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

Streaming approximation resistance of every ordering CSP

Noah Singer* Madhu Sudan† Santhoshini Velusamy‡

Abstract

An ordering constraint satisfaction problem (OCSP) is given by a positive integer k and a constraint predicate Π mapping permutations on $\{1, \ldots, k\}$ to $\{0, 1\}$. Given an instance of OCSP(Π) 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 Π. 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 Π, OCSP(Π) is approximation-resistant to $o(\sqrt{n})$-space streaming algorithms, i.e., algorithms using $o(\sqrt{n})$ space cannot distinguish streams where almost every constraint is satisfiable from streams where no ordering beats the random ordering by a noticeable amount. In the case of MAS our result shows that for every $\varepsilon > 0$, MAS is not $1/2 + \varepsilon$-approximable. The previous best inapproximability result only ruled out a $3/4$ approximation.

Our results build on a recent work of Chou, Golovnev, Sudan, and Velusamy who show tight inapproximability results for some constraint satisfaction problems over arbitrary (finite) alphabets. We show that the hard instances from this earlier work have the following "small-set expansion" property: in every partition of the hypergraph formed by the constraints into small blocks, most of the hyperedges are incident on vertices from distinct blocks. By exploiting this combinatorial property, in combination with a natural reduction from CSPs over large finite alphabets to OCSPs, we give optimal inapproximability results for all OCSPs.

1 Introduction

In this work we consider the complexity of solving "ordering constraint satisfaction problems (OCSP)" 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,

*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

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.

We let $[n]$ denote the set $\{0, \ldots, n-1\}$ and 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 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 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 Π 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 $\boldsymbol{\sigma} : [n] \to [n]$ 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 Π to the variables $\mathbf{j}(i) = (j(i)_0, \ldots, j(i)_{k-1})$. (We omit Π 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 Ψ is defined as $\mathsf{val}_\Psi = \max_{\boldsymbol{\sigma} \in \mathsf{S}_n} \{\mathsf{val}_\Psi(\boldsymbol{\sigma})\}$.

Two simple examples of $\mathsf{Max\text{-}OCSP}$ problems are the *maximum acyclic subgraph* (MAS) problem and the $\mathsf{Betweenness}$ problem. MAS corresponds to the ordering constraint function $\Pi_{\mathsf{MAS}} : \mathsf{S}_2 \to \{0,1\}$ given by $\Pi_{\mathsf{MAS}}([0\ 1]) = 1$ and $\Pi_{\mathsf{MAS}}([1\ 0]) = 0$. If we re-interpret the constraints as directed edges in a graph on n vertices, the problem asks for an ordering of the vertices which maximizes the number of forward edges (which form an acyclic subgraph). The $\mathsf{Betweenness}$ problem corresponds to the ordering constraint function $\Pi_{\mathsf{Betweenness}} : \mathsf{S}_3 \to \{0,1\}$ given by $\Pi_{\mathsf{Betweenness}}([0\ 1\ 2]) = 1, \Pi_{\mathsf{Betweenness}}([2\ 1\ 0]) = 1$, and $\Pi_{\mathsf{Betweenness}}(\boldsymbol{\pi}) = 0$ for all other $\boldsymbol{\pi} \in \mathsf{S}_3$. Here, a constraint (i,j,k) reads as "x_j lies between x_i and x_k", and the goal is again to find a permutation $\boldsymbol{\sigma} \in \mathsf{S}_n$ maximizing the number of satisfied constraints.

1.2 Approximability and Streaming Algorithms

In this work we consider the "approximability" of $\mathsf{Max\text{-}OCSP}(\Pi)$ in the "streaming setting". We define these terms next starting with the latter.

In the (single-pass) "streaming setting" an instance $\Psi = (C_0, \ldots, C_{m-1})$ of $\mathsf{Max\text{-}OCSP}(\Pi)$ is presented as a stream of constraints with the ith 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 $A(\Psi) \in [0,1]$. 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 ($\Omega(n^\varepsilon)$ for $\varepsilon > 0$) in the input length.

We say that A is an *α-approximation algorithm* if for every Ψ, $\alpha \cdot \mathsf{val}_\Psi \leq A(\Psi) \leq \mathsf{val}_\Psi$ with probability at least $2/3$ over the internal coin tosses of A. Thus our approximation factors α 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 Π is satisfied by a random ordering. Every instance of Ψ satisfies $\mathsf{val}_\Psi \geq \rho(\Pi)$ and thus the algorithm that always outputs $\rho(\Pi)$ is a $\rho(\Pi)$-approximation algorithm for $\textsf{Max-OCSP}(\Pi)$. We say that a problem is *approximable (in the streaming setting)* if we can beat this trivial algorithm by a positive factor. Specifically $\textsf{Max-OCSP}(\Pi)$ 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 $\textsf{Max-OCSP}(\Pi)$, We say $\textsf{Max-OCSP}(\Pi)$ is *approximation-resistant (in the streaming setting)* otherwise.

1.3 Main result and comparison to prior works

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

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

Our theorem parallels a result of Guruswami, Håstad, Manokaran, Raghavendra, and Charikar [GHM+11] who prove approximation resistance with respect to polynomial time algorithms based on the unique games conjecture. In our setting of streaming algorithms the only problem that seems to have been explored in the literature before was $\textsf{MAS}$, and even in this case a tight result was not known. Guruswami, Velingker, and Velusamy [GVV17] proved that for every $\varepsilon > 0$, $\textsf{MAS}$ is not $(\frac{7}{8}+\varepsilon)$-approximable in $o(\sqrt{n})$ space. A stronger hardness for $3/4$ approximation for $\textsf{MAS}$ is indicated in the work of Guruswami and Tao [GT19] who suggest that their hardness of unique games, an "unordered" CSP problem, could be converted to such a hardness for $\textsf{MAS}$. As far as we know our result is the first tight hardness result for $\textsf{Max-OCSP}(\Pi)$ for any non-constant Π, while yielding tight hardness results for every Π.

1.4 Techniques

We start by describing our proof technique for the special case of the $\textsf{MAS}$ problem. Later we describe the general case.

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 and Velusamy [CGSV21]. Thus whereas previous approaches towards proving hardness of $\textsf{MAS}$, for example, were unable to get optimal hardness results for $\textsf{MAS}$ 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.

Recall that $\Pi_{\textsf{MAS}}([0\ 1]) = 1$ while $\Pi_{\textsf{MAS}}([1\ 0]) = 0$. For a large constant q, we define the constraint function $f^q_{\textsf{MAS}} : [q]^2 \to \{0,1\}$ by $f^q_{\textsf{MAS}}(x,y) = 1$ iff $x < y$. $\textsf{Max-CSP}(f^{\textsf{MAS}}_q)$, the problem of maximizing $f^{\textsf{MAS}}_q$ constraints applied to variables which take values in $[q]$, aims to capture a "q-coarsening" of $\Pi_{\textsf{MAS}}$. Specifically we think of an ordering $\boldsymbol{\sigma}$ of n variables as dividing the n variables into q blocks with variables $\sigma_0, \ldots, \sigma_{n/q-1}$ being in the first block, $\sigma_{n/q}, \ldots, \sigma_{2n/q-1}$ being in the second block and so on. $f^q_{\textsf{MAS}}$ is defined so that if the $[q]$-assignment to the variables based on which block they belong to satisfies an $f^q_{\textsf{MAS}}$ constraint, then the underlying $\Pi_{\textsf{MAS}}$ constraint will be satisfied by $\boldsymbol{\sigma}$.

We can get an optimal hardness result for $\mathsf{Max\text{-}CSP}(f^q_{\mathsf{MAS}})$ from the work of [CGSV21] — we can use their results to show that $o(\sqrt{n})$ space algorithms cannot distinguish "**YES** instances" whose $\mathsf{Max\text{-}CSP}(f^q_{\mathsf{MAS}})$ value is $1 - 1/q$ from "**NO** instances" instances whose $\mathsf{Max\text{-}CSP}(f^q_{\mathsf{MAS}})$ value is $1/2$. (We remark that even to get this result we need to choose some "distributions" carefully and this is not immediate from the previous work, but once these choices are made, the lower bound follows from the previous work.) However this does not immediately imply a hardness result for the original OCSP problem $\mathsf{Max\text{-}OCSP}(\Pi_{\mathsf{MAS}})$: By definition of f^q_{MAS} it follows that the **YES** instances of MAS have $\mathsf{Max\text{-}OCSP}(\Pi_{\mathsf{MAS}})$ values at least $1-1/q$ and they are indistinguishable to small space algorithms from the **NO** instances, but the **NO** instances may now have $\mathsf{Max\text{-}OCSP}(\Pi_{\mathsf{MAS}})$ value much higher than $1/2$.

To get hardness of $\mathsf{Max\text{-}OCSP}(\Pi_{\mathsf{MAS}})$ we can no longer use the main theorems of [CGSV21] as a black box. Instead we need to delve into their reduction and notice that the hard instances (in the **NO** case) not only have small $\mathsf{Max\text{-}CSP}(f^q_{\mathsf{MAS}})$ values but also 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. This additional property allows us to prove that the reduction from the coarsened problem $\mathsf{Max\text{-}CSP}(f^q_{\mathsf{MAS}})$ to the ordering problem $\mathsf{Max\text{-}OCSP}(\Pi_{\mathsf{MAS}})$ preserves values approximately (to within an additive $o(1)$ amount).

Extending the idea to other OCSPs involves two additional steps. We define f^q_{Π} analogously to f^q_{MAS} (the definition is completely determined by Π and q), but we still need to find the right "distributions" that allow us to apply the results of [CGSV21]. We describe this process in Section 3.1. Having done this we now need an analysis of the **NO** instances arising from the construction in [CGSV21]. 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 two vertices from the same block. This allows us to show that the q-coarsened unordered instances have roughly the same $\mathsf{Max\text{-}CSP}(f^q_{\Pi})$ and $\mathsf{Max\text{-}OCSP}(\Pi)$ values (in the **NO** case) and this allows us to get optimal hardness results for all ordering CSPs.

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 CSP value after coarsening are somewhat different: Their analysis involves more sophisticated tools such as influence of variables and Gaussian noise stability. Our analysis in contrast is a more elementary analysis of the type common with random graphs.

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 $\mathsf{supp}(\Pi) = \{\boldsymbol{\pi} \in \mathsf{S}_k | \Pi(\boldsymbol{\pi}) = 1\}$.

A (directed, 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* (i.e., ordered k-tuples of vertices), such that no vertex appears in the same k-hyperedge twice. 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 *α-partial k-hypermatching* is a k-hypermatching which contains αn k-hyperedges.

A *q-partition* of V is a map $\mathcal{P} : V \to [q]$. Importantly, q-partitions are *ordered* objects; that is, composing a q-partition $\mathcal{P}$ with a nontrivial permutation on $[q]$ leads to a new q-partition which we treat as distinct. Given a q-partition $\mathcal{P} : V \to [q]$ of V and $i \in [q]$, we define the *i-th block* $\mathcal{P}_i$ as the set $\mathcal{P}^{-1}(i) \subseteq V$.

Given an instance Ψ 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 Ψ (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 $X_0, \dots, X_{m-1}$ be (not necessarily independent) $\{0,1\}$-valued random variables, such that for some $p \in (0,1)$, $\mathsf{E}[X_i \mid X_0, \dots, X_{i-1}] \leq p$ for every $i \in [m]$. Then if $\mu := pm$,*

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

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 $\Pi : \mathsf{S}_k \to \{0,1\}$, $\mathsf{Max\text{-}OCSP}(\Pi)$ is approximation resistant in the (single-pass) streaming setting. In particular for every $\varepsilon > 0$, every $\rho(\Pi) + \varepsilon$ approximation algorithm A for $\mathsf{Max\text{-}OCSP}(\Pi)$ requires $\Omega(\sqrt{n})$ 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, α 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 3.6. 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 3.7). Finally we assert in Lemma 3.8 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 $\Omega(\sqrt{n})$ space.

Assuming Lemma 3.6, Lemma 3.7, and Lemma 3.8 the proof of Theorem 1.1 is straightforward and proved at the end of this section. Proofs of Lemma 3.6 and Lemma 3.7 are in Section 4 and of Lemma 3.8 in Section 5.

3.1 Distribution of hard instances

The work of [CGSV21] reduces the task of building hard instances of k-ary CSPs over alphabets of size q in the streaming setting to the task of defining two distributions supported on $[q]^k$ satisfying certain properties. Following the same approach, to define $\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}(\Pi)$ and $\mathcal{G}^N_{q,n,\alpha,T}(\Pi)$, we first define a pair of distributions on $[q]^k$, where k is the arity of Π, which are denoted $\mathcal{D}^{Y,\boldsymbol{\pi}}_q(\Pi)$ and $\mathcal{D}^N_q(\Pi)$. Later, in Definition 3.5, we use these distributions to define $\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}(\Pi)$ and $\mathcal{G}^N_{q,n,\alpha,T}(\Pi)$.

For $i \in [q]$, define the k-tuple of "contiguous" values $\mathbf{v}^i_q = (i, \ldots, i+k-1 \pmod q)$. 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 $(a_{\boldsymbol{\pi}^{-1}(0)}, \ldots, a_{\boldsymbol{\pi}^{-1}(k-1)})$. 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. Let $\boldsymbol{\tau} = \mathsf{ord}(\mathbf{a})$, so that $\boldsymbol{\tau}$ is the unique permutation such that $a_{\boldsymbol{\tau}(0)} < \cdots < a_{\boldsymbol{\tau}(k-1)}$. 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. $\square$

Now the distributions supported on $[q]^k$ are defined as follows:

Definition 3.2 ($\mathcal{D}^{Y,\boldsymbol{\pi}}_q(\Pi)$ and $\mathcal{D}^N_q(\Pi)$)**.** *Let Π be a Max-OCSP of arity k. For $q \in \mathbb{N}$ and $\boldsymbol{\pi} \in \mathsf{S}_k$, $\mathcal{D}^{Y,\boldsymbol{\pi}}_q(\Pi)$ is the uniform distribution over the set $\{(\mathbf{v}^i_q)_{\boldsymbol{\pi}} : i \in [q]\}$. For $q \in \mathbb{N}$, $\mathcal{D}^N_q$ is the uniform distribution over all k-tuples in $[q]^k$.*

For a distribution $\mathcal{D}$ supported on $[q]^k$ and index $i \in [k]$ we define its *ith marginal* to be the distribution $\mathcal{D}_i$ supported on $[q]$ sampled by picking $\mathbf{a} = (a_0, \ldots, a_{k-1}) \sim \mathcal{D}$ and outputting a_i. We say that a distribution $\mathcal{D}$ has *uniform marginals* if $\mathcal{D}_i$ is the uniform distribution on $[q]$ for every $i \in [k]$.

The following proposition follows immediately from the definition of the $\mathcal{D}^{Y,\boldsymbol{\pi}}_q(\Pi)$ and $\mathcal{D}^N_q(\Pi)$.

Proposition 3.3. *For every Π, $\boldsymbol{\pi}$, k and q, the distributions $\mathcal{D}^{Y,\boldsymbol{\pi}}_q(\Pi)$ and $\mathcal{D}^N_q(\Pi)$ have uniform marginals.*

Definition 3.4 (Uniform distribution over partial hypermatchings)**.** *Let $\mathcal{H}_{n,\alpha}$ denote the uniform distribution over all α-partial k-hypermatchings on $[n]$.*

We now formally define our **YES** and **NO** distributions for $\mathsf{Max\text{-}OCSP}(\Pi)$. See Figure 1 below for a visual interpretation in the case of MAS.

Definition 3.5 ($\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}(\Pi)$ and $\mathcal{G}^{N}_{q,n,\alpha,T}(\Pi)$)**.** *Let $q,n,T \in \mathbb{N}$, $\alpha > 0$, and let $B = N$ or $B = (Y,\boldsymbol{\pi})$ for some $\boldsymbol{\pi} \in \mathsf{S}_k$. We define the distribution $\mathcal{G}^{B}_{q,n,\alpha,T}$, over n-variable $\mathsf{Max\text{-}OCSP}(\Pi)$ instances, as follows:*

1. *Sample a uniformly random q-partition $\mathcal{P} : [n] \to [q]$.*
2. *Sample T hypermatchings independently $\widetilde{G}_0, \ldots, \widetilde{G}_{T-1} \sim \mathcal{H}_{n,\alpha}$.*
3. *For each $\ell \in [T]$, do the following. Let $\widetilde{G}_\ell$ be an empty k-hypergraph on $[n]$. For each k-hyperedge $\mathbf{e} = (j_0, \ldots, j_{k-1}) \in E(\widetilde{G}_\ell)$, sample a tuple $\mathbf{i} = (i_0, \ldots, i_{k-1}) \sim \mathcal{D}^B_q$, and add the k-hyperedge $\mathbf{e}$ to $\widetilde{G}_\ell$ if and only if $(\mathcal{P}(j_0), \ldots, \mathcal{P}(j_{k-1})) = \mathbf{i}$.*
4. *Let $G := G_0 \cup \cdots \cup G_{T-1}$.*
5. *Return the $\mathsf{Max\text{-}OCSP}(\Pi)$ instance Ψ on n variables given by the constraint hypergraph G.*

We say that an algorithm $\mathbf{ALG}$ achieves advantage δ 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

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

In the following section we state lemmas which highlight the main properties of the distributions above.

3.2 Statement of key lemmas

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

Lemma 3.6 ($\mathcal{G}^Y$ has high $\mathsf{Max\text{-}OCSP}(\Pi)$ values)**.** *For every ordering constraint satisfaction function Π, every $\boldsymbol{\pi} \in \mathsf{supp}(\Pi)$, and $\Psi \sim \mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}(\Pi)$, we have $\mathsf{val}_\Psi \geq 1 - \frac{k-1}{q}$ (i.e., this occurs with probability 1).*

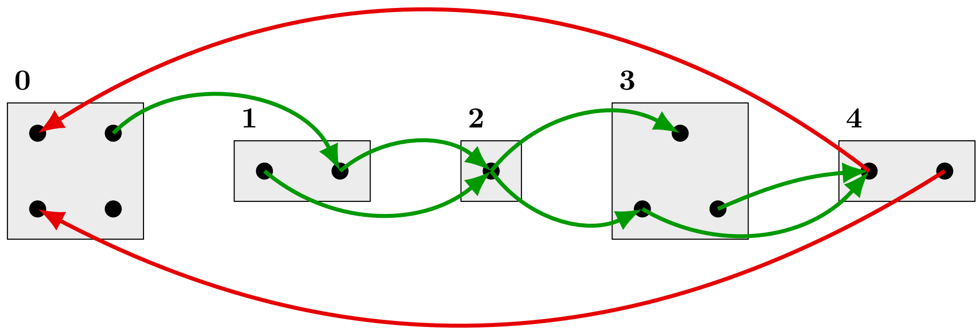
We prove Lemma 3.6 in Section 4.2. Next we assert that $\mathcal{G}^N$ is supported mostly on instances of low value.

Lemma 3.7 ($\mathcal{G}^N$ has low $\mathsf{Max\text{-}OCSP}(\Pi)$ 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*

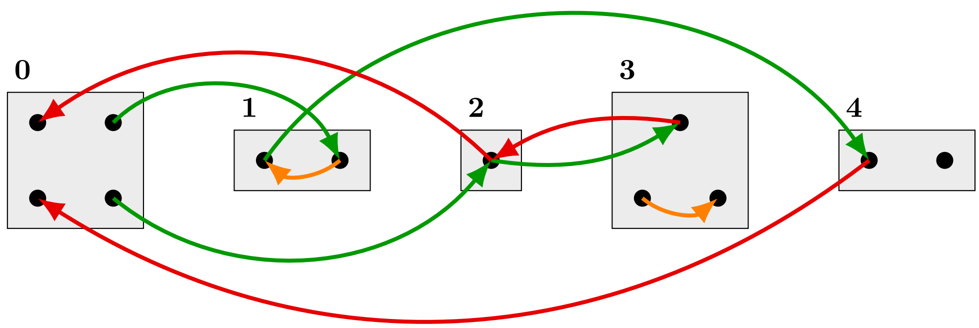
$$\Pr_{\Psi \sim \mathcal{G}^{N}_{q,n,\alpha,T}}\left[\mathsf{val}_\Psi \geq \rho(\Pi) + \frac{\varepsilon}{2}\right] \leq 0.01.$$

We prove Lemma 3.7 in Section 4.3. We note that this lemma is more technically involved than Lemma 3.6 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 [CGSV21].

Lemma 3.8. *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 $\boldsymbol{\pi} \in \mathsf{supp}(\Pi)$, every streaming algorithm $\mathbf{ALG}$ distinguishing $\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}(\Pi)$ from $\mathcal{G}^{N}_{q,n,\alpha,T}(\Pi)$ with advantage $1/8$ for all lengths n uses space $\Omega(\sqrt{n})$.*



(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 Π supported only on $[0\ 1]$. According to the definition of $\mathcal{G}^Y$ (see Definition 3.2 and Definition 3.5, with $\boldsymbol{\pi} = [0\ 1]$), instances are sampled by first sampling a q-partition $\mathcal{P} : [n] \to [q]$, and then sampling some edges; every sampled edge (u, v) must satisfy $\mathcal{P}(v) = \mathcal{P}(u)+1 \pmod q$. On the other hand, there are no requirements on $(\mathcal{P}(u), \mathcal{P}(v))$ for instances sampled from $\mathcal{G}^N$. Above, the blocks of the partition $\mathcal{P}$ 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 $\mathcal{P}(v) > \mathcal{P}(u)$, $\mathcal{P}(v) = \mathcal{P}(u)$, or $\mathcal{P}(v) < \mathcal{P}(u)$, 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 3.6 and Lemma 3.7, 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.

3.3 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 $\mathsf{Max\text{-}OCSP}(\Pi)$ that uses space s. Fix $\boldsymbol{\pi} \in \mathsf{supp}(\Pi)$. Consider the algorithm $\mathbf{ALG}$ defined as follows: on input Ψ, an instance of $\mathsf{Max\text{-}OCSP}(\Pi)$, if $A(\Psi) \geq \rho(\Pi) + \frac{\varepsilon}{2}$, then $\mathbf{ALG}$ outputs 1, else, it outputs 0. Observe that $\mathbf{ALG}$ uses $O(s)$ space. Set $q_0 \geq \frac{2(k-1)}{\varepsilon}$ such that the condition of Lemma 3.7 holds. Set $\alpha_0 \in (0, \alpha_0(k)]$ such that the conditions of Lemma 3.7 holds. Consider any $q \geq q_0$ and $\alpha \leq \alpha_0$: let T_0 be set as in Lemma 3.7. Consider any $T \geq T_0$: since $q \geq \frac{2(k-1)}{\varepsilon}$, it follows from Lemma 3.6 that for $\Psi \sim \mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}(\Pi)$, we have $\mathsf{val}_\Psi \geq 1 - \frac{\varepsilon}{2}$, and hence with probability at least $2/3$, $A(\Psi) \geq \rho(\Pi) + \frac{\varepsilon}{2}$. Therefore, $\mathbb{E}_{\Psi \sim \mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}(\Pi)}[\mathbf{ALG}(\Psi) = 1] \geq 2/3$. Similarly, by the choice of q_0, α_0, T_0, it follows from Lemma 3.7 that

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

and hence, $\mathbb{E}_{\Psi \sim \mathcal{G}^N_{q,n,\alpha,T}(\Pi)}[\mathbf{ALG}(\Psi) = 1] \leq \frac{1}{3} + 0.01$. Therefore, $\mathbf{ALG}$ distinguishes $\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}(\Pi)$ from $\mathcal{G}^N_{q,n,\alpha,T}(\Pi)$ with advantage $1/8$. By applying Lemma 3.8, we conclude that the space complexity of A is at least $\Omega(\sqrt{n})$. □

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

The goal of this section is to prove our technical lemmas which lower bound the $\mathsf{Max\text{-}OCSP}(\Pi)$ values of $\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}$ (Lemma 3.6) and upper bound the $\mathsf{Max\text{-}OCSP}(\Pi)$ values of $\mathcal{G}^N_{q,n,\alpha,T}$ (Lemma 3.7).

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 S_n), and define an operation called *q-coarsening* on $\mathsf{Max\text{-}OCSP}$'s, which restricts the solution space from S_n to $[q]^n$.

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

Definition 4.1 (q-coarsening)**.** *Let Π be a k-ary $\mathsf{Max\text{-}OCSP}$ and let $q \in \mathbb{N}$. The* q-coarsening *of Π is the k-ary $\mathsf{Max\text{-}CSP}$ problem $\mathsf{Max\text{-}CSP}(f^q_\Pi)$ 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 $\Pi(\mathit{ord}(\mathbf{a})) = 1$. The* q-coarsening of an instance *Ψ of $\mathsf{Max\text{-}OCSP}(\Pi)$ is the instance Φ of $\mathsf{Max\text{-}CSP}(f^q_\Pi)$ 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 $q \in \mathbb{N}$, Ψ is an instance of $\mathsf{Max\text{-}OCSP}(\Pi)$, and Φ is the q-coarsening of Ψ, then $\mathsf{val}_\Psi \geq \mathsf{val}_\Phi^q$.*

Proof. We will show that for every assignment $\mathbf{b} \in [q]^n$ to Φ, we can construct an assignment $\boldsymbol{\sigma} \in \mathsf{S}_n$ to Ψ such that $\mathsf{val}_\Psi(\boldsymbol{\sigma}) \geq \mathsf{val}_\Phi^q(\mathbf{b})$. Specifically, given an assignment $\mathbf{b} \in [q]^n$ to Φ, for $i \in [q]$, let $S_i \subseteq [n]$ be the sequence of indices with assigned value q, enumerated in some arbitrary order. Next, let $\boldsymbol{\sigma}$ be the ordering on $[n]$ given by placing $S_0, \ldots, S_{q-1}$ in order. Consider any constraint $C = (j_0, \ldots, j_{k-1})$ in Φ which is satisfied by $\mathbf{b}$. Since $f_\Pi^q(b_{j_0}, \ldots, b_{j_{k-1}}) = 1$, $\Pi(\mathsf{ord}(b_{j_0}, \ldots, b_{j_{k-1}})) = 1$. By construction, since $b_{j_0}, \ldots, b_{j_{k-1}}$ are distinct, $\mathsf{ord}(b_{j_0}, \ldots, b_{j_{k-1}}) = \mathsf{ord}(\boldsymbol{\sigma}(j_0), \ldots, \boldsymbol{\sigma}(j_{k-1}))$. Hence C is also satisfied by $\boldsymbol{\sigma}$ in Ψ, and so $\mathsf{val}_\Psi(\boldsymbol{\sigma}) \geq \mathsf{val}_\Psi^q(\mathbf{b})$. □

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

In this section, we prove Lemma 3.6, which states that the $\mathsf{Max\text{-}OCSP}(\Pi)$ values of instances Ψ drawn from $\mathcal{G}_{q,n,\alpha,T}^{Y,\boldsymbol{\pi}}$ are large. For convenience, we restate it here:

Lemma 3.6 ($\mathcal{G}^Y$ has high $\mathsf{Max\text{-}OCSP}(\Pi)$ values)**.** *For every ordering constraint satisfaction function Π, every $\boldsymbol{\pi} \in \mathsf{supp}(\Pi)$, and $\Psi \sim \mathcal{G}_{q,n,\alpha,T}^{Y,\boldsymbol{\pi}}(\Pi)$, we have $\mathsf{val}_\Psi \geq 1 - \frac{k-1}{q}$ (i.e., this occurs with probability 1).*

Note that we prove a bound for *every* instance Ψ in the support of $\mathcal{G}_{q,n,\alpha,T}^{Y,\boldsymbol{\pi}}$, although it would suffice for our application to prove that such a bound holds with high probability over the choice of Ψ.

To prove Lemma 3.6, if Φ is the q-coarsening of Ψ, by Lemma 4.2, it suffices to show that $\mathsf{val}_\Phi^q \geq 1 - \frac{k-1}{q}$. One natural approach is to consider the q-partition $\mathcal{P} : [n] \to [q]$ sampled when sampling Ψ, and define the assignment $\mathbf{b}_\Psi$ to Φ by $(\mathbf{b}_\Psi)_i = \mathcal{P}(i)$. Consider any constraint $C = \mathbf{j} = (j_0, \ldots, j_{k-1})$ in Ψ; by the definition of $\mathcal{G}^{Y,\boldsymbol{\pi}}$ (Definition 3.5), we have $(\mathcal{P}(j_0), \ldots, \mathcal{P}(j_{k-1})) = (\mathbf{v}_q^\ell)_{\boldsymbol{\pi}}$ for some (unique) $\ell \in [q]$, which we term the *identifier* of C (recall, we defined $\mathbf{v}_q^\ell$ as the k-tuple $(\ell, \ldots, \ell + k - 1 \pmod q) \in [q]^k$). Now $\mathbf{b}_\Psi|_\mathbf{j} = (\mathbf{v}_q^\ell)_{\boldsymbol{\pi}}$. Hence, C is satisfied by $\mathbf{b}_\Psi$ 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 $\mathbf{b}_\Psi$ 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 $\mathbf{b}_\Psi$.[1] The proof is as follows:

Proof of Lemma 3.6. For $t \in [q]$, define the assignment $\mathbf{b}_\Psi^t$ to Φ as $(\mathbf{b}_\Psi^t)_i = \mathcal{P}(i) + t \pmod q$ for $i \in [n]$.

Fix $t \in [q]$. Then we claim that $\mathbf{b}_\Psi^t$ satisfies any constraint C with identifier ℓ such that $\ell + t \pmod q \in \{0, \ldots, q-k\}$. Indeed, if $C = \mathbf{j}$ is a constraint with identifier ℓ, since $(\mathcal{P}(j_0), \ldots, \mathcal{P}(j_{k-1})) = (\mathbf{v}_q^\ell)_{\boldsymbol{\pi}}$, then we have $\mathbf{b}_\Psi^t|_\mathbf{j} = (\mathbf{v}_q^{\ell+t})_{\boldsymbol{\pi}}$; as long as $\ell + t \in \pmod q \in \{0, \ldots, q-k\}$, then $\mathsf{ord}(\mathbf{v}_q^{\ell+t}) = [0 \ \cdots \ k-1]$, and so $\mathsf{ord}((\mathbf{v}_q^{\ell+t})_{\boldsymbol{\pi}}) = \boldsymbol{\pi}$ and $\Pi(\mathsf{ord}((\mathbf{v}_q^{\ell+t})_{\boldsymbol{\pi}})) = 1$.

Now (no longer fixing t), for each $\ell \in [q]$, let w_ℓ be the fraction of constraints in Ψ with identifier ℓ. By the above claim, for each $t \in [q]$, we have $\mathsf{val}_\Phi^q(\mathbf{b}_\Psi^t) \geq \sum_{\ell : \ell+t \pmod q \in \{0,\ldots,q-k\}} w_\ell$. On the

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

other hand, $\sum_{\ell=0}^{q-1} w_\ell = 1$ (since every constraint has some (unique) identifier). Hence

$$\sum_{t=0}^{q-1} \mathsf{val}_\Phi(b_\Psi^t) \geq \sum_{t=0}^{q-1} \left(\sum_{\ell:\ell+t \pmod q \in \{0,\dots,q-k\}} w_\ell \right) = q-(k-1),$$

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

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

In this section, we prove Lemma 3.7, 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. Restated:

Lemma 3.7 ($\mathcal{G}^N$ has low $\mathsf{Max\text{-}OCSP}(\Pi)$ 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*

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

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 3.7, 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 $\mathcal{P} : V \to [q]$, a k-hyperedge $\mathbf{e} \in E$* congregates *on $\mathcal{P}$ if it lies on one of the blocks $\mathcal{P}_i$.*

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 (γ, δ)-*small set hypergraph expander (SSHE) *if it has the following property: For every subset $S \subseteq V$ of size at most $\gamma|V|$, $N(G, S) \leq \delta|E|$ (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 (γ, δ)-*small partition hypergraph expander (SPHE) *if it has the following property: For every partition $\mathcal{P} : V \to [q]$ where each block $\mathcal{P}_i$ has size at most $\gamma|V|$, the number of k-hyperedges in E which congregate on $\mathcal{P}$ 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 3.7. It will be proved formally later in this section.

Proof sketch of Lemma 3.7. 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 Ψ is satisfied. □

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 S_n directly.

In the remainder of this section, we prove the necessary lemmas and then give a formal proof of Lemma 3.7. 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 (γ, δ)-SSHE, then it is a $\left(\gamma, \delta(\frac{2}{\gamma} + 1)\right)$-SPHE.*

Proof. Let $n = |V|$. Consider any partition $\mathcal{P} : V \to [\ell]$ of V where each block has size at most γn. WLOG, all but one block $\mathcal{P}_i$ 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 $\mathcal{P}$). Hence $\ell \leq \frac{2}{\gamma} + 1$.[2] By the SSHE property, there are at most δm k-hyperedges which lie on each block; hence there are at most $\delta(\frac{2}{\gamma} + 1)m$ constraints which congregate on $\mathcal{P}$. □

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

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

Proof. We will show that for every assignment $\boldsymbol{\sigma} \in \mathsf{S}_n$ to Ψ, we can construct an assignment $\mathbf{b} = (b_0, \ldots, b_{n-1}) \in [q]^n$ to Φ such that $\mathsf{val}_\Psi(\boldsymbol{\sigma}) \leq \mathsf{val}_\Phi^q(\mathbf{b}) + \delta$. 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 Φ. 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 $\mathbf{b} \in [q]^n$ as a q-partition $\mathcal{P}_\mathbf{b} : [n] \to [q]$ (given by $\mathcal{P}_\mathbf{b}(i) = b_i$). Consider the constraint hypergraph of Ψ (which is the same as the constraint hypergraph of Φ). 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 $\mathcal{P}_\mathbf{b}$. If C is good, then $\mathcal{P}_\mathbf{b}(j_0), \ldots, \mathcal{P}_\mathbf{b}(j_{k-1})$ are all distinct; together with monotonicity of $\mathbf{b}$, we conclude that if C is good, then $\mathsf{ord}(b_{j_0}, \ldots, b_{j_{k-1}}) = \mathsf{ord}(\boldsymbol{\sigma}(j_0), \ldots, \boldsymbol{\sigma}(j_{k-1}))$.

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

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

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 $\mathsf{Max\text{-}OCSP}(\Pi)$ 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. We observe that the following process samples an instance from $\mathcal{G}^N_{q,n,\alpha,T}$. First, sample T hypermatchings $\widetilde{G}_0, \ldots, \widetilde{G}_{T-1} \sim \mathcal{H}_{n,\alpha}$ independently, and let $\widetilde{G} := \widetilde{G}_0 \cup \cdots \cup \widetilde{G}_{T-1}$. Then, throw away every k-hyperedge in $\widetilde{G}$ with probability $1 - \frac{1}{q^k}$ independently to get a new k-hypergraph G, and return the $\mathsf{Max\text{-}OCSP}(\Pi)$ instance whose constraint hypergraph is G. Hence, the number of constraints in Ψ is distributed as the sum of $n\alpha T$ independent Bernoulli$(1/q^k)$ random variables. The desired bound then follows by applying the Chernoff bound. $\square$

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 γn vertices, we have*

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

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

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

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

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

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

□

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

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

Proof. Let $\alpha_0, \ldots, \alpha_{T-1} \geq 0$ be such that $\frac{\alpha T}{2q^k} \leq \alpha_0 + \cdots + \alpha_{T-1} \leq \alpha T$. 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 3.5). This is equivalent to simply sampling each $G_i \sim \mathcal{H}_{n,\alpha_i}$ independently.

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

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

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 $\leq 2^n$ possible subsets $S \subseteq [n]$ gives the desired bound. □

4.3.2 $\mathcal{G}^N$ has low coarsened $\textsf{Max-CSP}(f^q_\Pi)$ values with high probability

For $G \sim \mathcal{H}_{n,\alpha}$, we define an instance $\Phi(G)$ of $\textsf{Max-CSP}(f^q_\Pi)$ 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 $\textsf{Max-CSP}(f^q_\Pi)$)**.** *For every n, $\alpha, \eta > 0$, and $\mathbf{b} \in [q]^n$,*

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

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., $f^q_\Pi(\mathbf{b}|_{\mathbf{j}(i)}) = 1$. 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, $\mathbb{E}[X_i \mid X_0, \ldots, X_{i-1}] \le \rho(\Pi)$. 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 k-tuple $\mathbf{j}^{\boldsymbol{\pi}} = (j_{\boldsymbol{\pi}(0)}, \ldots, j_{\boldsymbol{\pi}(k-1)})$ is not satisfied by $\mathbf{b}$. On the other hand, if $b_{j_0}, \ldots, b_{j_{k-1}}$ are all distinct, then

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

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

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

as desired. ☐

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

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

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 Φ). ☐

We finally give the proof of Lemma 3.7.

Proof of Lemma 3.7. Let $q_0 := \left\lceil \frac{192k^2}{\varepsilon} \right\rceil$ and let $\alpha_0 := \frac{1}{2k}$. Suppose $\alpha \le \alpha_0$ and $q \ge q_0$. 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 $T \ge T_0$; 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 $\alpha \le \alpha_0 = \frac{1}{2k}$); taking a union bound (and then conditioning on $m(\Psi) \ge \frac{n\alpha T}{2q^k}$), 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 $\mathsf{val}^q_\Phi \ge \rho(\Pi) + \eta$ (where Φ is the q-coarsening of Ψ) and $G(\Psi)$ is a (γ, δ)-SSHE. If $G(\Psi)$ is a (γ, δ)-SSHE, by Lemma 4.7 it is also a (γ, δ')-SPHE, where $\delta' := \frac{3\delta}{\gamma} \ge \delta(\frac{2}{\gamma} + 1)$. Note that $\delta' = 24k^2\gamma = \frac{\varepsilon}{4}$. Now since $q \ge q_0 \ge \frac{2}{\gamma}$, we can apply Lemma 4.8, and conclude that for sufficiently large n, with probability ≥ 0.99 over the choice of $\Psi \sim \mathcal{G}^N_{q,n,\alpha,T}$, we have

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

as desired. ☐

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

In this section we prove Lemma 3.8. This indistinguishability follows directly from the work of [CGSV21], who introduce a 2-player communication problem called "Signal Detection (SD)", and a related streaming problem called "Streaming SD". Both problems are parameterized by two distributions $\mathcal{D}^Y$ and $\mathcal{D}^N$ supported on $[q]^k$. If both distributions have uniform marginals then they show that the corresponding streaming-SD problem requires $\Omega(\sqrt{n})$ space to solve. Our lower bound on the distinguishability of $\mathcal{G}^Y$ and $\mathcal{G}^N$ follows immediately.

In order to state their result that we use, we recall their definition of the Streaming-SD problem, which relies in turn on two distributions they define as part of the SD problem. We define their distributions below, and then define the Streaming-SD problem and then state their space lower bound. The following definition is based on [CGSV21, Definition 5.3].

Definition 5.1 (Signal Detection (SD) Distributions)**.** *Let $n, k, q \in \mathbb{N}, \alpha \in (0,1)$, where k, q and α are constants with respect to n, and αn is an integer less than n/k. For a pair $\mathcal{D}^Y$ and $\mathcal{D}^N$ of distributions over $[q]^k$ we define two distributions $\mathcal{Y}$ and $\mathcal{N}$ over triples $(\mathbf{x}^*, M, \mathbf{z})$ where $\mathbf{x}^* \in [q]^n$, $M \in \{0,1\}^{k\alpha n \times n}$ and $\mathbf{z} \in \{0,1\}^{\alpha n}$.*

- *In the **YES** case, the triple $(\mathbf{x}^*, M, \mathbf{z}) \sim \mathcal{Y}$ is sampled as follows:*
  1. *$\mathbf{x}^* \sim \mathsf{Unif}([q]^n)$.*
  2. *$M \in \{0,1\}^{k\alpha n \times n}$ is chosen uniformly among all matrices with exactly one 1 in each row and at most one 1 in each column. We let $M = (M_0, \ldots, M_{\alpha n - 1})$ where $M_i \in \{0,1\}^{k \times n}$ is the ith block of rows of M, where each block has exactly k rows.*
  3. *$\mathbf{b} = (\mathbf{b}(0), \ldots, \mathbf{b}(\alpha n - 1))$ is sampled by sampling each $\mathbf{b}(i) \in [q]^k$ independently according to $\mathcal{D}^Y$.*
  4. *$\mathbf{z} = (\mathbf{z}(0), \ldots, \mathbf{z}(\alpha n - 1))$ is determined from M, $\mathbf{x}^*$ and $\mathbf{b}$ as follows. For each i, we define $\mathbf{z}(i) = 1$ if $M_i \mathbf{x}^* = \mathbf{b}(i)$, and $\mathbf{z}(i) = 0$ otherwise.*
- *The **NO** case is similar. To sample $(\mathbf{x}^*, M, \mathbf{z}) \sim \mathcal{N}$ we sample $\mathbf{x}^*$ and M as in the **YES** case. We now sample each $\mathbf{b}(i)$ independently according to $\mathcal{D}^N$ for $i \in [\alpha n]$, and let $\mathbf{z}(i) = 1$ if $M_i \mathbf{x}^* = \mathbf{b}(i)$, and $\mathbf{z}(i) = 0$ otherwise.*

We now define the Streaming-SD problem.

Definition 5.2 (Streaming-SD, [CGSV21, Definition 5.5])**.** *Let $n, q, T \in \mathbb{N}, \alpha \in (0,1)$, where q, T, and α are constants with respect to n. For a pair $\mathcal{D}^Y$ and $\mathcal{D}^N$ of distributions over $[q]^k$, the $(\mathcal{D}^Y, \mathcal{D}^N, T)$-streaming-SD problem is the task of distinguishing, for every n, $\boldsymbol{\sigma} \sim \mathcal{Y}_{\mathrm{strm},n}$ from $\boldsymbol{\sigma} \sim \mathcal{N}_{\mathrm{strm},n}$ where for a given length parameter n, the distributions $\mathcal{Y}_{\mathrm{strm}} = \mathcal{Y}_{\mathrm{strm},n}$ and $\mathcal{N}_{\mathrm{strm}} = \mathcal{N}_{\mathrm{strm},n}$ are defined as follows:*

- *Let $\mathcal{Y}$ be the distribution over **YES**-instances of length n, i.e., triples $(\mathbf{x}^*, M, \mathbf{z})$, from **YES**-case of the definition of $(\mathcal{D}^Y, \mathcal{D}^N)$-SD (Definition 5.1). For $\mathbf{x} \in [q]^n$, let $\mathcal{Y}|_{\mathbf{x}}$ denote the distribution $\mathcal{Y}$ conditioned on $\mathbf{x}^* = \mathbf{x}$. The stream $\boldsymbol{\sigma} \sim \mathcal{Y}_{\mathrm{strm}}$ is sampled as follows:*
  - *Sample $\mathbf{x}^*$ uniformly from $[q]^n$.*
  - *Let $(M^{(0)}, \mathbf{z}^{(0)}), \ldots, (M^{(T-1)}, \mathbf{z}^{(T-1)})$ be sampled independently according to $\mathcal{Y}|_{\mathbf{x}^*}$.*
  - *Let $\boldsymbol{\sigma}^{(t)}$ be the pair $(M^{(t)}, \mathbf{z}^{(t)})$ presented as a stream of edges with labels in $\{0,1\}$.*

– *Specifically for $t \in [T]$ and $i \in [\alpha n]$, let $\boldsymbol{\sigma}^{(t)}(i) = (\mathbf{e}^{(t)}(i), \mathbf{z}^{(t)}(i))$ where $\mathbf{e}^{(t)}(i)$ is the i-th hyperedge of $M^{(t)}$, i.e., $\mathbf{e}^{(t)}(i) = (j^{(t)}(ki), \ldots, j^{(t)}(ki+k-1)$ and $j^{(t)}(\ell)$ is the unique index j such that $M^{(t)}_{j,\ell} = 1$.*

– *Let $\boldsymbol{\sigma} = \boldsymbol{\sigma}^{(0)} \circ \cdots \circ \boldsymbol{\sigma}^{(T-1)}$ be the concatenation of the $\boldsymbol{\sigma}^{(t)}$s.*

- *$\boldsymbol{\sigma} \sim \mathcal{N}_{\mathrm{strm}}$ is sampled similarly except we now sample $(M^{(0)}, \mathbf{z}^{(0)}), \ldots, (M^{(T-1)}, \mathbf{z}^{(T-1)})$ independently according to $\mathcal{N}|_{\mathbf{x}^*}$ where $\mathcal{N}|_{\mathbf{x}}$ is the distribution $\mathcal{N}$ of **NO**-instances conditioned on $\mathbf{x}^* = \mathbf{x}$.*

We say that an algorithm **ALG** *solves $(\mathcal{D}_q^{Y,\boldsymbol{\pi}}, \mathcal{D}_q^N, T)$-streaming-SD with advantage δ if there exists an n_0 such that for all $n \geq n_0$, we have*

$$\mathop{\mathbb{E}}_{\sigma \sim \mathcal{Y}_{\mathrm{strm}}}[\mathbf{ALG}(\sigma) = 1] - \mathop{\mathbb{E}}_{\sigma \sim \mathcal{N}_{\mathrm{strm}}}[\mathbf{ALG}(\sigma) = 1] \geq \delta.$$

The following theorem from [CGSV21] states that every streaming algorithm **ALG** solving $(\mathcal{D}^Y, \mathcal{D}^N, T)$-streaming-SD for distributions $\mathcal{D}^Y$ and $\mathcal{D}^N$ with uniform marginal distributions, with some constant advantage for all lengths n, uses space $\Omega(\sqrt{n})$.

Theorem 5.3 ([CGSV21, Lemma 5.14])**.** *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: If $\mathcal{D}^Y, \mathcal{D}^N$ are distributions supported on $[q]^k$ with uniform marginals, then every streaming algorithm* **ALG** *solving $(\mathcal{D}^Y, \mathcal{D}^N, T)$-streaming-SD with advantage $1/8$ for all lengths n uses space $\Omega(\sqrt{n})$.*

We note that Lemma 5.14 in [CGSV21] actually states a more general (and somewhat harder to state) result that effectively allows the use of some pairs of distributions that do not have uniform marginals. But it is straightforward to see that Theorem 5.3 is a special case of their Lemma 5.14 and we do not state the more general form here.

We are now ready to prove Lemma 3.8 which is restated for convenience below.

Lemma 3.8. *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 $\boldsymbol{\pi} \in \mathsf{supp}(\Pi)$, every streaming algorithm* **ALG** *distinguishing $\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}(\Pi)$ from $\mathcal{G}^{N}_{q,n,\alpha,T}(\Pi)$ with advantage $1/8$ for all lengths n uses space $\Omega(\sqrt{n})$.*

Proof of Lemma 3.8. We prove the lemma for the same α_0 as in Theorem 5.3.

Suppose **ALG** distinguishes $\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}(\Pi)$ from $\mathcal{G}^{N}_{q,n,\alpha,T}(\Pi)$ with advantage $1/8$ for all lengths n. In Definition 3.2, we constructed two distributions $\mathcal{D}_q^{Y,\boldsymbol{\pi}}$ and $\mathcal{D}_q^N$ supported on $[q]^k$ with uniform marginals. We now show how to use **ALG** to get an algorithm **ALG′** solving $(\mathcal{D}_q^{Y,\boldsymbol{\pi}}, \mathcal{D}_q^N, T)$-streaming-SD with advantage $1/8$ for all lengths n. The $\Omega(\sqrt{n})$ space lower bound then follows from Theorem 5.3.

Let $\mathcal{Y}_{\mathrm{strm},n}$ and $\mathcal{N}_{\mathrm{strm},n}$ denote the distributions of **YES** and **NO** instances of $(\mathcal{D}_q^Y(\pi), \mathcal{D}^N, T)$-streaming-SD of length n. Given an instance $\boldsymbol{\sigma}$ of streaming-SD, which is a sequence $(\boldsymbol{\sigma}(0), \ldots, \boldsymbol{\sigma}(\alpha T n - 1))$ where each $\boldsymbol{\sigma}(i) = (\mathbf{j}(i), \mathbf{z}(i))$ with $\mathbf{j}(i) \in [n]^k$ and $\mathbf{z}(i) \in \{0,1\}$, the algorithm **ALG′** produces (a stream representing) an instance $\Psi(\boldsymbol{\sigma})$ of $\mathsf{Max\text{-}OCSP}(\Pi)$ with n variables. The variables of Ψ are $\mathbf{x} = (x_0, \ldots, x_{n-1})$, where $\mathbf{x} \in \mathsf{S}_n$, and the constraints $C_0, \ldots, C_{m-1}$ of Ψ are constructed as follows. For each $\boldsymbol{\sigma}(i) = (\mathbf{j}(i), \mathbf{z}(i))$ with $\mathbf{z}(i) \in \{0,1\}$, if $\mathbf{z}(i) = 1$ we add the constraint $\mathbf{j}(i)$ to Ψ, otherwise if $\mathbf{z}(i) = 0$, we don't add the corresponding constraint to Ψ. Observe that for $\boldsymbol{\sigma} \sim \mathcal{Y}_{\mathrm{strm},n}$, we have $\Psi(\boldsymbol{\sigma}) \sim \mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}(\Pi)$, and for $\boldsymbol{\sigma} \sim \mathcal{N}_{\mathrm{strm},n}$, we have

$\Psi(\boldsymbol{\sigma}) \sim \mathcal{G}^{N}_{q,n,\alpha,T}(\Pi)$, where $\mathcal{G}^{Y,\boldsymbol{\pi}}_{q,n,\alpha,T}(\Pi)$ and $\mathcal{G}^{N}_{q,n,\alpha,T}(\Pi)$ are the distributions on the instances of Max-OCSP(Π) that were defined in Definition 3.5. $\mathbf{ALG}'$ now runs $\mathbf{ALG}$ on Ψ and outputs what $\mathbf{ALG}$ outputs. It is straightforward to see that $\mathbf{ALG}'$ achieves the same advantage as $\mathbf{ALG}$, thus proving the lemma. □

References

[CGSV21] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, and Santhoshini Velusamy. Approximability of all finite CSPs in the dynamic streaming setting. In *ECCC*, 3rd May 2021.

[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.

[GT19] Venkatesan Guruswami and Runzhou Tao. Streaming hardness of unique games. In *APPROX 2019*, pages 5:1–5:12. LIPIcs, 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.

[KK19] Michael Kapralov and Dmitry Krachun. An optimal space lower bound for approximating MAX-CUT. In *STOC 2019*, pages 277–288. ACM, 2019.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il