

STREAMING APPROXIMATION RESISTANCE OF EVERY ORDERING CSP

NOAH G. SINGER, MADHU SUDAN,
AND SANTHOSHINI VELUSAMY

Abstract. An ordering constraint satisfaction problem (OCSP) is defined by a family \mathcal{F} of predicates mapping permutations on $\{1, \dots, k\}$ to $\{0, 1\}$. An instance of $\text{Max-OCSP}(\mathcal{F})$ on n variables consists of a list of constraints, each consisting of a predicate from \mathcal{F} applied on k distinct variables. The goal is to find an ordering of the n variables that maximizes the number of constraints for which the induced ordering on the k variables satisfies the predicate. OCSPs capture well-studied problems including ‘maximum acyclic subgraph’ (MAS) and “maximum betweenness”. In this work, we consider the task of approximating the maximum number of satisfiable constraints in the (single-pass) streaming setting, when an instance is presented as a stream of constraints. We show that for every \mathcal{F} , $\text{Max-OCSP}(\mathcal{F})$ 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 $\epsilon > 0$, MAS is not $(1/2 + \epsilon)$ -approximable in $o(n)$ space. The previous best inapproximability result, due to Guruswami & Tao (2019), only ruled out $3/4$ -approximations in $o(\sqrt{n})$ space. Our results build on recent works of Chou *et al.* (2022b, 2024) who provide a tight, linear-space inapproximability theorem for a broad class of “standard” (i.e., non-ordering) constraint satisfaction problems (CSPs) over arbitrary (finite) alphabets. Our results are obtained by building a family of appropriate standard CSPs (one for every alphabet size q) from any given OCSP and applying their theorem to

this family of CSPs. To convert the resulting hardness results for standard CSPs back to our OCSP, we show that the hard instances from this earlier theorem have the following “partition expansion” property with high probability: For every partition of the n variables into small blocks, for most of the constraints, all variables are in distinct blocks.

Keywords. Streaming algorithms, Approximation resistance, Constraint satisfaction problems, Ordering constraints

1. Introduction

In this work, we consider the complexity of “approximating” “ordering constraint satisfaction problems (OCSPs)” in the “streaming model”. 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 Traveling Salesperson Problem and most forms of scheduling problems fit this description, though our work considers a more concrete class of problems, namely *ordering constraint satisfaction problems (OCSPs)*. OCSPs as a class were first defined by [Guruswami et al. \(2011\)](#). To describe them here, we first set up some notation and terminology and then give some examples.

We let $[n]$ denote the set $\{1, \dots, n\}$ and \mathfrak{S}_n denote the space of all permutations in $[n]^n$, i.e.,

$$\mathfrak{S}_n \stackrel{\text{def}}{=} \{\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_n) \in [n]^n : \forall i \neq j, \sigma_i \neq \sigma_j\}.$$

We interpret each element $\boldsymbol{\sigma} \in \mathfrak{S}_n$ as a schedule for n tasks, labeled $1, \dots, n$, such that task i is scheduled in position σ_i . We use bold type to denote vectors (e.g., $\boldsymbol{\sigma}$), parenthetical indices for sequences of vectors (e.g., $\boldsymbol{\sigma}(1), \dots, \boldsymbol{\sigma}(m)$), and normal type to denote scalar entries (e.g., σ_i).

Given k distinct integers a_1, \dots, a_k , we define $\text{ord}(a_1, \dots, a_k) \in \mathfrak{S}_k$ as the unique $\boldsymbol{\pi} = (\pi_1, \dots, \pi_k) \in \mathfrak{S}_k$ such that $\pi_i < \pi_j$ iff $a_i < a_j$ for all $i \neq j$. If a_1, \dots, a_k are not all distinct, we write $\text{ord}(a_1, \dots, a_k) = \perp$, and thus, we can view ord as a map $\mathbb{Z}^k \rightarrow \mathfrak{S}_k \cup \{\perp\}$.

$\{\perp\}$. Given $\sigma = (\sigma_1, \dots, \sigma_n) \in \mathfrak{S}_n$ and k indices $\mathbf{j} = (j_1, \dots, j_k) \in [n]^n$, we let $\sigma|_{\mathbf{j}}$ denote $(\sigma_{j_1}, \dots, \sigma_{j_k}) \in [n]^n$.

The solution space of OCSPs is precisely \mathfrak{S}_n . A *k-ary ordering constraint predicate* is a function $\Pi : \mathfrak{S}_k \rightarrow \{0, 1\}$. An *ordering constraint application* (Π, \mathbf{j}) on n variables is given by a predicate Π and a k -tuple $\mathbf{j} \in [n]^n$ of distinct indices, and (Π, \mathbf{j}) is *satisfied* by an assignment $\sigma \in \mathfrak{S}_n$ iff $\Pi(\text{ord}(\sigma|_{\mathbf{j}})) = 1$. In the interest of brevity, we will often skip the term “ordering” below and further refer to constraint predicates as “predicates” and constraint applications as “constraints”.

A *maximum ordering constraint satisfaction problem*, denoted $\text{Max-OCSP}(\mathcal{F})$, is specified by a (finite) family of ordering constraint predicates $\mathcal{F} \subseteq \bigcup_{k \in \mathbb{N}} \{\Pi : \mathfrak{S}_k \rightarrow \{0, 1\}\}$. An *instance* of $\text{Max-OCSP}(\mathcal{F})$ on n variables is given by m constraints C_1, \dots, C_m where $C_i = (\Pi_i, \mathbf{j}(i))$ and $\Pi_i \in \mathcal{F}$. (We will typically specialize to the case where the family \mathcal{F} contains only a single predicate Π ; in this case, we write the problem as $\text{Max-OCSP}(\Pi)$ and omit Π from constraint descriptions. We’ll see below that for proving inapproximability results, it is sufficient to consider this case; see the Remark in [Section 1.2](#) and the proof of [Theorem 1.2](#) in [Section 3](#).) The *value* of an ordering $\sigma \in \mathfrak{S}_n$ on the instance Ψ , denoted $\text{ocsp-val}_{\Psi}(\sigma)$, is the fraction of constraints satisfied by σ , i.e.,

$$\text{ocsp-val}_{\Psi}(\sigma) \stackrel{\text{def}}{=} \frac{1}{m} \sum_{i \in [m]} \Pi_i(\text{ord}(\sigma|_{\mathbf{j}(i)})).$$

The optimal value of Ψ is defined as

$$\text{ocsp-val}_{\Psi} \stackrel{\text{def}}{=} \max_{\sigma \in \mathfrak{S}_n} \{\text{ocsp-val}_{\Psi}(\sigma)\}.$$

The canonical problem that fits the Max-OCSP framework is the *maximum acyclic subgraph* (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 that maximizes 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* backward edges); however, [Karp \(1972\)](#), in his famous list

of 21 **NP**-complete problems, proved the **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, **MAS** can be viewed as a 2-ary OCSP $\mathbf{MAS} = \mathbf{Max-OCSP}(\Pi_{\mathbf{MAS}})$, where $\Pi_{\mathbf{MAS}} : \mathfrak{S}_2 \rightarrow \{0, 1\}$ denotes the predicate given by $\Pi_{\mathbf{MAS}}(1, 2) = 1$ and $\Pi_{\mathbf{MAS}}(2, 1) = 0$, and we associate vertices with variables and edges with constraints. Indeed, a constraint (j_1, j_2) (where $j_1, j_2 \in [n]$ are distinct variables) will be satisfied by an ordering $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_n) \in \mathfrak{S}_n$ iff $\Pi_{\mathbf{MAS}}(\text{ord}(\boldsymbol{\sigma}|_{(j_1, j_2)})) = 1$, or equivalently, iff $\sigma_{j_1} < \sigma_{j_2}$. In the “scheduling” interpretation of OCSPs, a constraint (j_1, j_2) expresses “precedence” of event j_1 over event j_2 , since it is satisfied iff j_1 is scheduled before j_2 .

A second natural Max-OCSP problem is the *maximum betweenness* (**Max-Btwn**) problem. This is a 3-ary OCSP in which an ordering $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_n)$ satisfies a constraint (j_1, j_2, j_3) iff σ_{j_2} is between σ_{j_1} and σ_{j_3} , i.e., iff $\sigma_{j_1} < \sigma_{j_2} < \sigma_{j_3}$ or $\sigma_{j_1} > \sigma_{j_2} > \sigma_{j_3}$, and the goal is again to find the maximum number of satisfiable constraints. Thus, $\mathbf{Max-Btwn} = \mathbf{Max-OCSP}(\Pi_{\mathbf{Btwn}})$ where we define the constraint predicate $\Pi_{\mathbf{Btwn}} : \mathfrak{S}_3 \rightarrow \{0, 1\}$ by $\Pi_{\mathbf{Btwn}}(1, 2, 3) = 1$, $\Pi_{\mathbf{Btwn}}(3, 2, 1) = 1$, and $\Pi_{\mathbf{Btwn}}(\boldsymbol{\pi}) = 0$ for all other $\boldsymbol{\pi} \in \mathfrak{S}_3$. The complexity **Max-Btwn** was originally studied by [Opatrny \(1979\)](#), who proved that even deciding whether a set of betweenness constraints is perfectly satisfiable (i.e., whether the value of an instance Ψ is 1) is **NP**-complete.

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

Given an ordering predicate $\Pi : \mathfrak{S}_k \rightarrow \{0, 1\}$, let $\rho(\Pi) = \frac{|\{\boldsymbol{\pi} \in \mathfrak{S}_k : \Pi(\boldsymbol{\pi}) = 1\}|}{k!}$ denote the probability that Π is satisfied by a random ordering. Given a (finite) family of predicates \mathcal{F} , let $\rho(\mathcal{F}) =$

$\min_{\Pi \in \mathcal{F}} \{\rho(\Pi)\}$. Every instance Ψ of $\text{Max-OCSP}(\mathcal{F})$ satisfies

$$\text{ocsp-val}_{\Psi} \geq \rho(\mathcal{F})$$

(since the right-hand side is a lower bound on the expected value of a random assignment). Thus, the trivial algorithm that always outputs $\rho(\mathcal{F})$ is a $\rho(\mathcal{F})$ -approximation algorithm for $\text{Max-OCSP}(\mathcal{F})$. Under what conditions it is possible to beat this “trivial” approximation is a major open question.

REMARK 1.1. We define $\rho(\mathcal{F}) = \min_{\Pi \in \mathcal{F}} \{\rho(\Pi)\}$ to be the “trivial” approximability threshold for $\text{Max-OCSP}(\mathcal{F})$ because for every $\epsilon > 0$ there are instances of $\text{Max-OCSP}(\mathcal{F})$ with value at most $\rho(\mathcal{F}) + \epsilon$. This is a consequence, for instance, of [Lemma 3.4](#), which holds a priori for the single-predicate case $|\mathcal{F}| = 1$, but can be extended to general finite families \mathcal{F} by taking the minimum over $\Pi \in \mathcal{F}$ of $\rho(\Pi)$, since every instance of $\text{Max-OCSP}(\Pi)$ is also an instance of $\text{Max-OCSP}(\mathcal{F})$ with the same value.

A problem is said to be *approximation-resistant* with respect to a given class of algorithms if the trivial algorithm is essentially the best. Specifically for $\text{Max-OCSP}(\mathcal{F})$, we say it is approximation-resistant for a class of algorithms if for every $\epsilon > 0$, no algorithm in the class $(\rho(\mathcal{F}) + \epsilon)$ -approximates $\text{Max-OCSP}(\mathcal{F})$. Ordering CSP problems were shown to be approximation-resistant with respect to the class of polynomial-time algorithms by [Guruswami et al. \(2011\)](#) (assuming the unique games conjecture (UGC) of [\(Khot 2002\)](#)). In this work, we consider the analogous question with respect to “sublinear-space streaming algorithms”, which we define next.

1.3. Streaming algorithms. A (single-pass) streaming algorithm for OCSPs is defined as follows. In $\text{Max-OCSP}(\mathcal{F})$, an instance Ψ is presented as a stream (C_1, \dots, C_m) , where each stream element is a constraint $C_i = (\Pi_i, \mathbf{j}(i))$. A streaming algorithm Alg updates its state with each element of the stream and at the end produces an output $\text{Alg}(\Psi) \in [0, 1]$ (which is supposed to estimate ocsp-val_{Ψ}). The measure of complexity of interest to us is the space used by Alg measured as a function of n , the number of variables in

Ψ . (This is a somewhat non-standard choice in general but standard in the CSP literature. This choice is consistent with that of measuring the complexity of graph algorithms as a function of the number of vertices in the input instance.) In particular, we distinguish between algorithms that use space polylogarithmic in n and space that grows polynomially ($\Omega(n^\delta)$ for $\delta > 0$). (Note that for this coarse level of distinction, measuring space as a function of n or as a function of the input length would be qualitatively equivalent. However, our main result is more detailed and tight up to polylogarithmic factors when viewed as a function of n .)

We say that a problem $\text{Max-OCSP}(\mathcal{F})$ is *(streaming) approximable* if we can beat the trivial $\rho(\mathcal{F})$ -approximation algorithm by a positive constant factor. Specifically, $\text{Max-OCSP}(\mathcal{F})$ is said to be approximable if for every $\delta > 0$ there exists $\epsilon > 0$ and a space $O(n^\delta)$ algorithm that $(\rho(\mathcal{F}) + \epsilon)$ -approximates $\text{Max-OCSP}(\mathcal{F})$. We say $\text{Max-OCSP}(\Pi)$ is *(streaming) approximation-resistant* 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 (Chou *et al.* 2022a,b, 2021, 2024, 2020; Guruswami & Tao 2019; Guruswami *et al.* 2017; Kapralov *et al.* 2015, 2017; Kapralov & Krachun 2019; Kogan & Krauthgamer 2015; Saxena *et al.* 2023a,b). Most of these papers study approximability, not of *ordering* CSPs, but of “standard” CSPs where the variables can take values in a finite alphabet. ((Guruswami *et al.* 2017) and (Guruswami & Tao 2019) are the exceptions, and we will discuss them below.) Single-pass streaming algorithms with subpolynomial space are not formally a subclass of polynomial-time algorithms.¹ However, as far as we are aware, all known sublinear-space streaming algorithms for CSP approximation can be implemented as polynomial-time (often even linear-time!) algorithms. Indeed, there is essentially only one family of techniques for achieving nontrivial approximation ratios for CSPs via sublinear-space streaming algorithms, namely algorithms counting “biases” (see (Chou *et al.* 2021, 2020; Guruswami *et al.* 2017; Saxena *et al.* 2023a)), and these algorithms

¹The models are incomparable in the $\Omega(n^\delta)$ space regime since the streaming model has no time complexity or uniformity assumptions.

can be implemented as linear-time classical algorithms and only give nontrivial guarantees for small classes of CSPs. In particular, [Saxena *et al.* \(2023a\)](#) showed that the **Max-2AND** problem is 0.483-approximable in the streaming setting (whereas the trivial approximation is a $\frac{1}{4}$ -approximation). For more background on CSPs in the streaming model, see the surveys of [Velusamy \(2023\)](#), [Singer \(2022\)](#), and [Sudan \(2022\)](#).

However, this success in obtaining non-trivial algorithms has not extended to any OCSP problem. Indeed, given the known (UGC-)hardness of OCSPs with respect to polynomial-time algorithms ([Guruswami *et al.* 2011](#)), and the empirically observed phenomenon that subpolynomial space streaming algorithms are linear time simulatable, it would be extremely surprising to find a non-trivial approximation algorithm for an OCSP using subpolynomial space. This work confirms this expectation formally, and unconditionally, by showing that there are no non-trivial sublinear (in n) space streaming algorithms for approximating OCSPs.

1.4. Results. In this paper, we prove the following theorem:

THEOREM 1.2 (Main theorem). *For every (finite) family of ordering predicates \mathcal{F} , **Max-OCSP**(\mathcal{F}) is approximation-resistant (to single-pass streaming algorithms). In particular, for every $\epsilon > 0$, every $(\rho(\mathcal{F}) + \epsilon)$ -approximation algorithm for **Max-OCSP**(\mathcal{F}) requires $\Omega(n)$ space.*

In particular, for every $\epsilon > 0$, **MAS** is not $(1/2 + \epsilon)$ -approximable and **Max-Btwn** is not $(1/3 + \epsilon)$ -approximable. [Theorem 1.2](#) is proved in [Section 3](#), modulo several technical lemmas proven in later sections.

The linear-space bound in [Theorem 1.2](#) is optimal, up to logarithmic factors:

THEOREM 1.3 ($\tilde{O}(n)$ -space algorithm). *Let \mathcal{F} denote any (finite) family of ordering predicates. For all $c > 0$ and $\epsilon > 0$, there exists a single-pass streaming algorithm **Alg** which, given an instance Ψ of **Max-OCSP**(\mathcal{F}) with n variables and $m \leq n^c$ constraints, outputs a $(1 - \epsilon)$ -approximation to **ocsp-val** $_{\Psi}$ in $O(n \log^3 n / \epsilon^2)$ bits of space.*

The algorithm in [Theorem 1.3](#) is the analogue for OCSPs of a well-known algorithm in the setting of streaming CSPs (see, ([Chou et al. 2022b](#); [Kapralov & Krachun 2019](#); [Kogan & Krauthgamer 2015](#))): It simply sparsifies the input instance down to $\tilde{O}(n/\epsilon^2)$ constraints and then solves the $\text{Max-OCSP}(\mathcal{F})$ problem exactly on the sparsified instance. For completeness, we prove [Theorem 1.3](#) in [Appendix A](#).

1.5. Related works. As far as we know, in the streaming setting, [Theorem 1.2](#) is the first tight inapproximability result for $\text{Max-OCSP}(\mathcal{F})$ for *any* constraint family \mathcal{F} in $\Omega(n^\delta)$ space for any $\delta > 0$, and it yields tight approximability results for *every* family in *linear* space.

[Theorem 1.2](#) parallels the classical result of [Guruswami et al. \(2011\)](#), who prove that $\text{Max-OCSP}(\Pi)$ is approximation-resistant with respect to *polynomial-time* algorithms, for every Π , assuming the unique games conjecture.² 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 et al. \(2017\)](#) proved that for every $\epsilon > 0$, **MAS** is not $(7/8 + \epsilon)$ -approximable in $o(\sqrt{n})$ space using a gadget reduction from the Boolean hidden matching problem ([Gavinsky et al. 2008](#)). [Guruswami & Tao \(2019\)](#) indicated that $3/4$ -approximating **MAS** is hard in $o(\sqrt{n})$ space. Their proof first establishes $o(\sqrt{n})$ -space approximation resistance for a (non-ordering) CSP called **Max-UniqueGames** and then reduces from **Max-UniqueGames** to **MAS**, though this reduction is not fully analyzed.

[Chakrabarti et al. \(2020\)](#) recently also studied directed graph

²Without relying on the unique games conjecture, some weaker **NP**-hardness results are known. For **Max-Btwn**, since $\rho(\Pi_{\text{Btwn}}) = \frac{1}{3}$, the trivial algorithm is a $\frac{1}{3}$ -approximation. [Chor & Sudan \(1998\)](#) showed that $(\frac{47}{48} + \epsilon)$ -approximating **MaxBtwn** is **NP**-hard, for every $\epsilon > 0$. The $\frac{47}{48}$ factor was improved to $\frac{1}{2}$ by [Austrin et al. \(2015\)](#). For **MAS**, the trivial algorithm is a $\frac{1}{2}$ -approximation. [Newman \(2000\)](#) showed that $(\frac{65}{66} + \epsilon)$ -approximating **MAS** is **NP**-hard, for every $\epsilon > 0$. [Austrin et al. \(2015\)](#) improved the $\frac{65}{66}$ to $\frac{14}{15}$, and [Bhangale & Khot \(2019\)](#) further improved the factor to $\frac{2}{3}$.

ordering problems (e.g., acyclicity testing, (s, t) -connectivity, topological sorting) in the streaming setting. For the problems they consider, they give *super-linear* space lower bounds even for multi-pass streaming algorithms. In contrast, as we mentioned above, every OCSP can be approximated arbitrarily well by simple $\tilde{O}(n)$ -space algorithms, even in a single pass. However, one of the problems considered by Chakrabarti *et al.* (2020) is close enough to **MAS** to merit a more detailed comparison: 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 the **MFAS** value of a graph and the **MAS** value of the graph is exactly one. Chakrabarti *et al.* (2020) proved that for every $\kappa > 1$, κ -approximating³ 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). We remark that, as is typical for pairs of minimization and maximization problems defined in this way (that is, such that the values sum to 1), the hard instances involved in proving optimal inapproximability are incomparable: In particular, proving κ -inapproximability for **MFAS** involves constructing indistinguishable (distributions of) instances with **MAS** values $\approx 1 - \epsilon$ vs. $\approx 1 - \kappa\epsilon$ and thus does not imply any hardness of approximation for **MAS**.

A recent work of Chatziafratis & Makarychev (2023) studies another variant of CSPs, called *phylogenetic CSPs*, where the solution space is a set of *trees*,⁴ analogously to how for ordering CSPs, the solution space was a set of permutations. The authors prove approximation resistance of phylogenetic CSPs against polynomial-time algorithms assuming the UGC, and their proof is via value-preserving reductions from ordering CSPs, which were

³For minimization problems, a κ -approximation to a value v is in the interval $[v, \kappa v]$. Thus approximation factors are larger than 1.

⁴Given n variables, the ordering space to a phylogenetic CSP consists of rooted binary, or more generally k -ary, trees with n labeled leaves, and constraints specify “hierarchical structure” among the leaves. For instance, in the “triple reconstruction” problem, the constraint (a, b, c) expresses that a and b are closer to each other than either is to c , where distance is measured as path length in the tree. See (Chatziafratis & Makarychev 2023, §3,§9) for full definitions.

themselves proven approximation-resistant under the UGC by [Guruswami *et al.* \(2011\)](#). It would be interesting to see whether the techniques in the current work could help prove streaming approximation lower bounds for phylogenetic CSPs or other CSP variants.

1.6. Techniques. Our general approach to prove hardness of Max-OCSP problems is the following: We choose a family of (standard) CSPs where hardness results are known and then reduce these CSPs to the OCSPs at hand. While this general approach is not new, in order to achieve optimal streaming hardness results for OCSPs, we need to choose the “source” CSPs carefully, so that we can *both* (i) apply previously known optimal streaming hardness results for these CSPs (in our setting, we use results due to [Chou, Golovnev, Sudan, Velingker & Velusamy \(2022b\)](#)) and (ii) design *streaming* reductions from these CSPs to Max-OCSPs which produce instances with (almost) optimal ratios in value between **YES** and **NO** instances. In contrast, previous approaches ([Guruswami & Tao 2019; Guruswami *et al.* 2017](#)) toward proving hardness of Max-OCSPs (in particular **MAS**) were unable to achieve optimal streaming hardness results *despite* starting with optimal hardness results for the source CSP **Max-UniqueGames**, because of issues in designing streaming reductions which produce sufficiently large value gaps. In the remainder of this section, we describe and motivate this approach toward proving the approximation resistance of Max-OCSPs.

1.6.1. Special case: The intuition for **MAS.** We start by describing our proof technique for the special case of the **MAS** problem. Let $+_q$ denote the modular addition operator on $[q] = \{1, \dots, q\}$: For $a, b \in [q]$, $a +_q b$ denotes the unique $c \in [q]$ such that $a + b \equiv c \pmod{q}$. Thus, for instance, $1 +_q q = 1$.

Similarly to earlier work in the setting of streaming approximability (e.g., the work of [Kapralov *et al.* \(2015\)](#)), we prove inapproximability of **MAS** by exhibiting a pair of distributions over **MAS** instances, which we denote \mathcal{Y} (the “**YES** instances”) and \mathcal{N} (the “**NO** instances”), satisfying the following two properties:

1. \mathcal{Y} and \mathcal{N} are “indistinguishable” to streaming algorithms (in a sense we define formally below).

2. With high probability, \mathcal{Y} has high **MAS** values (≈ 1) and \mathcal{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 construct distributions \mathcal{Y} and \mathcal{N} satisfying these properties?

The “recipe” which has proved successful in past works for proving streaming approximation resistance for “standard” CSPs is roughly to let the \mathcal{N} instances be completely random, while \mathcal{Y} instances are sampled with “hidden structure” which guarantees a very good assignment. Then, 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 (Gavinsky *et al.* 2008; Verbin & Yu 2011)). In the OCSP setting, we might hope that the hidden structure could simply be an ordering; that is, we could hope to define \mathcal{Y} by first sampling a random ordering of the variables, then sampling constraints that go forward with respect to this ordering, and then perhaps adding some noise. But unfortunately, we don’t know how to directly prove communication lower bounds for such problems.

Hence, instead of seeking to prove indistinguishability directly, we turn back to earlier streaming hardness-of-approximation results proven in the context of standard CSPs. In this setting, variables take on values in a finite alphabet $[q]$ (i.e., the solution space is $[q]^n$), and k -ary predicates $f : [q]^k \rightarrow \{0, 1\}$ can be applied to small subsets of variables to form constraints. We make two observations about this definition. Firstly, in a CSP, two variables may be assigned the same value in $[q]$, whereas in an OCSP, every variable must get a distinct value in $[n]$. Secondly, for a CSP or OCSP defined by a single binary predicate, each constraint simply specifies a pair (j_1, j_2) of distinct indices in $[n]$; by extension, instances can be viewed equivalently as directed graphs on $[n]$ (allowing multiple edges). Thus, we can view instances of binary CSPs as instances of **MAS**, and vice versa.

The plan is as follows. We’ll define a binary predicate denoted $\Pi_{\text{MAS}}^{\downarrow q} : [q]^2 \rightarrow \{0, 1\}$. Let **Max-CSP**($\Pi_{\text{MAS}}^{\downarrow q}$) denote the problem of

maximizing $\Pi_{\text{MAS}}^{\downarrow q}$ constraints applied to assignments in $[q]^n$. The hope is that for a careful choice of the alphabet size q and the predicate $\Pi_{\text{MAS}}^{\downarrow q}$, we can *reuse* indistinguishable **YES/NO** distributions for $\text{Max-CSP}(\Pi_{\text{MAS}}^{\downarrow q})$ — in particular, those constructed in the recent work of [Chou, Golovnev, Sudan, Velingker & Velusamy \(2022b\)](#) — as **YES/NO** distributions for **MAS**. This requires us to relate the values of an **MAS** instance and the corresponding $\text{Max-CSP}(\Pi_{\text{MAS}}^{\downarrow q})$ instance. To be precise, for an **MAS** instance Ψ , let $\Psi^{\downarrow q}$ denote the $\text{Max-CSP}(\Pi_{\text{MAS}}^{\downarrow q})$ instance with the exact same list of constraints and let $\text{csp-val}_{\Psi^{\downarrow q}}$ denote the value of this instance. We choose q and $\Pi_{\text{MAS}}^{\downarrow q}$ so as to imply the following four properties about the indistinguishable distributions \mathcal{Y} and \mathcal{N} “given to us” by [Chou *et al.* \(2022b\)](#):

1. With high probability over $\Psi \sim \mathcal{Y}$, $\text{csp-val}_{\Psi^{\downarrow q}} \approx 1$.
2. With high probability over $\Psi \sim \mathcal{N}$, $\text{csp-val}_{\Psi^{\downarrow q}} \approx \frac{1}{2}$.
3. For all Ψ , $\text{ocsp-val}_{\Psi} \geq \text{csp-val}_{\Psi^{\downarrow q}}$.
4. With high probability over $\Psi \sim \mathcal{N}$, ocsp-val_{Ψ} is not much larger than $\text{csp-val}_{\Psi^{\downarrow q}}$.

Together, these items will suffice to prove the theorem since [2](#) and [4](#) together imply that with high probability over $\Psi \sim \mathcal{N}$, $\text{ocsp-val}_{\Psi} \approx \frac{1}{2}$, while [1](#) and [3](#) together imply that with high probability over $\Psi \sim \mathcal{Y}$, $\text{ocsp-val}_{\Psi} \approx 1$.

In order to satisfy these criteria, we define the CSP predicate as follows. Recall that $\Pi_{\text{MAS}}(1, 2) = 1$ while $\Pi_{\text{MAS}}(2, 1) = 0$. We define the constraint predicate $\Pi_{\text{MAS}}^{\downarrow q} : [q]^2 \rightarrow \{0, 1\}$ by $\Pi_{\text{MAS}}^{\downarrow q}(b_1, b_2) = 1$ iff $b_1 < b_2$. We call this the *q-coarsening* of Π_{MAS} , and it gives $\text{Max-CSP}(\Pi_{\text{MAS}}^{\downarrow q})$ the following “scheduling” interpretation. Recall, the goal of **MAS** is to schedule n tasks, each task i is assigned a distinct position $\sigma_i \in [n]$ in the schedule, and the goal is to maximize constraints of the form (j_1, j_2) requiring that task j_1 takes place before task j_2 , i.e., $\sigma_{j_1} < \sigma_{j_2}$. In $\text{Max-CSP}(\Pi_{\text{MAS}}^{\downarrow q})$, the goal is to schedule n tasks in q *batches*: Each task i receives a (not necessarily distinct!) batch $\sigma_i \in [q]$, and constraints (j_1, j_2) still require

that $\sigma_{j_1} < \sigma_{j_2}$, that is, j_1 ’s assigned batch is earlier than j_2 . 3 follows immediately in this interpretation: Given any batched schedule $\mathbf{b} \in [q]^n$, we can immediately “lift” to a non-batched schedule $\mathbf{b}^\uparrow \in \mathfrak{S}_n$ by arbitrarily ordering the tasks in each batch, which can only increase the number of satisfied constraints.

Proving 4 is the meat of the argument. Note that if we set $q = n$, $\mathbf{Max-CSP}(\Pi_{\mathbf{MAS}}^{\downarrow q})$ becomes the same problem as **MAS**, and hence, 4 is trivial! However, we can only apply the inapproximability results of [Chou et al. \(2022b\)](#) (specifically, the indistinguishability of their distributions \mathcal{Y} and \mathcal{N}) when q is a constant. Briefly, the [\(Chou et al. 2022b\)](#) results roughly state that a predicate $f : [q]^k \rightarrow \{0, 1\}$ is inapproximable when its support satisfies a property that they call *width*: it contains most of a “diagonal”, in the sense that for some $\mathbf{a} \in [q]^k$, the set $\{c \in [q] : f(\mathbf{a} +_q (c, \dots, c)) = 1\}$ is large. Luckily for us, $\Pi_{\mathbf{MAS}}^{\downarrow q}$ has this property with $\mathbf{a} = (1, 2)$; indeed, $\Pi_{\mathbf{MAS}}^{\downarrow q}(1 +_q c, 2 +_q c) = 1$ unless $c = q - 1$ (in which case $1 +_q c = q$ while $2 +_q c = 1$).

To actually prove 4, then, we can no longer use the results of [Chou et al. \(2022b\)](#) as a black box. Specifically, we need to understand the structure of the **NO** distribution \mathcal{N} (beyond 2 and its indistinguishability from \mathcal{Y}). We show that instances drawn from \mathcal{N} are (with high probability) “small partition expanders” in a specific sense: for every partition of the set of variables into q blocks of roughly equal size, very few constraints, specifically a $o(1)$ fraction, involve two variables in the same block. (See [Definition 4.8](#).) Now, we think of a “schedule” $\boldsymbol{\sigma} \in \mathfrak{S}_n$ as giving rise to a “batched schedule” $\boldsymbol{\sigma}^{\downarrow q} \in [q]^n$ in the following way: If task i is scheduled in position $\sigma_i \in [n]$, then we place it in batch $\approx \sigma_i q / n$. Thus, the first $\approx n/q$ scheduled tasks are placed in batch 1, the next $\approx n/q$ in batch 2, etc. Hence, whenever a constraint (j_1, j_2) is satisfied by $\boldsymbol{\sigma}$ (as an **MAS** constraint), it will also be satisfied by $\boldsymbol{\sigma}^{\downarrow q}$ (as a $\mathbf{Max-CSP}(\Pi_{\mathbf{MAS}}^{\downarrow q})$ constraint), *unless* j_1 and j_2 end up in the same batch, but by the small partition expansion condition, this happens only for $o(1)$ fraction of the constraints. Hence, $\text{ocsp-val}_{\Psi} \leq \text{csp-val}_{\Psi^{\downarrow q}} + o(1)$.

1.6.2. Extending to general ordering CSPs. Extending the idea to other OCSPs follows the same basic outline. Given the

constraint predicate $\Pi : \mathfrak{S}_k \rightarrow \{0, 1\}$ (of arity k) and positive integer q , we define $\Pi^{\downarrow q} : [q]^k \rightarrow \{0, 1\}$ analogously to $\Pi_{\text{MAS}}^{\downarrow q} : \Pi^{\downarrow q}(\mathbf{b})$ is $\Pi(\text{ord}(\mathbf{b}))$ if $\text{ord}(\mathbf{b}) \neq \perp$ (i.e., \mathbf{b} 's entries are all distinct), and 0 otherwise. We then describe the **YES** and **NO** distributions of $\text{Max-CSP}(\Pi^{\downarrow q})$ which the general theorem of [Chou et al. \(2022b\)](#) shows are indistinguishable to $o(n)$ space algorithms, again taking advantage of the fact that Π^q 's support mostly contains a diagonal. Finally, we give an analysis of the partition expansion in the **NO** instances arising from the construction in [\(Chou et al. 2022b\)](#). Specifically, we show that the instances are now a “small partition hypergraph expander”, in the sense that for every partition of the n variables into q blocks of roughly equal size, very few constraints involve even two vertices from the same block.

1.6.3. Further remarks. Our notion of coarsening is somewhat similar to, but not the same as, that used in previous works, notably [\(Guruswami et al. 2011\)](#). In particular, the techniques used to compare the OCSP value (before coarsening) with the standard CSP value (after coarsening) are somewhat different: Their analysis involves more sophisticated tools such as influence of variables and Gaussian noise stability. The proof in our setting, in contrast, uses a more elementary analysis of the type common with random graphs.

In the rest of the paper, in the interest of self-containedness, we will avoid invoking the work of [Chou et al. \(2022b\)](#) on linear-space streaming CSP inapproximability where possible. Instead, we will explicitly define the distributions \mathcal{Y} and \mathcal{N} over $\text{Max-OCSP}(\Pi)$ instances for arbitrary ordering predicates Π and analyze them directly, without invoking any prior analyses of their coarsened CSP values (which would require formally defining the notion of the “width” of predicates in [\(Chou et al. 2022b\)](#)). Hence, we'll only need to invoke [\(Chou et al. 2022b\)](#) in the context of using communication lower bounds to prove indistinguishability of \mathcal{Y} and \mathcal{N} . We also manage to prove a stronger statement about the coarsened \mathcal{Y} distribution (though it is unnecessary for our application): Its value is high with probability 1, as opposed to just $1 - o(1)$ (which would be implied by the analysis in [\(Chou et al. 2022b\)](#)).

Organization of the rest of the paper. In Section 2, we introduce some additional notation and background material. In Section 3, we introduce two distributions on $\text{Max-OCSP}(\Pi)$ instances, the **YES** distribution \mathcal{Y} and the **NO** distribution \mathcal{N} ; 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, and then prove Theorem 1.2 modulo these lemmas. Finally, 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. Additional notation. The *support* of an ordering constraint predicate $\Pi : \mathfrak{S}_k \rightarrow \{0, 1\}$ is the set $\text{supp}(\Pi) = \{\boldsymbol{\pi} \in \mathfrak{S}_k : \Pi(\boldsymbol{\pi}) = 1\}$.

We first define a notion of “ k -hypergraphs”. (These are k -uniform ordered hypergraphs with multiple hyperedges and without self-loops.) Given a finite set V , an (ordered, self-loop-free) k -hyperedge $\mathbf{j} = (j_1, \dots, j_k)$ is a sequence of k distinct elements $j_1, \dots, j_k \in V$. We stress that the ordering of vertices within an edge is important to us. An k -hypergraph $G = (V, E)$ is given by a set of vertices V and a multiset $E \subseteq V^k$ of k -hyperedges on V . A k -hyperedge \mathbf{j} is *incident* on a vertex v if v appears in \mathbf{j} . Let $\Gamma(\mathbf{j}) \subseteq V$ denote the set of vertices to which a k -hyperedge \mathbf{j} 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. We let $\mathcal{M}_{k,a}(n)$ denote the uniform distribution over all k -hypermatchings on $[n]$ with a n edges.

A vector $\mathbf{b} = (b_1, \dots, b_n) \in [q]^n$ may be viewed as a q -partition of $[n]$ into blocks $\mathbf{b}^{-1}(1), \dots, \mathbf{b}^{-1}(q)$, where the i -th block $\mathbf{b}^{-1}(i)$ is defined as the set of indices $\{j \in [n] : b_j = i\}$. Given $\mathbf{b} = (b_1, \dots, b_n) \in [q]^n$ and an indexing vector $\mathbf{j} = (j_1, \dots, j_k) \in [n]^k$, we define $\mathbf{b}|\mathbf{j} = (b_{j_1}, \dots, b_{j_k}) \in [q]^k$.

Given an instance Ψ of $\text{Max-OCSP}(\Pi)$ on n variables, we define its *constraint hypergraph* $G(\Psi)$ to be the k -hypergraph on $[n]$

consisting of the k -hyperedge \mathbf{j} for each constraint (Π, \mathbf{j}) in Ψ . We also let $m(\Psi)$ denote the number of constraints in Ψ (equiv., the number of k -hyperedges in $G(\Psi)$).

2.2. Concentration bounds. We require the following Azuma-style concentration inequality for (not necessarily independent) Bernoulli random variables with bounded conditional expectations taken from [Kapralov & Krachun \(2019\)](#):

LEMMA 2.1 ([Kapralov & Krachun 2019](#), Lemma 2.5). *Let $0 < p < 1$. Let X_1, \dots, X_m be $\{0, 1\}$ -valued random variables such that for every $i \in [m]$, $\mathbb{E}[X_i | X_1, \dots, X_{i-1}] \leq p$. Then for every $\eta > 0$,*

$$\Pr \left[\sum_{i=1}^m X_i \geq (p + \eta)m \right] \leq \exp \left(- \left(\frac{\eta^2}{2(p + \eta)} \right) m \right).$$

We also require standard Chernoff bounds for sums of independent Bernoulli variables which we state here for completeness:

LEMMA 2.2 (Chernoff bounds). *Let $0 < p < 1$. Let X_1, \dots, X_m be $\{0, 1\}$ -valued random variables such that for every $i \in [m]$, $\mathbb{E}[X_i] = p$. Then,*

(i) *For every $\eta > 0$,*

$$\Pr \left[\sum_{i=1}^m X_i \leq (p - \eta)m \right] \leq \exp \left(- \left(\frac{\eta^2}{2p} \right) m \right).$$

(ii) *For all $\eta > 0$,*

$$\Pr \left[\left| \sum_{i=1}^m X_i - pm \right| \geq \eta m \right] \leq 2 \exp \left(- \left(\frac{\eta^2}{3p} \right) m \right).$$

(Note that the lower bounds in these lemmas are trivial if $\eta > p$.)

2.3. Stirling's approximation. Finally, we state a standard form of Stirling's bound for the factorial:

LEMMA 2.3 (Stirling approximation). *For all $n \in N$,*

$$\sqrt{2\pi n}(n/e)^n < n! < 2\sqrt{2\pi n}(n/e)^n.$$

3. The streaming space lower bound

In this section, we prove our main theorem (Theorem 1.2), modulo some lemmas that we prove in later sections. We focus first on the following special case for single-predicate families:

THEOREM 3.1 (Main theorem (single-predicate case)). *For every $k \in \mathbb{N}$ and every predicate $\Pi : \mathfrak{S}_k \rightarrow \{0, 1\}$, $\text{Max-OCSP}(\Pi)$ is approximation-resistant (to single-pass streaming algorithms). In particular, for every $\epsilon > 0$, every $(\rho(\Pi) + \epsilon)$ -approximation algorithm for $\text{Max-OCSP}(\Pi)$ requires $\Omega(n)$ space.*

Indeed, given Theorem 3.1, Theorem 1.2 follows immediately:

PROOF (Proof of Theorem 1.2). Given any family \mathcal{F} of predicates, let Π have minimal random assignment value ρ over predicates in \mathcal{F} , so that $\rho(\Pi) = \rho(\mathcal{F})$. Then, since every instance of $\text{Max-OCSP}(\Pi)$ is also an instance of $\text{Max-OCSP}(\mathcal{F})$, Theorem 3.1 immediately implies $(\rho(\mathcal{F}) + \epsilon)$ -approximations for $\text{Max-OCSP}(\mathcal{F})$ require $\Omega(n)$ space. \square

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 $1 - \epsilon/2$, from **NO** instances with OCSP value at most $\rho(\Pi) + \epsilon/2$. Such a statement is in turn proved by exhibiting two distributions, the **YES** distribution \mathcal{Y} and the **NO** distribution \mathcal{N} , and showing these are indistinguishable. Specifically, we carefully choose some parameters q, T, α and a permutation $\pi \in \mathfrak{S}_k$, and define two distributions $\mathcal{Y} = \mathcal{Y}_{q, \alpha, T}^{\Pi, \pi}(n)$ and $\mathcal{N} = \mathcal{N}_{q, \alpha, T}^{\Pi}(n)$ over n -variable instances of $\text{Max-OCSP}(\Pi)$. We claim that for our choice of parameters \mathcal{Y} is supported on instances with value at least $1 - \epsilon/2$ — this is asserted in Lemma 3.3 below. Similarly, we claim that \mathcal{N} is mostly supported (with probability $1 - o(1)$) on instances with value at most $\rho(\Pi) + \epsilon/2$ (see Lemma 3.4). Finally, we assert in Lemma 3.5 that any algorithm

that distinguishes \mathcal{Y} from \mathcal{N} with “advantage” at least $1/8$ (i.e., accepts $\Psi \sim \mathcal{Y}$ with probability $1/8$ more than $\Psi \sim \mathcal{N}$) requires $\Omega(n)$ space.

Assuming [Lemma 3.3](#), [Lemma 3.4](#), and [Lemma 3.5](#), the proof of [Theorem 3.1](#) is straightforward and given at the end of this section. We prove [Lemma 3.3](#) and [Lemma 3.4](#) in [Section 4](#) and [Lemma 3.5](#) in [Section 5](#).

3.1. Distribution of hard instances. We now formally define our **YES** and **NO** distributions for $\text{Max-OCSP}(\Pi)$.

DEFINITION 3.2 ($\mathcal{Y}_{q,\alpha,T}^{\Pi,\pi}(n)$ and $\mathcal{N}_{q,\alpha,T}^{\Pi}(n)$). For $k \in \mathbb{N}$ and $\Pi : \mathfrak{S}_k \rightarrow \{0, 1\}$, let $q, n, T \in \mathbb{N}$ and $\alpha > 0$, $q \geq k$, and let $\pi \in \text{supp}(\Pi)$. We define two distributions over $\text{Max-OCSP}(\Pi)$ instances with n variables, the **YES** distribution $\mathcal{Y}_{q,\alpha,T}^{\Pi,\pi}(n)$ and the **NO** distribution $\mathcal{N}_{q,\alpha,T}^{\Pi}(n)$, as follows:

- (i) Sample a uniformly random q -partition $\mathbf{b} = (b_1, \dots, b_n) \in [q]^n$.
- (ii) Sample T hypermatchings $\tilde{G}_1, \dots, \tilde{G}_T \sim \mathcal{M}_{k,\alpha}(n)$ independently.
- (iii) For each $t \in [T]$, do the following:
 - Let G_t be an empty k -hypergraph on $[n]$.
 - For each k -hyperedge $\mathbf{j} = (j_1, \dots, j_k) \in E(\tilde{G}_t)$:
 - **YES** case: If there exists $c \in [q]$ such that $\mathbf{b}|_{\mathbf{j}} = \pi +_q (c, \dots, c)$, add \mathbf{j} to G_t with probability $1/q$. (Here π is viewed as k -tuple in $[k]^k \subseteq [q]^k$.)
 - **NO** case: Add \mathbf{j} to G_t with probability $\frac{1}{q^k}$.
- (iv) Set $G \leftarrow G_1 \cup \dots \cup G_T$.
- (v) Return the $\text{Max-OCSP}(\Pi)$ instance Ψ on n variables given by the constraint hypergraph G .

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

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

We make several remarks on this definition. Firstly, note that the constraints within $\mathcal{Y}_{q,\alpha,T}^{\Pi,\pi}(n)$ and $\mathcal{N}_{q,\alpha,T}^{\Pi}(n)$ do not directly depend on Π . We still parameterize the distributions by Π , since they are formally distributions over $\text{Max-OCSP}(\Pi)$ instances; Π also determines the arity k and the set of allowed permutations π in the **YES** case. Secondly, we note that when sampling an instance from $\mathcal{N}_{q,\alpha,T}^{\Pi}(n)$, the partition \mathbf{b} is ignored, and so $\mathcal{N}_{q,\alpha,T}^{\Pi}(n)$ is “random”. Hence, these instances fit into the typical streaming lower bound “recipe” 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 3.1](#) for a visual interpretation of the distributions in the case of MAS.

3.2. Statement of key lemmas. Our first lemma shows that \mathcal{Y} is supported on instances of high value.

LEMMA 3.3 (\mathcal{Y} has high $\text{Max-OCSP}(\Pi)$ values). *For every ordering constraint predicate $\Pi : \mathfrak{S}_k \rightarrow \{0, 1\}$, every $\pi \in \text{supp}(\Pi)$ and $\Psi \sim \mathcal{Y}_{q,\alpha,T}^{\Pi,\pi}(n)$, we have $\text{ocsp-val}_\Psi \geq 1 - \frac{k-1}{q}$ (i.e., this occurs with probability 1).*

We prove [Lemma 3.3](#) in [Section 4.2](#). Next, we assert that \mathcal{N} is supported mostly on instances of low value.

LEMMA 3.4 (\mathcal{N} has low $\text{Max-OCSP}(\Pi)$ values). *For every ordering constraint predicate $\Pi : \mathfrak{S}_k \rightarrow \{0, 1\}$, and every $\epsilon > 0$, 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*

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

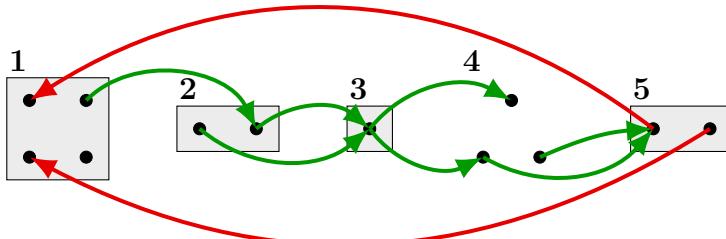
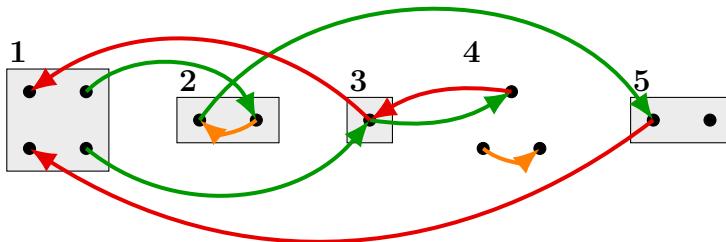
(a) Constraint graph of a sample MAS instance drawn from \mathcal{Y} (b) Constraint graph of a sample MAS instance drawn from \mathcal{N}

Figure 3.1: The constraint graphs of MAS instances which could plausibly be drawn from \mathcal{Y} and \mathcal{N} , respectively, for $q = 5$ and $n = 12$. Recall that MAS is a binary Max-OCSP with ordering constraint function Π_{MAS} supported only on $(1, 2)$. According to the definition of \mathcal{Y} (see [Definition 3.2](#), with $\pi = (1, 2)$), instances are sampled by first sampling a q -partition $\mathbf{b} = (b_1, \dots, b_n) \in [q]^n$, and then sampling some constraints; every sampled constraint (j_1, j_2) must satisfy $b_{j_2} \equiv b_{j_1} + 1 \pmod{q}$. On the other hand, there are no requirements on (b_{j_1}, b_{j_2}) for instances sampled from \mathcal{N} . Above, the blocks of the partition \mathbf{b} are labeled $1, \dots, 5$, 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 (j_1, j_2) green, orange, or red if $b_{j_2} > b_{j_1}$, $b_{j_2} = b_{j_1}$, or $b_{j_2} < b_{j_1}$, respectively. This visually suggests important elements of our proofs that \mathcal{Y} has MAS values close to 1 and \mathcal{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{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 green (i.e., all but those which go from block q to block 1). On the other hand, if we executed a similar process in \mathcal{N} , the resulting ordering would satisfy all green edges and some subset of the orange edges; however, in expectation, these account only for $\frac{q(q+1)}{2q^2} = \frac{q+1}{2q} \approx \frac{1}{2}$ fraction of the edges.

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{Y} and \mathcal{N} to small space streaming algorithms. We remark that this lemma follows directly from the work of [Chou et al. \(2022b\)](#), but we prove it for completeness in [Section 5](#).

LEMMA 3.5 (\mathcal{Y} and \mathcal{N} are indistinguishable). *For every $q \geq k \in \mathbb{N}$, 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 ordering constraint predicate $\Pi : \mathfrak{S}_k \rightarrow \{0, 1\}$ and $\pi \in \text{supp}(\Pi)$, every streaming algorithm distinguishing $\mathcal{Y}_{q, \alpha, T}^{\Pi, \pi}(n)$ from $\mathcal{N}_{q, \alpha, T}^{\Pi}(n)$ with advantage $1/8$ for all lengths n uses space $\Omega(n)$.*

3.3. Proof of [Theorem 3.1](#).

We now prove [Theorem 3.1](#).

PROOF (Proof of [Theorem 3.1](#)). Let Alg be an algorithm for $\text{Max-OCSP}(\Pi)$ that uses space $s(n)$ and achieves a $(\rho(\Pi) + \epsilon)$ -approximation. Fix $\pi \in \text{supp}(\Pi)$. Consider the algorithm Alg' defined as follows: on input Ψ , an instance of $\text{Max-OCSP}(\Pi)$, if $\text{Alg}(\Psi) \geq \rho(\Pi) + \frac{\epsilon}{2}$, then Alg' outputs 1, else, it outputs 0. Observe that Alg' uses $O(s(n))$ space. Set $q_0 \geq \frac{2(k-1)}{\epsilon}$ 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 $q \geq \frac{2(k-1)}{\epsilon}$, it follows from [Lemma 3.3](#) that for $\Psi \sim \mathcal{Y}_{q, \alpha, T}^{\Pi, \pi}(n)$, we have $\text{ocsp-val}_\Psi \geq 1 - \frac{\epsilon}{2}$, and hence with probability at least $2/3$, $\text{Alg}(\Psi) \geq \rho(\Pi) + \frac{\epsilon}{2}$. Therefore, $\mathbb{E}_{\Psi \sim \mathcal{Y}_{q, \alpha, T}^{\Pi, \pi}(n)}[\text{Alg}(\Psi) = 1] \geq 2/3$. Similarly, by the choice of q_0, α_0, T_0 , it follows from [Lemma 3.4](#) that

$$\Pr_{\Psi \sim \mathcal{N}_{q, \alpha, T}^{\Pi}(n)} \left[\text{ocsp-val}_\Psi \geq \rho(\Pi) + \frac{\epsilon}{2} \right] \leq 0.01,$$

and hence, $\mathbb{E}_{\Psi \sim \mathcal{N}_{q, \alpha, T}^{\Pi}(n)}[\text{Alg}(\Psi) = 1] \leq \frac{1}{3} + 0.01$. Therefore, Alg' distinguishes $\mathcal{Y}_{q, \alpha, T}^{\Pi, \pi}(n)$ from $\mathcal{N}_{q, \alpha, T}^{\Pi}(n)$ with advantage $1/8$. By applying [Lemma 3.5](#), we conclude that Alg uses $s(n) \geq \Omega(n)$ space. \square

4. Bounds on $\mathbf{Max-OCSP}(\Pi)$ values of \mathcal{Y} and \mathcal{N}

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

4.1. CSPs and coarsening. In order to prove the lemmas, we recall the definition of (standard) *constraint satisfaction problems (CSPs)*, whose solution spaces are $[q]^n$ (as opposed to \mathfrak{S}_n for OCSPs), and define an operation called *q-coarsening* on Max-OCSPs, which restricts the solution space from \mathfrak{S}_n to $[q]^n$.

A *maximum constraint satisfaction problem*, $\mathbf{Max-CSP}(f)$, is specified by a single constraint predicate $f : [q]^k \rightarrow \{0, 1\}$, for some positive integer k . An *instance* of $\mathbf{Max-CSP}(f)$ on n variables is given by m constraints C_1, \dots, C_m where $C_i = (f, \mathbf{j}(i))$, i.e., the application of the predicate f to the variables $\mathbf{j}(i) = (j(i)_1, \dots, j(i)_k)$. (Again, f is omitted when clear from context.) The *value* of an assignment $\mathbf{b} \in [q]^n$ on an instance $\Phi = (C_1, \dots, C_m)$, denoted $\mathbf{csp-val}_{\Phi}(\mathbf{b})$, is the fraction of constraints satisfied by \mathbf{b} , i.e., $\mathbf{csp-val}_{\Phi}(\mathbf{b}) = \frac{1}{m} \sum_{i \in [m]} f(\mathbf{b}|_{\mathbf{j}(i)})$, where (recall) $\mathbf{b}|_{\mathbf{j}} = (b_{j_1}, \dots, b_{j_k})$ for $\mathbf{b} = (b_1, \dots, b_n), \mathbf{j} = (j_1, \dots, j_k)$. The optimal value of Φ is defined as $\mathbf{csp-val}_{\Phi} = \max_{\mathbf{b} \in [q]^n} \{\mathbf{csp-val}_{\Phi}(\mathbf{b})\}$.

DEFINITION 4.1 (*q-coarsening*). *Let $\Pi : \mathfrak{S}_k \rightarrow \{0, 1\}$ be an ordering predicate. For $q \in \mathbb{N}$, the q -coarsening of Π is the predicate $\Pi^{\downarrow q} : [q]^k \rightarrow \{0, 1\}$ defined by $\Pi^{\downarrow q}(\mathbf{a}) = 1$ iff $\Pi(\mathbf{ord}(\mathbf{a})) = 1$ (if $\mathbf{ord}(\mathbf{a}) = \perp$, i.e., \mathbf{a} 's entries are not all distinct, we define $\Pi^{\downarrow q}(\mathbf{a}) = 0$). The q -coarsening of the problem $\mathbf{Max-OCSP}(\Pi)$ is the problem $\mathbf{Max-CSP}(\Pi^{\downarrow q})$, and the q -coarsening of an instance Ψ of $\mathbf{Max-OCSP}(\Pi)$ is the instance $\Psi^{\downarrow q}$ of $\mathbf{Max-CSP}(\Pi^{\downarrow q})$ given by the same constraint hypergraph.*

The following lemma captures the idea that coarsening restricts the space of possible solutions; compare to [Lemma 4.10](#).

LEMMA 4.2. *If $q \in \mathbb{N}$ and Ψ is an instance of $\mathbf{Max-OCSP}(\Pi)$, then $\mathbf{ocsp-val}_{\Psi} \geq \mathbf{csp-val}_{\Psi^{\downarrow q}}$.*

PROOF. We will show that for every assignment $\mathbf{b} \in [q]^n$ to $\Psi^{\downarrow q}$, we can construct an assignment $\mathbf{b}^\uparrow \in \mathfrak{S}_n$ to Ψ such that $\text{ocsp-val}_\Psi(\mathbf{b}^\uparrow) \geq \text{csp-val}_{\Psi^{\downarrow q}}(\mathbf{b})$. Consider an assignment $\mathbf{b} \in [q]^n$. Let \mathbf{b}^\uparrow be the ordering on $[n]$ constructed by placing the blocks $\mathbf{b}^{-1}(1), \dots, \mathbf{b}^{-1}(q)$ in order (within each block, we enumerate the indices arbitrarily). Consider any constraint $C = \mathbf{j} = (j_1, \dots, j_k)$ in Ψ which is satisfied by \mathbf{b} in $\Psi^{\downarrow q}$. Since $\Pi^{\downarrow q}(\mathbf{b}|_{\mathbf{j}}) = 1$, by definition of $\Pi^{\downarrow q}$ we have that $\Pi(\text{ord}(\mathbf{b}|_{\mathbf{j}})) = 1$ and in particular that b_{j_1}, \dots, b_{j_k} are distinct. The latter implies, by the construction of \mathbf{b}^\uparrow , that $\text{ord}(\mathbf{b}|_{\mathbf{j}}) = \text{ord}(\mathbf{b}^\uparrow|_{\mathbf{j}})$. Hence, $\Pi(\text{ord}(\mathbf{b}^\uparrow|_{\mathbf{j}})) = 1$, so \mathbf{b}^\uparrow satisfies C in Ψ . Hence, $\text{ocsp-val}_\Psi(\mathbf{b}^\uparrow) \geq \text{csp-val}_{\Psi^{\downarrow q}}(\mathbf{b})$. \square

4.2. \mathcal{Y} has high $\text{Max-OCSP}(\Pi)$ values. In this section, we prove [Lemma 3.3](#), which states that the $\text{Max-OCSP}(\Pi)$ values of instances Ψ drawn from $\mathcal{Y}_{q,\alpha,T}^{\Pi,\pi}(n)$ are large. Note that we prove a bound for *every* instance Ψ in the support of $\mathcal{Y}_{q,\alpha,T}^{\Pi,\pi}(n)$, 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 3.3](#), by [Lemma 4.2](#), it suffices to show that $\text{csp-val}_{\Psi^{\downarrow q}} \geq 1 - \frac{k-1}{q}$. One natural approach is to consider the q -partition $\mathbf{b} = (b_1, \dots, b_n) \in [q]^n$ sampled when sampling Ψ and view \mathbf{b} as an assignment for $\Psi^{\downarrow q}$. Consider any constraint $C = \mathbf{j} = (j_1, \dots, j_k)$ in Ψ ; by the definition of $\mathcal{Y}^{\Pi,\pi}$ ([Definition 3.2](#)), we have $\mathbf{b}|_{\mathbf{j}} = \mathbf{v}^{(c)} +_q \boldsymbol{\pi}$ for some (unique) $c \in [q]$, where $\mathbf{v}^{(c)}$ denotes the vector $(c, \dots, c) \in [q]^k$. We term c the *identifier* of C . Now we use the following simple fact:

FACT 4.3. *Let $\boldsymbol{\pi} \in \mathfrak{S}_k$. Then for every $c \in \{1, \dots, q-k\} \cup \{q\}$, $\text{ord}(\mathbf{v}^{(c)} +_q \boldsymbol{\pi}) = \boldsymbol{\pi}$.*

PROOF. Follows from the fact that for c in this range, and every $i, j \in [k] \subset [q]$, we have $i < j$ iff $i +_q c < j +_q c$. \square

Thus, C is satisfied by \mathbf{b} iff $\Pi(\text{ord}(\mathbf{v}^{(c)} +_q \boldsymbol{\pi})) = 1$. Hence, a sufficient condition for \mathbf{b} to satisfy C is that C 's identifier $c \in \{1, \dots, q-k\} \cup \{q\}$, since in this case by [Fact 4.3](#), we have $\text{ord}(\mathbf{v}^{(c)} +_q \boldsymbol{\pi}) = \boldsymbol{\pi}$. Unfortunately, when sampling the constraints, we might get “unlucky” and get a sample that over-represents the identifiers

$\{q - k + 1, \dots, q - 1\}$. We can resolve this issue using “shifted” versions of \mathbf{b} .⁵ The proof is as follows:

PROOF (Proof of [Lemma 3.3](#)). For $t \in [q]$, define the assignment $\mathbf{b}^{(t)} = (b_1^{(t)}, \dots, b_n^{(t)})$ to $\Psi^{\downarrow q}$ via $b_i^{(t)} = b_i +_q t$ for $i \in [n]$.

Fix $t \in [q]$. Then we claim that $\mathbf{b}^{(t)}$ satisfies any constraint C with identifier c such that $c +_q t \in \{1, \dots, q - k\} \cup \{q\}$. Indeed, if $C = \mathbf{j}$ is a constraint with identifier c , we have $\mathbf{b}^{(t)}|_{\mathbf{j}} = \mathbf{v}^{(c)} +_q \mathbf{v}^{(t)} +_q \boldsymbol{\pi} = \mathbf{v}^{(c+q t)} +_q \boldsymbol{\pi}$, and then, we use [Fact 4.3](#).

Now (no longer fixing t), for each $c \in [q]$, let $w^{(c)}$ be the fraction of constraints in Ψ with identifier c . By the previous paragraph, for each $t \in [q]$, we have $\text{csp-val}_{\Psi}^{\downarrow q}(\mathbf{b}^{(t)}) \geq \sum_{c: c+q t \in \{1, \dots, q - k\} \cup \{q\}} w^{(c)}$. On the other hand, $\sum_{c=1}^q w^{(c)} = 1$ (since every constraint has some (unique) identifier). Hence,

$$\sum_{t=1}^q \text{csp-val}_{\Psi^{\downarrow q}}(\mathbf{b}^{(t)}) \geq \sum_{t=1}^q \left(\sum_{c: c+q t \in \{1, \dots, q - k\} \cup \{q\}} w^{(c)} \right) = q - (k - 1),$$

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

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

REMARK 4.4. *Using concentration bounds (i.e., [Lemma 2.1](#)), one could show that the probability that a fixed solution $\boldsymbol{\sigma} \in \mathfrak{S}_n$ satisfies more than $\rho(\Pi) + 1/q$ constraints is exponentially small in n . However, taking a union bound over all $n!$ permutations $\boldsymbol{\sigma}$ would cause an unacceptable blowup in the probability (since by Stirling’s approximation, $n! \sim (n/e)^n$).*

⁵Alternatively, in expectation, $\text{csp-val}_{\Psi^{\downarrow q}}(\mathbf{b}) = 1 - \frac{k-1}{q}$. Hence, with probability at least $\frac{99}{100}$, $\text{csp-val}_{\Psi^{\downarrow q}}(\mathbf{b}) \geq 1 - \frac{100(k-1)}{q}$ by Markov’s inequality; this suffices for a “with-high-probability” statement.

Instead, to prove [Lemma 3.4](#), we take an indirect approach, involving bounding the **Max-CSP** value of the q -coarsening of a random instance and bounding the gap between the **Max-OCSP** value and the q -coarsened **Max-CSP** value. To do this, we define the following notions of small set expansion for k -hypergraphs:

DEFINITION 4.5 (Lying on a set). *Let $G = ([n], E)$ be a k -hypergraph. Given a set $S \subseteq [n]$, a k -hyperedge $\mathbf{j} \in E$ lies on S if it is incident on two (distinct) vertices in S (i.e., if $|\Gamma(\mathbf{j}) \cap S| \geq 2$).*

DEFINITION 4.6 (Congregating on a partition). *Let $G = ([n], E)$ be a k -hypergraph. Given a q -partition $\mathbf{b} \in [q]^n$, a k -hyperedge $\mathbf{j} \in E$ congregates on \mathbf{b} if it lies on a block $\mathbf{b}^{-1}(i) = \{j \in [n] : b_j = i\}$ for some $i \in [q]$.*

DEFINITION 4.7. (Small set hypergraph expansion (SSHE) property) *A k -hypergraph $G = ([n], E)$ is a (γ, δ) -small set hypergraph expander (SSHE) if it has the following property: For every subset $S \subseteq [n]$ of size at most γn , the number of k -hyperedges in E which lie on S is at most $\delta |E|$.*

DEFINITION 4.8. (Small partition hypergraph expansion (SPHE) property) *A k -hypergraph $G = ([n], E)$ is a (γ, δ) -small partition hypergraph expander (SPHE) if it has the following property: For every partition $\mathbf{b} \in [q]^n$ where each block $\mathbf{b}^{-1}(i) = \{j \in [n] : b_j = i\}$ has size at most γn , the number of k -hyperedges in E that congregate on \mathbf{b} is at most $\delta |E|$.*

In the context of [Figure 3.1](#), the SPHE property says that for *any* partition with small blocks, there cannot be too many “orange” edges.

In the remainder of this subsection, we state several lemmas and then give a formal proof of [Lemma 3.4](#). We begin with several short lemmas.

LEMMA 4.9 (Good SSHEs are good SPHEs). *For every $k \in \mathbb{N}$ and $\gamma, \delta > 0$, if a k -hypergraph $G = (V, E)$ is a (γ, δ) -SSHE, then it is a $(\gamma, \delta(2/\gamma + 1))$ -SPHE.*

PROOF. Let $n = |V|$. Consider any partition $\mathbf{b} \in [q]^n$ of V where each block $\mathbf{b}^{-1}(i)$ has size at most γn . WLOG, all but one block has size at least $\frac{\gamma n}{2}$ (if not, merge blocks until this happens, only increasing the number of k -hyperedges that congregate on \mathbf{b}). Hence, $\ell \leq \frac{2}{\gamma} + 1$.⁶ By the SSHE property, there are at most δm k -hyperedges that lie on each block; hence, there are at most $\delta(\frac{2}{\gamma} + 1)m$ constraints that congregate on \mathbf{b} . \square

LEMMA 4.10 (Coarsening roughly preserves value in SPHEs). *Let Ψ be a $\text{Max-OCSP}(\Pi)$ instance on n variables, and let $q \geq 2/\gamma$. Suppose that the constraint hypergraph $G(\Psi)$ of Ψ is a (γ, δ) -SPHE. Then for sufficiently large n ,*

$$\text{ocsp-val}_\Psi \leq \text{csp-val}_{\Psi^{\downarrow q}} + \delta.$$

PROOF. For every assignment $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_n) \in \mathfrak{S}_n$ to Ψ , we will construct an assignment $\boldsymbol{\sigma}^{\downarrow q} = (\sigma_1^{\downarrow q}, \dots, \sigma_n^{\downarrow q}) \in [q]^n$ to $\Psi^{\downarrow q}$ such that $\text{ocsp-val}_\Psi(\boldsymbol{\sigma}) \leq \text{csp-val}_{\Psi^{\downarrow q}}(\boldsymbol{\sigma}^{\downarrow q}) + \delta$. Fix $\boldsymbol{\sigma} \in \mathfrak{S}_n$. Define $\boldsymbol{\sigma}^{\downarrow q} \in [q]^n$ by $\sigma_i^{\downarrow q} = \lfloor \sigma_i / \gamma n \rfloor$ for each $i \in [n]$. We verify that since $\sigma_i \leq n$, we have

$$\sigma_i^{\downarrow q} \leq \lfloor n / \gamma n \rfloor \leq n / \gamma n \leq n / (2n/q) \leq n / (n/q) = q,$$

so $\boldsymbol{\sigma}^{\downarrow q}$ is a valid assignment to $\Psi^{\downarrow q}$. Also, $\boldsymbol{\sigma}^{\downarrow q}$ has the property that for every $i, j \in [n]$, if $\sigma_i < \sigma_j$ then $\sigma_i^{\downarrow q} \leq \sigma_j^{\downarrow q}$; we call this *monotonicity* of $\boldsymbol{\sigma}^{\downarrow q}$.

Now view $\boldsymbol{\sigma}^{\downarrow q}$ as a q -partition and consider the constraint hypergraph $G(\Psi)$ of Ψ (which is the same as the constraint hypergraph $G(\Psi^{\downarrow q})$ of $\Psi^{\downarrow q}$). Call a constraint $C = \mathbf{j} = (j_1, \dots, j_k)$ in Ψ *good* if it is both satisfied by $\boldsymbol{\sigma}$, and the k -hyperedge corresponding to it does not congregate on $\boldsymbol{\sigma}^{\downarrow q}$. If C is good, then $\sigma_{j_1}^{\downarrow q}, \dots, \sigma_{j_k}^{\downarrow q}$ are

⁶We 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.

all distinct; together with monotonicity of $\sigma^{\downarrow q}$, we conclude that if C is good, then $\text{ord}(\sigma^{\downarrow q}|_j) = \text{ord}(\sigma|_j)$, and hence C is also satisfied by $\sigma^{\downarrow q}$ in $\Psi^{\downarrow q}$.

Finally, we note that each block in $\sigma^{\downarrow q}$ has size at most γn by definition; hence, by the SPHE property of the constraint hypergraph of Ψ , at most δ -fraction of the constraints of Ψ correspond to k -hyperedges that congregate on $\sigma^{\downarrow q}$. Since $\text{ocsp-val}_\Psi(\sigma)$ -fraction of the constraints of Ψ are satisfied by σ , at least $(\text{ocsp-val}_\Psi(\sigma) - \delta)$ -fraction of the constraints of Ψ are good, and hence, $\sigma^{\downarrow q}$ satisfies at least $(\text{ocsp-val}_\Psi(\sigma) - \delta)$ -fraction of the constraints of $\Psi^{\downarrow q}$, as desired. \square

The construction in this lemma was called *coarsening* the assignment σ by [Guruswami et al. \(2011\)](#) (cf. ([Guruswami et al. 2011](#), Definition 4.1)).

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

LEMMA 4.11 (Most instances in \mathcal{N} have many constraints). *For every $n, \alpha, \gamma > 0$, and $q \in \mathbb{N}$,*

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

PROOF. 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 follows by applying a Chernoff bound ([Lemma 2.2](#)) with $\eta = p/2$. \square

Now we state the following pair of lemmas, whose more involved proofs we defer to [Section 4.3.1](#) and [Section 4.3.2](#), respectively:

LEMMA 4.12. *For every $n, \alpha, \gamma > 0$, and $q \in \mathbb{N}$ with $\alpha \leq 1/(2k)$,*

$$\begin{aligned} \Pr_{\Psi \sim \mathcal{N}_{q, \alpha, T}^{\text{II}}(n)} \left[G(\Psi) \text{ is not a } (\gamma, 8k^2\gamma^2)\text{-SSHE} \mid m(\Psi) \geq \frac{n\alpha T}{2q^k} \right] \\ \leq \exp \left(- \left(\frac{\gamma^2 \alpha T}{2q^k} - \ln 2 \right) n \right). \end{aligned}$$

LEMMA 4.13 (Satisfiability of random $\text{Max-CSP}(\Psi^{\downarrow q})$ instances).
For every $n, \alpha, \eta > 0$,

$$\begin{aligned} \Pr_{\Psi \sim \mathcal{N}_{q,\alpha,T}^{\Pi}(n)} \left[\text{csp-val}_{\Psi^{\downarrow q}} \geq \rho(\Pi) + \eta \mid m(\Psi) \geq \frac{n\alpha T}{2q^k} \right] \\ \leq \exp \left(- \left(\frac{\eta^2 \alpha T}{4(\rho(\Pi) + \eta)q^k} - \ln q \right) n \right). \end{aligned}$$

We remark here that the proofs of both lemmas only require union bounds over sets of size $(O_{\epsilon}(1))^n$ (the set of all small subsets of $[n]$ and of all solutions to the coarsened Max-CSP , respectively); this lets us avoid the issue, mentioned in the Remark at the beginning of this subsection, of union-bounding over the entire space \mathfrak{S}_n of super-exponential size $n!$ directly.

We finally give the proof of [Lemma 3.4](#).

PROOF (Proof of [Lemma 3.4](#)). Let $q_0 \stackrel{\text{def}}{=} \left\lceil \frac{192k^2}{\epsilon} \right\rceil$ and let $\alpha_0 \stackrel{\text{def}}{=} \frac{1}{2k}$. Suppose $\alpha \leq \alpha_0$ and $q \geq q_0$. Then let $\gamma \stackrel{\text{def}}{=} \frac{\epsilon}{96k^2}$ and $\eta \stackrel{\text{def}}{=} \frac{\epsilon}{4}$, and let

$$T_0 \stackrel{\text{def}}{=} \left\lceil \max \left\{ \frac{2(\ln 2)q^k}{\gamma^2 \alpha}, \frac{4(\rho(\Pi) + \eta)q^k(\ln q)}{\eta^2 \alpha} \right\} \right\rceil + 1.$$

Now, we will prove the desired bound for any $T \geq T_0$.

Let $\mathcal{E}_1, \mathcal{E}_2$, and \mathcal{E}_3 denote, respectively, the events “ $m(\Psi) \leq (\alpha T / (2q^k))n$ ”, “ $G(\Psi)$ is not a $(\gamma, 8k^2\gamma^2)$ -SSHE”, and “ $\text{csp-val}_{\Psi^{\downarrow q}} \geq \rho(\Pi) + \eta$ ”. Then, since $\alpha \leq \alpha_0$, [Lemma 4.11](#), [Lemma 4.12](#), and [Lemma 4.13](#) state that

$$\begin{aligned} \Pr[\mathcal{E}_1] &\leq \exp(-C_1 n), \\ \Pr[\mathcal{E}_2 \mid \overline{\mathcal{E}_1}] &\leq \exp(-C_2 n), \\ \Pr[\mathcal{E}_3 \mid \overline{\mathcal{E}_1}] &\leq \exp(-C_3 n), \end{aligned}$$

respectively, where we define the constants $C_1 = \frac{\alpha T}{8q^k}$, $C_2 = \frac{\gamma^2 \alpha T}{2q^k} - \ln 2$, and $C_3 = \frac{\eta^2 \alpha T}{4(\rho(\Pi) + \eta)q^k} - \ln q$, and all probabilities are over the choice of $\Psi \sim \mathcal{N}_{q,\alpha,T}^{\Pi}(n)$. Now observe that for our choice of T ,

C_1, C_2, C_3 are all positive and do not depend on n , so for sufficiently large n , all three probabilities are smaller than $1/1000$. This implies that $\Pr[\mathcal{E}_2 \vee \mathcal{E}_3] \leq 1/100$: Indeed,

(total probability)

$$\Pr[\mathcal{E}_2 \vee \mathcal{E}_3] = \Pr[\mathcal{E}_2 \vee \mathcal{E}_3 \mid \overline{\mathcal{E}_1}] \Pr[\overline{\mathcal{E}_1}] + \Pr[\mathcal{E}_2 \vee \mathcal{E}_3 \mid \mathcal{E}_1] \Pr[\mathcal{E}_1]$$

(probabilities ≤ 1)

$$\leq \Pr[\mathcal{E}_2 \vee \mathcal{E}_3 \mid \overline{\mathcal{E}_1}] + \Pr[\mathcal{E}_1]$$

(union bound) $\leq \Pr[\mathcal{E}_2 \mid \overline{\mathcal{E}_1}] + \Pr[\mathcal{E}_3 \mid \overline{\mathcal{E}_1}] + \Pr[\mathcal{E}_1]$

and this is $\leq 3/1000 < 1/100$ by assumption.

Finally, we show that when neither \mathcal{E}_2 nor \mathcal{E}_3 occurs, we have $\text{ocsp-val}_\Psi \leq \rho(\Pi) + \epsilon/2$. Indeed, if $G(\Psi)$ is a $(\gamma, 8k^2\gamma^2)$ -SSHE, by Lemma 4.9 it is also a (γ, δ) -SPHE for $\delta = 8k^2\gamma^2(2/\gamma + 1) \leq 8k^2\gamma^2(3/\gamma) = 24k^2\gamma = \epsilon/4$. Now since $q \geq q_0 \geq 2/\gamma$, we can apply Lemma 4.10 and conclude that

$$\text{ocsp-val}_\Psi \leq \rho(\Pi) + \eta + \delta \leq \rho(\Pi) + \frac{\epsilon}{2},$$

as desired. \square

4.3.1. \mathcal{N} is a good SSHE with high probability: Proving Lemma 4.12. Recall that for a k -hypergraph $G = (V, E)$ and $S \subseteq V(G)$, we define $\Gamma(\mathbf{j}) \subseteq V$ as the set of vertices incident on \mathbf{j} .

PROOF (Proof of Lemma 4.12). Let \tilde{G}_t denote the t -th hypermatching sampled when sampling Ψ (as in Definition 3.2). Let $\beta = \frac{m(\Psi)}{Tn}$, and for each $t \in [T]$, let $\beta_t = \frac{m(\tilde{G}_t)}{n}$, so that $\beta = \frac{1}{T} \sum_{t=1}^T \beta_t$. By our conditioning assumption, $\beta \geq \frac{\alpha}{2q^k}$, and $\beta_t \leq \alpha$ for each $t \in [T]$. It suffices to prove the lemma conditioned on fixed values for β_1, \dots, β_T . This is equivalent to simply sampling hypermatchings $G_t \sim \mathcal{M}_{k, \beta_t}(n)$ independently and including all of their k -hyperedges as constraints.

Fix any set $S \subseteq [n]$ of size at most γn . For each $t \in [T]$, label the k -hyperedges of G_t as $\mathbf{j}(t, 1), \dots, \mathbf{j}(t, \beta_t n)$. Consider the collection of $m(\Psi) = \beta Tn$ random variables $\{X_{t,i}\}_{t \in [T], i \in [\beta_t n]}$, each of which is the indicator for the event that $\mathbf{j}(t, i)$ lies on S . Define

$X = \sum_{t=1}^T \sum_{i=1}^{\beta_t n} X_{t,i}$ as the number of such “lying on” edges we observe. Since $m(G(\Psi)) = \beta T n$, it suffices to bound the probability that $X \geq 8k^2\gamma^2 m$ and then take the union bound over all subsets S .

For fixed $t \in [T]$, we first bound $\mathbb{E}[X_{t,i} \mid X_{t,1}, \dots, X_{t,i-1}]$ for each $i \in [\beta_t n]$. Conditioned on $\mathbf{j}_{t,1}, \dots, \mathbf{j}_{t,i-1}$, the k -hyperedge $\mathbf{j}(t, i)$ is uniformly distributed over the set of all k -hyperedges on $[n] \setminus (\Gamma(\mathbf{j}_{t,1}) \cup \dots \cup \Gamma(\mathbf{j}_{t,i-1}))$. It suffices to union-bound, over distinct pairs $\{a, b\} \in \binom{[k]}{2}$, the probability that the a -th and b -th vertices of $\mathbf{j}(t, i)$ are in S (conditioned on $X_{t,1}, \dots, X_{t,i-1}$). We can sample the a -th and b -th vertices of $\mathbf{j}(t, i)$ first (uniformly over the remaining vertices) and then ignore the remaining vertices. Hence, we have the upper bound

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

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

Now, we want to apply [Lemma 2.1](#) to the random variables $(X_{t,i})$. Consider enumerating the $X_{t,i}$ ’s lexicographically as

$$X_{1,1}, \dots, X_{1,\beta_1 n}, X_{2,1}, \dots, X_{2,\beta_2 n}, \dots, X_{T,1}, \dots, X_{T,\beta_T n}.$$

Now since the hypermatchings G_t are sampled independently for all $t \in [T]$, the collections of random variables $(X_{t,i})_{i \in [\beta_t n]}$ are independent across $t \in [T]$. So, for all $t \in [T]$ and $i \in [\beta_t n]$, the expectation of $X_{t,i}$ conditioned on the lexicographically-earlier $X_{t',i'}$ ’s is still at most $4k^2\gamma^2$. Thus, [Lemma 2.1](#) (at $p = \eta = 4k^2\gamma^2$) implies that

$$\Pr_{\Psi \sim \mathcal{N}_{q,\alpha,T}^{\Pi}(n)} [X \geq 8k^2\gamma^2 m] \leq \exp(-k^2\gamma^2 m) \leq \exp(-\gamma^2 m).$$

Finally, we assumed $m \geq \frac{\alpha T}{2q^k} n$, combined with the union-bound over the $\leq 2^n$ possible subsets $S \subseteq [n]$, gives the desired bound. \square

4.3.2. \mathcal{N} has low coarsened **Max-CSP**($\Psi^{\downarrow q}$) values (w.h.p.): Proving [Lemma 4.13](#)

PROOF (Proof of [Lemma 4.13](#)). We consider the same setup as in the first paragraph of the proof of [Lemma 4.12](#) in the previous subsection: We set $\beta = \frac{m(\Psi)}{Tn}$ and $\beta_t = \frac{m(G_t)}{n}$. Abbreviating $m = m(\Psi)$, Ψ contains $m = \beta Tn$ constraints. We condition on fixed β_1, \dots, β_T and consider sampling hypermatchings $G_t \sim \mathcal{M}_{k, \beta_t}(n)$ and we have $\frac{1}{T} \sum_{t=1}^T \beta_t = \beta \geq \alpha/2q^k$. We label the k -hyperedges of G_t as $\mathbf{j}(t, 1), \dots, \mathbf{j}(t, \beta_t n)$. Now, we view these k -hyperedges as the constraints of $\Psi^{\downarrow q}$ for $\Psi \sim \mathcal{N}_{q, \alpha, T}^{\Pi}(n)$.

Let $X_{t,i}$ for $t \in [T], i \in [\beta_t n]$ be the indicator for the event that, as a constraint in $\Psi^{\downarrow q}$, $\mathbf{j}(t, i)$ is satisfied by \mathbf{b} , i.e., $\Pi^{\downarrow q}(\mathbf{b} |_{\mathbf{j}(t, i)}) = 1$. Again, like in the proof of [Lemma 4.12](#), we first fix $t \in [T]$ and bound $\mathbb{E}[X_{t,i} | X_{t,1}, \dots, X_{t,i-1}]$, for each $i \in [\beta_t n]$. Conditioned on $\mathbf{j}(t, 1), \dots, \mathbf{j}(t, i-1)$, the k -hyperedge $\mathbf{j}(t, i)$ is uniformly distributed over the set of all k -hyperedges on $[n] \setminus (\Gamma(\mathbf{j}(t, 1)) \cup \dots \cup \Gamma(\mathbf{j}(t, i-1)))$. Now, we claim that $\mathbb{E}[X_{t,i} | X_{t,1}, \dots, X_{t,i-1}] \leq \rho(\Pi)$. Indeed, the set of possible k -hyperedges on $[n] \setminus (\Gamma(\mathbf{j}(t, 1)) \cup \dots \cup \Gamma(\mathbf{j}(t, i-1)))$ may be partitioned into blocks of size $k!$ by mapping each k -hyperedge to the set of vertices on which it is incident. That is, we can consider the $k!$ possible k -hyperedges resulting from permuting the vertices of a given k -hyperedge $\mathbf{j} = (j_1, \dots, j_k)$. If b_{j_1}, \dots, b_{j_k} are not all distinct, then $\text{ord}(\mathbf{b} |_{\mathbf{j}}) = \perp$ and so none of the k -hyperedges are satisfied by \mathbf{b} as constraints in $\Psi^{\downarrow q}$; otherwise, $\text{ord}(\mathbf{b} |_{\mathbf{j}})$ is some ordering in \mathfrak{S}_k , and so exactly $|\text{supp}(\Pi)| = \rho(\Pi) \cdot k!$ permutations of \mathbf{j} are satisfied as constraints in $\Psi^{\downarrow q}$.

Now, we again apply [Lemma 2.1](#) to the random variables $(X_{t,i})$. Using the same lexicographic enumeration of variables as in the proof of [Lemma 4.12](#), and the same observation that the hypermatchings G_t are sampled independently, we conclude that the expectation of $X_{t,i}$ conditioned on the lexicographically earlier $X_{t',i'}$'s is at most $\rho(\Pi)$. Again, we apply [Lemma 2.1](#) (now with $p = \rho(\Pi)$) to deduce that

$$\Pr \left[\sum_{t=1}^T \sum_{i=1}^{\beta_t n} X_i \geq (\rho(\Pi) + \eta)m \right] \leq \exp \left(- \left(\frac{\eta^2}{2(\rho(\Pi) + \eta)} \right) m \right).$$

Finally, we assumed $m \geq n\alpha T/2q^k$, so the RHS is at most

$$\exp\left(-\left(\frac{\eta^2\alpha T}{4(\rho(\Pi) + \eta)}\right)n\right);$$

then, we union bound over $\mathbf{b} \in [q]^n$ to yield the desired conclusion. \square

5. Streaming indistinguishability of \mathcal{Y} and \mathcal{N}

In this section, we prove [Lemma 3.5](#) establishing the streaming indistinguishability of the distributions \mathcal{Y} and \mathcal{N} . This indistinguishability follows directly from the work of [Chou et al. \(2022b\)](#), who introduce a T -player communication problem called *implicit randomized mask detection (IRMD)*. Once we properly situate our instances \mathcal{Y} and \mathcal{N} within the framework of [Chou et al. \(2022b\)](#), [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 ([Chou et al. 2022b](#), Definition 3.1). In [Chou et al. \(2022b\)](#), 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 problem) *Let $q, k, n, T \in \mathbb{N}, \alpha \in (0, 1/k)$ be parameters. $\text{IRMD}_{q,k,\alpha,T}(n)$ is a T -player one-way communication game. The T players are denoted $\text{Player}_1, \dots, \text{Player}_T$. The input to the players is drawn either from a “**YES** distribution” or a “**NO** distribution”, and their collective goal is to distinguish these two cases. Each player Player_t receives two inputs: (i) a uniform k -hypermatching $\tilde{G}_t \sim \mathcal{M}_{k,\alpha}(n)$ on n vertices with αn hyperedges $\mathbf{j}(t, 1), \dots, \mathbf{j}(t, \alpha n)$, and (ii) a vector $\mathbf{z}_t = (\mathbf{z}_{t,1}, \dots, \mathbf{z}_{t,\alpha n}) \in ([q]^k)^{\alpha n}$ labeling the k -hyperedges of \tilde{G}_t . There is also a random partition $\mathbf{b} \sim \text{Unif}([q]^n)$ which is hidden, i.e., not given to the players directly. The difference between the **YES** and **NO** distributions is in how \mathbf{z}_t is determined by \tilde{G}_t and \mathbf{b} :*

- o **YES** case: Each $\mathbf{z}_{t,i} = \mathbf{b}|_{\mathbf{j}(t,i)} +_q \mathbf{y}_{t,i}$ where $\mathbf{y}_{t,i} \sim \text{Unif}(\{\mathbf{v}^{(c)} = (c, \dots, c) : c \in [q]\})$ independently.
- o **NO** case: Each $\mathbf{z}_{t,i} = \mathbf{b}|_{\mathbf{j}(t,i)} +_q \mathbf{y}_{t,i}$ where $\mathbf{y}_{t,i} \sim \text{Unif}([q]^k)$ independently.

Player_t sends a private message to the Player_{t+1} after receiving a message from Player_{t-1} . The goal is for Player_T to decide whether $(\mathbf{z}_t)_{t \in T}$ has been chosen from the **YES** or **NO** distribution, and the advantage of a protocol is defined as

$$\left| \Pr_{\text{YES case}}[\text{Player}_T \text{ outputs 1}] - \Pr_{\text{NO case}}[\text{Player}_T \text{ outputs 1}] \right|.$$

Note that the definition of the IRMD problem does not depend on an underlying predicate (beyond fixing q and k). Nevertheless, we will be able to leverage IRMD's hardness to prove [Lemma 3.5](#) (and indeed, all hardness results in [\(Chou et al. 2022b\)](#) itself stem from hardness for the IRMD problem). The following theorem of [Chou et al. \(2022b\)](#) gives a lower bound on the communication complexity of the IRMD problem:

THEOREM 5.2 ([Chou et al. 2022b](#), 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 $\text{IRMD}_{q,k,\alpha,T}(n)$ with advantage δ requires τn bits of communication.*

Now, we use this hardness result to prove [Lemma 3.5](#). The following proof is based on the proof of [\(Chou et al. 2022b, Theorem 4.3\)](#). However, [\(Chou et al. 2022b, Theorem 4.3\)](#) 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 (Proof of [Lemma 3.5](#)). We prove the lemma for the same α_0 as in [Theorem 5.2](#).

Suppose Alg is a $s(n)$ -space streaming algorithm that distinguishes $\mathcal{Y}_{q,\alpha,T}^{\Pi,\pi}(n)$ from $\mathcal{N}_{q,\alpha,T}^{\Pi}(n)$ with advantage $1/8$ for all lengths

n. We now show how to use Alg to construct a protocol $\text{Prot} = (\text{Prot}_1, \dots, \text{Prot}_T)$ (where Prot_t defines the behavior of the Player_t) solving $\text{IRMD}_{q,k,\alpha,T}(n)$ with advantage $1/8$ for $n \geq n_0$, which uses only $s(n)$ bits of communication; [Theorem 5.2](#) provides a constant $\tau \in (0, 1)$ yielding the desired contradiction if we set $s(n) \leq \tau n$. As is standard, this reduction will involve the players collectively running the streaming algorithm Alg . That is, Prot is defined as follows: For $t \in [T-2]$, Prot_t adds some constraints to the stream (in a manner to be specified below) and then sends the state of Alg on to Player_{t+1} . Finally, Prot_T terminates the streaming algorithm and outputs the output of Alg .

Recall, Player_t 's input is a pair $(\tilde{G}_t, \mathbf{z}_t)$ consisting of a hypermatching \tilde{G}_t and a vector $\mathbf{z}_t = (\mathbf{z}_{t,1}, \dots, \mathbf{z}_{t,\alpha n})$ of labels $\mathbf{z}_{t,i} \in [q]^k$ for each hyperedge in M_t . We define Prot_t 's behavior as follows: It adds each hyperedge $\mathbf{j}(i)$ in \tilde{G}_t to the stream iff $\mathbf{z}_{t,i} = \boldsymbol{\pi}$.

Let $\mathcal{Y}'(n)$ and $\mathcal{N}'(n)$ denote the distributions of $\text{Max-OCSP}(\Pi)$ instances constructed by Prot in the **YES** and **NO** cases, respectively. The crucial claim is that $\mathcal{Y}'(n)$ and $\mathcal{Y}_{q,\alpha,T}^{\Pi, \boldsymbol{\pi}}(n)$ are identical distributions, and similarly with $\mathcal{N}'(n)$ and $\mathcal{N}_{q,\alpha,T}^{\Pi}(n)$. This claim suffices to prove the lemma since the constructed stream of constraints is fed into Alg , which is an $s(n)$ -space streaming algorithm distinguishing $\mathcal{Y}_{q,\alpha,T}^{\Pi, \boldsymbol{\pi}}(n)$ from $\mathcal{N}_{q,\alpha,T}^{\Pi}(n)$; hence, we can conclude that Prot is a protocol for IRMD using $s(n) \leq \tau n$ bits of communication.

It remains to prove the claim. We first consider the **NO** case. $\mathcal{N}'(n)$ and $\mathcal{N}_{q,\alpha,T}^{\Pi}(n)$ are both sampled by independently sampling T hypermatchings from $\mathcal{M}_{k,\alpha}(n)$ and then (independently) selecting some subset of k -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. For each $t \in [T]$, Prot_t adds a hyperedge $\mathbf{j}(t, i)$ iff $\mathbf{z}_{t,i} = \boldsymbol{\pi}$. But in the **NO** case (even conditioned on all other $\mathbf{z}_{t,i}$'s, on the hidden partition \mathbf{b} , and on $\mathbf{j}(t, i)$ itself), $\mathbf{z}_{t,i}$ is a uniform value in $[q]^k$, and hence $\mathbf{j}(t, i)$ is added to the instance with probability $1/q^k$. This is exactly how we defined $\mathcal{N}_{q,\alpha,T}^{\Pi}(n)$ to sample constraints.

Similarly, in the **YES** case, we consider the sampled q -partition $\mathbf{b} = (b_1, \dots, b_n) \in [q]^n$ and a hyperedge $\mathbf{j}(t, i) = (j_1, \dots, j_k)$. In this

case, by the definition of IRMD, we have $\mathbf{z}_{t,i} = \mathbf{b}|_{\mathbf{j}(t,i)} +_q \mathbf{v}^{(c)}$ where $c \sim [q]$ is uniform and $\mathbf{v}^{(c)} = (c, \dots, c)$. Hence, Prot_t will add $\mathbf{j}(t, i)$ iff $\mathbf{b}|_{\mathbf{j}(t,i)} = \boldsymbol{\pi} +_q \mathbf{v}^{(c')}$ where $c' \in [q]$ is the unique value such that $c' +_q c = q$. Consider the event \mathcal{E}_1 that there exists any $c' \in [q]$ such that $\mathbf{b}|_{\mathbf{j}(t,i)} = \boldsymbol{\pi} +_q \mathbf{v}^{(c')}$ and the event \mathcal{E}_2 that $c +_q c' = q$. Note that if \mathcal{E}_1 does not occur, then $\mathbf{j}(t, i)$ is never added, while if \mathcal{E}_1 does occur, with probability $1/q$ over the choice of c , \mathcal{E}_2 occurs and hence $\mathbf{j}(t, i)$ is added. (Again, this holds even conditioned on all other $\mathbf{z}_{t,i'}$'s, on $\mathbf{j}(t, i)$, and on \mathbf{b} .) This is exactly how we defined $\mathcal{Y}_{q,\alpha,T}^{\Pi,\boldsymbol{\pi}}(n)$ to sample constraints. \square

A. Good approximations in $\tilde{O}(n)$ space via sparsification

In this appendix, we prove [Theorem 1.3](#), which gives a simple $(1 - \epsilon)$ -approximation algorithm for all **Max-OCSP** problems given $\tilde{O}(n)$ space. To prove [Theorem 1.3](#), we first develop a simple sampling lemma that states that sparsifying down to $\Omega(n \log n / \epsilon^2)$ constraints preserves the value of a **Max-OCSP** instance up to an additive $\pm \epsilon$:

LEMMA A.1 ($\tilde{O}(n)$ -space sampling lemma). *Let $\mathcal{F} = \bigcup_{k \in \mathbb{N}} \{\Pi : \mathfrak{S}_k \rightarrow \{0, 1\}\}$ denote the (infinite) family of all ordering predicates. For all $\epsilon > 0$ and sufficiently large n , let Ψ be an instance of **Max-OCSP**(\mathcal{F}) on n variables. Let $\tilde{m} \geq 10n \log n / \epsilon^2$. Consider the random instance $\tilde{\Psi}$ with n variables and \tilde{m} constraints which is sampled by sampling each constraint independently and uniformly from the constraints of Ψ . Then w.p. 99% over the choice of $\tilde{\Psi}$, $\text{ocsp-val}_{\Psi} - \epsilon \leq \text{ocsp-val}_{\tilde{\Psi}} \leq \text{ocsp-val}_{\Psi} + \epsilon$.*

PROOF. First, fix an assignment $\boldsymbol{\sigma} \in \mathfrak{S}_n$ to $\tilde{\Psi}$. Let $\mathcal{E}_{\boldsymbol{\sigma}}$ denote the event that $|\text{ocsp-val}_{\tilde{\Psi}}(\boldsymbol{\sigma}) - \text{ocsp-val}_{\Psi}(\boldsymbol{\sigma})| \geq \epsilon$. We claim that

$$\Pr[\mathcal{E}_{\boldsymbol{\sigma}}] \leq 2 \exp(-100n \log n).$$

Indeed, for $j \in [\tilde{m}]$, let C_j denote the j -th constraint sampled when sampling $\tilde{\Psi}$, and let X_j denote the indicator for the event that C_j is satisfied by $\boldsymbol{\sigma}$. So $\tilde{m} \cdot \text{ocsp-val}_{\tilde{\Psi}}(\boldsymbol{\sigma}) = \sum_{i=1}^{\tilde{m}} X_i$. Multiplying both

sides of the desired inequality by \tilde{m} , \mathcal{E}_σ is the event that $|\sum_{i=1}^{\tilde{m}} X_i - \tilde{m} \cdot \text{ocsp-val}_\Psi(\sigma)| \geq \epsilon \tilde{m}$. Since C_j is a uniform constraint from Ψ , $\mathbb{E}[X_j] = \text{ocsp-val}_\Psi$. Since the C_j 's are sampled independently, by the Chernoff bound (Lemma 2.2) with $\eta = \epsilon$ and $p = \text{ocsp-val}_\Psi$ we have

$$\begin{aligned} \Pr \left[\left| \sum_{i=1}^{\tilde{m}} X_i - \tilde{m} \cdot \text{ocsp-val}_\Psi \right| \geq \epsilon \tilde{m} \right] &\leq 2 \exp(-2\epsilon^2 \tilde{m} / \text{ocsp-val}_\Psi) \\ &\leq 2 \exp(-2\epsilon^2 \tilde{m}). \end{aligned}$$

We assumed $\tilde{m} \geq 10n \log n / \epsilon^2$. Therefore, the RHS is at most $2 \exp(-100n \log n)$, as desired.

Now, let $\mathcal{E} = \bigcup_{\sigma \in \mathfrak{S}_n} \mathcal{E}_\sigma$ denote the event that \mathcal{E}_σ occurs for any $\sigma \in \mathfrak{S}_n$. To bound $\Pr[\mathcal{E}]$, we use a union bound:

$$\Pr[\mathcal{E}] < 2|\mathfrak{S}_n| \exp(-100n \log n).$$

But by Stirling's approximation (Lemma 2.3), $|\mathfrak{S}_n| \leq 3\sqrt{n}(n/e)^n$, and $\exp(-100n \log n) \geq n^{-100n}$ (since $\log n > \ln n$), so the error probability is $(3\sqrt{n}(n/e)^n)(2n^{-100n}) < 6n^{-99n+\frac{1}{2}}$ which is less than $1/100$ for sufficiently large n .

Finally, we show that conditioned on \mathcal{E} , the desired inequality $\text{ocsp-val}_\Psi - \epsilon \leq \text{ocsp-val}_{\tilde{\Psi}} \leq \text{ocsp-val}_\Psi + \epsilon$ holds. Indeed, we have

$$\begin{aligned} \text{ocsp-val}_{\tilde{\Psi}} &= \max_{\sigma \in \mathfrak{S}_n} \text{ocsp-val}_{\tilde{\Psi}}(\sigma) \\ &\leq \max_{\sigma \in \mathfrak{S}_n} (\text{ocsp-val}_\Psi(\sigma) + \epsilon) = \text{ocsp-val}_\Psi + \epsilon \end{aligned}$$

and similarly for $\text{ocsp-val}_{\tilde{\Psi}} \geq \text{ocsp-val} - \epsilon$. □

REMARK A.2. For OCSPs, the $\log n$ factor in \tilde{m} is required because the solution space has size $\Omega((n/e)^n)$ (by Stirling's approximation) and we take a union bound over all solutions. In contrast, for standard CSPs over an alphabet size of q , the solution space has size only q^n , and in the analogous analysis, there would only be a $\log q$ factor in \tilde{m} .

Given Lemma A.1, we can now prove Theorem 1.3:

PROOF (Proof of [Theorem 1.3](#)). Let $\epsilon' = \epsilon\rho(\Pi)/2$, and let $\tilde{m} = \lceil 10n \log n / (\epsilon')^2 \rceil$. Consider the following streaming algorithm to sample an instance $\tilde{\Psi}$ on n variables and \tilde{m} constraints: Initialize a buffer $(C_1, \dots, C_{\tilde{m}})$ of constraints, and then, when the i -th constraint C of Ψ arrives, set $C_j \leftarrow C$ with probability $1/i$ (otherwise C_j remains the same) independently for each $j \in [\tilde{m}]$. After the stream, let $\tilde{\Psi}$ denote the instance formed by $(C_1, \dots, C_{\tilde{m}})$, and output $\text{ocsp-val}_{\tilde{\Psi}} - \epsilon'$.

After all constraints of Ψ arrive, each constraint C_j is an independent, uniformly chosen constraint from Ψ , and so we can apply [Lemma A.1](#) to $\tilde{\Psi}$ to deduce that w.h.p. $\text{ocsp-val}_{\Psi} - \epsilon' \leq \text{ocsp-val}_{\tilde{\Psi}} \leq \text{ocsp-val}_{\Psi} + \epsilon'$. Conditioning on this event, we deduce that

$$\text{ocsp-val}_{\Psi} - 2\epsilon' \leq \text{ocsp-val}_{\tilde{\Psi}} \leq \text{ocsp-val}_{\Psi}.$$

The LHS is

$$\text{ocsp-val}_{\Psi} - 2\epsilon' = \text{ocsp-val}_{\Psi} - \epsilon\rho(\Pi) \geq (1 - \epsilon)\text{ocsp-val}_{\Psi}$$

since $\text{ocsp-val}_{\Psi} \geq \rho(\Pi)$, yielding the desired conclusion. \square

Acknowledgements

A previous version of this paper appeared in APPROX 2021 ([Singer et al. 2021a](#)). The results starting from that version improve on an earlier version of the paper ([Singer et al. 2021b](#)) that gave only $\Omega(\sqrt{n})$ space lower bounds for all OCSPs. Our improvement to $\Omega(n)$ space lower bounds comes by invoking the more recent results of [Chou et al. \(2022b\)](#), whereas our previous version used the strongest lower bounds for CSPs that were available at the time from an earlier work of [Chou et al. \(2024\)](#).⁷

We would like to thank the anonymous referees at *Computational Complexity* and APPROX for their helpful comments.

⁷The conference version of ([Chou et al. 2024](#)) appeared in 2021. The results in ([Chou et al. 2024](#)) are quantitatively weaker for the problems considered in ([Chou et al. 2022b](#)), 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 ([Chou et al. 2022b](#)) suffices.

N.G.S. was supported by an NSF Graduate Research Fellowship (Award DGE2140739). Work was done in part when swa hor was an undergraduate student at Harvard University.

M.S. was supported in part by a Simons Investigator Award and NSF Award CCF 1715187.

S.V. was supported in part by a Google Ph.D. Fellowship, a Simons Investigator Award to Madhu Sudan, and NSF Award CCF 2152413. Work was done in part when the author was an graduate student at Harvard University.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Funding Open access funding provided by Carnegie Mellon University.

References

PER AUSTRIN, RAJSEKAR MANOKARAN & CENNY WENNER (2015). On the NP-hardness of Approximating Ordering-Constraint Satisfaction Problems. *Theory of Computing* **11**, 257–283. Conference version in APPROX 2013.

AMEY BHANGALE & SUBHASH KHOT (2019). UG-Hardness to NP-Hardness by Losing Half. In *34th Computational Complexity Conference (CCC 2019)*, volume 137 of *LIPICS*. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, New Brunswick, NJ, USA, August 18-20, 2019.

AMIT CHAKRABARTI, PRANTAR GHOSH, ANDREW MCGREGOR & SOFYA VOROTNIKOVA (2020). Vertex Ordering Problems in Directed Graph Streams. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2020)*, 1786–1802. Society for Industrial and Applied Mathematics, Salt Lake City, UT, USA, January 5-9, 2020.

VAGGOS CHATZIAFRATIS & KONSTANTIN MAKARYCHEV (2023). Triplet Reconstruction and All Other Phylogenetic CSPs Are Approximation Resistant. In *IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS 2023)*, 253–284. IEEE Computer Society, Santa Cruz, CA, USA, November 6-9, 2023.

BENNY CHOR & MADHU SUDAN (1998). A Geometric Approach to Betweenness. *SIAM Journal on Discrete Mathematics* **11**(4), 511–523. Conference version in Algorithms, ESA 1995.

CHI-NING CHOU, ALEXANDER GOLOVNEV, AMIRBEHSHAD SHAHRASBI, MADHU SUDAN & SANTHOSHINI VELUSAMY (2022a). Sketching Approximability of (Weak) Monarchy Predicates. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX 2022)*, AMIT CHAKRABARTI & CHAITANYA SWAMY, editors, volume 245 of *LIPICS*, 35:1–35:17. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, virtual, September 19-21, 2022.

CHI-NING CHOU, ALEXANDER GOLOVNEV, MADHU SUDAN, AMEYA VELINGKER & SANTHOSHINI VELUSAMY (2022b). Linear Space Streaming Lower Bounds for Approximating CSPs. In *Proceedings of the 54th Annual ACM Symposium on Theory of Computing (STOC 2022)*. Association for Computing Machinery, Rome, Italy, June 20-24, 2022.

CHI-NING CHOU, ALEXANDER GOLOVNEV, MADHU SUDAN & SANTHOSHINI VELUSAMY (2021). Approximability of All Boolean CSPs with Linear Sketches. URL <https://arxiv.org/abs/2102.12351v7>. Subset of results in [Chou et al. \(2024\)](#).

CHI-NING CHOU, ALEXANDER GOLOVNEV, MADHU SUDAN & SANTHOSHINI VELUSAMY (2024). Approximability of All Finite CSPs with Linear Sketches. *Journal of the ACM Just Accepted*. Conference version in FOCS 2021.

CHI-NING CHOU, ALEXANDER GOLOVNEV & SANTHOSHINI VELUSAMY (2020). 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)*, 330–341. IEEE Computer Society, virtual, November 16-19, 2020.

DMITRY GAVINSKY, JULIA KEMPE, IORDANIS KERENIDIS, RAN RAZ & RONALD DE WOLF (2008). Exponential Separation for One-Way Quantum Communication Complexity, with Applications to Cryptography. *SIAM Journal on Computing* **38**(5), 1695–1708. ISSN 0097-5397. Conference version in STOC 2007.

VENKATESAN GURUSWAMI, JOHAN HÅSTAD, RAJSEKAR MANOKARAN, PRASAD RAGHAVENDRA & MOSES CHARIKAR (2011). Beating the Random Ordering Is Hard: Every Ordering CSP Is Approximation Resistant. *SIAM Journal on Computing* **40**(3), 878–914. Conference version in FOCS 2008.

VENKATESAN GURUSWAMI & RUNZHOU TAO (2019). Streaming Hardness of Unique Games. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX 2019)*, DIMITRIS ACHLIOPTAS & LÁSZLÓ A. VÉGH, editors, volume 145 of *LIPICS*, 5:1–5:12. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, Cambridge, MA, USA, September 20-22, 2019.

VENKATESAN GURUSWAMI, AMEYA VELINGKER & SANTHOSHINI VELUSAMY (2017). Streaming Complexity of Approximating Max 2CSP and Max Acyclic Subgraph. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX 2017)*, KLAUS JANSEN, JOSÉ D. P. ROLIM, DAVID WILLIAMSON & SANTOSH S. VEMPALA, editors, volume 81 of *LIPICS*, 8:1–8:19. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, Berkeley, CA, USA, August 16-18, 2017.

MICHAEL KAPRALOV, SANJEEV KHANNA & MADHU SUDAN (2015). Streaming Lower Bounds for Approximating MAX-CUT. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2015)*, 1263–1282. Society for Industrial and Applied Mathematics, San Diego, CA, USA, January 4-6, 2015.

MICHAEL KAPRALOV, SANJEEV KHANNA, MADHU SUDAN & AMEYA VELINGKER (2017). $(1 + \omega(1))$ -Approximation to MAX-CUT Requires

Linear Space. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2017)*, 1703–1722. Society for Industrial and Applied Mathematics, Barcelona, Spain, January 16-19, 2017.

MICHAEL KAPRALOV & DMITRY KRACHUN (2019). An Optimal Space Lower Bound for Approximating MAX-CUT. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC 2019)*, 277–288. Association for Computing Machinery, Phoenix, AZ, USA, June 23-26, 2019.

RICHARD M. KARP (1972). Reducibility among Combinatorial Problems. In *Complexity of Computer Computations*, R.E. MILLER, J.W. THATCHER & J.D. BOHLINGER, editors, The IBM Research Symposia Series (IRSS), 85–103. Springer.

SUBHASH KHOT (2002). On the Power of Unique 2-Prover 1-Round Games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC 2002)*, 767–775. Association for Computing Machinery, Québec, Canada, May 19-21, 2002.

DMITRY KOGAN & ROBERT KRAUTHGAMER (2015). Sketching Cuts in Graphs and Hypergraphs. In *Proceedings of the 6th Annual Conference on Innovations in Theoretical Computer Science (ITCS 2015)*, 367–376. Association for Computing Machinery, Rehovot, Israel, January 11-13, 2015.

ALANTHA NEWMAN (2000). *Approximating the Maximum Acyclic Subgraph*. Master’s thesis, Massachusetts Institute of Technology.

JAROSLAV OPATRNÝ (1979). Total Ordering Problem. *SIAM Journal on Computing* 8(1), 111–114.

RAGHUVANSH R. SAXENA, NOAH SINGER, MADHU SUDAN & SANTHOSHINI VELUSAMY (2023a). Improved Streaming Algorithms for Maximum Directed Cut via Smoothed Snapshots. In *63rd Annual Symposium on Foundations of Computer Science (FOCS 2023)*, 855–870. IEEE Computing Society, Santa Cruz, CA, USA, November 6-9, 2023.

RAGHUVANSH R. SAXENA, NOAH G. SINGER, MADHU SUDAN & SANTHOSHINI VELUSAMY (2023b). Streaming Complexity of CSPs with Randomly Ordered Constraints. In *Proceedings of the 2023 Annual*

ACM-SIAM Symposium on Discrete Algorithms (SODA 2023), 4083–4103. Society for Industrial and Applied Mathematics, Florence, Italy, January 22–25, 2023.

NOAH SINGER (2022). *On Streaming Approximation Algorithms for Constraint Satisfaction Problems*. Undergraduate thesis, Harvard University, Cambridge, MA. URL <https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37371750>.

NOAH SINGER, MADHU SUDAN & SANTHOSHINI VELUSAMY (2021a). Streaming Approximation Resistance of Every Ordering CSP. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX 2021)*, MARY WOOTTERS & LAURA SANITÀ, editors, volume 207 of *LIPICS*, 17:1–17:19. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, virtual, August 16–18, 2021.

NOAH SINGER, MADHU SUDAN & SANTHOSHINI VELUSAMY (2021b). Streaming Approximation Resistance of Every Ordering CSP. Original version of this paper; proved only $o(\sqrt{n})$ space lower bounds.

MADHU SUDAN (2022). Streaming and Sketching Complexity of CSPs: A Survey (Invited Talk). In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, MIKOŁAJ BOJAŃCZYK, EMANUELA MERELLI & DAVID P. WOODRUFF, editors, volume 229 of *LIPICS*, 5:1–5:20. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, Paris, France, July 4–8, 2022.

SANTHOSHINI VELUSAMY (2023). *Approximability of Constraint Satisfaction Problems in the Streaming Setting*. Ph.D. thesis, Harvard University, Cambridge, MA. URL <https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37377430>.

ELAD VERBIN & WEI YU (2011). 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)*, 11–25. Society for Industrial and Applied Mathematics, San Francisco, CA, USA, January 23–25, 2011.

NOAH G. SINGER
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15232, USA
ngsinger@cs.cmu.edu
<https://noahsinger.org>

MADHU SUDAN
School of Engineering and Applied
Sciences
Harvard University
Cambridge, MA 02138, USA
madhu@cs.harvard.edu
<http://madhu.seas.harvard.edu>

SANTHOSHINI VELUSAMY
Toyota Technological Institute
Chicago, IL 60637, USA
santhoshini@ttic.edu
<https://sites.google.com/view/santhoshinivelusamy>