{b} \in [q]^n$. Let $\boldsymbol{\sigma}$ be the ordering on $[n]$ given by placing the blocks $\mathbf{b}^{-1}(0), \ldots, \mathbf{b}^{-1}(q-1)$ in order (within each block, we enumerate the indices arbitrarily). Consider any constraint $C = \mathbf{j} = (j_0, \ldots, j_{k-1})$ in $\Phi$ which is satisfied by $\mathbf{b}$ in $\Phi$. Since $f^q_\Pi(\mathbf{b}|_{\mathbf{j}}) = 1$, by definition of $f^q_\Pi$ we have that $\Pi(\mathsf{ord}(\mathbf{b}|_{\mathbf{j}})) = 1$ and $b_{j_0}, \ldots, b_{j_{k-1}}$ are distinct. The latter implies, by construction of $\boldsymbol{\sigma}$, that $\mathsf{ord}(\mathbf{b}|_{\mathbf{j}}) = \mathsf{ord}(\boldsymbol{\sigma}|_{\mathbf{j}})$. Hence $\Pi(\mathsf{ord}(\boldsymbol{\sigma}|_{\mathbf{j}})) = 1$, so $\boldsymbol{\sigma}$ satisfies $C$ in $\Psi$. Hence $\mathsf{val}_\Psi(\boldsymbol{\sigma}) \geq \overline{\mathsf{val}}^q_\Phi(\mathbf{b})$. □

## 4.2 $\mathcal{G}^Y$ has high $\textsf{Max-OCSP}(\Pi)$ values

In this section, we prove Lemma 3.3, which states that the $\textsf{Max-OCSP}(\Pi)$ values of instances $\Psi$ drawn from $\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}$ are large. Note that we prove a bound for *every* instance $\Psi$ in the support of $\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}$, although it would suffice for our application to prove that such a bound holds with high probability over the choice of $\Psi$.

To prove Lemma 3.3, if $\Phi$ is the $q$-coarsening of $\Psi$, by Lemma 4.2, it suffices to show that $\overline{\mathsf{val}}^q_\Phi \geq 1 - \frac{k-1}{q}$. One natural approach is to consider the $q$-partition $\mathbf{b} = (b_0, \ldots, b_{n-1}) \in [q]^n$ sampled when sampling $\Psi$ and view $\mathbf{b}$ as an assignment to $\Phi$. Consider any constraint $C = \mathbf{j} = (j_0, \ldots, j_{k-1})$

in $\Psi$; by the definition of $\mathcal{G}^{Y,\boldsymbol{\pi}}$ (Definition 3.2), we have $\mathbf{b}|_{\mathbf{j}} = (\mathbf{v}_q^{(\ell)})_{\boldsymbol{\pi}}$ for some (unique) $\ell \in [q]$, which we term the *identifier* of $C$ (recall, we defined $\mathbf{v}_q^{(\ell)}$ as the $k$-tuple $(\ell, \ldots, \ell+k-1) \in [q]^k$). In other words, $\mathbf{b}|_{\mathbf{j}} = (\mathbf{v}_q^{(\ell)})_{\boldsymbol{\pi}}$. Hence, $C$ is satisfied by $\mathbf{b}$ iff $\Pi(\mathsf{ord}((\mathbf{v}_q^{(\ell)})_{\boldsymbol{\pi}})) = 1$. By Proposition 3.1 above, $\mathsf{ord}((\mathbf{v}_q^{(\ell)})_{\boldsymbol{\pi}}) = \mathsf{ord}(\mathbf{v}_q^{(\ell)}) \circ \boldsymbol{\pi}$. Hence a sufficient condition for $\mathbf{b}$ to satisfy $C$ (which is in fact necessary in the case $|\mathsf{supp}(\Pi)| = 1$) is that $\mathsf{ord}(\mathbf{v}_q^{(\ell)}) = [0 \ \cdots \ k-1]$ (since then $\mathsf{ord}((\mathbf{v}_q^{(\ell)})_{\boldsymbol{\pi}}) = \boldsymbol{\pi}$); this happens iff $C$'s identifier $\ell \in \{0, \ldots, q-k\}$. Unfortunately, when sampling the constraints $C$, we might get "unlucky" and get a sample which over-represents the constraints $C$ with identifier $\ell \in \{q-k+1, \ldots, q-1\}$. We can resolve this issue using "shifted" versions of $\mathbf{b}$.[4] The proof is as follows:

*Proof of Lemma 3.3.* For $t \in [q]$, define the assignment $\mathbf{b}^{(t)} = (b_0^{(t)}, \ldots, b_{n-1}^{(t)})$ to $\Phi$ via $b_i^{(t)} = b_i + t$ for $i \in [n]$.

Fix $t \in [q]$. Then we claim that $\mathbf{b}^{(t)}$ satisfies any constraint $C$ with identifier $\ell$ such that $\ell + t \in \{0, \ldots, q-k\}$. Indeed, if $C = \mathbf{j}$ is a constraint with identifier $\ell$, since $\mathbf{b}|_{\mathbf{j}} = (\mathbf{v}_q^{(\ell)})_{\boldsymbol{\pi}}$, then we have $\mathbf{b}^{(t)}|_{\mathbf{j}} = (\mathbf{v}_q^{(\ell+t)})_{\boldsymbol{\pi}}$; as long as $\ell + t \in \{0, \ldots, q-k\}$, then $\mathsf{ord}(\mathbf{v}_q^{(\ell+t)}) = [0 \ \cdots \ k-1]$, and so by Proposition 3.1, $\mathsf{ord}((\mathbf{v}_q^{(\ell+t)})_{\boldsymbol{\pi}}) = \boldsymbol{\pi}$. Thus, $\Pi(\mathsf{ord}((\mathbf{v}_q^{(\ell+t)})_{\boldsymbol{\pi}})) = 1$.

Now (no longer fixing $t$), for each $\ell \in [q]$, let $w^{(\ell)}$ be the fraction of constraints in $\Psi$ with identifier $\ell$. By the above claim, for each $t \in [q]$, we have $\overline{\mathsf{val}}_\Phi^q(\mathbf{b}^{(t)}) \geq \sum_{\ell : \ell+t \in \{0,\ldots,q-k\}} w^{(\ell)}$. On the other hand, $\sum_{\ell=0}^{q-1} w^{(\ell)} = 1$ (since every constraint has some (unique) identifier). Hence

$$\sum_{t=0}^{q-1} \mathsf{val}_\Phi(b^{(t)}) \geq \sum_{t=0}^{q-1} \left( \sum_{\ell : \ell+t \in \{0,\ldots,q-k\}} w^{(\ell)} \right) = q - (k-1),$$

since each term $w^{(\ell)}$ appears exactly $q-(k-1)$ times in the expanded sum. Hence by averaging, there exists some $t \in [q]$ such that $\overline{\mathsf{val}}_\Phi^q(\mathbf{b}^{(t)}) \geq 1 - \frac{k-1}{q}$, and so $\overline{\mathsf{val}}_\Phi^q \geq 1 - \frac{k-1}{q}$, as desired. □

### 4.3 $\mathcal{G}^N$ has low $\mathsf{Max\text{-}OCSP}(\Pi)$ values

In this section, we prove Lemma 3.4, which states that the $\mathsf{Max\text{-}OCSP}(\Pi)$ value of an instance drawn from $\mathcal{G}^N$ does not significantly exceed the random ordering threshold $\rho(\Pi)$, with high probability.

Using concentration bounds (i.e., Lemma 2.1), one could show that a fixed solution $\boldsymbol{\sigma} \in \mathsf{S}_n$ satisfies more than $\rho(\Pi) + \frac{1}{q}$ constraints with probability which is exponentially small in $n$. However, taking a union bound over all $n!$ permutations $\boldsymbol{\sigma}$ would cause an unacceptable blowup in the probability. Instead, to prove Lemma 3.4, we take an indirect approach, involving bounding the $\mathsf{Max\text{-}CSP}$ value of the $q$-coarsening of a random instance and bounding the gap between the $\mathsf{Max\text{-}OCSP}$ value and the $q$-coarsenened $\mathsf{Max\text{-}CSP}$ value. To do this, we define the following notions of small set expansion for $k$-hypergraphs:

**Definition 4.3** (Lying on a set)**.** *Let $G = (V, E)$ be a $k$-hypergraph. Given a set $S \subseteq V$, a $k$-hyperedge $\mathbf{e} \in E$* lies on *$S$ if it is incident on two (distinct) vertices in $S$ (i.e., if $|\Gamma(\mathbf{e}) \cap S| \geq 2$).*

**Definition 4.4** (Congregating on a partition)**.** *Let $G = (V, E)$ be a $k$-hypergraph. Given a $q$-partition $\mathbf{b} \in [q]^n$, a $k$-hyperedge $\mathbf{e} \in E$* congregates *on $\mathbf{b}$ if it lies on one of the blocks $\mathbf{b}^{-1}(i)$.*

[4] Alternatively, in expectation, $\overline{\mathsf{val}}_\Phi^q(\mathbf{b}) = 1 - \frac{k-1}{q}$. Hence with probability at least $\frac{99}{100}$, $\overline{\mathsf{val}}_\Phi^q(\mathbf{b}) \geq 1 - \frac{100(k-1)}{q}$ by Markov's inequality; this suffices for a "with-high-probability" statement.

We denote by $N(G,S)$ the number of $k$-hyperedges of $G$ which lie on $S$.

**Definition 4.5** (Small set hypergraph expansion (SSHE) property)**.** *A $k$-hypergraph $G=(V,E)$ is a $(\gamma,\delta)$-*small set hypergraph expander (SSHE) *if it has the following property: For every subset $S \subseteq V$ of size at most $\gamma|V|$, $N(G,S) \le \delta|E|$ (i.e., the number of $k$-hyperedges in $E$ which lie on $S$ is at most $\delta|E|$).*

**Definition 4.6** (Small partition hypergraph expansion (SPHE) property)**.** *A $k$-hypergraph $G=(V,E)$ is a $(\gamma,\delta)$-*small partition hypergraph expander (SPHE) *if it has the following property: For every partition $\mathbf{b} \in [q]^n$ where each block $\mathbf{b}^{-1}(i)$ has size at most $\gamma|V|$, the number of $k$-hyperedges in $E$ which congregate on $\mathbf{b}$ is at most $\delta|E|$.*

In the context of Figure 1, the SPHE property says that for *any* partition with small blocks, there cannot be too many "orange" edges.

Having defined the SSHE and SPHE properties, we now sketch the proof of Lemma 3.4. It will be proved formally later in this section.

*Proof sketch of Lemma 3.4.* For sufficiently large $q$, with high probability, the $\mathsf{Max\text{-}CSP}$ value of the $q$-coarsening of a random $\mathsf{Max\text{-}OCSP}(\Pi)$ instance drawn from $\mathcal{G}_q^N$ is not much larger than $\rho(\Pi)$ (Lemma 4.13 below). The constraint hypergraph for a random $\mathsf{Max\text{-}OCSP}(\Pi)$ instance drawn from $\mathcal{G}_q^N$ is a good SSHE with high probability (Lemma 4.11 below). Hypergraphs which are good SSHEs are also (slightly worse) SPHEs (Lemma 4.7 below). Finally, if the constraint hypergraph of a $\mathsf{Max\text{-}OCSP}(\Pi)$ instance is a good SPHE, its $\mathsf{Max\text{-}OCSP}(\Pi)$ value cannot be much larger than its $q$-coarsened $\mathsf{Max\text{-}CSP}$ value (Lemma 4.8 below); intuitively, this is because if we "coarsen" an optimal ordering $\boldsymbol{\sigma}$ for the $\mathsf{Max\text{-}OCSP}$ by lumping vertices together in small groups to get an assignment $\mathbf{b}$ for the coarsened $\mathsf{Max\text{-}CSP}$, we can view this assignment $\mathbf{b}$ as a partition on $V$, and for every $k$-hyperedge in $G(\Psi)$ which does not congregate on this partition, the corresponding constraint in $\Psi$ is satisfied. $\square$

We remark that the bounds on $\mathsf{Max\text{-}CSP}$ values of coarsened random instances (Lemma 4.13 below) and on SSHE in random instances (Lemma 4.11 below) both use concentration inequalities (i.e., Lemma 2.1) and union bound over a space of size only $(O_\varepsilon(1))^n$ (the space of all solutions to the coarsened $\mathsf{Max\text{-}CSP}$ and the space of all small subsets of $[n]$, respectively); this lets us avoid the issue of union-bounding over the entire space $\mathsf{S}_n$ directly.

In the remainder of this section, we prove the necessary lemmas and then give a formal proof of Lemma 3.4. We begin with several short lemmas.

**Lemma 4.7** (Good SSHEs are good SPHEs)**.** *For every $\gamma,\delta > 0$, if a $k$-hypergraph $G=(V,E)$ a $(\gamma,\delta)$-SSHE, then it is a $\left(\gamma, \delta(\frac{2}{\gamma}+1)\right)$-SPHE.*

*Proof.* Let $n=|V|$. Consider any partition $\mathbf{b} \in [q]^n$ of $V$ where each block has size at most $\gamma n$. WLOG, all but one block $\mathbf{b}^{-1}(i)$ has size at least $\frac{\gamma n}{2}$ (if not, merge blocks until this happens, only increasing the number of $k$-hyperedges which congregate on $\mathbf{b}$). Hence $\ell \le \frac{2}{\gamma}+1$.[5] By the SSHE property, there are at most $\delta m$ $k$-hyperedges which lie on each block; hence there are at most $\delta(\frac{2}{\gamma}+1)m$ constraints which congregate on $\mathbf{b}$. $\square$

[5] We include the $+1$ to account for the extra block which may have arbitrarily small size. Excluding this block, there are at most $\frac{n}{\lceil \gamma n/2 \rceil} \le \frac{n}{\gamma n/2}$ blocks remaining.

**Lemma 4.8** (Coarsening roughly preserves value in SPHEs)**.** *Let $\Psi$ be a Max-OCSP$(\Pi)$ instance on $n$ variables. Suppose that the constraint hypergraph of $\Psi$ is a $(\gamma, \delta)$-SPHE. Let $\Phi$ be the $q$-coarsening of $\Psi$. Then for sufficiently large $n$, if $q \geq \frac{2}{\gamma}$,*

$$\mathsf{val}_\Psi \leq \overline{\mathsf{val}}^q_\Phi + \delta.$$

*Proof.* We will show that for every assignment $\boldsymbol{\sigma} \in \mathsf{S}_n$ to $\Psi$, we can construct an assignment $\mathbf{b} = (b_0, \ldots, b_{n-1}) \in [q]^n$ to $\Phi$ such that $\mathsf{val}_\Psi(\boldsymbol{\sigma}) \leq \overline{\mathsf{val}}^q_\Phi(\mathbf{b}) + \delta$. Fix $\boldsymbol{\sigma} \in \mathsf{S}_n$. Define $\mathbf{b} \in [q]^n$ by $b_i = \lfloor \boldsymbol{\sigma}(i)/\lfloor \gamma n \rfloor \rfloor$ for each $i \in [n]$. Observe that since $\boldsymbol{\sigma}(i) \leq n-1$, we have $b_i \leq \lfloor (n-1)/\lfloor \gamma n \rfloor \rfloor < q$, hence $\mathbf{b}$ is a valid assignment to $\Phi$. Also, $\mathbf{b}$ has the property that for every $i, j \in [n]$, if $\boldsymbol{\sigma}(i) < \boldsymbol{\sigma}(j)$ then $b_i \leq b_j$; we call this *monotonicity* of $\mathbf{b}$.

View $\mathbf{b}$ as a $q$-partition and consider the constraint hypergraph of $\Psi$ (which is the same as the constraint hypergraph of $\Phi$). Call a constraint $C = (j_0, \ldots, j_{k-1})$ *good* if it is both satisfied by $\boldsymbol{\sigma}$, *and* the $k$-hyperedge corresponding to it does not congregate on $\mathbf{b}$. If $C$ is good, then $b_{j_0}, \ldots, b_{j_{k-1}}$ are all distinct; together with monotonicity of $\mathbf{b}$, we conclude that if $C$ is good, then $\mathsf{ord}(\mathbf{b}|_\mathbf{j}) = \mathsf{ord}(\boldsymbol{\sigma}(j_0), \ldots, \boldsymbol{\sigma}(j_{k-1}))$.

Finally, we note that each block in $\mathbf{b}$ has size at most $\gamma n$ by definition; hence by the SPHE property of the constraint hypergraph of $\Psi$, at most $\delta$-fraction of the constraints of $\Psi$ correspond to $k$-hyperedges which congregate on $\mathbf{b}$. Since $\mathsf{val}_\Psi(\boldsymbol{\sigma})$ fraction of the constraints of $\Psi$ are satisfied by $\boldsymbol{\sigma}$, at least $(\mathsf{val}_\Psi(\boldsymbol{\sigma}) - \delta)$-fraction of the constraints of $\Psi$ are good, and hence $\mathbf{b}$ satisfies at least $(\mathsf{val}_\Psi(\boldsymbol{\sigma}) - \delta)$-fraction of the constraints of $\Phi$, as desired. □

The construction in this lemma was called *coarsening* the assignment $\boldsymbol{\sigma}$ by [GHM+11] (cf. [GHM+11, Definition 4.1]).

We also include the following helpful lemma, which lets us restrict to the case where our sampled Max-OCSP$(\Pi)$ instance has many constraints.

**Lemma 4.9** (Most instances in $\mathcal{G}^N$ have many constraints)**.** *For every $n$, $\alpha, \gamma > 0$, and $q \in \mathbb{N}$,*

$$\Pr_{\Psi \sim \mathcal{G}^N_{q,n,\alpha,T}} \left[ m(\Psi) \leq \frac{n\alpha T}{2q^k} \right] \leq \exp\left( -\frac{n\alpha T}{8q^k} \right).$$

*Proof.* The number of constraints in $\Psi$ is distributed as the sum of $n\alpha T$ independent Bernoulli$(1/q^k)$ random variables. The desired bound follows by applying the Chernoff bound. □

#### 4.3.1 $\mathcal{G}^N$ is a good SSHE with high probability

Recall that for a $k$-hypergraph $G = (V, E)$ and $S \subseteq V(G)$, we define $N(G, S)$ to be the number of $k$-hyperedges in $G$ that lie on $S$, and for an $k$-hyperedge $\mathbf{e} \in E$, we define $\Gamma(\mathbf{e}) \subseteq V$ as the set of vertices incident on $\mathbf{e}$.

**Lemma 4.10** (Random hypermatchings barely lie on small sets)**.** *For every $n$ and $\alpha, \gamma > 0$ with $\alpha \leq \frac{1}{2k}$, and every subset $S \subseteq [n]$ of at most $\gamma n$ vertices, we have*

$$\Pr_{G \sim \mathcal{H}_{k,n,\alpha}} [N(G, S) \geq 8k^2\gamma^2\alpha n] \leq \exp\left(-\gamma^2 \alpha n\right).$$

*Proof.* Label the hyperedges of $G$ as $\mathbf{e}_0, \ldots, \mathbf{e}_{\alpha n-1}$. For $i \in [\alpha n]$, let $X_i$ be the indicator for the event that $\mathbf{e}_i$ lies on $S$. We have $N(G,S) = X_0 + \cdots + X_{\alpha n-1}$.

We first bound $\mathbb{E}[X_i \mid X_0, \ldots, X_{i-1}]$ for each $i$. Conditioned on $\mathbf{e}_0, \ldots, \mathbf{e}_{i-1}$, the $k$-hyperedge $\mathbf{e}_i$ is uniformly distributed over the set of all $k$-hyperedges on $[n] \setminus (\Gamma(\mathbf{e}_0) \cup \cdots \cup \Gamma(\mathbf{e}_{i-1}))$. It suffices to union-bound, over distinct pairs $j_1 < j_2 \in \binom{[k]}{2}$, the probability that the $j_1$-st and $j_2$-nd vertices of $\mathbf{e}_i$ are in $S$ (conditioned on $X_0, \ldots, X_{i-1}$). We can sample the $j_1$-st and $j_2$-nd vertices of $\mathbf{e}_i$ first (uniformly over remaining vertices outside of $S$) and then sample the remaining vertices (uniformly over remaining vertices). Hence we have the upper-bound

$$\begin{aligned}\mathbb{E}[X_i \mid X_0, \ldots, X_{i-1}] &\le \binom{k}{2} \cdot \frac{|S|(|S|-1)}{(n-ki)(n-ki-1)} \\ &\le \binom{k}{2} \cdot \left(\frac{|S|}{n-ki}\right)^2 \\ &\le \binom{k}{2} \cdot \left(\frac{|S|}{n-k\alpha n}\right)^2 \le 4k^2\gamma^2\,,\end{aligned}$$

since $\alpha \le \frac{1}{2k}$.

Now, we apply the concentration bound in Lemma 2.1 to conclude that:

$$\Pr_{G\sim\mathcal{H}_{k,n,\alpha}}\left[X_0 + \cdots + X_{\alpha n-1} \ge 8k^2\gamma^2\alpha n\right] \le \exp\left(-2k^2\gamma^2\alpha n\right) \le \exp(-\gamma^2\alpha n).$$

□

**Lemma 4.11.** *For every $n$, $\alpha, \gamma > 0$, and $q \in \mathbb{N}$ with $\alpha \le \frac{1}{2k}$,*

$$\Pr_{\Psi\sim\mathcal{G}^N_{q,n,\alpha,T}}\left[G(\Psi) \text{ is not a } (\gamma, 8k^2\gamma^2)\text{-SSHE} \,\middle|\, m(\Psi) \ge \frac{n\alpha T}{2q^k}\right] \le \exp\left(-\left(\frac{\gamma^2\alpha T}{2q^k} - \ln 2\right)n\right).$$

*Proof.* Let $\alpha_0, \ldots, \alpha_{T-1} \ge 0$ be such that $\frac{\alpha T}{2q^k} \le \alpha_0 + \cdots + \alpha_{T-1} \le \alpha T$. It suffices to prove the bound, for every such sequence $\alpha_0, \ldots, \alpha_{T-1}$, conditioned on the event that for every $i \in [T]$, $m(G_i) = \alpha_i n$ (where $G_i$ is defined as in Definition 3.2). This is equivalent to simply sampling each $G_i \sim \mathcal{H}_{k,n,\alpha_i}$ independently.

Fix any set $S \subseteq [n]$ of size at most $\gamma n$. Applying Lemma 4.10, and the fact that each hypermatching $G_i$ in $G$ is sampled independently, we conclude that

$$\begin{aligned}&\Pr_{\Psi\sim\mathcal{G}^N_{q,n,\alpha,T}}\left[\exists i \in [T] \text{ s.t. } N(G_i,S) \ge 8k^2\gamma^2\alpha_i n \,\middle|\, \forall i \in [T], m(G_i) = \alpha_i n\right] \\ &\le \exp\left(-\gamma^2(\alpha_0 + \cdots + \alpha_{T-1})n\right) \\ &\le \exp\left(-\frac{\gamma^2\alpha T n}{2q^k}\right).\end{aligned}$$

Hence by averaging, the total fraction of $k$-hyperedges in $G$ which lie on $S$ is at most $8k^2\gamma^2$. Taking the union-bound over the $\le 2^n$ possible subsets $S \subseteq [n]$ gives the desired bound. □

### 4.3.2 $\mathcal{G}^N$ has low coarsened $\mathsf{Max\text{-}CSP}(f_\Pi^q)$ values with high probability

For $G \sim \mathcal{H}_{k,n,\alpha}$, we define an instance $\Phi(G)$ of $\mathsf{Max\text{-}CSP}(f_\Pi^q)$ on $n$ variables $x_0, \ldots, x_{n-1}$ naturally as follows: for each $k$-hyperedge $\mathbf{j} = (j_0, \ldots, j_{k-1}) \in E(G) \subseteq [n]^k$, we add the constraint $\mathbf{j}$ to $\Phi(G)$.

**Lemma 4.12** (Satisfiability of random instances of $\mathsf{Max\text{-}CSP}(f_\Pi^q)$)**.** *For every $n$, $\alpha, \eta > 0$, and $\mathbf{b} \in [q]^n$,*

$$\Pr_{G \sim \mathcal{H}_{k,n,\alpha}}[\overline{\mathsf{val}}^q_{\Phi(G)}(\mathbf{b}) \geq \rho(\Pi) + \eta] \leq \exp\left(-\left(\frac{\eta^2\alpha}{2(\rho(\Pi)+\eta)}\right)n\right).$$

*Proof.* Let the $k$-hyperedges of $G$ be labelled as $\mathbf{e}_0, \ldots, \mathbf{e}_{\alpha n-1}$ and the corresponding constraints of $\Phi(G)$ be denoted by $\mathbf{j}(0), \ldots, \mathbf{j}(\alpha n - 1)$. For $i \in [\alpha n]$, let $X_i$ be the indicator for the event that the constraint $\mathbf{j}(i)$ is satisfied by $\mathbf{b}$, i.e., $f_\Pi^q(\mathbf{b}|_{\mathbf{j}(i)}) = 1$. Again, like in the proof of Lemma 4.10, we bound $\mathbb{E}[X_i \mid X_0, \ldots, X_{i-1}]$, for each $i$. Conditioned on $\mathbf{e}_0, \ldots, \mathbf{e}_{i-1}$, the $k$-hyperedge $\mathbf{e}_i$ is uniformly distributed over the set of all $k$-hyperedges on $[n] \setminus (\Gamma(\mathbf{e}_0) \cup \cdots \cup \Gamma(\mathbf{e}_{i-1}))$. Hence, $\mathbb{E}[X_i \mid X_0, \ldots, X_{i-1}] \leq \rho(\Pi)$. Indeed, the set of possible $k$-hyperedges on $[n] \setminus (\Gamma(\mathbf{e}_0) \cup \cdots \cup \Gamma(\mathbf{e}_{i-1}))$ may be partitioned into blocks of size $k!$ by mapping each $k$-hyperedge to the set of vertices on which it is incident. For each subset $J = \{j_0, \ldots, j_{k-1}\} \subseteq [n]$, if $b_{j_0}, \ldots, b_{j_{k-1}}$ are not all distinct, then for every $\boldsymbol{\pi} \in \mathsf{S}_k$, the constraint corresponding to the permuted $k$-tuple $\mathbf{j}_{\boldsymbol{\pi}}$ is not satisfied by $\mathbf{b}$. On the other hand, if $b_{j_0}, \ldots, b_{j_{k-1}}$ are all distinct, then

$$\left|\{\boldsymbol{\pi} \in \mathsf{S}_k : f_\Pi^q(\mathbf{b}|_{\mathbf{j}_{\boldsymbol{\pi}}}) = 1\}\right| = |\mathsf{supp}(\Pi)| = \rho(\Pi) \cdot k!\,.$$

Finally, we again apply the concentration bound in Lemma 2.1 to conclude that:

$$\Pr_{G \sim \mathcal{H}_{k,n,\alpha}}[X_0 + \cdots + X_{\alpha n-1} \geq (\rho(\Pi)+\eta)\alpha n] \leq \exp\left(-\left(\frac{\eta^2\alpha}{2(\rho(\Pi)+\eta)}\right)n\right),$$

as desired. □

**Lemma 4.13.** *For every $n$ and $\alpha, \eta > 0$,*

$$\Pr_{\Psi \sim \mathcal{G}^N_{q,n,\alpha,T}}\left[\overline{\mathsf{val}}^q_\Phi \geq \rho(\Pi) + \eta, \text{ where } \Phi \text{ is the } q\text{-coarsening of } \Psi \;\middle|\; m(\Psi) \geq \frac{n\alpha T}{2q^k}\right]$$
$$\leq \exp\left(-\left(\frac{\eta^2\alpha T}{4(\rho(\Pi)+\eta)q^k} - \ln q\right)n\right).$$

*Proof.* Identical to the proof of Lemma 4.11 (using Lemma 4.12 instead of Lemma 4.10), but now union-bounding over a set of size $q^n$ (i.e., the set of possible assignments $\mathbf{b} \in [q]^n$ for $\Phi$). □

We finally give the proof of Lemma 3.4.

*Proof of Lemma 3.4.* Let $q_0 := \left\lceil \frac{192k^2}{\varepsilon} \right\rceil$ and let $\alpha_0 := \frac{1}{2k}$. Suppose $\alpha \leq \alpha_0$ and $q \geq q_0$. Then let $\gamma := \frac{\varepsilon}{96k^2}$ and $\eta := \frac{\varepsilon}{4}$, and let

$$T_0 := \max\left\{\frac{4(\ln 2)q^k}{\gamma^2\alpha}, \frac{8(\rho(\Pi)+\eta)q^k(\ln q)}{\eta^2\alpha}\right\}.$$

Consider any $T \geq T_0$; we will prove the desired bound. Let $\delta := 8k^2\gamma^2$. Then the multiplicative factors in the exponents of the error terms in Lemma 4.9, Lemma 4.11, and Lemma 4.13 are all positive (the latter two lemmas may be applied since $\alpha \leq \alpha_0 = \frac{1}{2k}$); taking a union bound (and then conditioning on $m(\Psi) \geq \frac{n\alpha T}{2q^k}$), for sufficiently large $n$, we can conclude that with probability at least 0.99 over $\Psi \sim \mathcal{G}^N_{q,n,\alpha,T}$, we have $\overline{\mathsf{val}}^q_\Phi \geq \rho(\Pi) + \eta$ (where $\Phi$ is the $q$-coarsening of $\Psi$) and $G(\Psi)$ is a $(\gamma, \delta)$-SSHE. If $G(\Psi)$ is a $(\gamma, \delta)$-SSHE, by Lemma 4.7 it is also a $(\gamma, \delta')$-SPHE, where $\delta' := \frac{3\delta}{\gamma} \geq \delta(\frac{2}{\gamma} + 1)$. Note that $\delta' = 24k^2\gamma = \frac{\varepsilon}{4}$. Now since $q \geq q_0 \geq \frac{2}{\gamma}$, we can apply Lemma 4.8, and conclude that for sufficiently large $n$, with probability $\geq 0.99$ over the choice of $\Psi \sim \mathcal{G}^N_{q,n,\alpha,T}$, we have

$$\mathsf{val}_\Psi \geq \rho(\Pi) + \eta + \delta' = \rho(\Pi) + \frac{\varepsilon}{2},$$

as desired. □

# 5 Streaming indistinguishability of $\mathcal{G}^Y$ and $\mathcal{G}^N$

In this section we prove Lemma 3.5. This indistinguishability follows directly from the work of [CGS+21], who introduce a $T$-player communication problem called *implicit randomized mask detection (IRMD)*. Once we properly situate our instances $\mathcal{G}^Y$ and $\mathcal{G}^N$ within the framework of [CGS+21], Lemma 3.5 follows immediately.

We first recall their definition of the IRMD problem, and state their lower bound. The following definition is based on [CGS+21, Definition 3.1]. In [CGS+21] the IRMD game is parametrized by two distributions $\mathcal{D}_Y$ and $\mathcal{D}_N$, but hardness is proved for a specific pair of distributions which suffices for our purpose; these distributions will thus be "hardcoded" into the definition we give.

**Definition 5.1** (Implicit randomized mask detection (IRMD) problem)**.** *Let $q, k, n, T \in \mathbb{N}, \alpha \in (0, 1/k)$ be parameters. In the $\mathsf{IRMD}_{\alpha,T}$ game, there are $T$ players, indexed from 0 to $T-1$, and a hidden partition encoded by a random $\mathbf{b} \in [q]^n$. The $t$-th player has two inputs: (a.) $M_t \in \{0,1\}^{\alpha kn \times n}$, the hypermatching matrix corresponding to a uniform $\alpha$-partial $k$-hypermatching on $n$ vertices (i.e., drawn from $\mathcal{H}_{n,\alpha}$), and (b.) a vector $\mathbf{z}_t \in [q]^{\alpha kn}$ that can be generated from one of two different distributions:*

- ***(YES)*** $\mathbf{z}_t = M_t\mathbf{b} + \mathbf{y}_t \pmod q$ *where* $\mathbf{y}_t \in [q]^{\alpha kn}$ *is of the form* $\mathbf{y}_t = (\mathbf{y}_{t,0}, \ldots, \mathbf{y}_{t,\alpha n-1})$ *and each* $\mathbf{y}_{t,i} \in [q]^k$ *is sampled as* $(a, \ldots, a)$ *where* $a$ *is sampled uniformly from* $[q]$.
- ***(NO)*** $\mathbf{z}_t = M_t\mathbf{b} + \mathbf{y}_t \pmod q$ *where* $\mathbf{y}_t \in [q]^{\alpha kn}$ *is of the form* $\mathbf{y}_t = (\mathbf{y}_{t,0}, \ldots, \mathbf{y}_{t,\alpha n-1})$ *and each* $\mathbf{y}_{t,i} \in [q]^k$ *is sampled as* $(a_0, \ldots, a_{k-1})$ *where each* $a_j$ *is sampled uniformly and independently from* $[q]$.

*This is a one-way game where the $t$-th player can send a private message to the $(t+1)$-st player after receiving a message from the previous player. The goal is for the $(T-1)$-st player to decide whether the $\{\mathbf{z}_t\}$ have been chosen from the **YES** or **NO** distribution, and the advantage of a protocol is defined as*

$$\left| \Pr_{\textbf{\textit{YES}}\ \textit{case}}[\textit{the } (T-1)\textit{-st player outputs 1}] - \Pr_{\textbf{\textit{NO}}\ \textit{case}}[\textit{the } (T-1)\textit{-st player outputs 1}] \right|.$$

Note that the definition of the IRMD problem does not depend on an underlying family of constraints. Nevertheless, we will be able to leverage its hardness to prove Lemma 3.5 (and indeed, all hardness results in [CGS+21] itself stem from hardness for the IRMD problem). The following theorem from [CGS+21] gives a lower bound on the communication complexity of the IRMD problem:

**Theorem 5.2** ([CGS+21, Theorem 3.2])**.** *For every $q, k \in \mathbb{N}$ and $\delta \in (0, 1/2)$, $\alpha \in (0, 1/k)$, $T \in \mathbb{N}$ there exists $n_0 \in \mathbb{N}$ and $\tau \in (0, 1)$ such that the following holds. For all $n \geq n_0$, every protocol for* $\mathsf{IRMD}_{\alpha,T}$ *on $n$ vertices with advantage $\delta$ requires $\tau n$ bits of communication.*

Now, we use this hardness result to prove Lemma 3.5. The following proof is based on the proof of [CGS+21, Theorem 4.3], which introduces a notion called the *width* of a constraint family, which we briefly discuss. For our purposes, it suffices to define the width $\omega(f) \in [0, 1]$ of a single constraint $f : [q]^k \to \{0, 1\}$ as

$$\omega(f) = \max_{\mathbf{b} \in [q]^k} \left\{ \Pr_{\ell \in [q]} [f(\mathbf{b} + \ell) = 1] \right\},$$

where $\mathbf{b} + \ell$ denotes adding $\ell$ to each component of $\mathbf{b}$. [CGS+21, Theorem 4.3] states that for every $f$ and $\varepsilon > 0$, $\mathsf{Max\text{-}CSP}(f)$ cannot be $(\rho(f)/\omega(f) + \varepsilon)$-approximated by a sublinear-space single-pass streaming algorithm, where $\rho(f) = \Pr_{\mathbf{b} \in [q]^k}[f(\mathbf{b}) = 1]$ is the random assignment value for $f$. In other words, whenever $\omega(f)$ is close to 1, $\mathsf{Max\text{-}CSP}(f)$ is difficult to approximate. In our setting, we have $\omega(f^q_\Pi) \geq 1 - \frac{k-1}{q}$; indeed, simply take $\mathbf{b} = (\boldsymbol{\pi}^{-1}(0), \ldots, \boldsymbol{\pi}^{-1}(k-1))$, and then for any $\ell \in \{0, \ldots, q-k\}$, we have $f^q_\Pi(\mathbf{b} + \ell) = 1$ (by the same reasoning as in Section 4.2). The fact that $\omega(f^q_\Pi) \approx 1$ for large $q$ is precisely what enables us to apply [CGS+21]'s lower bounds to get optimal lower bounds in our setting. However, [CGS+21, Theorem 4.3] as written contains both the streaming-to-communication reduction and an analysis of the CSP values of **YES** and **NO** instances; in the following, we reprove only the former (and adapt the language to our setting).

*Proof of Lemma 3.5.* We prove the lemma for the same $\alpha_0$ as in Theorem 5.2.

Suppose **ALG** is a $O(n)$-space streaming algorithm which distinguishes $\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}$ from $\mathcal{G}^{N}_{q,n,\alpha,T}$ with advantage $1/8$ for all lengths $n$. We now show how to use **ALG** to construct a protocol **ALG′** solving $\mathsf{IRMD}_{\alpha,T}$ with advantage $1/8$ for $n \geq n_0$, which uses only $O(n)$ bits of communication; this contradicts Theorem 5.2. As is standard, this reduction will involve the players collectively running the streaming algorithm **ALG**. That is, **ALG′** is defined as follows: For $t = 0, \ldots, T-1$, the $t$-th player $P_t$ will add some constraints to the stream and then send the state of **ALG** on to the next player. Finally, the last player $P_{T-1}$ terminates the streaming algorithm and outputs the output of **ALG**.

Which constraints does $P_t$ add to the stream in **ALG′**? $P_t$'s input is $(M_t, \mathbf{z}_t)$, with $\mathbf{z}_t = (\mathbf{z}_{t,0}, \ldots, \mathbf{z}_{t,\alpha n-1})$, and each $\mathbf{z}_{t,i} \in [q]^k$. Above, we defined $\mathbf{v}^{(\ell)}_q = (\ell, \ldots, \ell + k - 1 \pmod q) \in [q]^k$, so that $(\mathbf{v}^{(0)}_q)_{\boldsymbol{\pi}} = (\boldsymbol{\pi}^{-1}(0), \ldots, \boldsymbol{\pi}^{-1}(k-1))$ (see Section 3.1). $P_t$ simply examines each $i \in [\alpha n]$ and the corresponding hyperedge $\widetilde{\mathbf{e}}_i$ in $M_t$. If $\mathbf{z}_{t,i} = (\mathbf{v}^{(0)}_q)_{\boldsymbol{\pi}}$, $P_t$ adds the constraint corresponding to $\widetilde{\mathbf{e}}_i$ to the stream.

Let $\mathcal{S}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}$ and $\mathcal{S}^{N}_{q,n,\alpha,T}$ denote the distributions of $\mathsf{Max\text{-}OCSP}(\Pi)$ instances constructed by **ALG′** in the **YES** and **NO** cases, respectively. The crucial claim is that $\mathcal{S}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}$ and $\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}$ are identical distributions, and similarly with $\mathcal{S}^{N}_{q,n,\alpha,T}$ and $\mathcal{G}^{N}_{q,n,\alpha,T}$. This claim suffices to prove the lemma, since the constructed stream of constraints is fed into **ALG**, which is an $O(n)$-space streaming algorithm distinguishing $\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}$ from $\mathcal{G}^{N}_{q,n,\alpha,T}$; hence we can conclude that **ALG′** is a protocol for $\mathsf{IRMD}$ using $O(n)$ bits of communication.

It remains to prove the claim. We first consider the **NO** case. $\mathcal{S}^{N}_{q,n,\alpha,T}$ and $\mathcal{G}^{N}_{q,n,\alpha,T}$ are both sampled by independently sampling $T$ hypermatchings from $\mathcal{H}_{n,\alpha}$ and then (independently) selecting some subset of hyperedges from each hypermatching to add as constraints. It suffices by independence to prove equivalence of how the subset of each hypermatching is sampled in each case. In the $t$-th hypermatching in $\mathcal{S}^N_{q,n,\alpha,T}$, $P_t$ adds a hyperedge $\widetilde{\mathbf{e}}_i$ iff $\mathbf{z}_{t,i} = (\mathbf{v}_q^{(0)})_{\boldsymbol{\pi}}$. But (even conditioned on all other $\mathbf{z}_{t,i'}$'s and on $\widetilde{\mathbf{e}}_i$ itself), $\mathbf{z}_{t,i}$ is a uniform value in $[q]^k$, and hence $\widetilde{\mathbf{e}}_i$ is added to the instance with probability $\frac{1}{q}^k$ (independently of every other hyperedge). This is exactly how we defined $\mathcal{G}^N_{q,n,\alpha,T}$ to sample constraints.

Similarly, we consider the **YES** case, analyzing the $t$-th hypermatching in $\mathcal{S}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}$. Consider the sampled $q$-partition $\mathbf{b} = (b_0, \ldots, b_{n-1}) \in [q]^n$. Again consider a hyperedge $\widetilde{\mathbf{e}}_i = (j_0, \ldots, j_{k-1})$. In this case, $\mathbf{z}_{t,i}$ is a *uniform translation* of $\mathbf{b}|_{\mathbf{j}}$, i.e., it equals $\mathbf{b}|_j + \ell'$ where $\ell' \in [q]$ is uniform and the sum denotes adding $\ell'$ to each component of $\mathbf{b}|_j$. Hence $P_t$ will add $\widetilde{\mathbf{e}}_i$ iff (1) $\mathbf{b}|_{\mathbf{j}} = (\mathbf{v}_q^{(\ell)})_{\boldsymbol{\pi}}$ for some $\ell \in [q]$ *and* (2) $\ell + \ell' = 0 \pmod q$. The latter event occurs with probability $\frac{1}{q}$, even conditioned on all other $\mathbf{z}_{t,i'}$'s and on $\widetilde{\mathbf{e}}_i$. Hence $\widetilde{\mathbf{e}}_i$ is added to the instance with probability $\frac{1}{q}$, as long as $\mathbf{b}|_{\mathbf{j}} = (\mathbf{v}_q^{(\ell)})_{\boldsymbol{\pi}}$ for some $\ell \in [q]$. This is exactly how we defined $\mathcal{G}^Y_{q,n,\alpha,T}$ to sample constraints. $\square$

# References


[AMW15] Per Austrin, Rajsekar Manokaran, and Cenny Wenner. On the NP-hardness of approximating ordering-constraint satisfaction problems. *Theory of Computing*, 11:257–283, 2015. Conference version in APPROX 2013.

[BK19] Amey Bhangale and Subhash Khot. UG-Hardness to NP-Hardness by Losing Half. In *34th Computational Complexity Conference (CCC 2019, New Brunswick, New Jersey, USA, August 18-20, 2019)*, volume 137 of *LIPIcs*. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2019.

[CGMV20] Amit Chakrabarti, Prantar Ghosh, Andrew McGregor, and Sofya Vorotnikova. Vertex ordering problems in directed graph streams. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2020, Salt Lake City, Utah, USA, January 5-9, 2020)*, pages 1786–1802. Society for Industrial and Applied Mathematics, January 2020.

[CGS+21] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, Ameya Velingker, and Santhoshini Velusamy. Linear Space Streaming Lower Bounds for Approximating CSPs. June 2021.

[CGSV21a] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, and Santhoshini Velusamy. Approximability of all Boolean CSPs in the dynamic streaming setting. April 2021.

[CGSV21b] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, and Santhoshini Velusamy. Approximability of all finite CSPs in the dynamic streaming setting. June 2021.

[CGV20] Chi-Ning Chou, Alexander Golovnev, and Santhoshini Velusamy. Optimal Streaming Approximations for all Boolean Max-2CSPs and Max-$k$SAT. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS 2020, November 16-19, 2020)*, pages 330–341. IEEE Computer Society, November 2020.

[CS98] Benny Chor and Madhu Sudan. A geometric approach to betweenness. *SIAM Journal on Discrete Mathematics*, 11(4):511–523, 1998. Conference version in Algorithms, ESA 1995.

[GHM+11] Venkatesan Guruswami, Johan Håstad, Rajsekar Manokaran, Prasad Raghavendra, and Moses Charikar. Beating the Random Ordering is Hard: Every ordering CSP is approximation resistant. *SIAM Journal on Computing*, 40(3):878–914, 2011. Conference version in FOCS 2008.

[GKK+08] Dmitry Gavinsky, Julia Kempe, Iordanis Kerenidis, Ran Raz, and Ronald de Wolf. Exponential separation for one-way quantum communication complexity, with applications to cryptography. *SIAM Journal on Computing*, 38(5):1695–1708, December 2008. Conference version in STOC 2007.

[GT19] Venkatesan Guruswami and Runzhou Tao. Streaming Hardness of Unique Games. In Dimitris Achlioptas and László A. Végh, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019) (APPROX 2019, Cambridge, MA, USA, September 20-22, 2019)*, volume 145 of *LIPIcs*, pages 5:1–5:12. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, September 2019.

[GVV17] Venkatesan Guruswami, Ameya Velingker, and Santhoshini Velusamy. Streaming Complexity of Approximating Max 2CSP and Max Acyclic Subgraph. In Klaus Jansen, José D. P. Rolim, David Williamson, and Santosh S. Vempala, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017) (APPROX 2017, Berkeley, CA, USA, August 16-18, 2017)*, volume 81 of *LIPIcs*, pages 8:1–8:19. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, August 2017.

[Hås01] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

[Kar72] Richard M. Karp. Reducibility among Combinatorial Problems. In R.E. Miller, J.W. Thatcher, and J.D. Bohlinger, editors, *The IBM Research Symposia Series*, pages 85–103. Springer, 1972.

[Kho02] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC 2002, Québec, Canada, May 19-21, 2002)*, pages 767–775. Association for Computing Machinery, May 2002.

[KK15] Dmitry Kogan and Robert Krauthgamer. Sketching cuts in graphs and hypergraphs. In *Proceedings of the 6th Annual Conference on Innovations in Theoretical Computer Science (ITCS 2015, Rehovot, Israel, January 11-13, 2015)*, pages 367–376. Association for Computing Machinery, 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 (STOC 2019, Phoenix, AZ, USA, June 23-26, 2019)*, pages 277–288. Association for Computing Machinery, June 2019.

[KKS15] Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Streaming lower bounds for approximating MAX-CUT. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2015, San Diego, California, USA, January 4-6, 2015)*, pages 1263–1282. Society for Industrial and Applied Mathematics, January 2015.

[KKSV17] Michael Kapralov, Sanjeev Khanna, Madhu Sudan, and Ameya Velingker. $(1 + \omega(1))$-approximation to MAX-CUT requires linear space. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2017, Barcelona, Spain, January 16-19, 2017)*, pages 1703–1722. Society for Industrial and Applied Mathematics, January 2017.

[New00] Alantha Newman. *Approximating the Maximum Acyclic Subgraph.* Masters Thesis, Massachusetts Institute of Technology, 2000.

[Opa79] Jaroslav Opatrny. Total Ordering Problem. *SIAM Journal on Computing*, 8(1):111–114, February 1979.

[SSV21] Noah Singer, Madhu Sudan, and Santhoshini Velusamy. Streaming approximation resistance of every ordering CSP. May 2021.

[VY11] Elad Verbin and Wei Yu. The streaming complexity of cycle counting, sorting by reversals, and other problems. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2011, San Francisco, California, USA, January 23-25, 2011)*, pages 11–25. Society for Industrial and Applied Mathematics, 2011.



ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il