Sketching Approximability of All Finite CSPs

CHI-NING CHOU, School of Engineering and Applied Sciences, Harvard University, Cambridge, USA
ALEXANDER GOLOVNEV, Department of Computer Science, Georgetown University, Washington, USA
MADHU SUDAN, School of Engineering and Applied Sciences, Harvard University, Cambridge, USA
SANTHOSHINI VELUSAMY, Toyota Technological Institute, Chicago, USA

A constraint satisfaction problem (CSP), Max-CSP($\mathcal{F}$), is specified by a finite set of constraints $\mathcal{F} \subseteq \{[q]^k \to \{0,1\}\}$ for positive integers q and k. An instance of the problem on n variables is given by m applications of constraints from $\mathcal{F}$ to subsequences of the n variables, and the goal is to find an assignment to the variables that satisfies the maximum number of constraints. In the (γ,β)-approximation version of the problem for parameters $0 \le \beta < \gamma \le 1$, the goal is to distinguish instances where at least γ fraction of the constraints can be satisfied from instances where at most β fraction of the constraints can be satisfied.

In this work, we consider the approximability of this problem in the context of sketching algorithms and give a dichotomy result. Specifically, for every family $\mathcal{F}$ and every $\beta < \gamma$, we show that either a linear sketching algorithm solves the problem in polylogarithmic space or the problem is not solvable by any sketching algorithm in $o(\sqrt{n})$ space. In particular, we give non-trivial approximation algorithms using polylogarithmic space for infinitely many constraint satisfaction problems.

We also extend previously known lower bounds for general streaming algorithms to a wide variety of problems, and in particular the case of $q = k = 2$, where we get a dichotomy, and the case when the satisfying assignments of the constraints of $\mathcal{F}$ support a distribution on $[q]^k$ with uniform marginals.

Prior to this work, other than sporadic examples, the only systematic classes of CSPs that were analyzed considered the setting of Boolean variables $q = 2$, binary constraints $k = 2$, and singleton families $|\mathcal{F}| = 1$ and only considered the setting where constraints are placed on literals rather than variables.

Our positive results show wide applicability of bias-based algorithms used previously by [47] and [41], which we extend to include richer norm estimation algorithms, by giving a systematic way to discover biases. Our negative results combine the Fourier analytic methods of [56], which we extend to a wider class of CSPs, with a rich collection of reductions among communication complexity problems that lie at the heart of the negative results. In particular, previous works used Fourier analysis over the Boolean cube to initiate their results and the results seemed particularly tailored to functions on Boolean literals (i.e., with negations). Our techniques surprisingly allow us to get to general q-ary CSPs without negations by appealing to the same Fourier analytic starting point over Boolean hypercubes.

C.-N. Chou is supported by NSF Awards CCF 1565264 and CNS 1618026.
M. Sudan is supported in part by a Simons Investigator Award and NSF Awards CCF 1715187 and CCF 2152413.
S. Velusamy is supported in part by a Google Ph.D. Fellowship, a Simons Investigator Award to Madhu Sudan, and NSF Awards CCF 1715187 and CCF 2152413.
Authors' addresses: C.-N. Chou and M. Sudan, School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA; e-mails: chiningchou@g.harvard.edu, madhu@cs.harvard.edu; A. Golovnev, Department of Computer Science, Georgetown University, Washington, DC, USA; e-mail: alexgolovnev@gmail.com; S. Velusamy, Toyota Technological Institute, Chicago, IL, USA; e-mail: santhoshini@ttic.edu.

ACM 0004-5411/2024/04-ART15
https://doi.org/10.1145/3649435

CCS Concepts: • **Theory of computation → Sketching and sampling**; **Communication complexity**; **Approximation algorithms analysis;**

Additional Key Words and Phrases: Streaming algorithms, communication lower bound, inapproximability, constraint satisfaction problem

ACM Reference Format:
Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, and Santhoshini Velusamy. 2024. Sketching Approximability of All Finite CSPs. *J. ACM* 71, 2, Article 15 (April 2024), 74 pages. https://doi.org/10.1145/3649435

1 INTRODUCTION

In this article we give a complete characterization of the approximability of **constraint satisfaction problems (CSPs)** by sketching algorithms. We describe the exact class of problems below and give a brief history of previous work before giving our results.

1.1 CSPs

For positive integers q and k, a q-ary CSP is given by a (finite) set of constraints $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$. A constraint C on $x_1, \ldots, x_n$ is given by a pair $(f, \mathbf{j})$, with $f \in \mathcal{F}$ and $\mathbf{j} = (j_1, \ldots, j_k) \in [n]^k$, where the coordinates of $\mathbf{j}$ are all distinct.[1] An assignment $\mathbf{b} \in [q]^n$ satisfies $C = (f, \mathbf{j})$ if $f(b_{j_1}, \ldots, b_{j_k}) = 1$. To every finite set $\mathcal{F}$, we associate a maximization problem Max-CSP($\mathcal{F}$) that is defined as follows: An instance Ψ of Max-CSP($\mathcal{F}$) consists of m constraints $C_1, \ldots, C_m$ applied to n variables $x_1, x_2, \ldots, x_n$ along with m non-negative integer weights $w_1, \ldots, w_m$. The value of an assignment $\mathbf{b} \in [q]^n$ on an instance $\Psi = (C_1, \ldots, C_m; w_1, \ldots, w_m)$, denoted $\mathrm{val}_\Psi(\mathbf{b})$, is the fraction of weight of constraints satisfied by $\mathbf{b}$. The goal of the *exact* problem is to compute the maximum, over all assignments, of the value of the assignment on the input instance, i.e., to compute, given Ψ, the quantity $\mathrm{val}_\Psi = \max_{\mathbf{b} \in [q]^n} \{\mathrm{val}_\Psi(\mathbf{b})\}$.

In this work we consider the approximation version of Max-CSP($\mathcal{F}$), which we study in terms of the "gapped promise problems." Specifically, given $0 \le \beta < \gamma \le 1$, the (γ, β)-approximation version of Max-CSP($\mathcal{F}$), abbreviated (γ, β)-Max-CSP($\mathcal{F}$), is the task of distinguishing between instances from $\Gamma = \{\Psi | \, \mathrm{opt}(\Psi) \ge \gamma\}$ and instances from $B = \{\Psi | \, \mathrm{opt}(\Psi) \le \beta\}$. It is well known that this distinguishability problem is a refinement of the usual study of approximation, which usually studies the ratio of γ/β for tractable versions of (γ, β)-Max-CSP($\mathcal{F}$). See Proposition 2.5 for a formal statement in the context of streaming approximability of Max-CSP($\mathcal{F}$) problems.

1.2 Streaming Algorithms

We study the complexity of (γ, β)-Max-CSP($\mathcal{F}$) in the setting of randomized streaming algorithms. Here, an instance $\Psi = (C_1, \ldots, C_m)$ is presented as a stream $\sigma_1, \sigma_2, \ldots, \sigma_m$, with $\sigma_i = (f(i), \mathbf{j}(i))$ representing the ith constraint. We study the space required to solve the (γ, β)-approximation version of Max-CSP($\mathcal{F}$). Specifically, we consider algorithms that are allowed to use internal randomness and s bits of space. The algorithms output a single bit at the end. They are said to solve the (γ, β)-approximation problem correctly if they output the correct answer with probability at least $2/3$ (i.e., they err with probability at most $1/3$).

A sketching algorithm is a special class of a streaming algorithm, where the algorithm's output is determined by a small sketch it produces of the input stream, and the sketch itself has the property

[1] To allow repeated variables in a constraint, note that one can turn $\mathcal{F}$ into $\mathcal{F}'$ by introducing new functions corresponding to all the possible replications of variables of functions in $\mathcal{F}$.

that the sketch of the concatenation of two streams can be computed from the sketches of the two component streams. (See Definition 2.3 for a formal definition.)

For over a decade now, there has been active research on designing streaming and sketching algorithms for combinatorial optimization problems in various settings. See, for example:

— [6, 7, 10–12, 14, 15, 18, 20, 24, 33, 44, 47, 54, 57, 58] for results in the *single-pass* setting, where the algorithm is allowed only a single pass through the stream;
— [4, 9, 13, 16, 17, 30–32, 61] for results on *multi-pass* streaming algorithms, which are allowed a constant number of passes through the stream; and
— [5, 8, 19, 34, 55, 56] for results in the *random-ordering* setting, where the input is randomly shuffled in the stream.

We primarily focus on single-pass streaming algorithms, and our main dividing line is between algorithms that work with space $\text{poly}(\log n)$ versus algorithms that require space at least n^{ε} for some $\varepsilon > 0$. In informal usage we refer to a streaming problem as "easy" if it can be solved with polylogarithmic space (the former setting) and "hard" if it requires polynomial space for sketching algorithms. We note that all the positive results (algorithms) given in this article are linear sketching algorithms, which are more restrictive than general sketching algorithms. We also note that many of our lower bounds work against general streaming algorithms, and we elaborate on this in Section 1.4.

1.3 Past Work

To our knowledge, streaming algorithms for CSPs have not been investigated extensively. Here we cover the few results we are aware of, all of which consider only the Boolean ($q = 2$) setting. On the positive side, it may be surprising that there exists any non-trivial algorithm at all. (Briefly, we say that an algorithm that outputs a constant value independent of the input is "trivial.")

It turns out that there do exist some non-trivial approximation algorithms for Boolean CSPs. This was established by the work of Guruswami et al. [47], who, in our notation, gave an algorithm for the $(\gamma, 2\gamma/5 - \varepsilon)$-approximation version of Max-2AND, for every $\gamma \in [0, 1]$ (Max-2AND is the Max-CSP($\mathcal{F}$) problem corresponding to $\mathcal{F} = \{f_{c,d} | c, d \in \{0, 1\}\}$, where $f_{c,d}(a, b) = 1$ if $a = c$ and $b = d$ and $f_{c,d}(a, b) = 0$ otherwise). A central ingredient in their algorithm is the ability of streaming algorithms to approximate the ℓ_1 norm of a vector in the turnstile setting, which allows them to estimate the "bias" of n variables (how often they occur positively in constraints, as opposed to negatively). Subsequently, the work of Chou et al. [41] further established the utility of such algorithms, which we refer to as bias-based algorithms, by giving optimal algorithms for all Boolean CSPs on two variables. In particular, they give a better (optimal!) analysis of bias-based algorithms for Max-2AND and show that Max-2SAT also has an optimal algorithm based on bias.

On the negative side, the problem that has been explored the most is Max-CUT, or in our language Max-2XOR, which corresponds to $\mathcal{F} = \{f\}$ and $f(x, y) = x \oplus y$. Kapralov et al. [56] showed that Max-2XOR does not have a $(1, 1/2 + \varepsilon)$-approximation algorithm using $o(\sqrt{n})$-space, for any $\varepsilon > 0$. This was subsequently improved upon by Kapralov et al. [57] and Kapralov and Krachun [58]. The final paper [58] completely resolves Max-CUT showing that $(1, 1/2 + \varepsilon)$-approximation for these problems requires $\Omega(n)$ space. Turning to other problems, the work by [47] notices that the $(1, 1/2 + \varepsilon)$-inapproximability of Max-2XOR immediately yields $(1, 1/2 + \varepsilon)$-inapproximability of Max-2AND as well. In [41] more sophisticated reductions are used to improve the inapproximability result for Max-2AND to a $(\gamma, 4\gamma/9 + \varepsilon)$-inapproximability for some positive γ, which turns out to be the optimal ratio by their algorithm and analysis. As noted earlier, their

work gives algorithms for Max-CSP($\mathcal{F}$) for all $\mathcal{F} \subseteq \{f : \{0,1\}^2 \to \{0,1\}\}$,[2] which are optimal if $\mathcal{F}$ is closed under literals (i.e., if $f(x,y) \in \mathcal{F}$, then so are the functions $f(\neg x, y)$ and $f(\neg x, \neg y)$).

1.4 Results

Our main theorem is a decidable dichotomy theorem for (γ,β)-Max-CSP($\mathcal{F}$) with sketching algorithms.

Theorem 1.1 (Succinct Version). *For every $q, k \in \mathbb{N}$, $0 \le \beta < \gamma \le 1$ and $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$, one of the following two conditions holds: Either (γ,β)-Max-CSP($\mathcal{F}$) can be solved with $O(\log^3 n)$ space by linear sketches or, for every $\varepsilon > 0$, every sketching algorithm for $(\gamma-\varepsilon,\beta+\varepsilon)$-Max-CSP($\mathcal{F}$) requires $\Omega(\sqrt{n})$-space. Furthermore, there is a polynomial space algorithm that decides which of the two conditions holds, given γ, β, and $\mathcal{F}$.*

Theorem 1.1 combines the more detailed Theorem 3.3 with the polynomial space decidability coming from Theorem 3.4.

The first-order message of the theorem statement is that the known non-trivial approximation algorithms for streaming CSPs (i.e., the algorithms for Max-2AND and Max-2SAT from [36]) can *potentially* be extended to infinitely many problems. To confirm this potential, one needs to be able to identify an infinite subclass of CSPs for which the decidability condition for non-trivial (γ,β) pairs can be analytically shown to be "solvable in polylog space." While we do not find such explicit families in this article, subsequent work has succeeded in getting such an analysis [26, 39]. We elaborate further on this in Section 1.7 but note that the subsequent work [26] shows that Max-kAND (the generalization of Max-2AND to k literals) for every $k \in \mathbb{N}$ has non-trivial approximation algorithms, thereby confirming this potential! We believe this in itself may be a surprising result to some given that the bias-based algorithms and their analysis did appear tailored to the structure of Max-2AND and Max-2SAT.

The next main message is that when the class of algorithms we use cannot be used to solve a (γ,β)-approximation problem, then there is an inherent hurdle and no sketching-based algorithm can work. Indeed, in many cases our results rule out completely general streaming algorithms, though we do not get a dichotomy for general streaming.

Finally, we highlight some of the descriptive strengths of the class of problems captured by Theorem 1.1 above; we note that previous works could only handle the special case where (1) $\mathcal{F}$ contains a single function f, (2) $q = 2$, (3) constraints are placed on "literals" rather than variables, and (4) they only capture a single-parameter approximation problem, not the more refined two-parameter ("gapped") version considered in this work. The difference in expressivity due to conditions (1) through (3) is significant: To capture a problem such as Max-3SAT, one needs to go beyond restriction (1) to allow different constraints for clauses of length 1, 2, and 3. This is a quantitatively significant restriction in that the approximability in this case is "smaller" than that of Max-CSP(f) for any of the constituent functions. So hard instances do involve a mix of constraints! The lack of expressiveness induced by the second restriction of Boolean variables is perhaps more obvious. Natural examples of CSPs that require larger alphabets are Max-q-Coloring and Unique Games. Next we turn to restriction (3)—the inability to capture CSP problems over variables. This restriction prevents previous works from capturing some very basic problems including Max-CUT and Max-DICUT. Furthermore, the notion of "literals" is natural only in the setting of Boolean variables—so overcoming this restriction seems crucial to eliminating the restriction of the Booleanity of the variables. Notice that while for families with a single function $\mathcal{F} = \{f\}$,

[2]Note that when $q = 2$, we switch to using $\{0, 1\}$ or $\{-1, 1\}$ as the domain (as opposed to $\{1, 2\}$) depending on convenience.

going from constraints on literals to constraints on variables does not lead to greater expressivity, once we study Max-CSP($\mathcal{F}$) for all sets $\mathcal{F}$, the study does get formally richer. Finally, the two-parameter versions allow us to also understand the approximability of satisfiable and nearly satisfiable instances of Max-CSP, a quest that is quite common in the literature. (See, for instance, the works on robust satisfiability [22, 42, 63].)

In particular, Theorem 1.1 allows us also to capture the extreme case of hard problems where no "non-trivial" algorithms exist. Such problems are usually referred to as approximation-resistant problems. In the study of Boolean CSPs, with constraints placed on literals, "non-triviality" is defined as "beating a random assignment," and approximation resistance in the setting of polynomial time algorithms is a well-studied topic [21, 45, 48]. Extending the definition to the setting where constraints are placed on variables rather than literals requires some thought. We propose a definition in this article (see Definition 3.5) that uses the notion that algorithms outputting a constant value are trivial, and a problem is approximation resistant if beating this trivial algorithm is hard. Specifically, $\mathcal{F}$ is said to be approximation resistant if for every $\beta < \gamma$ either (γ, β)-Max-CSP($\mathcal{F}$) is solved by a "constant function" or it requires $n^{\Omega(1)}$ space. We then show how Theorem 1.1 (or its more detailed version Theorem 3.3) leads to a characterization of approximation resistance in the streaming setting as well. (See Theorem 3.8.)

As mentioned earlier, the results above (and in particular the negative results) apply only to sketching algorithms for streaming CSPs. For a general streaming algorithm, we get some partial results. To describe our next result, we define the notion of a function supporting a one-wise independent distribution. We say that f supports *one-wise independence* if there exists a distribution $\mathcal{D}$ supported on $f^{-1}(1)$ whose marginals are uniform on $[q]$. We say that $\mathcal{F}$ supports one-wise independence if every $f \in \mathcal{F}$ supports one-wise independence.

Theorem 1.2 (Informal). *If $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$ supports one-wise independence, then it is approximation resistant in the streaming setting.*

Theorem 1.2 is formalized as Theorem 3.12 in Section 3.3.2. We also give theorems capturing hardness in the streaming setting beyond the one-wise independent case. Stating the full theorem requires more notions (see Section 3.3.2), but as a consequence we get the following extension of theorem of [41].

Theorem 1.3. *Let $q = k = 2$. Then, for every family $\mathcal{F} \subseteq \{f : [q]^2 \to \{0,1\}\}$, and for every $0 \le \beta < \gamma \le 1$, at least one of the following always holds:*

(1) (γ, β)-Max-CSP($\mathcal{F}$) has an $O(\log^3 n)$-space linear sketching algorithm.
(2) For every $\varepsilon > 0$, every streaming algorithm that solves $(\gamma - \varepsilon, \beta + \varepsilon)$-Max-CSP($\mathcal{F}$) requires $\Omega(\sqrt{n})$ space. If $\gamma = 1$, then $(1, \beta + \varepsilon)$-Max-CSP($\mathcal{F}$) requires $\Omega(\sqrt{n})$ space.

Furthermore, for every $\ell \in \mathbb{N}$, there is an algorithm using space $\mathrm{poly}(\ell)$ that decides which of the two conditions holds given the truth-tables of functions in $\mathcal{F}$, and γ and β as ℓ-bit rationals.

Theorem 1.3 is proved in Section 3.3.2. [41] study the setting where constraints are applied to literals and $\mathcal{F}$ contains a single function and get a tight characterization of the approximability of Max-CSP($\mathcal{F}$).[3]

Our work extends theirs by allowing constraints to be applied only to variables, by allowing families of constraint functions, and by determining the complexity of every (γ, β)-Max-CSP($\mathcal{F}$) (and not just studying the optimal ratio of β/γ).

[3]By approximability of Max-CSP($\mathcal{F}$) we refer to the quantity $\inf_\beta \sup_\gamma \{\{\beta/\gamma\}\}$ over polylog space solvable (γ, β)-Max-CSP($\mathcal{F}$) problems.

For the sake of completeness we also give a simple characterization of the Max-CSP($\mathcal{F}$) problems that are solvable *exactly* in polylogarithmic space.

Theorem 1.4 (Succinct Version). *For every $q, k \in \mathbb{N}$ and $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$, the Max-CSP($\mathcal{F}$) problem is solvable exactly in deterministic logarithmic space if and only if there is a constant $\sigma \in [q]$ such that every satisfiable function in $\mathcal{F}$ is satisfied by the all σ-assignment. All remaining families $\mathcal{F}$ require $\Omega(n)$ space to solve exactly.*

The proof of this theorem is by elementary reductions from standard communication complexity problems and is included in Section 9.

This version: This version of the article subsumes the works [36–38]. The paper [36], now withdrawn, claimed a restriction of Theorem 1.1 in the streaming setting, but that version had an error and the status of Theorem 1.1 in [36] is currently open. [37] proves the results of this article for the special cases of $\mathcal{F} = \{f\}$, $q = 2$ and constraints being applied to literals rather than variables. [38] essentially contains the same results as this article but builds upon [37]. The conference version of [38] appeared in the proceedings of FOCS 2021 [35]. This article combines [37] and [38].

1.5 Contrast with Dichotomies in the Polynomial Time Setting

The literature on polynomial time dichotomies of Max-CSP(f) problems is vast. One broad family of results here [27, 71, 76] considers the exact satisfiability problems (corresponding to distinguishing between instances from $\{\Psi | \operatorname{opt}(\Psi) = 1\}$ and instances from $\{\Psi | \operatorname{opt}(\Psi) < 1\}$). Another family of results [21, 60, 67] considers the approximation versions of Max-CSP(f) and gets "near dichotomies" along the lines of this article—i.e., they either show that the (γ, β)-approximation is easy (in polynomial time) or, for every $\varepsilon > 0$, the $(\gamma - \varepsilon, \beta + \varepsilon)$-approximation version is hard (in some appropriate sense). Our work resembles the latter series of works both in terms of the nature of results obtained and the kinds of characterizations used to describe the "easy" and "hard" classes and in the proof approaches (though of course the sketching setting is much easier to analyze, allowing for simpler proofs overall and unconditional results). We summarize their results, giving comparisons to our theorem, and then describe a principal contrast.

In a seminal work, Raghavendra [67] gave a characterization of the polynomial time approximability of the Max-CSP(f) problems based on the unique games conjecture [59]. Our Theorem 1.1 is analogous to his theorem. A characterization of approximation-resistant functions is given by Khot et al. [60]. Our Theorem 1.2 is analogous to this. Austrin and Mossel [21] show that all functions supporting a pairwise independent distribution are approximation resistant. Our Theorem 3.12 is analogous to this theorem.

While our results run in parallel to the work on polynomial time approximability, our characterizations are not immediately comparable. Indeed, there are some significant differences, which we highlight below. Of course there is the obvious difference that our negative results are unconditional (and not predicated on a complexity theoretic assumption like the unique games conjecture or P≠NP). But more significantly our characterization is a bit more "explicit" than those of [67] and [60]. In particular, the former only shows decidability of the problem, which takes ε as an input (in addition to γ, β, and f) and distinguishes (γ, β)-approximable problems from $(\gamma - \varepsilon, \beta + \varepsilon)$-inapproximable problems. The running time of their decision procedure grows with $1/\varepsilon$. In contrast, our distinguishability is sharper and separates (γ, β)-approximability from "$\forall \varepsilon > 0$, $(\gamma - \varepsilon, \beta + \varepsilon)$-inapproximability," so our algorithm does not require ε as an input—it merely takes γ, β, and f as input. Indeed, this difference is key to the understanding of approximation resistance. Due to the stronger form of our main theorem (Theorem 1.1), our characterization of streaming

approximation resistance is explicit (decidable in PSPACE), whereas a decidable characterization of approximation resistance in the polynomial time setting seems to be still open.

Our characterizations also seem to differ from the previous versions in terms of the features being exploited to distinguish the two classes. This leads to some strange gaps in our knowledge. For instance, it would be natural to suspect that (conditional) inapproximability in the polynomial time setting should also lead to (unconditional) inapproximability in the streaming setting. But we do not have a formal theorem proving this. (Of course, if this were false, it would be a breakthrough result, giving a quasi-polynomial time (even polylog space) algorithm for the unique games!)

1.6 Overview of Our Analysis

At the heart of our characterization is a family of linear sketching algorithms for Max-CSP($\mathcal{F}$). We will describe this family soon, but the main idea of our proof is that if no algorithm in this family solves (γ, β)-Max-CSP($\mathcal{F}$), then we can extract a pair of instances, roughly a family of γ-satisfiable "yes" instances and a family of at most β-satisfiable "no" instances, that certify this inability. We then show how this pair of instances can be exploited as gadgets in a negative result. Up to this part, our approach resembles that in [67] (though of course, all the steps are quite different). The main difference is that we are able to use the structure of the algorithm and the lower bound construction to show that we can afford to consider only instances on k variables. (This step involves a non-trivial choice of definitions that we elaborate on shortly.) This bound on the number of variables allows us to get a "decidable" separation between approximable and inapproximable problems. Specifically, we show that the distinction between the approximable setting and the inapproximable one can be expressed by a quantified formula over the reals with a constant number of quantifiers over 2^k variables and equations—a problem that is known to be solvable in PSPACE. We give more details below. To simplify the discussion we consider a singleton function family $\mathcal{F} = \{f\}$. Extending to multiple functions is not much harder (though as stressed by the Max-3SAT example, this is not trivial either). We start by giving some intuition into our framework before actually describing the framework. We remark that while this intuition may be helpful, it is not necessary for any of our proofs.

Intuition. Our starting point is the belief that streaming algorithms working with polylogarithmic space can essentially extract the "bias profile" of an instance, while algorithms with much more (specifically $o(\sqrt{n})$) space cannot do much more. Here, by bias profile of an instance Φ on n variables, we mean the $n \times k$ matrix $B = B(\Phi)$, with $B_{i,j}$ representing the fraction of constraints of Φ that have x_i as the jth variable. If our belief were to be true, then the only obstacle to deciding (γ, β)-Max-CSP(f) in $o(\sqrt{n})$ space would be two instances Φ_Y and Φ_N on the same set of variables with $\mathsf{val}(\Phi_Y) \geq \gamma$ and $\mathsf{val}(\Phi_N) \leq \beta$ while the instances have the same bias profile, i.e., $B(\Phi_Y) = B(\Phi_N)$.

To convert our belief into a proof of Theorem 1.1, we need to do three things: (1) Given γ, β, and f, show that the existence of such a pair of instances Φ_Y and Φ_N can be decided (in finite time); (2) show that if no pair of such instances exist, then (γ, β)-Max-CSP(f) can be decided by a polylogarithmic space sketching algorithm; and (3) if such a pair of instances exists, then no $o(\sqrt{n})$ space sketching algorithm can solve (γ, β)-Max-CSP(f).

While step (3) ends up taking most of the technical work in this article, it is also perhaps the most believable. Roughly hard instances of arbitrary length can be extracted from Φ_Y and Φ_N by doing "random lifts"; i.e., creating many copies of each variable in Φ_Y and applying constraints randomly among these copies according to Φ_Y or Φ_N roughly preserves the values; and the fact that the bias profiles match can be converted into a hardness result for sketching algorithms using communication complexity-based arguments. We expand on this more below.

The less believable steps (in our estimate) are steps (1) and (2), and it turns out that understanding the challenge in (1) better leads to a solution to both steps. The challenge behind (1) is of course the fact that a priori the number of variables in Φ_Y or Φ_N cannot be bounded and so there is no finite upper bound on the time it would take to decide their existence. The key to resolving this is the fact (that we will argue below) that the information contained in Φ_Y and Φ_N can be compressed into smaller instances on kq variables.

To establish this, let us suppose (without loss of generality) that Φ_Y and Φ_N are instances on $n \times q$ variables $\{X_{i,\sigma}\}_{i\in[n],\sigma\in[q]}$. Further suppose the assignment that establishes $\mathrm{val}(\Phi_Y) \geq \gamma$ is the assignment $a_{i,\sigma} = \sigma$. For permutations $\pi_1, \ldots, \pi_q : [n] \to [n]$, let $\Phi_Y^{\pi_1,\ldots,\pi_q}$ be a copy of Φ_Y with variables renamed to $\{X_{\pi_\sigma(i),\sigma}\}$. Similarly define $\Phi_N^{\pi_1,\ldots,\pi_q}$. Note that renaming the variables preserves the values and the bias profiles still match, and furthermore the assignment that yields a value of γ to $\Phi_Y^{\pi_1,\ldots,\pi_q}$ is still $a_{i,\sigma} = \sigma$. Thus, if we now consider the instances $\widetilde{\Phi_Y}$ obtained by concatenating all the constraints of $\Phi_Y^{\pi_1,\ldots,\pi_q}$ over all choices of $\pi_1, \ldots, \pi_q$, and similarly define $\widetilde{\Phi_N}$, then the resulting instances still have matching bias profiles and they still satisfy $\mathrm{val}(\widetilde{\Phi_Y}) \geq \gamma$ and $\mathrm{val}(\widetilde{\Phi_Y}) \leq \beta$. The gain with all these transformations is that $\widetilde{\Phi_Y}$ and $\widetilde{\Phi_N}$ are very symmetric instances with only q equivalence classes of variables (as opposed to n general variables). And a random constraint just picks a uniform variable from an equivalence class, conditioned on picking a variable from that class, in any given position. (Recall that by our assumption, every constraint is applied on k *distinct* variables.) Thus, the instances $\widetilde{\Phi_Y}$ and $\widetilde{\Phi_N}$ are effectively given by a distribution supported on $[q]^k$, where the probability of $(\sigma_1, \ldots, \sigma_k)$ measures the frequency of constraints on k-tuples of variables of the form $(X_{*,\sigma_1}, \ldots, X_{*,\sigma_k})$.

Thus, the instances revealing the gap between γ and β are finitely specified (or at least are distributions over a finite space), but it is still unclear how to search for (specifications of) such instances of value at least γ or at most β. To address this challenge one may try to reduce the entire instance $\widetilde{\Phi_Y}$ into an "equivalent" instance on just q variables (by replacing all variables $X_{i,\sigma}$ for $i \in [n]$ with a single variable Z_i), but this may result in constraints where all variables are not distinct. To exclude this possibility we replace the collection of variables $X_{i,\sigma}$ with k variables $Z_{\ell,\sigma}$ for $\ell \in [k]$ and now compress $\widetilde{\Phi_Y}$ by replacing all occurrences of $X_{i,\sigma}$ as the ℓth variable in a constraint with $Z_{\ell,\sigma}$. This leads to a compressed instance Φ'_Y on just kq variables. We can do a similar reduction with $\widetilde{\Phi_N}$ to get an instance Φ'_N. These resulting instances also have matching bias profiles. The reduction in the variables ensures $\mathrm{val}(\Phi'_Y) \geq \gamma$ since the assignment $Z_{\ell,\sigma} = \sigma$ still satisfies a γ fraction of the constraints. However, it is no longer true that $\mathrm{val}(\Phi'_N) \leq \beta$. This is so since the assignment to a variable $Y_{i,\sigma}$ might depend on i, which was not a possibility considered when bounding $\mathrm{val}(\widetilde{\Phi_N})$. What we would like at this stage is a succinct way to capture the fact that if we try to reverse engineer $\widetilde{\Phi_N}$ from Φ'_N, then we would have $\mathrm{val}(\widetilde{\Phi_N}) \leq \beta$. It turns out one succinct way to capture this is to consider only those distributions on assignments to the variables $Z_{\ell,\sigma}$ that are independent across variables and furthermore the distributions of $Z_{\ell,\sigma}$ and $Z_{\ell',\sigma}$ are identical. If we require that Φ'_N has value at most β in expectation over all such distributions of assignments to its variables, then we effectively capture the constraint $\mathrm{val}(\widetilde{\Phi_N}) \leq \beta$.

Thus, the search for instances Φ_Y and Φ_N can be reduced to a search for instances Φ'_Y and Φ'_N on just kq variables whose bias profiles must match and whose values satisfy some constraints. Since the marginals of distributions supported on $[q]^k$ are captured by vectors in $[0,1]^{kq} \subseteq \mathbb{R}^{kq}$, we get that the space of marginals of all yes instances (of the special type we care about) is given by a subset of points in $\mathbb{R}^{kq}$, which we denote $K_\gamma^Y(\mathcal{F})$. Similarly, the space of the marginals of the no instances is also a subset of $\mathbb{R}^{kq}$, denoted $K_\beta^N(\mathcal{F})$. It turns out these sets are bounded, closed, and convex and actually described by some polynomial conditions. Thus, solving step (1) reduces

to the task of determining if $K_\gamma^Y(\mathcal{F})$ and $K_\beta^N(\mathcal{F})$ intersect. And when they do not intersect, the separating hyperplane gives us a clue on how to solve the problem from step (2), i.e., how to solve (γ, β)-Max-CSP($\mathcal{F}$) with polylogarithmic space.

To show that this framework works, we need to explain what our algorithms are, why they lead to these special instances when they fail, and how to use the failure of the algorithms (or equivalently the intersection of $K_\gamma^Y(\mathcal{F})$ and $K_\beta^N(\mathcal{F})$) to get the hardness of (γ, β)-Max-CSP($\mathcal{F}$). We attempt to explain this below.

Bias-based algorithms. The class of algorithms we use are what we call "bias-based algorithms," which extend algorithms used for Max-DICUT and other problems in [41, 47]. Roughly, these algorithms work by inspecting constraints one at a time and (linearly) updating the "preference/bias" of variables involved in the constraint for a given assignment. This update depends on the location of the variable within the constraint (and if there are multiple functions in the family, also on the function itself). Thus, implicitly these algorithms maintain an n-dimensional bias vector and at the end use some property of this vector to estimate a lower bound on the value of the instance. If this property is computable efficiently in the turnstile streaming model, then this leads to a space-efficient streaming algorithm.

The key questions for us are: (1) How to update the bias? and (2) What property of the vector yields a lower bound? When dealing with specific functions as in previous papers, there are some natural candidates for bias, and the most natural one turns out to be both useful and computable efficiently using ℓ_1 norm estimators. For the property, one has to devise a "rounding scheme" that takes the bias vector and uses it to create an assignment that achieves a large value (or value related to the property being estimated).

In our case, obviously "inspection" of natural candidates will not work for item (1)—we have infinitely many problems to inspect. But it turns out that the convex set framework, somewhat surprisingly, completely solves both parts (1) and (2) for us. If $K_\gamma^Y(\mathcal{F})$ and $K_\beta^N(\mathcal{F})$ do not intersect, then there is a linear separator in $\mathbb{R}^{kq}$ separating the two sets and the coefficients of this separator are interpretable as giving kq "biases"—for $i \in [k]$ and $\sigma \in [k]$ the (i, σ)-th coefficient can be viewed as the bias/preference of the ith variable in a constraint for taking the assignment $\sigma \in [q]$. This gives us an $n \times q$ bias matrix at the end that captures all the biases of variables from the whole instance. Turning to (2), a natural property to consider at this stage is the one-infinity norm of this matrix (i.e., the ℓ_1 norm of the n-dimensional vector whose coordinates are the ℓ_∞ norms of the rows of the bias matrix). Informally, this corresponds to each variable acting independently according to its bias. It turns out this norm is one of many that is known to be computable with small space in the turnstile streaming setting, and in particular we use a result of Andoni et al. [3] to compute this. Finally, we need a relationship between this property and a lower bound on the value, and once again the fact that the bias came from a separating hyperplane (and the exact definition of the sets in the convex set framework) allows us to distinguish instances with value at least γ from instances of value at most β. (Note that these constants are already baked into our sets and hence the separating hyperplane.) We remark that we do not give an explicit rounding procedure for our approximation algorithm, though one can probably be extracted from the definitions of the convex sets and analyses of the correctness of our algorithms.

Lower bounds. Finally, we turn to the lower bounds. Once again we restrict our overview to the setting of $|\mathcal{F}| = 1$ for simplicity. Both our lower bounds for sketching algorithms and for general streaming algorithms have a common starting point. Recall we are given that there are two distributions $\mathcal{D}_Y$ and $\mathcal{D}_N$ on constraints that have the same one-wise marginals, and these can be viewed as distributions on $[q]^k$.

For every pair of such distributions $\mathcal{D}_Y$ and $\mathcal{D}_N$ in $[q]^k$ we define a two-player communication problem we call $(\mathcal{D}_Y, \mathcal{D}_N)$-**signal detection (SD)**. (So in effect these are infinitely many different communication problems, roughly corresponding to the infinitely many different Max-CSP($\mathcal{F}$) problems we wish to analyze.) We show that if $\mathcal{D}_Y$ and $\mathcal{D}_N$ have the same marginals, then the communication problem requires $\Omega(\sqrt{n})$ communication. We give further details below but now explain the path from this communication lower bound to the streaming lower bounds. To get these lower bounds, we convert our SD lower bound into lower bounds on some T-players games, for all large constants T. Instances of the T-player games immediately correspond to instances of Max-CSP($\mathcal{F}$) and furthermore the properties of the sets $K^Y_\gamma(\mathcal{F})$ and $K^N_\beta(\mathcal{F})$ translate into the value of these Max-CSP($\mathcal{F}$) instances.

Turning to the T-player games: In the lower bound for sketching algorithms, we first convert the SD lower bound into a lower bound on a T-player *simultaneous* communication game. This conversion is relatively standard in the streaming literature [12, 49, 53, 62]: Reduce the two-player communication game to the T-player communication game by letting Bob play the role of one of the players and Alice play the role of the remaining $T-1$ players. By turning a sketching algorithm into a protocol for the communication game, we can get a space $\sqrt{n}$ lower bound for every (γ, β)-Max-CSP($\mathcal{F}$) against any sketching algorithms whenever the corresponding K^Y and K^N intersect. (See Theorem 5.1.) For the hardness result in the streaming setting, the lower bound on the simultaneous communication problem no longer suffices. So here we craft our own reduction to a T-player *one-way* communication problem, which reduces in turn to (γ, β)-Max-CSP($\mathcal{F}$) in the streaming setting. (This step follows the same path as [41, 56].) Unfortunately, this step works only in some restricted cases (for instance, if $\mathcal{D}_N$ is the uniform distribution on $[q]^k$), and this yields our lower bound (Theorem 3.12) in the streaming setting.

We now turn to our family of communication problems (SD), which is a distributional one-way communication problem. In the $(\mathcal{D}_Y, \mathcal{D}_N)$-SD problem with length parameter n, Alice gets a random string $\mathbf{x}^* \in [q]^n$ and Bob gets a hypermatching $\mathbf{J} = (\mathbf{j}(1), \ldots, \mathbf{j}(m))$ with $m = \alpha n$ edges (where $\alpha > 0$ is a constant of our choice independent of n). In other words, $\mathbf{j}(i)$ is a sequence of k distinct elements of $[n]$ and furthermore $\mathbf{j}(i)$ and $\mathbf{j}(i')$ are disjoint for every $i \neq i' \in [m]$. In addition, Bob also gets m bits $\mathbf{z} = (z(1), \ldots, z(m))$, where $z(i)$ is obtained by sampling $\mathbf{b}(i) \sim \mathcal{D}_Y$ in the **YES** case (and $\mathbf{b}(i) \sim \mathcal{D}_N$ in the **NO** case) independently for $i \in [m]$ and letting $z(i) = 1$ iff $\mathbf{x}^*|_{\mathbf{j}(i)} = \mathbf{b}(i)$. The goal of the communication problem is for Alice to send a message to Bob that allows Bob to guess whether this is a **YES** instance or a **NO** instance. The minimum length (over all protocols solving SD) of Alice's message is the complexity of the $(\mathcal{D}_Y, \mathcal{D}_N)$-SD. It is straightforward from the definition to get a $O_{\mathcal{D}_Y, \mathcal{D}_N, \alpha}(1)$-bit communication protocol achieving constant advantage if $\mathcal{D}_Y$ and $\mathcal{D}_N$ do not have the same marginals. Our lower bound shows that whenever the marginals match, the communication is at least $\Omega(\sqrt{n})$. (It is again straightforward to show distributions with matching marginals where $O(\sqrt{n})$ bits of communication suffice to distinguish the two cases.)

Before giving some details on our lower bound proof of the SD problem, we briefly give some context to the problem itself. We note that our communication game is different from those in previous works: Specifically the problem studied in [43, 56] is called the ***Boolean Hidden Matching (BHM)*** problem from [43] and the works [57, 58] study a variant called the *Implicit Hidden Partition* problem. While these problems are similar, they are less expressive than our formulation, and specifically do not seem to capture all Max-CSP($\mathcal{F}$) problems. We note that the BHM problem is essentially well suited only for the setting $k = q = 2$. In particular, the definition and analysis of BHM relies on the Fourier analysis over $\mathbb{F}_q$. Increasing k leads to several possible extensions that seem more naturally suited to CSPs on literals rather than variables. And increasing q leads to further complications since we do not have a natural field to work with. Thus, the choice of SD is

made carefully to allow both expressibility (we need to capture all Max-CSP($\mathcal{F}$)s) and the ability to prove lower bounds.

Turning to our lower bound, it comes in two major steps. In the first step we resort to a different communication problem that we call the **"Randomized Mask Detection Problem with advice" (Advice-RMD)**. In this problem, defined only for $q = 2$, Alice and Bob are given more information than in SD. Specifically, Alice is given as "advice" a partition of $[n]$ into k parts with the promise that the ℓth variable in every constraint is from the ℓth part for every $\ell \in [k]$. And Bob is given the vectors $(\mathbf{z}(1), \ldots, \mathbf{z}(m))$, where $\mathbf{z}(i) = \mathbf{x}^*|_{\mathbf{j}(i)} \oplus \mathbf{b}(i)$ for $i \in [m]$. This problem is closest both in definition and in analyzability to the previous problems. Indeed, we are able to extend previous Fourier-analytic lower bounds, in the special case where the marginals of $\mathcal{D}_Y$ and $\mathcal{D}_N$ over $\{-1, 1\}$ are *uniform*, to give an $\Omega(\sqrt{n})$ lower bound on the communication complexity of this problem. (See Theorem 6.2.) This immediately yields a hardness of the SD problem when $\mathcal{D}_Y$ and $\mathcal{D}_N$ are distributions over $\{-1, 1\}^k$ with uniform marginals, but we need more.

To extend the lower bound to all q and to non-uniform marginals, we use more combinatorial methods. Specifically, we show that we can move $\mathcal{D}_Y$ to $\mathcal{D}_N$ in a series of steps $\mathcal{D}_Y = \mathcal{D}_1, \ldots, \mathcal{D}_L = \mathcal{D}_N$, where for every i, the difference between $\mathcal{D}_i$ and $\mathcal{D}_{i+1}$ is "captured" (in a sense we do not elaborate here) by two distributions with uniform marginals over $\{a, b\}^k$ for some $a, b \in [q]$. We refer to each of these L steps as a "polarization step." Showing that L, the number of polarization steps, is finite leads to an interesting problem we solve in Section 7.1. (The bound depends on q and k but not $\mathcal{D}_Y, \mathcal{D}_N, \alpha$, or n. We remark that any dependence on the first three would have been fine for our application.) Finally we show that the lower bound on the Advice-RMD mentioned above, in the Boolean uniform marginal setting, suffices to show that the $(\mathcal{D}_i, \mathcal{D}_{i+1})$-SD problem also requires $\Omega(\sqrt{n})$ communication. (See Theorems 6.4 and 7.4.) By a triangle inequality it follows that $(\mathcal{D}_Y, \mathcal{D}_N)$-SD requires $\Omega(\sqrt{n})$ communication. (See Theorem 5.4.)

1.7 Subsequent Results

Subsequent to the first announcement of this work, several follow-up results have extended and strengthened the results of this article. We report on some of these below.

Explicit Families of Easy and Hard Problems. One of the main drawbacks of our result in Theorem 1.1 is that the decision criterion is not completely explicit. This is of course natural given the richness of the class of problems, but it is still natural to ask whether there are some clean families of problems that can be shown to be non-trivially approximable or not by further analyzing the tractability condition. Two subsequent works have addressed this question for infinite classes of problems, and we report on these below.

One class of works by the authors with Shahrasbi [39] explores the "monarchy" and "weak monarchy" predicates. The monarchy predicate is the function $f_{\text{monarchy}} : \{-1, 1\}^k \to \{0, 1\}$ given by $f(x_1, \ldots, x_k) = \text{sign}((k-2)x_1 + \sum_{i=2}^{k} x_i)$. In other words, $f_{\text{monarchy}}(\mathbf{x}) = 1$ if $x_1 = 1$ and at least one other x_i is 1, or if $x_2 = \cdots = x_k = 1$. The monarchy family $\mathcal{F}_{\text{monarchy}}$ is given by applying the monarchy predicate to literals, i.e., $\mathcal{F}_{\text{monarchy}} = \{f^{\mathbf{b}}_{\text{monarchy}} | \mathbf{b} \in \{-1, 1\}^k\}$, where $f^{\mathbf{b}}_{\text{monarchy}}(\mathbf{x}) = f_{\text{monarchy}}(\mathbf{x} \odot \mathbf{b})$. The monarchy CSP (Max-CSP($\mathcal{F}$)$_{\text{monarchy}}$) is known to be approximable in the polynomial time setting for every k [66]. In contrast, their work [39] shows that for $k \geq 5$, the monarchy CSP is approximation resistant in the sketching setting. This is of particular interest since this is a family that is not one-wise independent but remains approximation resistant in the sketching setting. The approximation resistance of this class for general streaming algorithms remains open. [39] also explores weak monarchy CSPs, i.e., CSPs on functions of the form $f_{k,j}(\mathbf{x}) = \text{sign}(jx_1 + \sum_{i=2}^{k} x_i)$ applied to literals. They show that for every j for all sufficiently large k, the weak monarchy CSP based on $f_{k,j}(\mathbf{x})$ is non-trivially approximable in the sketching setting.

Another work deriving explicit bounds for infinite families is due to Boyland et al. [26]. They derive the exact form of the optimal sketching approximation ratios for several symmetric Boolean CSPs including Max-kAND and Th_k^{k-1} (the "weight-at-least-$(k-1)$" threshold function on k variables). In both cases they show that there are non-trivial approximation algorithms, thus establishing infinitely many problems for which the exact approximation ratio can be determined using (and further analyzing) our framework. (As an example they show that the approximation ratio for Max-kAND is exactly $2^{-(k-1)}(1-k^{-2})^{(k-1)/2}$ for odd $k \geq 3$ for sketching algorithms.) Their work further analyzes our streaming lower bound in Theorem 3.10 and shows that for the threshold function Th_4^3, our streaming and sketching lower bounds match. (This is analogous to our result for Max-DICUT in Section 3.4.)

$o(n)$-Space algorithms. In a work of the authors with Velingker [40], the space lower bound in Theorem 3.12 is improved to $\Omega(n)$ for a subclass of function families that support one-wise independence. In particular, they show that the subclass they consider is approximation resistant with respect to $o(n)$-space streaming algorithms. We do not describe the exact subclass here but mention that it suffices for them to get an "approximate" classification of all approximation problems, Namely, for every given γ, β, and $\mathcal{F}$ over a q-ary alphabet, they show that either (γ,β)-Max-CSP($\mathcal{F}$) is trivial or $(\gamma/q,\beta)$-Max-CSP($\mathcal{F}$) requires $\Omega(n)$ space to solve. Their work suggests some inherent barriers in extending the full classification of the problems considered in the current article to $o(n)$-space algorithms. This was later confirmed in a work of Saxena et al. [70] where they give an $\tilde{O}(\sqrt{n})$ space algorithm for Max-DICUT that beats the best $o(\sqrt{n})$ space algorithm. Singer [74] partially extends this result to obtain an $O(n^{1-1/k})$ space algorithm for Max-kAND that beats the optimal $o(\sqrt{n})$ space algorithm on "bounded-degree" instances.

Random-ordering streaming setting. While Kapralov et al. [56] show that Max-CUT is inapproximable by $o(\sqrt{n})$ space streaming algorithms even in the random-ordering setting, Saxena et al. [69] give an $O(\log n)$ space streaming algorithm in this setting that beats the optimal $o(\sqrt{n})$ space algorithm for Max-DICUT in the adversarial-ordering setting. Singer [74] extends this result to obtain $O(\log n)$ space random-order streaming algorithms that beat the best $o(\sqrt{n})$ space adversarial-order algorithms for Max-kAND, for all k!

Multi-pass streaming setting. The random-order streaming algorithms in [69, 74] can be trivially extended to obtain $O(\log n)$ space two-pass adversarial-order streaming algorithms with the same approximation ratio. A recent result due to Kol et al. [61] gives a complete characterization for the exact computability of every Boolean Max-CSP(f) in the multi-pass streaming setting and subsumes our Theorem 1.4 for this family. In particular, for every Boolean predicate f, they give an $\tilde{O}(n^{\deg(f)})$ space single-pass streaming algorithm that solves Max-CSP(f) exactly, where $\deg(f)$ is the degree of f when viewed as multilinear polynomial, and show that any *constant*-pass streaming algorithm requires at least $\Omega(n^{\deg(f)})$ space.

Variations of CSPs. It turns out that our work on CSPs also is helpful in analyzing some variations of CSPs. In particular, Singer et al. [73] consider the space of "ordering CSPs" where the challenge is to find an ordering of n variables that satisfy some specified ordering constraints. An example is the **Maximum Acyclic Subgraph (MAS)** problem where the goal is to find an ordering of n variables $x_1,\ldots,x_n$ that, given many constraints of the form $x_i < x_j$, satisfies as many constraints as possible. Prior to the work of [73], no problem (including MAS) was tightly analyzed. [73] show that no ordering CSP has a non-trivial streaming algorithm with $o(\sqrt{n})$ space. Their work crucially relies on the framework from this article and uses the approximation resistance of some CSPs considered in this article. (See Section 3.4 for further details.) Since the problems needed in their

work fall within the subclass of problems considered in [40], their streaming lower bound actually improves to $\Omega(n)$-space.

1.8 Structure of Rest of the Article

Section 2 contains some of the preliminary background used in the rest of the article. In Section 3, we describe our results in detail. In particular, we build our convex set framework and give an explicit criterion to distinguish the easy and hard Max-CSP($\mathcal{F}$) problems. We also describe sufficient conditions for the hardness of some streaming problems in the streaming setting. In Section 4, we describe and analyze our algorithm that yields our easiness result. In Section 5, we define the "Signal Detection" problem and show how the communication complexity of this problem leads to the streaming space lower bounds claimed in Section 3. In Section 6, we introduce and analyze the Advice-RMD problem. In Section 7, we prove our general lower bound for SD assuming that a single polarization step is hard. In Section 8, we complete this remaining step by using the Advice-RMD lower bound to show hardness of a single polarization step, thus concluding our main lower bound. Finally, in Section 9 we give the dichotomy for the exact computability of Max-CSP($\mathcal{F}$).

2 PRELIMINARIES

In this section we introduce notations, definitions, and some standard tools that will be used in the rest of this article. Specifically, we define constraint satisfaction problems and some promise problems related to their approximation (Section 2.1). Then we formally describe the streaming and sketching models of computation along with some variants and background material (Section 2.2). In Section 2.2.1 we explain the folklore relationship between the promise problems defined in Section 2.1 and the standard single-parameter version of approximations, in the context of streaming algorithms. Section 2.3 has some basic notions from probability and some tools we will use. Section 2.4 recalls notions from Fourier analysis and mentions the tools used from this area. Finally, Section 2.5 defines notions and results from the quantified theory of reals. We start with some notation.

We let $\mathbb{N}$ denote the set of positive integers. We let $[n]$ denote the set $\{1, \ldots, n\}$. For a finite set Ω, let $\Delta(\Omega)$ denote the space of all probability distributions over Ω, i.e.,

$$\Delta(\Omega) = \left\{ \mathcal{D} : \Omega \to \mathbb{R}^{\geq 0} \mid \sum_{\omega \in \Omega} \mathcal{D}(\omega) = 1 \right\}.$$

We view $\Delta(\Omega)$ as being contained in $\mathbb{R}^{|\Omega|}$. We use $X \sim \mathcal{D}$ to denote a random variable drawn from the distribution $\mathcal{D}$. By default, a Boolean variable in this article takes value in $\{-1, 1\}$. For every $p \in [0, 1]$, Bern(p) denotes the Bernoulli distribution that takes value 1 with probability p and takes value -1 with probability $1 - p$.

We will follow the convention that n denotes the number of variables in CSPs, m denotes the number of constraints, and k denotes the arity of the CSP.

For variables of a vector form, we write them in boldface, e.g., $\mathbf{x} \in [q]^n$, and its ith entry is written without boldface, e.g., x_i. For a variable being a vector of vectors, we write it, for example, as $\mathbf{b} = (\mathbf{b}(1), \mathbf{b}(2), \ldots, \mathbf{b}(m))$, where $\mathbf{b}(i) \in [q]^k$. The jth entry of the ith vector of $\mathbf{b}$ is then written as $\mathbf{b}(i)_j$. Let $\mathbf{x}$ and $\mathbf{y}$ be two vectors of the same length; $\mathbf{x} \odot \mathbf{y}$ denotes the entry-wise product of them.

2.1 Approximate Constraint Satisfaction

Max-CSP($\mathcal{F}$) is specified by a family of constraints $\mathcal{F}$, where each constraint function $f \in \mathcal{F}$ is such that $f : [q]^k \to \{0, 1\}$, for a fixed positive integer k. Given n variables $x_1, x_2, \ldots, x_n$, an application of the constraint function f to these variables, which we term simply a *constraint*, is given

by a k-tuple $\mathbf{j} = (j_1, \ldots, j_k) \in [n]^k$, where the j_is are distinct and represent the application of the constraint function f to the variables $x_{j_1}, \ldots, x_{j_k}$. We use $C_{\mathcal{F},n}$ to denote the set of all constraints of Max-CSP($\mathcal{F}$) on n variables. (Note that $C_{\mathcal{F},n}$ is a finite set.) Specifically, an assignment $\mathbf{b} \in [q]^n$ satisfies a constraint given by $(f, \mathbf{j})$ if $f(b_{j_1}, \ldots, b_{j_k}) = 1$.

An instance Ψ of Max-CSP($\mathcal{F}$) consists of m constraints $C_1, \ldots, C_m$ with non-negative weights $w_1, \ldots, w_m$, where $C_i = (f_i, \mathbf{j}(i)) \in C_{\mathcal{F},n}$ and $w_i \in \mathbb{R}$ for each $i \in [m]$. For an assignment $\mathbf{b} \in [q]^n$, the value $\mathrm{val}_\Psi(\mathbf{b})$ of $\mathbf{b}$ on Ψ is the fraction of weight of constraints satisfied by $\mathbf{b}$, i.e., $\mathrm{val}_\Psi(\mathbf{b}) = \frac{1}{W} \sum_{i \in [m]} w_i \cdot f_i(\mathbf{b}|_{\mathbf{j}(i)})$, where $W = \sum_{i=1}^m w_i$. The optimal value of Ψ is defined as $\mathrm{val}_\Psi = \max_{\mathbf{b} \in [q]^n} \{\mathrm{val}_\Psi(\mathbf{b})\}$. The approximation version of Max-CSP($\mathcal{F}$) is defined as follows.

Throughout this article we will only consider the case of Max-CSP($\mathcal{F}$) instances with integer weights bounded by a polynomial in n.

Definition 2.1 ((γ, β)-Max-CSP($\mathcal{F}$)). Let $\mathcal{F}$ be a constraint family and $0 \le \beta < \gamma \le 1$. For each $m \in \mathbb{N}$, let $\Gamma_m = \{\Psi = (C_1, \ldots, C_m; w_1, \ldots, w_m) \mid \mathrm{val}_\Psi \ge \gamma\}$ and $B_m = \{\Psi = (C_1, \ldots, C_m; w_1, \ldots, w_m) \mid \mathrm{val}_\Psi \le \beta\}$.

The task of (γ, β)-Max-CSP($\mathcal{F}$) is to distinguish between instances from $\Gamma = \cup_{m \le \mathrm{poly}(n)} \Gamma_m$ and instances from $B = \cup_{m \le \mathrm{poly}(n)} B_m$. Specifically, we desire algorithms that output 1 w.p. at least $2/3$ on inputs from Γ and output 1 w.p. at most $1/3$ on inputs from B.

2.2 Streaming and Sketching Algorithms

We now define streaming and sketching algorithms in the context of Max-CSP($\mathcal{F}$). Note that the input to both algorithms are sequences of weighted constraints. Rather than explicitly including the weight, we will simply allow the sequence to repeat constraints (not necessarily successively). The implied weight of a constraint will thus be the number of times it is repeated. (Note that we only consider integer polynomially bounded weights. Thus, this representation only blows up the input by a polynomial factor.) A stream is thus an element of $(C_{\mathcal{F},n})^*$ and we use λ to denote the empty stream.

Definition 2.2 (Streaming Algorithm). A deterministic space s streaming algorithm **ALG** for Max-CSP($\mathcal{F}$) on n variables is given by a (state-evolution) function $S : \{0,1\}^s \times C_{\mathcal{F},n} \to \{0,1\}^s$ and an (output) function $v : \{0,1\}^s \to [0,1]$. Let $\widetilde{S} : (C_{\mathcal{F},n})^* \to \{0,1\}^s$ given by $\widetilde{S}(\lambda) = 0^s$ and $\widetilde{S}(\sigma_1, \ldots, \sigma_m) = S(\widetilde{S}(\sigma_1, \ldots, \sigma_{m-1}), \sigma_m)$ denote the iterated state-evolution map. Then the output of **ALG** on input $\sigma = (\sigma_1, \ldots, \sigma_m)$ is $v(\widetilde{S}(\sigma))$.

In a *uniform randomized* space s streaming algorithm the evolution map is given by $S : \{0,1\}^s \times C_{\mathcal{F},n} \times \{0,1\}^r \to \{0,1\}^s$ for some $r \le s$ and its iterate evolution map is a random variable given by $\widetilde{S}(\sigma_1, \ldots, \sigma_m) = S(\widetilde{S}(\sigma_1, \ldots, \sigma_{m-1}), \sigma_m, R_m)$, where $R_m \sim \mathrm{Unif}(\{0,1\}^r)$ is independent of $\sigma_1, \ldots, \sigma_m$ and $R_1, \ldots, R_{m-1}$.

A *non-uniform randomized* space s streaming algorithm is simply a distribution on deterministic space s streaming algorithms.

We note that non-uniform randomized algorithms can simulate uniform ones but may be much stronger since they allow algorithms to "remember" all previous random coins without being charged for the memory. All our upper bounds are in the uniform randomized model. Our lower bounds are in the non-uniform randomized model (and use this extra power in the reductions).

Sketching algorithms are a special class of streaming algorithms that have been widely used in both upper bounds and lower bounds. For the definition of sketching algorithms below, we adopt Definition 5.21 in [28].

Definition 2.3 (Sketching Algorithms). A deterministic space s streaming algorithm $\mathbf{ALG} = (S, \upsilon)$ is a sketching algorithm if there exists a compression function $\mathsf{SKETCH} : (\mathcal{C}_{\mathcal{F},n})^* \to \{0,1\}^s$ and a combination function $\mathsf{COMB} : \{0,1\}^s \times \{0,1\}^s \to \{0,1\}^s$ such that the following hold:

— $S(z, C) = \mathsf{COMB}(z, \mathsf{SKETCH}(C))$ for every $z \in \{0,1\}^s$ and $C \in \mathcal{C}_{\mathcal{F},n}$.

— For every pair of streams $\sigma, \tau \in (\mathcal{C}_{\mathcal{F},n})^*$, we have

$$\mathsf{COMB}(\mathsf{SKETCH}(\sigma), \mathsf{SKETCH}(\tau)) = \mathsf{SKETCH}(\sigma \circ \tau),$$

where $\sigma \circ \tau$ represents the concatenation of the streams σ and τ.

A uniform randomized sketching algorithm is similarly defined with $\mathsf{COMB} : \{0,1\}^s \times \{0,1\}^s \times \{0,1\}^r \to \{0,1\}^s$ and $S(z, C, R) = \mathsf{COMB}(z, \mathsf{SKETCH}(C), R)$ for every z, C, R, where $r \leq s$. A randomized algorithm **ALG** is a non-uniform randomized sketching algorithm if it is a distribution over deterministic sketching algorithms.

We remark that there can be several variants to the streaming problem above involving the possibility of weighted constraints, deletion of constraints, and length of the input stream.

(1) Dynamic streams: In this setting constraints may be inserted, even multiple times, and later deleted. In this setting algorithms are required to be correct on the final instance, under the promise that constraints were deleted fewer times than they were inserted at all intermediate stages of the streaming process. The input stream can be unboundedly large in this setting even while maintaining polynomially bounded integer weights (e.g., by inserting and deleting the same constraint an arbitrary number of times). Thus, algorithms may have restrictions on the length of input streams or have complexity growing with the length of the stream.

All our lower bounds work in the insertion-only setting. Our upper bounds work on dynamic streams provided they have length polynomial in n.

(2) Weighted instances: Variations of Max-CSP($\mathcal{F}$) allow constraints to have non-negative real weights. We do not explicitly consider this setting in this article, but standard techniques (involving rounding weights to nearby rationals) allow algorithms for polynomially bounded integer weights to be extended to apply to this setting also.

(3) Linear Sketching: An instance Ψ of Max-CSP($\mathcal{F}$) can be viewed as a vector in $\mathbb{R}^{\mathcal{C}_{\mathcal{F},n}}$ with the Cth coordinate representing the weight of the constraint C in Ψ. A linear sketching algorithm is one whose state is a linear function of this representation of the instance. Note that in this representation, the stream can be viewed as a sequence of linear updates. Thus, if the state is a linear function, the updates to the state can be computed knowing only the previous state and the update to Ψ, thus leading to a natural streaming algorithm. Furthermore, it can be seen that this streaming algorithm also satisfies the notion of sketchability.

The space complexity of such a sketching algorithm deserves special mention. The space requirement of linear sketching is the space needed to represent t real numbers, where t is the rank of the linear map used to sketch the inputs. When the weights are integers bounded by a polynomial in n, this can be used to show that the real numbers arising in the sketch can be represented by $O(\log n)$ bit rationals and so this translates to a small space sketch. This possibility goes away if the input is not polynomially bounded.

All our algorithms are linear sketching algorithms as defined above.

Remark 2.4. We note that [1, 64] have shown that algorithms that work on dynamic streams are also linear sketching algorithms. Thus, the assertion above that our algorithms are linear sketching algorithms (Item 3) seems redundant in view of the claim that they work in the dynamic setting (Item 1). However, the results in [1, 64] only apply to the case where the input streams are superpolynomially long (even requiring doubly exponential length). This is even necessary as

proved by [51]. Our results, on the other hand, only hold for polynomial length streams. Thus, in our setting, dynamic streams and linear sketching are not equivalent.

2.2.1 Relation to Single-parameter Approximability. The traditional study of approximation algorithms typically focuses on a single-parameter problem. Specifically, for $\alpha \in [0,1]$, Max-CSP($\mathcal{F}$) is said to be *α-approximable* in space s in the streaming setting if there is a space s algorithm that on input of a stream representing instance Ψ of Max-CSP($\mathcal{F}$) outputs a number in $[\alpha \cdot \mathrm{val}_\Psi, \mathrm{val}_\Psi]$. The connection between this single-parameter approximability and the gapped problems we study is folklore. For the sake of completeness we describe the algorithmic implication below.

Proposition 2.5. *Given $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$, a space complexity measure $s : \mathbb{N} \to \mathbb{N}$, and sets Easy, Hard $\subseteq [0,1] \times [0,1]$ such that for every $(\gamma,\beta) \in$ Easy, (γ,β)-Max-CSP($\mathcal{F}$) is solvable in $s(n)$-space in the sketching model, and for every $(\gamma,\beta) \in$ Hard, (γ,β)-Max-CSP($\mathcal{F}$) is not solvable in $s(n)$-space in the sketching model. Then for*

$$\alpha = \inf_{\beta \in [0,1]} \left\{ \sup_{\gamma \in (\beta,1] \text{ s.t } (\gamma,\beta) \in \textsc{Easy}} \{\beta/\gamma\} \right\},$$

and for every $\varepsilon > 0$, there is an $(\alpha-\varepsilon)$-approximation algorithm for Max-CSP($\mathcal{F}$) that uses $O_{k,q,\varepsilon}(s(n))$ space in the sketching model. Conversely, for

$$\alpha = \inf_{\beta \in [0,1]} \left\{ \sup_{\gamma \in (\beta,1] \text{ s.t } (\gamma,\beta) \notin \textsc{Hard}} \{\beta/\gamma\} \right\},$$

and every $\varepsilon > 0$, every $(\alpha+\varepsilon)$-approximation sketching algorithm for Max-CSP($\mathcal{F}$) requires $s(n)$ space.

Proof. The negative result is simple. We prove it in the contrapositive form by showing that if Max-CSP($\mathcal{F}$) has an $(\alpha+\varepsilon)$-approximation algorithm using $s(n)$ space, then for every (γ,β) with $\beta \le \alpha\gamma$, (γ,β)-Max-CSP($\mathcal{F}$) is solvable in $s(n)$ space (and so $(\gamma,\beta) \notin$ Hard). Suppose Max-CSP($\mathcal{F}$) has an $(\alpha+\varepsilon)$ approximation algorithm A using $s(n)$-space in the sketching model. Given γ, β with $\beta/\gamma \ge \alpha$, we can use A to solve the (γ,β)-Max-CSP($\mathcal{F}$) on input Ψ as follows: Compute $A(\Psi)$ and output YES if $A(\Psi) \ge \beta$ and NO otherwise. Since $\beta \le \alpha\gamma < (\alpha+\varepsilon)\gamma$, it follows that if $\mathrm{val}(\Psi) \ge \gamma$, then $A(\Psi)$ will output some number greater than β and our algorithm will output YES. If $\mathrm{val}(\Psi) \le \beta$, then $A(\Psi)$ will output some number less than or equal to β and our algorithm outputs NO. This yields the negative result.

For the positive result, we assume that Easy is monotone in the following sense: If $(\gamma,\beta) \in$ Easy and $\beta' \le \beta$, then $(\gamma,\beta') \in$ Easy. (Note that we can assume this since an algorithm solving the (γ,β)-Max-CSP($\mathcal{F}$) problem also solves the (γ,β')-Max-CSP($\mathcal{F}$) problem.) We also assume that every constraint in $\mathcal{F}$ has at least one satisfying assignment. (If not we can simply remove unsatisfiable constraints from $\mathcal{F}$ and ignore them in the input stream.) Due to this assumption, we have that a random assignment satisfies at least $\rho \triangleq q^{-k}$ fraction of the constraints. Let $\tau \triangleq \varepsilon \cdot \rho/2$ and let

$$A_\tau = \{(i\tau, j\tau) \in [0,1]^2 \mid i,j \in \mathbb{Z}^{\ge 0}, (i\tau, j\tau) \in \textsc{Easy}\}.$$

Thus, for every $(\gamma,\delta) \in A_\tau$ there is an $s(n)$-space algorithm for (γ,β)-Max-CSP($\mathcal{F}$) with error probability $1/3$. By repeating this algorithm $O(\log(1/\tau))$ times and taking majority, we may assume the error probability is at most $1/(10\tau^2)$. We refer to this amplified algorithm as the (γ,β)-distinguisher below. In the following we consider the case where all $O(\tau^{-2})$ distinguishers output correct answers, which happens with probability at least $2/3$.

Our $O_\tau(s(n))$ space $(\alpha-\varepsilon)$-approximation algorithm for Max-CSP($\mathcal{F}$) is the following: On input Ψ, run in parallel all the (γ,β)-distinguishers on Ψ, for every $(\gamma,\beta)\in A_\tau$. Let

$$\beta_0 = \arg\max_\beta [\exists \gamma \text{ such that the } (\gamma,\beta)\text{-distinguisher outputs YES on } \Psi].$$

Output $\beta' = \max\{\rho, \beta_0\}$.

We now prove that this is an $(\alpha-\varepsilon)$-approximation algorithm. First note that by the correctness of the distinguisher we have $\beta' \le \mathrm{val}_\Psi$. Let γ_0 be the smallest multiple of τ satisfying $\gamma_0 \ge (\beta_0+\tau)/\alpha$. By the definition of α, we have that $(\gamma_0, \alpha\gamma_0) \in$ Easy and so by the monotonicity assumption on Easy we have $(\gamma_0, \beta_0+\tau) \in$ Easy. So $(\gamma_0, \beta_0+\tau) \in A_\tau$ and so the $(\gamma_0, \beta_0+\tau)$-distinguisher must have output NO on Ψ (by the maximality of β_0). By the correctness of this distinguisher we conclude $\mathrm{val}_\Psi \le \gamma_0 \le (\beta_0+\tau)/\alpha + \tau \le (\beta'+\tau)/\alpha + \tau$. We now verify that $(\beta'+\tau)/\alpha+\tau \le \beta'/(\alpha-\varepsilon)$ and this gives us the desired approximation guarantee. We have

$$(\beta'+\tau)/\alpha+\tau \le (\beta'+2\tau)/\alpha \le (\beta'/\alpha)\cdot(1+2\tau/\rho) = (\beta'/\alpha)(1+\varepsilon) \le (\beta'/(\alpha(1-\varepsilon))),$$

where the first inequality uses $\alpha \le 1$, the second uses $\beta' \ge \rho$, the equality comes from the definition of τ, and the final inequality uses $(1+\varepsilon)(1-\varepsilon) \le 1$. This concludes the positive result. □

2.3 Probabilistic Notions and Tools

We recall some standard notions from probability theory and mention some results we will use.

2.3.1 Total Variation Distance. The total variation distance between probability distributions plays an important role in our analysis.

Definition 2.6 (Total Variation Distance of Discrete Random Variables). Let Ω be a finite probability space and X, Y be random variables with support Ω. The total variation distance between X and Y is defined as follows:

$$\|X-Y\|_{tvd} := \frac{1}{2}\sum_{\omega\in\Omega} |\Pr[X=\omega] - \Pr[Y=\omega]| .$$

We will use the triangle and data processing inequalities for the total variation distance.

Proposition 2.7 (E.g., [56, Claim 6.5]). *For random variables X, Y, and W:*

- *(Triangle inequality) $\|X-Y\|_{tvd} \ge \|X-W\|_{tvd} - \|Y-W\|_{tvd}$.*
- *(Data processing inequality) If W is independent of both X and Y, and f is a function, then $\|f(X,W)-f(Y,W)\|_{tvd} \le \|X-Y\|_{tvd}$.*

2.3.2 A Concentration Inequality. We will use the following concentration inequality, which is essentially an Azuma-Hoeffding-style inequality for submartingales. The form we use is based on [58, Lemma 2.5] and allows for variables with different expectations. The analysis is a very slight modification of theirs.

Lemma 2.8. *Let $X = \sum_{i\in[N]} X_i$, where X_i are Bernoulli random variables such that for every $k \in [N]$, $\mathbb{E}[X_k \mid X_1, \ldots, X_{k-1}] \le p_k$ for some $p_k \in (0,1)$. Let $\mu = \sum_{k=1}^N p_k$. For every $\Delta > 0$, we have*

$$\Pr[X \ge \mu + \Delta] \le \exp\left(-\frac{\Delta^2}{2\mu+2\Delta}\right).$$

Proof. Let $v = \Delta/(\mu+\Delta)$ and $u = \ln(1+v)$. We have

$$\mathbb{E}[e^{uX}] = \mathbb{E}\left[\prod_{k=1}^N e^{uX_k}\right] \le (1+p_N(e^u-1))\cdot\mathbb{E}\left[\prod_{k=1}^{N-1} e^{uX_k}\right] \le \prod_{i=1}^N (1+p_k(e^u-1)) = \prod_{i=1}^N (1+p_k v) \le e^{v\mu},$$

where the final inequality uses $1+x \le e^x$ for every x (and the definition of μ). Applying Markov's inequality to the above, we have

$$\Pr[X \ge \mu+\Delta] = \Pr\left[e^{uX} \ge e^{u(\mu+\Delta)}\right] \le \mathbb{E}[e^{uX}]/e^{u(\mu+\Delta)} \le e^{v\mu-u\mu-u\Delta}.$$

From the inequality $e^{v-v^2/2} \le 1+v$ we infer $u \ge v - v^2/2$ and so the final expression above can be bounded as

$$\Pr[X \ge \mu+\Delta] \le e^{v\mu-u\mu-u\Delta} \le e^{\frac{v^2}{2}(\mu+\Delta)-v\Delta} = e^{-\frac{\Delta^2}{2(\mu+\Delta)}},$$

where the final equality comes from our choice of v. □

2.4 Fourier Analysis

We will need the following basic notions from Fourier analysis over the Boolean hypercube (see, for instance, [65]). For a Boolean function $f : \{-1,1\}^k \to \mathbb{R}$ its Fourier coefficients are defined by $\widehat{f}(\mathbf{v}) = \mathbb{E}_{\mathbf{a}\in\{-1,1\}^k}[f(\mathbf{a})\cdot(-1)^{\mathbf{v}^\top\mathbf{a}}]$, where $\mathbf{v} \in \{0,1\}^k$. We need the following two important tools.

Lemma 2.9 (Parseval's Identity). *For every function $f\{-1,1\}^k \to \mathbb{R}$,*

$$\|f\|_2^2 = \frac{1}{2^k}\sum_{\mathbf{a}\in\{-1,1\}^k} f(\mathbf{a})^2 = \sum_{\mathbf{v}\in\{0,1\}^k} \widehat{f}(\mathbf{v})^2 .$$

Note that for every distribution f on $\{-1,1\}^k$, $\widehat{f}(0^k) = 2^{-k}$. For the uniform distribution U on $\{-1,1\}^k$, $\widehat{U}(\mathbf{v}) = 0$ for every $\mathbf{v} \ne 0^k$. Thus, by Lemma 2.9, for any distribution f on $\{-1,1\}^k$:

$$\|f-U\|_2^2 = \sum_{\mathbf{v}\in\{0,1\}^k}\left(\widehat{f}(\mathbf{v})-\widehat{U}(\mathbf{v})\right)^2 = \sum_{\mathbf{v}\in\{0,1\}^k\setminus\{0^k\}} \widehat{f}(\mathbf{v})^2 . \tag{2.10}$$

Next, we will use the following consequence of hypercontractivity for Boolean functions as given in [43, Lemma 6], which in turns relies on a lemma from [50].

Lemma 2.11. *Let $f : \{-1,1\}^n \to \{-1,0,1\}$ and $A = \{\mathbf{a} \in \{-1,1\}^n \mid f(\mathbf{a}) \ne 0\}$. If $|A| \ge 2^{n-c}$ for some $c \in \mathbb{N}$, then for every $\ell \in \{1,\dots,4c\}$, we have*

$$\frac{2^{2n}}{|A|^2}\sum_{\substack{\mathbf{v}\in\{0,1\}^n\\ \|\mathbf{v}\|_1=\ell}} \widehat{f}(\mathbf{v})^2 \le \left(\frac{4\sqrt{2}c}{\ell}\right)^\ell .$$

2.5 Quantified Theory of Reals

The decidability of several characterizations in this article follows from the decidability of the "quantified theory of the reals." We describe the main problem and result here.

Definition 2.12 (Quantified Polynomial Sentence). A quantified polynomial sentence over K variables, S polynomials of degree D of quantifier width w is given by (1) a Boolean formula $\Psi(Y_1,\dots,Y_S)$ on S Boolean variables; (2) a set $\mathcal{P}$ of S polynomials $\mathcal{P} = \{P_i(X_1,\dots,X_K) \mid i \in [S]\}$, with each P_i being a polynomial with real coefficients and of degree at most D in K variables; and (3) a partition $\Pi = (X_{[1]},\dots,X_{[w]})$ of the set $\{X_1,\dots,X_K\}$ and w quantifiers $Q = (Q_1,\dots,Q_w)$ with $Q_j \in \{\exists,\forall\}$ for every $j \in w$. The sentence $(\Psi,\mathcal{P},\Pi,Q)$ is defined to be TRUE if $Q_1X_{[1]}Q_2X_{[2]}\dots Q_wX_{[w]}\Psi(Y_1(X_1,\dots,X_K),\dots,Y_S(X_1,\dots,X_K))$ is true, where $Y_i(X_1,\dots,X_K) =$ TRUE if and only if $P_i(X_1,\dots,X_S) \le 0$.

Note that the syntax is rich enough to express conditions such as $P(X) \ge 0$ and $P(X) < 0$ by use of arithmetic negations ($-P(X) \le 0$) and logical negations NOT($P(X) \ge 0$), where the logical

negation is inserted into the Boolean formula Ψ. As an example, the sentence "Every positive number can be written as the square of a real number" can be expressed as the sentence $\forall\alpha\exists\beta(-\alpha \geq 0) \vee ((\alpha - \beta^2) \geq 0) \vee (-(\alpha - \beta^2)) \geq 0$, which is a quantified sentence with two quantifiers, two variables partitioned into $\{\alpha\}$ and $\{\beta\}$ with quantifiers $Q_1 = \forall$ and $Q_2 = \exists$, and three polynomials of degree at most 2. This sentence happens to be TRUE.

Theorem 2.13 ([23, Theorem 14.14, see also Remark 13.10]). *The truth of a quantified formula with w quantifiers over K variables and S degree D polynomial (potentially strict) inequalities can be decided in space $K^{O(w)}\log(SD)$ and time $(SD)^{K^{O(w)}}$.*

Specifically, Theorem 14.14 in [23] asserts the time complexity above, and Remark 13.10 yields the space complexity.

3 RESULTS

In this section we introduce our convex set framework that makes our classification of "easy" vs. "hard" sketching problems explicit. The sets are introduced in Section 3.1. We then state our main dichotomy theorem and also state its decidability in Section 3.2. Other results of this article, including some strengthenings to the streaming setting, are stated in Section 3.3. We work out some example applications of the dichotomy theorem and strengthenings in Section 3.4. Finally, in Section 3.5 we include proofs of all the simple results and corollaries of this section, leaving only the proofs of Theorem 3.3, Theorem 3.10, and Theorem 3.16 to later sections.

3.1 The Convex Set Framework

The main objects that allow us to derive our characterization are the space of distributions on constraints that allow either a large number of constraints to be satisfied or only a few constraints to be satisfied. To see where the distributions come from, note that distributions of constraints over n variables can naturally be identified with instances of the weighted constraint satisfaction problem (where the weight associated with a constraint is simply its probability).

In this part we consider distributions of constraints over a set of kq variables denoted $\mathbf{x} = (x_{i,\sigma} \mid i \in [k], \sigma \in [q])$. (We think of the variables as sitting in a $k \times q$ matrix with i indexing the rows and σ indexing the columns.) For $f \in \mathcal{F}$ and $\mathbf{a} \in [q]^k$, let $C(f, \mathbf{a})$ denote the constraint $f(x_{1,a_1}, \ldots, x_{k,a_k})$. For an assignment $\mathbf{b} = (b_{i,\sigma} \mid i \in [k], \sigma \in [q]) \in [q]^{kq}$ we use the notation $C(f, \mathbf{a})(\mathbf{b})$ to denote the value $f(b_{1,a_1}, \ldots, b_{k,a_k})$. We let $\mathbb{I} \in [q]^{kq}$ denote the assignment $\mathbb{I}_{i,\sigma} = \sigma$. (In the following section we will use $\mathbb{I}$ as our planted assignment.)

We now turn to defining the "marginals" of distributions. For $\mathcal{D} \in \Delta(\mathcal{F} \times [q]^k)$, we let $\boldsymbol{\mu}(\mathcal{D}) = (\mu_{f,i,\sigma})_{f\in\mathcal{F}, i\in[k], \sigma\in[q]}$ be given by $\mu_{f,i,\sigma} = \Pr_{(g,\mathbf{a})\sim\mathcal{D}}[g = f \text{ and } a_i = \sigma]$. Thus, the marginal $\boldsymbol{\mu}(\mathcal{D})$ lies in $\mathbb{R}^{|\mathcal{F}|\times qk}$.

We often reduce our considerations to families $\mathcal{F}$ containing a single element. In such cases we simplify the notion of a distribution to $\mathcal{D} \in \Delta([q]^k)$. For $\mathcal{D} \in \Delta([q]^k)$, we let $\boldsymbol{\mu}(\mathcal{D}) = (\mu_{i,\sigma})_{i\in[k],\sigma\in[q]}$ be given by $\mu_{i,\sigma} = \Pr_{\mathbf{a}\sim\mathcal{D}}[a_i = \sigma]$.

Next we introduce our family of distributions that capture our "Yes" and "No" instances. "Yes" instances are highly satisfied by our planted assignment, while "No" instances are not very satisfied by any "column-symmetric," independent, probabilistic assignment. The fact that we only consider distributions on kq variables makes this a set in a finite-dimensional space.

Definition 3.1 (Space of YES/NO Distributions). For $q, k \in \mathbb{N}$, $\gamma \in [0,1]$, and $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$, we let

$$S^Y_\gamma(\mathcal{F}) = \left\{ \mathcal{D} \in \Delta(\mathcal{F} \times [q]^k) \mid \mathop{\mathbb{E}}_{(f,\mathbf{a})\sim\mathcal{D}}[C(f,\mathbf{a})(\mathbb{I})] \geq \gamma \right\}.$$

For $\beta \in [0,1]$, we let

$$S^N_\beta(\mathcal{F}) = \left\{ \mathcal{D} \in \Delta(\mathcal{F} \times [q]^k) \mid \forall (\mathcal{P}_\sigma \in \Delta([q]))_{\sigma \in [q]}, \mathop{\mathbb{E}}_{(f,\mathbf{a})\sim\mathcal{D}} \left[\mathop{\mathbb{E}}_{\mathbf{b}, b_{i,\sigma}\sim\mathcal{P}_\sigma} [C(f,\mathbf{a})(\mathbf{b})] \right] \leq \beta \right\}.$$

By construction, for $\beta < \gamma$, the sets $S^Y_\gamma(\mathcal{F})$ and $S^N_\beta(\mathcal{F})$ are disjoint. (In particular, for any $\mathcal{D} \in S^Y_\gamma(\mathcal{F})$, $\mathbb{I}$ corresponds to a (deterministic!) column symmetric assignment that satisfies $\gamma > \beta$ fraction of constraints, so $\mathcal{D} \notin S^N_\beta(\mathcal{F})$.) The key to the analysis of low-space sketching algorithms is that they only seem to be able to estimate the marginals of a distribution—so we turn to exploring the marginals of the sets above.

Definition 3.2 (Marginals of YES/NO Distributions). For $\gamma, \beta \in [0,1]$, and $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$, we let

$$K^Y_\gamma(\mathcal{F}) = \{\boldsymbol{\mu}(\mathcal{D}) \in \mathbb{R}^{|\mathcal{F}|kq} \mid \mathcal{D} \in S^Y_\gamma(\mathcal{F})\} \text{ and } K^N_\beta(\mathcal{F}) = \{\boldsymbol{\mu}(\mathcal{D}) \in \mathbb{R}^{|\mathcal{F}|kq} \mid \mathcal{D} \in S^N_\beta(\mathcal{F})\}.$$

See Section 3.4 for some examples of the sets $S^Y_\gamma(\mathcal{F}), S^N_\beta(\mathcal{F}), K^Y_\gamma(\mathcal{F}), K^N_\beta(\mathcal{F})$.

3.2 The Dichotomy for Sketching Algorithms

The following theorem now formalizes the informal statement that low space sketching algorithms (see Definition 2.3) can only capture the marginals of distributions.

Theorem 3.3 (Dichotomy for Sketching Algorithms). *For every $q, k \in \mathbb{N}$, every family of functions $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$, and every $0 \leq \beta < \gamma \leq 1$, the following hold:*

1. *If $K^Y_\gamma(\mathcal{F}) \cap K^N_\beta(\mathcal{F}) = \emptyset$, then (γ,β)-Max-CSP($\mathcal{F}$) admits a uniform randomized linear sketching algorithm that uses $O(\log^3 n)$ space[4] on instances on n variables.*
2. *If $K^Y_\gamma(\mathcal{F}) \cap K^N_\beta(\mathcal{F}) \neq \emptyset$, then for every $\varepsilon > 0$, every (non-uniform randomized) sketching algorithm for the $(\gamma-\varepsilon, \beta+\varepsilon)$-Max-CSP($\mathcal{F}$) requires $\Omega(\sqrt{n})$ space[5] on instances on n variables. Furthermore, if $\gamma = 1$, then every sketching algorithm for $(1, \beta+\varepsilon)$-Max-CSP($\mathcal{F}$) requires $\Omega(\sqrt{n})$ space.*

We remark that Part 1 of Theorem 3.3 is actually stronger and holds even for dynamic streams where constraints are added and deleted, provided the total length of the stream is polynomial in n. Theorem 3.3 is proved in two parts: Theorem 4.1 proves Theorem 3.3, Part 1, while Theorem 5.1 proves Theorem 3.3, Part 2.

We now complement Theorem 3.3 by showing that the condition "$K^Y_\gamma(\mathcal{F}) \cap K^N_\beta(\mathcal{F}) = \emptyset$?" can be decided in polynomial space given γ and β as ratios of ℓ-bit integers and members of $\mathcal{F}$ as truth tables. (So the input is of size $O(\ell + |\mathcal{F}| \cdot q^k)$ and our algorithm needs space polynomial in this quantity.)

Theorem 3.4. *For every $k, q \in \mathbb{N}$ $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$, and ℓ-bit rationals $\beta, \gamma \in [0,1]$ (i.e., β and γ are expressible as the ratio of two integers in $\{-2^\ell, \ldots, 2^\ell\}$), the condition "$K^Y_\gamma(\mathcal{F}) \cap K^N_\beta(\mathcal{F}) = \emptyset$?" can be decided in space* $\mathrm{poly}(|\mathcal{F}|, q^k, \ell)$ *given truth tables of all elements of $\mathcal{F}$ and γ and β as ℓ-bit rationals.*

We include a proof of Theorem 3.4 in Section 3.5.1.

[4] In particular, the space complexity is $O(\log^3 n)$ bits, or $O(\log^2 n)$ cells, where each cell is $O(\log n)$ bits long. Crucially, while the constant in the $O(\cdot)$ depends on k, γ, and β, the exponent is a universal constant.
[5] Again, the constant hidden in the Ω notation depends on k, γ, and β.

3.3 Other Results

3.3.1 Approximation Resistance of Sketching Algorithms. We now turn to the notion of "approximation resistant" Max-CSP($\mathcal{F}$) problems. We start with a discussion where $\mathcal{F} = \{f\}$. In the setting where constraints are applied to literals rather than variables, the notion of approximation resistance is used to refer to problems where it is hard to outperform a uniform random assignment. In other words, if $\rho(f)$ is defined to be the probability that a random assignment satisfies f, then Max-CSP(f) is defined to be approximation resistant if $(1-\varepsilon, \rho(f)+\varepsilon)$-Max-CSP($f$) is hard. In our setting, however, where constraints are applied to variables, this notion is a bit more nuanced. Here it may be possible to construct functions where a random assignment does poorly and yet every instance has a much higher value.[6] In our setting, the correct notion is to simply consider the infimum value achieved over instances of Max-CSP(f). If this quantity is ρ, then it is trivial to get a ρ-approximation for Max-CSP(f)—namely, the algorithm that outputs the constant ρ is always correct and gives a ρ-approximation. (Equivalently, (γ, β)-Max-CSP(f) can be decided by the algorithm that always outputs YES if $\beta < \rho$.) And if $(1-\varepsilon, \rho(f)+\varepsilon)$-Max-CSP($f$) is hard for every $\varepsilon > 0$, then we can say that Max-CSP(f) is approximation resistant.

The only catch with the above notion of approximation resistant is that ρ may not be computable. To resolve this problem we introduce an alternate definition of this quantity ρ and prove that it is equivalent and computable. We start with the definitions, generalized for all $\mathcal{F}$.

Definition 3.5 (Approximation Resistance for Streaming/Sketching Algorithms). For $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$, we define

$$\rho_{\min}(\mathcal{F}) = \lim \inf_{\Psi \text{ instance of Max-CSP}(\mathcal{F})} \{\mathsf{val}_\Psi\}.$$

We say that Max-CSP($\mathcal{F}$) is *approximation resistant* for streaming algorithms (resp. sketching algorithms) if for every $\varepsilon > 0$ there exists $\delta > 0$ such that every streaming (resp. sketching) algorithm for $(1-\varepsilon, \rho_{\min}(\mathcal{F})+\varepsilon)$-Max-CSP($\mathcal{F}$) requires $\Omega(n^\delta)$ space. We also define

$$\rho(\mathcal{F}) = \min_{\mathcal{D}_\mathcal{F} \in \Delta(\mathcal{F})} \left\{ \max_{\mathcal{D} \in \Delta([q])} \left\{ \mathop{\mathbb{E}}_{f \sim \mathcal{D}_\mathcal{F}, \mathbf{a} \sim \mathcal{D}^k} [f(\mathbf{a})] \right\} \right\}.$$

The following proposition asserts the equivalence of $\rho_{\min}(\mathcal{F})$ and $\rho(\mathcal{F})$.

Proposition 3.6. *For every $q, k \in \mathbb{N}$, $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$ we have $\rho_{\min}(\mathcal{F}) = \rho(\mathcal{F})$.*

Proposition 3.6 allows us to show that $\rho(\mathcal{F})$ is computable as asserted below.

Theorem 3.7. *There is an algorithm A that, on input $\mathcal{F} \subseteq \{[q]^k \to \{0,1\}\}$ presented as $|\mathcal{F}|$ truth tables and $\tau \in \mathbb{R}$ presented as an ℓ-bit rational, answers the question "Is $\rho_{\min}(\mathcal{F}) \le \tau$?" in space* $\mathsf{poly}(|\mathcal{F}|, q^k, \ell)$.

Theorem 3.3 immediately yields a decidable characterization of Max-CSP($\mathcal{F}$) problems that are approximation resistant with respect to sketching algorithms.

Theorem 3.8 (Classification of Sketching Approximation Resistance). *For every $q, k \in \mathbb{N}$, for every family $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$, Max-CSP($\mathcal{F}$) is approximation resistant with respect to sketching algorithms if and only if $K_1^Y(\mathcal{F}) \cap K^N_{\rho(\mathcal{F})}(\mathcal{F}) \neq \emptyset$. Furthermore, if Max-CSP($\mathcal{F}$) is approximation resistant with respect to sketching algorithms, then for every $\varepsilon > 0$ we have that $(1, \rho(\mathcal{F})+\varepsilon)$-Max-CSP($\mathcal{F}$) requires $\Omega(\sqrt{n})$ space for non-uniform randomized sketching algorithms.*

[6] Take, for instance, $f(x_1) = 1$ iff $x_1 = 1$. The random assignment satisfies f with probability $1/q$, while every instance is satisfiable!

If Max-CSP($\mathcal{F}$) is not approximation resistant with respect to sketching algorithms, then there exists $\varepsilon > 0$ such that $(1-\varepsilon, \rho(\mathcal{F})+\varepsilon)$-Max-CSP($\mathcal{F}$) can be solved in polylogarithmic space by a uniform randomized linear sketching algorithm. Finally, given the truth table of the functions in $\mathcal{F}$, there is an algorithm running in space $\mathrm{poly}(q^k|\mathcal{F}|)$ *that decides whether or not Max-CSP($\mathcal{F}$) is approximation resistant with respect to sketching algorithms.*

Proposition 3.6 and Theorems 3.7 and 3.8 are proved in Section 3.5.2.

3.3.2 Lower Bounds in the Streaming Setting. We now turn to some special classes of CSPs where we can prove lower bounds in the streaming setting as opposed to only ruling out sketching algorithms. To describe these classes we need some definitions.

We start by defining the notion of a "one-wise independent" distribution $\mathcal{D} \in \Delta(\mathcal{F} \times [q]^k)$. (We note that this is somewhat related to, but definitely not the same as, the notion of a family $\mathcal{F}$ that *supports* one-wise independence, which was defined informally in Section 1. We will recall that notion shortly.) We also define a broader notion of a "padded one-wise pair" of distributions.

Definition 3.9 (One-wise Independence and Padded One-wise Independence of Distributions). For $\mathcal{D} \in \Delta(\mathcal{F}\times[q]^k)$ we say that $\mathcal{D}$ is *one-wise independent* (or has "uniform marginals") if its marginal $\boldsymbol{\mu}(\mathcal{D}) = (\mu_{f,i,\sigma})_{f\in\mathcal{F}, i\in[k], \sigma\in[q]}$ satisfies $\mu_{f,i,\sigma} = \mu_{f,i,\sigma'}$ for every $f \in \mathcal{F}$, $i \in [k]$ and $\sigma, \sigma' \in [q]$. (In other words for every $f_0 \in \mathcal{F}$ and $i \in [k]$, the random variable a_i obtained by sampling $(f, (a_1, \ldots, a_k)) \sim \mathcal{D}$ conditioned on $f = f_0$ and projecting to a_i is uniformly distributed over $[q]$.)

We say that a pair of distributions $(\mathcal{D}_1, \mathcal{D}_2)$ forms a *padded one-wise pair* if there exist $\mathcal{D}_0, \mathcal{D}_1', \mathcal{D}_2'$, and $\tau \in [0,1]$ such that for every $i \in \{1,2\}$ we have that $\mathcal{D}_i'$ is one-wise independent and $\mathcal{D}_i = \tau\mathcal{D}_0 + (1-\tau)\mathcal{D}_i'$.

Our main lower bound in the streaming setting asserts that if $S_\gamma^Y(\mathcal{F}) \times S_\beta^N(\mathcal{F})$ contains a padded one-wise pair $(\mathcal{D}_Y, \mathcal{D}_N)$, then (γ,β)-Max-CSP($\mathcal{F}$) requires $\Omega(\sqrt{n})$-space.

Theorem 3.10 (Streaming Lower Bound). *For every $q, k \in \mathbb{N}$, every family of functions $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$ and for every $0 \le \beta < \gamma \le 1$, if there exists a padded one-wise pair of distributions $\mathcal{D}_Y \in S_\gamma^Y(\mathcal{F})$ and $\mathcal{D}_N \in S_\beta^N(\mathcal{F})$ then, for every $\varepsilon > 0$, every non-uniform randomized streaming algorithm that solves the $(\gamma-\varepsilon, \beta+\varepsilon)$-Max-CSP($\mathcal{F}$) problem requires $\Omega(\sqrt{n})$ space. Furthermore, if $\gamma = 1$, then $(1, \beta+\varepsilon)$-Max-CSP(f) requires $\Omega(\sqrt{n})$ space.*

Theorem 3.10 is proved in Section 5.2.4. As stated above, the theorem is more complex to apply than, say, Theorem 3.3, owing to the fact that the condition for hardness depends on the entire distribution (and the sets S_γ^Y and S_β^N) rather than just marginals (or the sets K_γ^Y and K_β^N). However, it can be used to derive some clean results, specifically Theorems 3.12 and 1.3, that do depend only on the marginals. We state these below after defining a notion of a function family supporting one-wise independence.

Definition 3.11 ((Weakly/Strongly) Supporting One-wise Independence). We say that a function $f : [q]^k \to \{0,1\}$ supports *one-wise independence* if there exists a distribution $\mathcal{D}$ supported on $f^{-1}(1)$ whose marginals are uniform on $[q]$. We say that a family $\mathcal{F}$ *strongly supports one-wise independence* if every function $f \in \mathcal{F}$ supports one-wise independence. We say that a family $\mathcal{F}$ *weakly supports one-wise independence* if there exists $\mathcal{F}' \subseteq \mathcal{F}$ satisfying $\rho(\mathcal{F}') = \rho(\mathcal{F})$ such that every function $f \in \mathcal{F}'$ supports one-wise independence.

Theorem 3.12. *For every $q, k \in \mathbb{N}$ and $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$ such that $\mathcal{F}$ weakly supports one-wise independence, Max-CSP($\mathcal{F}$) is approximation resistant with respect to streaming algorithms. In particular, for every $\varepsilon > 0$, every non-uniform randomized streaming algorithm for $(1, \rho(\mathcal{F})+\varepsilon)$-Max-CSP($\mathcal{F}$) requires $\Omega(\sqrt{n})$ space.*

Remark 3.13. We note that Theorem 1.2 differs from Theorem 3.12 in that Theorem 1.2 asserted hardness for $\mathcal{F}$ that strongly supports one-wise independence, whereas Theorem 3.12 asserts hardness for $\mathcal{F}$ that weakly supports one-wise independence. Thus, Theorem 3.12 is stronger and implies Theorem 1.2.

Finally, we turn to Theorem 1.3. Below we assert a more detailed version of the theorem along the lines of Theorem 3.3 in this case.

THEOREM 3.14. *For every family $\mathcal{F} \subseteq \{f : [2]^2 \to \{0,1\}\}$, and for every $0 \le \beta < \gamma \le 1$, the following hold:*

(1) If $K^Y_\gamma(\mathcal{F}) \cap K^N_\beta(\mathcal{F}) = \emptyset$, then (γ,β)-Max-CSP($\mathcal{F}$) admits a uniform randomized linear sketching algorithm that uses $O(\log^3 n)$ space.

(2) If $K^Y_\gamma(\mathcal{F}) \cap K^N_\beta(\mathcal{F}) \neq \emptyset$, then for every $\varepsilon > 0$, $(\gamma-\varepsilon,\beta+\varepsilon)$-Max-CSP($\mathcal{F}$) in the streaming setting requires $\Omega(\sqrt{n})$ space.[7] Furthermore, if $\gamma = 1$, then $(1,\beta+\varepsilon)$-Max-CSP($\mathcal{F}$) in the streaming setting requires $\Omega(\sqrt{n})$ space for non-uniform randomized streaming algorithms.

Theorem 3.14 clearly implies Theorem 1.3. We prove Theorems 3.12 and 3.14 in Section 3.5.3.

3.3.3 Classification of Exact Computability. Finally, for the sake of completeness we show that all "non-trivial" CSPs are hard to solve exactly. "Trivial" families are those where all satisfiable constraints are satisfied by a constant assignment, as defined precisely below.

Definition 3.15 (Constant Satisfiable). For $\sigma \in [q]$ and $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$ we say that $\mathcal{F}$ is σ-satisfiable if for every $f \in \mathcal{F} \setminus \{\mathbf{0}\}$ we have that $f(\sigma^k) = 1$. We say $\mathcal{F}$ is constant satisfiable if there exists $\sigma \in [q]$ such that $\mathcal{F}$ is σ-satisfiable.

Our theorem below asserts that constant-satisfiable families are the only ones that are solvable exactly. And for additive ε approximations to the maximum fraction of satisfiable constraints, they require space growing polynomially in ε^{-1}.

THEOREM 3.16. *For every $q, k \in \mathbb{N}$, and every family of functions $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$, the following hold:*

(1) If $\mathcal{F}$ is constant satisfiable, then there exists a deterministic linear sketching algorithm that uses $O(\log n)$ space and solves Max-CSP($\mathcal{F}$) exactly optimally.

(2) If $\mathcal{F}$ is not constant satisfiable, then the following hold in the streaming setting:

(a) Every probabilistic algorithm solving Max-CSP($\mathcal{F}$) exactly requires $\Omega(n)$ space.

(b) For every $\varepsilon = \varepsilon(n) > 0$, $(1,1-\varepsilon)$-Max-CSP($\mathcal{F}$) requires $\Omega(\min\{n, \varepsilon^{-1}\})$-space[8] on sufficiently large inputs.

(c) For $\rho_{\min}(\mathcal{F})$ defined in Definition 3.5, for every $\rho_{\min}(\mathcal{F}) < \gamma < 1$ and every $\varepsilon = \varepsilon(n) > 0$, $(\gamma, \gamma-\varepsilon)$-Max-CSP($\mathcal{F}$) requires $\Omega(\min\{n, \varepsilon^{-2}\})$-space[8] on sufficiently large inputs.

Theorem 3.16 is proved in Section 9.

3.4 Some Examples

We consider three basic examples of general q-CSP and illustrate how to apply Theorem 3.10 to determine their approximability.

[7] The constant hidden in the Ω notation may depend on k and ε.

[8] The constant hidden in the Ω depends on $\mathcal{F}$, but (obviously) not on ε.

The first example is Max-DICUT described below.

Example 1 (Max-DICUT).

Let $f(x,y) : [2]^2 \to \{0,1\}$ with $f(x,y) = 1$ if and only if $x = 2$ and $y = 1$. Note that Max-DICUT = Max-CSP($\{f\}$) with $q = k = 2$. Observe that for every distribution $\mathcal{D} \in \Delta([q]^k)$ with probability density vector $\boldsymbol{\phi}(\mathcal{D}) = (\phi_{22}, \phi_{21}, \phi_{12}, \phi_{11})$, we have for every $0 \le \gamma, \beta \le 1$:

$$S^Y_\gamma(\mathcal{F}) = \{\mathcal{D} \mid \phi_{21} \ge \gamma\}$$

and

$$S^N_\beta(\mathcal{F}) = \left\{\mathcal{D} \mid \max_{p,q\in[0,1]} p(1-p)\cdot\phi_{22} + pq\cdot\phi_{21} + (1-q)(1-p)\cdot\phi_{12} + (1-q)q\cdot\phi_{11} \le \beta\right\}.$$

Also, note that the marginal vector $\boldsymbol{\mu}(\mathcal{D}) = (\mu_{22}, \mu_{21}, \mu_{12}, \mu_{11})$ and $\boldsymbol{\phi}(\mathcal{D})$ satisfy the following relations:

$$\begin{cases} \mu_{22} = \phi_{12} + \phi_{22} \\ \mu_{21} = \phi_{11} + \phi_{21} \\ \mu_{12} = \phi_{21} + \phi_{22} \\ \mu_{11} = \phi_{11} + \phi_{12}\,. \end{cases}$$

Note that for every $\mathcal{D} \in \Delta([q]^k)$, we have $\mathcal{D} \in S^N_{1/4}$. In particular, the uniform distribution $\mathsf{Unif}([2]^2) \in S^N_{1/4}$. Since the distribution given by the density vector $(\phi_{22} = 0, \phi_{21} = 1/2, \phi_{12} = 1/2, \phi_{11} = 0)$ also has uniform marginals and belongs to $S^Y_{1/2}$, we have that for every $\beta \ge 1/4$, $K^Y_{1/2} \cap K^N_\beta(\mathcal{F}) \ne \emptyset$. So it suffices to focus on the case where $\gamma \ge 1/2$.

Fix $\gamma \ge 1/2$; we want to compute the minimum β such that $K^Y_\gamma(\mathcal{F}) \cap K^N_\beta(\mathcal{F}) \ne \emptyset$. The kernel of the mapping from probability density $\boldsymbol{\phi}$ to the marginal vector $\boldsymbol{\mu}$ is spanned by $(1,-1,-1,1)$. Then simple calculations show that the minimum β is achieved when $\boldsymbol{\mu} = (1-\gamma, \gamma, \gamma, 1-\gamma)$ with $(0, \gamma, 1-\gamma, 0) \in S^Y_\gamma(\mathcal{F})$ and $(1-\gamma, 2\gamma-1, 0, 1-\gamma) \in S^N_\beta(\mathcal{F})$. Specifically,

$$\begin{aligned}\beta &= \max_{p,q\in[0,1]} (p(1-p) + q(1-q))\cdot(1-\gamma) + pq\cdot(2\gamma-1) \\ &= \max_{p,q\in[0,1]} \frac{(1-\gamma)^2}{3-4\gamma} - \frac{3-4\gamma}{2}\cdot\left(\left(p + \frac{1-\gamma}{4\gamma-3}\right)^2 + \left(q + \frac{1-\gamma}{4\gamma-3}\right)^2\right) - \frac{(2\gamma-1)}{2}\cdot(p-q)^2\,.\end{aligned}$$

When $\gamma \ge 2/3$, the expression is maximized by $p = q = 1$ and hence $\beta = 2\gamma - 1$. When $1/2 \le \gamma \le 2/3$, the expression is maximized by $p = q = (1-\gamma)/(3-4\gamma)$ and hence $\beta = (1-\gamma)^2/(3-4\gamma)$.

We thus get that the set $H^\cap \triangleq \{(\gamma,\beta) \in [0,1]^2 | K^Y_\gamma \cap K^N_\beta \ne \emptyset\}$ (of *hard* problems) is given by (see also Figure 1)

$$\begin{aligned} H^\cap = &\left[0, \frac{1}{2}\right] \times \left[\frac{1}{4}, 1\right] \\ &\cup \left\{(\gamma,\beta) | \gamma \in \left[\frac{1}{2}, \frac{2}{3}\right], \beta \in \left[\frac{(1-\gamma)^2}{3-4\gamma}, 1\right]\right\} \\ &\cup \left\{(\gamma,\beta) | \gamma \in \left[\frac{2}{3}, 1\right], \beta \in [2\gamma-1, 1]\right\}. \end{aligned}$$

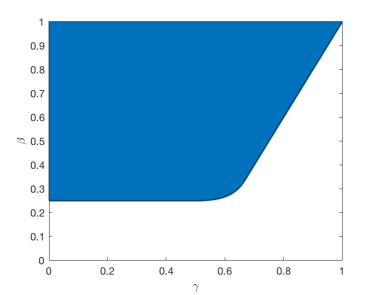


Fig. 1. A plot of $H^{\cap}$.

(We note that [37, Example 1] gives exactly the same set as the hard set of Max-2AND, which is a related but not identical result.)
Finally, over $\gamma \in [2/3, 1]$, β/γ is minimized at $(\gamma, \beta) = (2/3, 1/3)$ and $\beta/\gamma = 1/2$; over $\gamma \in [1/2, 2/3]$, β/γ is minimized at $(\gamma, \beta) = (3/5, 4/15)$ and $\beta/\gamma = 4/9$, yielding $4/9$ as the approximability threshold.
Specifically, Proposition 3.22 gives us that any pair of distributions $\mathcal{D}_Y, \mathcal{D}_N \in \Delta(\mathcal{F} \times [2]^2), \mathcal{D}_Y \in S^Y_{3/5}, \mathcal{D}_Y \in S^N_{4/15}$ witnessing $K^Y_{3/5} \cap K^Y_{4/15} \neq \emptyset$ forms a padded one-wise pair. Finally, Theorem 3.10, applied to the padded one-wise pair $(\mathcal{D}^Y, \mathcal{D}^N)$, implies that Max-DICUT cannot be approximated better with a factor $(4/9+\varepsilon)$ in space $o(\sqrt{n})$ in the streaming setting, which is consistent with the findings in [41] for the Max-DICUT problem.

Example 2 (Max-qUG).

Let $k = 2$ and $q \geq 2$. Let $\mathcal{F} = \{f : [q]^2 \to \{0, 1\} \mid f^{-1}(1) \text{ is a bijection}\}$. Note that Max-$q$UG = Max-CSP($\mathcal{F}$). We claim that the quantity $\alpha = \inf_\beta \alpha(\beta) = 1/q$, where $\alpha(\beta) = \sup_{\gamma | K^Y_\gamma \cap K^N_\beta = \emptyset} \{\beta/\gamma\}$. First, note that $\mathcal{D} \in S^N_{1/q}$ for every $\mathcal{D}$ and hence implies $\alpha \geq 1/q$.
For simplicity we work with the alphabet $\mathbb{Z}_q = \{0, \ldots, q-1\}$ instead of $[q]$. For $\tau \in \mathbb{Z}_q$ let $f_\tau \in \mathcal{F}$ be the constraint $f_\tau(x, y) = 1$ if and only if $x - y = \tau \pmod q$. Let $\mathcal{D}^Y$ be the uniform distribution over $\{(f_\tau, \sigma, \sigma + \tau) \mid \sigma, \tau \in \mathbb{Z}_q\}$. Note that obviously we have $\mathcal{D}^Y \in S^Y_1$. Now let $\mathcal{D}^N$ be the uniform distribution over $\{f_\tau \mid \tau \in \mathbb{Z}_q\} \times \mathbb{Z}^2_q$. Note that for any assignment to two variables $x_{1,\sigma_1}, x_{2,\sigma_2}$ the probability over τ that it satisfies $f_\tau(x_{1,\sigma_1}, x_{2,\sigma_2})$ is exactly $1/q$. If follows that any assignment to $(x_{i,\sigma})_{i,\sigma}$ satisfies exactly $1/q$ fraction of the constraints in $\mathcal{D}^N$ and so $\mathcal{D}^N \in S^N_{1/q}$. Observe that the marginals of $\mathcal{D}^Y$ and $\mathcal{D}^N$ are the same, i.e., $\boldsymbol{\mu}(\mathcal{D}^Y) = \boldsymbol{\mu}(\mathcal{D}^N) = \boldsymbol{\mu}(\text{Unif}(\{f_\tau\} \times \mathbb{Z}^2_q))$. This gives us $\boldsymbol{\mu}(\text{Unif}(\{f_\tau\} \times [q]^2)) \in K^Y_1 \cap K^N_{1/q}$, so we have $\alpha(\beta) = \beta$ for $\beta \geq 1/q$. Minimizing this over β, Theorem 3.10, applied to the one-wise independent distribution $\mathcal{D}^Y$ and $\mathcal{D}^N$, gives that the problem cannot be approximated better than $1/q$ in space $o(\sqrt{n})$ in the streaming setting, which is consistent with the findings in [46] for the Max-qUG problem.

Example 3 (Max-qCol)

Let $k = 2$ and $q \geq 2$. Let $\mathcal{F} = \{f_{\neq}\}$, where $f_{\neq} : [q]^2 \to \{0,1\}$ is given by $f_{\neq}(x,y) = 1 \Leftrightarrow x \neq y$. Note that Max-$q$Col = Max-CSP($\mathcal{F}$). We claim that the quantity $\alpha = \inf_\beta \alpha(\beta) = 1 - 1/q$, where $\alpha(\beta) = \sup_{\gamma \,|\, K_\gamma^Y \cap K_\beta^N = \emptyset}\{\beta/\gamma\}$. First, note that $\mathcal{D} \in S^N_{1-1/q}$ for every $\mathcal{D}$ and hence implies $\alpha \geq 1 - 1/q$. We now show this is also the upper bound by exhibiting $\mathcal{D}^Y$ and $\mathcal{D}^N$. Let $\mathcal{D}^Y$ be the uniform distribution over $\{(f_{\neq}, \sigma, \tau) \,|\, \sigma \neq \tau \in [q]\}$. Note that obviously we have $\mathcal{D}^Y \in S^Y_1$. Now let $\mathcal{D}^N$ be the uniform distribution over $\{f_{\neq}\} \times [q]^2$. This leads to $\beta = \max_{\mathcal{P}_\sigma}\{\mathbb{E}_{(f,a_1,a_2)\sim\mathcal{D}^N}[\mathbb{E}_{x\sim\mathcal{P}_{a_1}, y\sim\mathcal{P}_{a_2}}[f(x,y)]]\}$. The independence of a_1 and a_2 in $\mathcal{D}^N$ allows us to simplify this to $\max_{\mathcal{P}\in\Delta([q])}\{\mathbb{E}_{x,y\sim\mathcal{P}}[f_{\neq}(x,y)]\}$, and the latter is easily seen to be at most $1 - 1/q$. Thus, we conclude $\mathcal{D}^N \in S^N_{1-1/q}$. Since the marginals of $\mathcal{D}^Y$ and $\mathcal{D}^N$ are the same, i.e., $\boldsymbol{\mu}(\mathcal{D}^Y) = \boldsymbol{\mu}(\mathcal{D}^N) = \boldsymbol{\mu}(\mathsf{Unif}(\{f_{\neq}\} \times [2] \times [q]))$, this gives us $\boldsymbol{\mu}(\mathsf{Unif}(\{f_{\neq}\} \times [2] \times [q])) \in K^Y_1 \cap K^N_{1/q}$, so we have $\alpha(\beta) = \beta$ for $\beta \geq 1 - 1/q$. Minimizing this over β, Theorem 3.10, applied to the one-wise independent distribution $\mathcal{D}^Y$ and $\mathcal{D}^N$, gives that the problem cannot be approximated better than $1 - 1/q$ in space $o(\sqrt{n})$ in the streaming setting.

Another example along the same vein is analyzed in a subsequent work by Singer et al. [73], who show that $(1 - 1/q, (1/2)(1 - 1/q))$-Max-CSP($\mathcal{F}$) is hard for $\mathcal{F} = \{f_<\}$, where $f_< : [q]^2 \to \{0,1\}$ is given by $f_<(x,y) = 1$ if and only if $x < y$. This analysis forms a critical step in their improved analysis of the Maximum Acyclic Subgraph Problem (which is not captured in our framework).

3.5 Some Proofs of Theorems Asserted in This Section

In this subsection we prove all results asserted in Sections 3.2 and 3.3, with the exception of Theorems 3.3, 3.10, and 3.16.

3.5.1 Decidability of the Classification. We prove Theorem 3.4 in this section. The following lemma states some basic properties of the sets $S^Y_\gamma(\mathcal{F}), S^N_\beta(\mathcal{F}), K^Y_\gamma(\mathcal{F})$, and $K^N_\beta(\mathcal{F})$ and uses them to express the condition "$K^Y_\gamma(\mathcal{F}) \cap K^N_\beta(\mathcal{F}) = \emptyset$?" in the quantified theory of reals.

Lemma 3.17. *For every $k, q \in \mathbb{N}$ $\beta, \gamma \in [0,1]$ and $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$, the sets $S^Y_\gamma(\mathcal{F}), S^N_\beta(\mathcal{F})$, $K^Y_\gamma(\mathcal{F})$, and $K^N_\beta(\mathcal{F})$ are bounded, closed, and convex. Furthermore, the condition $K^Y_\gamma(\mathcal{F}) \cap K^N_\beta(\mathcal{F}) = \emptyset$ can be expressed in the quantified theory of reals with two quantifier alternations, $O(|\mathcal{F}|q^k + q^2)$ variables, and polynomials of degree at most $k + 1$.*

Proof. We start by observing that $\Delta(\mathcal{F} \times [q]^k)$ is a bounded convex polytope in $\mathbb{R}^{|\mathcal{F}|\times[q]^k}$. Furthermore, viewing $\mathcal{D}$ as a vector in $\mathbb{R}^{|\mathcal{F}|\times[q]^k}$, for any given $\mathbf{b} \in [q]^k$ the quantity $\mathbb{E}_{(f,a)\sim\mathcal{D}}[C(f,\mathbf{a})(\mathbf{b})]$ is linear in $\mathcal{D}$. Thus, $S^Y_\gamma(\mathcal{F})$ is given by a single linear constraint on $\Delta(\mathcal{F}\times[q]^k)$, making it a bounded convex polytope as well. $S^N_\beta(\mathcal{F})$ is a bit more complex, in that there are infinitely many linear inequalities defining it (one for every distribution $(\mathcal{P}_\sigma)_{\sigma\in[q]}$). Nevertheless, this leaves $S^N_\beta(\mathcal{F})$ bounded, closed (as infinite intersection of closed sets is closed), and convex (though it may no longer be a polytope). Finally, since $K^Y_\gamma(\mathcal{F})$ and $K^N_\beta(\mathcal{F})$ are linear projections of $S^Y_\gamma(\mathcal{F})$ and $S^N_\beta(\mathcal{F})$, respectively, they retain the features of being bounded, closed, and convex.

Finally, to get an effective algorithm for intersection detection, we express the intersection condition in the quantified theory of the reals. To get this, we note that $(\mathcal{P}_\sigma)_{\sigma\in[q]}$ can be expressed

by q^2 variables, specifically using variables $\mathcal{P}_\sigma(\tau)$ for every $\sigma, \tau \in [q]$, where $\mathcal{P}_\sigma(\tau)$ denotes the probability of τ in $\mathcal{P}_\sigma$. In terms of these variables (which will eventually be quantified over), the condition $\mathbb{E}_{(f,\mathbf{a})\sim\mathcal{D}}\left[\mathbb{E}_{\mathbf{b}, b_{i,\sigma}\sim\mathcal{P}_\sigma}[C(f,\mathbf{a})(\mathbf{b})]\right] \le \beta$ is a multivariate polynomial inequality in $(\mathcal{P}_\sigma)_\sigma$ and $\mathcal{D}$. (Specifically, we get a polynomial of total degree at most k in $(\mathcal{P}_\sigma)_\sigma$ and of total degree at most one in $\mathcal{D}$.) This allows us to use the following quantified system to express the condition $K_\gamma^Y(\mathcal{F}) \cap K_\beta^N(\mathcal{F}) \neq \emptyset$:

$$\exists \mathcal{D}_Y, \mathcal{D}_N \in \mathbb{R}^{|\mathcal{F}|\times q^k},\ \forall((\mathcal{P}_\sigma)_\sigma) \in \mathbb{R}^{q^2} \text{ s.t.}$$

$$\mathcal{D}_Y, \mathcal{D}_N, (\mathcal{P}_\sigma)_\sigma, \forall\sigma \in [q] \text{ are distributions,} \tag{3.18}$$

$$\forall f_0 \in \mathcal{F}, \forall i \in [k], \tau \in [q] \quad \Pr_{(f,\mathbf{a})\sim\mathcal{D}_Y}[f = f_0 \text{ and } a_i = \tau] = \Pr_{(f,\mathbf{a})\sim\mathcal{D}_N}[f = f_0 \text{ and } a_i = \tau], \tag{3.19}$$

$$\mathbb{E}_{(f,\mathbf{a})\sim\mathcal{D}_Y}[C(f,\mathbf{a})(\mathbb{I})] \ge \gamma, \tag{3.20}$$

$$\mathbb{E}_{(f,\mathbf{a})\sim\mathcal{D}_N}\left[\mathbb{E}_{\mathbf{b}, b_{i,\sigma}\sim\mathcal{P}_\sigma}[C(f,\mathbf{a})(\mathbf{b})]\right] \le \beta. \tag{3.21}$$

Note that Equations (3.18) to (3.20) are just linear inequalities in the variables $\mathcal{D}_Y, \mathcal{D}_N$.

As noticed above, Equation (3.21) is an inequality in the $\mathcal{P}_\sigma$s and $\mathcal{D}_N$, of total degree at most $k+1$.

We thus get that the intersection problem can be expressed in the quantified theory of the reals by an expression with two quantifier alternations, $2|\mathcal{F}|q^k + q^2$ variables, and $O(|\mathcal{F}|q^k + q^2)$ polynomial inequalities, with polynomials of degree at most $k+1$. (Most of the inequalities are of the form $\mathcal{D}_Y(\mathbf{b}) \ge 0$ or $\mathcal{D}_N(\mathbf{b}) \ge 0$. We also have $O(|\mathcal{F}|kq)$ equalities (saying probabilities must add to one and matching the marginals of $\mathcal{D}_Y$ and $\mathcal{D}_N$). Of the two remaining, Equation (3.20) is linear; only Equation (3.21) is a higher-degree polynomial. □

We are now ready to prove Theorem 3.4.

Proof of Theorem 3.4. The quantified polynomial system given by Lemma 3.17 yields parameters $K = O(|\mathcal{F}|q^k + q^2)$ for the number of variables and $w = 2$ for the number of alternations. Applying Theorem 2.13 with these parameters yields the theorem. □

3.5.2 Approximation Resistance. We start by proving Proposition 3.6, which asserts that $\rho(\mathcal{F}) = \rho_{\min}(\mathcal{F})$.

Proof of Proposition 3.6. We start by showing $\rho(\mathcal{F}) \le \rho_{\min}(\mathcal{F})$. Fix an instance Ψ of Max-CSP($\mathcal{F}$) and let $\mathcal{D}_\mathcal{F}$ be the distribution on $\mathcal{F}$ obtained by picking a random constraint of Ψ and looking at the function (while ignoring the variables that the constraint is applied to). By the definition of $\rho(\mathcal{F})$, there exists a distribution $\mathcal{D} \in \Delta([q])$ such that $\mathbb{E}_{f\sim\mathcal{D}_\mathcal{F}, \mathbf{a}\sim\mathcal{D}^k}[f(\mathbf{a})] \ge \rho(\mathcal{F})$. Now consider a random assignment to the variables of Ψ where variable x_j is assigned a value independently according to $\mathcal{D}$. It can be verified that $\mathbb{E}_\mathbf{x}[\mathrm{val}_\Psi(\mathbf{x})] \ge \rho(\mathcal{F})$ and so $\mathrm{val}_\Psi \ge \rho(\mathcal{F})$. We thus conclude that $\rho(\mathcal{F}) \le \mathrm{val}_\Psi$ for all Ψ and so $\rho(\mathcal{F}) \le \rho_{\min}(\mathcal{F})$.

We now turn to the other direction. We prove that for every $\varepsilon > 0$ we have $\rho_{\min}(\mathcal{F}) \le \rho(\mathcal{F}) + \varepsilon$ and the inequality follows by taking limits. Let $\mathcal{D}_\mathcal{F}$ be the distribution achieving the minimum in the definition of $\rho(\mathcal{F})$. Given $\varepsilon > 0$, let n be a sufficiently large integer and let $m = O(n^k/\varepsilon)$. Let Ψ be the instance of Max-CSP($\mathcal{F}$) on n variables with m constraints chosen as follows: For every $\mathbf{j} \in [n]^k$ with distinct coordinates and every $f \in \mathcal{F}$ we place $\lfloor \mathcal{D}_\mathcal{F}(f)/\varepsilon \rfloor$ copies of the constraint $(f, \mathbf{j})$.

We claim that the Ψ generated above satisfies $\mathrm{val}_\Psi \le \rho(\mathcal{F}) + \varepsilon/2 + O(1/n)$, and this suffices for the proposition. To see the claim, fix an assignment $\boldsymbol{\nu} \in [q]^n$ and let $\mathcal{D} \in \Delta([q])$ be the distribution induced by sampling $i \in [n]$ uniformly and outputting $\boldsymbol{\nu}_i$. On the one hand, we have

from the definition of $\rho(\mathcal{F})$ that $\mathbb{E}_{f\sim\mathcal{D}_\mathcal{F},\mathbf{a}\sim\mathcal{D}^k}[f(\mathbf{a})] \le \rho(\mathcal{F})$. On the other hand, we have that the distribution obtained by sampling a random constraint $(f,\mathbf{j})$ of Ψ and outputting $(f,\boldsymbol{\nu}|_\mathbf{j})$ is $\varepsilon/2 + O(1/n)$ close in total variation distance to sampling $f \sim \mathcal{D}_\mathcal{F}$ and $\mathbf{a} \sim \mathcal{D}^k$. (The $\varepsilon/2$ gap comes from the rounding down of each constraint to an integral number, and the $O(1/n)$ gap comes from the fact that $\mathbf{j}$ is sampled from $[n]$ without replacement.) We thus conclude that

$$\mathrm{val}_\Psi(\boldsymbol{\nu}) \le \mathop{\mathbb{E}}_{f\sim\mathcal{D}_\mathcal{F},\mathbf{a}\sim\mathcal{D}^k}[f(\mathbf{a})] + \varepsilon/2 + O(1/n) \le \rho(\mathcal{F}) + \varepsilon/2 + O(1/n) \le \rho(\mathcal{F}) + \varepsilon.$$

Since this holds for every $\boldsymbol{\nu}$, we conclude that this upper bounds val_Ψ as well, thus establishing the claim and hence the proposition. □

Now we prove Theorem 3.7, which asserts that $\rho(\mathcal{F})$ and thus $\rho_{\min}(\mathcal{F})$ is computable.

Proof of Theorem 3.7. By Proposition 3.6, we have

$$\rho_{\min}(\mathcal{F}) = \rho(\mathcal{F}) = \min_{\mathcal{D}_\mathcal{F}\in\Delta(\mathcal{F})}\left\{\max_{\mathcal{D}\in\Delta([q])}\left\{\mathop{\mathbb{E}}_{f\sim\mathcal{D}_\mathcal{F},\mathbf{a}\sim\mathcal{D}^k}[f(\mathbf{a})]\right\}\right\}.$$

Viewing $\mathcal{D}_\mathcal{F} \in \mathbb{R}^{|\mathcal{F}|}$ and $\mathcal{D} \in \mathbb{R}^q$ and noticing that the inner expectation is a degree $k+1$ polynomial in $\mathcal{D}_\mathcal{F}$ and $\mathcal{D}$, we get, again using Theorem 2.13, that there is a space $\mathrm{poly}(|\mathcal{F}|, q^k, \ell)$ algorithm answering the question "Is $\rho_{\min}(\mathcal{F}) \le \tau$?" □

Finally, we prove Theorem 3.8, which shows that the classification of approximation-resistant Max-CSP($\mathcal{F}$) problems is decidable.

Proof of Theorem 3.8. By Theorem 3.3, we have that Max-CSP($\mathcal{F}$) is approximation resistant if and only if $K^Y_{1-\varepsilon}(\mathcal{F}) \cap K^N_{\rho(\mathcal{F})+\varepsilon}(\mathcal{F}) \ne \emptyset$ for every small $\varepsilon > 0$. Taking limits as $\varepsilon \to 0$, this implies that Max-CSP($\mathcal{F}$) is approximation resistant if and only if $K^Y_1(\mathcal{F}) \cap K^N_{\rho(\mathcal{F})}(\mathcal{F}) \ne \emptyset$. If $K^Y_1(\mathcal{F}) \cap K^N_{\rho(\mathcal{F})}(\mathcal{F}) = \emptyset$, then by the property that these sets are closed (see Lemma 3.17), we have that there must exist $\varepsilon > 0$ such that $K^Y_{1-\varepsilon}(\mathcal{F}) \cap K^N_{\rho(f)+\varepsilon}(\mathcal{F}) = \emptyset$. In turn, this implies, again by Theorem 3.3, that the $(1-\varepsilon, \rho(\mathcal{F})+\varepsilon)$-approximation version of Max-CSP($\mathcal{F}$) can be solved by a streaming algorithm with $O(\log^3 n)$ space.

To get the decidability result, we combine the ingredients from the proof of Theorems 3.4 and 3.7. (We can't use them as blackboxes since $\rho_{\min}(\mathcal{F})$ may not be rational.) We create a quantified system of polynomial inequalities using a new variable called ρ and expressing the conditions $\rho = \rho(\mathcal{F})$ (with further variables for $\mathcal{D}_\mathcal{F}$ and $\mathcal{D}$ as in the proof of Theorem 3.7) and expressing the conditions $K^Y_1(\mathcal{F}) \cap K^N_\rho(\mathcal{F}) \ne \emptyset$ as in the proof of Theorem 3.4. The resulting expression is thus satisfiable if and only if $\mathcal{F}$ is approximation resistant, and this satisfiability can be decided in polynomial space in the input length $q^k|\mathcal{F}|$ by Theorem 2.13. □

3.5.3 Streaming Lower Bounds. We now prove Theorem 3.12 (assuming Theorem 3.10), which asserts that families that support one-wise independence are approximation resistant.

Proof of Theorem 3.12. Let $\mathcal{F}' \subseteq \mathcal{F}$ be a family satisfying $\rho(\mathcal{F}') = \rho(\mathcal{F})$ such that every function $f \in \mathcal{F}'$ supports one-wise independence. Let $\mathcal{D}_\mathcal{F} \in \Delta(\mathcal{F}')$ minimize $\max_{\mathcal{D}\in\Delta([q])}\left\{\mathbb{E}_{f\sim\mathcal{D}_\mathcal{F},\mathbf{a}\sim\mathcal{D}^k}[f(\mathbf{a})]\right\}$. For $f \in \mathcal{F}'$ let $\mathcal{D}_\mathcal{F} \in \Delta([q]^k)$ be the distribution with uniform marginals supported on $f^{-1}(1)$. Now let $\mathcal{D}_Y$ be the distribution where $(f,\mathbf{a}) \sim \mathcal{D}_Y$ is sampled by picking $f \in \mathcal{D}_\mathcal{F}$ (where $\mathcal{D}_\mathcal{F}$ is being viewed as an element of $\Delta(\mathcal{F})$) and then sampling $\mathbf{a} \sim \mathcal{D}_\mathcal{F}$. Now let $\mathcal{D}_N = \mathcal{D}_\mathcal{F} \times \mathrm{Unif}([q]^k)$. Note that $\mathcal{D}_Y$ and $\mathcal{D}_N$ are one-wise independent distributions with

$\boldsymbol{\mu}(\mathcal{D}_Y) = \boldsymbol{\mu}(\mathcal{D}_N)$. In particular, this implies that $(\mathcal{D}_Y, \mathcal{D}_N)$ are a padded one-wise pair. We claim that $\mathcal{D}_Y \in S_1^Y(\mathcal{F})$ and $\mathcal{D}_N \in S^N_{\rho(\mathcal{F})}(\mathcal{F})$. The theorem then follows immediately from Theorem 3.10.

To see the claim, first note that by definition we have that $(f, \mathbf{a}) \sim \mathcal{D}_Y$ satisfies $C(f, \mathbf{a})(\mathbb{I}) = f(\mathbf{a}) = f_0(\mathbf{a}) = 1$ with probability 1. Thus, we have $\mathbb{E}_{(f,\mathbf{a})\sim\mathcal{D}}[C(f, \mathbf{a})(\mathbb{I})] = 1$ and so $\mathcal{D}_Y \in S_1^Y(\mathcal{F})$. Now consider $(f, \mathbf{a}) \sim \mathcal{D}_N$. To show $\mathcal{D}_N \in S^N_{\rho(\mathcal{F})}(\mathcal{F})$ we need to show that for every family of distributions $(\mathcal{P}_\sigma \in \Delta([q]))_{\sigma\in[q]}$, the following holds: $\mathbb{E}_{(f,\mathbf{a})\sim\mathcal{D}}\left[\mathbb{E}_{\mathbf{b}, b_{i,\sigma}\sim\mathcal{P}_\sigma}[C(f, \mathbf{a})(\mathbf{b})]\right] \le \rho(\mathcal{F})$. Now let $\mathcal{P}$ be the distribution where $\tau \sim \mathcal{P}$ is sampled by picking $\sigma \sim \mathsf{Unif}([q])$ and then sampling $\tau \sim \mathcal{P}_\sigma$. We have

$$\begin{aligned}
\mathop{\mathbb{E}}_{(f,\mathbf{a})\sim\mathcal{D}}\left[\mathop{\mathbb{E}}_{\mathbf{b}, b_{i,\sigma}\sim\mathcal{P}_\sigma}[C(f, \mathbf{a})(\mathbf{b})]\right] &= \mathop{\mathbb{E}}_{f\sim\mathcal{D}_\mathcal{F}, \mathbf{a}\sim\mathsf{Unif}([q]^k)}\left[\mathop{\mathbb{E}}_{\mathbf{b}, b_{i,\sigma}\sim\mathcal{P}_\sigma}[C(f, \mathbf{a})(\mathbf{b})]\right] \\
&= \mathop{\mathbb{E}}_{f\sim\mathcal{D}_\mathcal{F}}\left[\mathop{\mathbb{E}}_{\mathbf{a}\sim\mathcal{P}^k}[f(\mathbf{a})]\right] \\
&\le \rho(\mathcal{F}') \\
&= \rho(\mathcal{F})\,.
\end{aligned}$$

This proves $\mathcal{D}_N \in S^N_{\rho(\mathcal{F})}(\mathcal{F})$ and thus proves the theorem. □

Next we turn to proving Theorem 3.14. To do so, we first prove the following simple proposition above distributions or pairs of Boolean variables.

Proposition 3.22. *If $\mathcal{D}_Y, \mathcal{D}_N \in \Delta(\mathcal{F} \times [2]^2)$ satisfy $\boldsymbol{\mu}(\mathcal{D}_Y) = \boldsymbol{\mu}(\mathcal{D}_N)$, then $(\mathcal{D}_Y, \mathcal{D}_N)$ form a padded one-wise pair.*

Proof. For $g \in \mathcal{F}$, let $P(g)$ denote the probability of sampling a constraint $(f, \mathbf{j}) \sim \mathcal{D}_Y$ with function $f = g$ and let P denote this distribution. Note that since $\boldsymbol{\mu}(\mathcal{D}_Y) = \boldsymbol{\mu}(\mathcal{D}_N)$, $\mathcal{D}_N$ also samples g with the same probability. Let $\mathcal{D}_{Y|g}$ denote $\mathcal{D}_Y$ conditioned on $f = g$. Similarly, let $\mathcal{D}_{N|g}$ denote $\mathcal{D}_N$ conditioned on $f = g$.

Now $\mathcal{D}_{Y|g}$ and $\mathcal{D}_{N|g}$ are distributions from $\Delta(\{g\} \times [2]^2)$ with matching marginals. We'll show that there exist $\mathcal{D}_{0|g}$, $\mathcal{D}'_{Y|g}$ and $\mathcal{D}'_{N|g}$, and τ_g such that (1) $\mathcal{D}_{Y|g} = \tau_g\mathcal{D}_{0|g} + (1 - \tau_g)\mathcal{D}'_{Y|g}$, (2) $\mathcal{D}_{N|g} = \tau_g\mathcal{D}_{0|g} + (1 - \tau_g)\mathcal{D}'_{N|g}$, and (3) $\mathcal{D}'_{Y|g}$ and $\mathcal{D}'_{N|g}$ are one-wise independent. Let $\mathcal{D}_{Y|g} = (p_{1,1}, p_{1,2}, p_{2,1}, p_{2,2})$, where $p_{i,j}$ denotes the probability $\Pr_{(a,b)\sim\mathcal{D}_{Y|g}}[a = i, b = j]$. If $\mathcal{D}_{N|g}$ has matching marginals with $\mathcal{D}_{Y|g}$, then there exists a $\delta_g \in [-1, 1]$ such that $\mathcal{D}_{N|g} = (p_{1,1}-\delta_g, p_{1,2}+\delta_g, p_{2,1}+\delta_g, p_{2,2}-\delta_g)$. Assume without loss of generality that $\delta_g \ge 0$. Let $\tau_g = 1-2\delta_g$, $\mathcal{D}_{0|g} = \frac{1}{1-2\delta_g}(p_{1,1} - \delta_g, p_{1,2}, p_{2,1}, p_{2,2} - \delta_g)$, $\mathcal{D}'_{Y|g} = (1/2, 0, 0, 1/2)$, and $\mathcal{D}'_{N|g} = (0, 1/2, 1/2, 0)$. It can be verified that $\mathcal{D}'_{Y|g}$ and $\mathcal{D}'_{N|g}$ are one-wise independent, $\mathcal{D}_{Y|g} = \tau_g\mathcal{D}_{0|g} + (1 - \tau_g)\mathcal{D}'_{Y|g}$ and $\mathcal{D}_{N|g} = \tau_g\mathcal{D}_{0|g} + (1 - \tau_g)\mathcal{D}'_{N|g}$.

Now let $\tau = \mathbb{E}_{f\sim P}[\tau_f]$, and $\mathcal{D}_0 \in \Delta(\mathcal{F}\times[q]^k)$ be the distribution where $\mathbf{a} = (f, \mathbf{b}) \in \{\mathcal{F}\}\times[2]^2$ is sampled with probability $\frac{P(f)\cdot\tau_f\cdot\mathcal{D}_{0|f}(\mathbf{a})}{\tau}$, where $\mathcal{D}_{0|f}(\mathbf{a})$ is the probability of sampling $\mathbf{a}$ from $\mathcal{D}_{0|f}$. Note that this is a valid probability distribution as

$$\sum_{f\in\mathcal{F}}\sum_{\mathbf{b}\in[2]^2}\frac{P(f)\cdot\tau_f\cdot\mathcal{D}_{0|f}(f, \mathbf{b})}{\tau} = \sum_{f\in\mathcal{F}}\frac{P(f)\cdot\tau_f}{\tau}\cdot\sum_{\mathbf{b}\in[2]^2}\mathcal{D}_{0|f}((f, \mathbf{b})) = 1\,.$$

Similarly, define $\mathcal{D}'_Y$ and $\mathcal{D}'_N$ such that $\mathbf{a}$ is sampled with probability $\frac{P(f)\cdot(1-\tau_f)\cdot\mathcal{D}'_{Y|f}(\mathbf{a})}{1-\tau}$ and probability $\frac{P(f)\cdot(1-\tau_f)\cdot\mathcal{D}'_{N|f}(\mathbf{a})}{1-\tau}$, respectively. It can be verified that these choices satisfy that (1) $\mathcal{D}_Y = \tau\mathcal{D}_0 + (1 - \tau)\mathcal{D}'_Y$, (2) $\mathcal{D}_N = \tau\mathcal{D}_0 + (1 - \tau)\mathcal{D}'_N$, and (3) $\mathcal{D}'_Y$ and $\mathcal{D}'_N$ are one-wise independent. It follows that $\mathcal{D}_Y$ and $\mathcal{D}_N$ form a padded one-wise pair. □

Combining Proposition 3.22 and Theorem 3.10, we immediately get the following theorem, which in turn implies Theorem 1.3.

Proof of Theorem 3.14. Part (1) is simply the specialization of Part (1) of Theorem 3.3 to the case $k = 2$. For Part (2), suppose $\boldsymbol{\mu} \in K^Y_\gamma \cap K^N_\beta$. Let $\mathcal{D}_Y \in S^Y_\gamma$ and $\mathcal{D}_N \in S^N_\beta$ be distributions such that $\boldsymbol{\mu}(\mathcal{D}_Y) = \boldsymbol{\mu}(\mathcal{D}_N) = \boldsymbol{\mu}$. Then, by Proposition 3.22, we have that $\mathcal{D}_Y$ and $\mathcal{D}_N$ form a padded one-wise pair, and so Theorem 3.10 can be applied to get Part (2). □

4 A STREAMING APPROXIMATION ALGORITHM FOR MAX-CSP($\mathcal{F}$)

In this section we give our main algorithmic result—an $O(\log^3 n)$-space linear sketching streaming algorithm for (γ, β)-Max-CSP($\mathcal{F}$) if $K^Y_\gamma = K^Y_\gamma(\mathcal{F})$ and $K^N_\beta = K^N_\beta(\mathcal{F})$ are disjoint. (See Definition 3.2.)

The algorithm in fact works in the (general) dynamic setting where the input instance $\Psi = (C_1, \ldots, C_m; w_1, \ldots, w_m)$ is obtained by inserting and deleting (unweighted) constraints, possibly with repetitions and thus leading to an (integer) weighted instance. Formally, the instance $\Psi = (C_1, \ldots, C_m; w_1, \ldots, w_m)$ is presented as a stream $\sigma_1, \ldots, \sigma_\ell$, where $\sigma_t = (C'_t, w'_t)$ and $w'_t \in \{-1, 1\}$ such that $w_i = \sum_{t \in [\ell]: C_i = C'_t} w'_t$. For the algorithmic result to hold, we require that w_is are non-negative at the end of the stream but the intermediate values can be arbitrary. Furthermore, the algorithm requires that the length of the stream be polynomial in n (or else there will be a logarithmic multiplicative factor in the length of the stream in the space usage).

We now state our main theorem of this section, which simply restates Part (1) of Theorem 3.3.

Theorem 4.1. *For every $q, k \in \mathbb{N}$, every family of functions $\mathcal{F} \subseteq \{f : [q]^k \to \{0, 1\}\}$, and for every $0 \le \beta < \gamma \le 1$, if $K^Y_\gamma(\mathcal{F}) \cap K^N_\beta(\mathcal{F}) = \emptyset$, then (γ, β)-Max-CSP($\mathcal{F}$) in the dynamic setting admits a probabilistic linear sketching streaming algorithm that uses $O(\log^3 n)$ space.*

We start with a brief overview of our algorithm. Roughly, given an instance Ψ on n variables with m constraints, our streaming algorithm (implicitly) works with an $n \times q$ bias non-negative matrix bias whose (i, σ)th entry tries to capture how much the ith variable would like to be assigned the value σ (according to our approximation heuristic). Note that any such matrix is too large for our algorithm, so the algorithm does not explicitly maintain this matrix. Our heuristic ensures that bias is updated linearly by every constraint and so the rich theory of norm approximations of matrices under linear updates can be brought into play to compute any desired norm of this matrix. Given the intuition that $\mathsf{bias}_{i,\sigma}$ represents the preference of variable i for value σ, a natural norm of interest to us is $\|\mathsf{bias}\|_{1,\infty} \triangleq \sum_{i=1}^n \{\max_{\sigma \in [q]} \{\mathsf{bias}_{i,\sigma}\}\}$. This norm, fortunately for us, is well known to be computable using $O(q \log^3 n)$ bits of space [3] (assuming bias is updated linearly), and we use this algorithm as a black box.

The question then turns to asking how bias should be defined. On input of a stream $\sigma_1, \ldots, \sigma_\ell$ representing an instance $\Psi = (C_1, \ldots, C_m)$ with $\sigma_i = (C'_i = (\mathbf{j}(i), \mathbf{b}(i)), w'_i)$, how should bias be updated? Presumably the ith update will only involve the rows $\mathbf{j}(i)_1, \ldots, \mathbf{j}(i)_k$, but how should these be updated and how should this update depend on the function f_i? Here is where the disjointness of K^Y and K^N comes into play. (We suppress $\mathcal{F}$ and γ and β in the notation of the sets S^Y_γ, S^N_β and K^Y_γ, K^N_β in this overview.) We show that these sets are convex and closed, and so there is a hyperplane (with margin) separating the two sets. Let $\boldsymbol{\lambda} = (\lambda_{f,i,\sigma})_{f \in \mathcal{F}, i \in [k], \sigma \in [q]}$ be the coefficients of this separating hyperplane, and let $\tau_N < \tau_Y$ be thresholds such that $\langle \boldsymbol{\lambda}, \boldsymbol{\mu} \rangle \ge \tau_Y$ for $\boldsymbol{\mu} \in K^Y$ and $\langle \boldsymbol{\lambda}, \boldsymbol{\mu} \rangle \le \tau_N$ for $\boldsymbol{\mu} \in K^N$. It turns out that the coefficients of $\boldsymbol{\lambda}$ give us exactly the right information to determine the update to the bias vector: Specifically, given an element σ_i of the stream with constraint $C'_i = (f_i, \mathbf{j}(i))$ and weight w'_i and $\ell \in [k]$ and $\sigma \in [q]$, we add $\lambda_{f_i,\ell,\sigma} \cdot w'_i$ to $\mathsf{bias}_{\mathbf{j}(i)_\ell, \sigma}$.

We are unable to provide intuition for why these updates work, but the proof that the algorithm works is nevertheless quite short!

We now turn to describing our algorithm. Recall by Lemma 3.17 that the set S^Y, S^N, K^Y, K^N is all convex and closed. This implies the existence of a separating hyperplane when K^Y and K^N do not intersect. We use a mild additional property to conclude that the coefficients of this hyperplane are non-negative, and we later use this crucially in the computation of the bias of the instance.

Proposition 4.2. *Let β, γ, and $\mathcal{F}$ be such that $0 \le \beta < \gamma \le 1$ and $K^Y_\gamma(\mathcal{F}) \cap K^N_\beta(\mathcal{F}) = \emptyset$. Then there exists a* non-negative *vector $\boldsymbol{\lambda} = (\lambda_{f,i,\sigma})_{f\in\mathcal{F}, i\in[k], \sigma\in[q]}$ and real numbers $\tau_Y > \tau_N$ such that*

$$\forall \boldsymbol{\mu} \in K^Y_\gamma(\mathcal{F}),\ \langle \boldsymbol{\lambda}, \boldsymbol{\mu}\rangle \ge \tau_Y \text{ and } \forall \boldsymbol{\mu} \in K^N_\beta(\mathcal{F}),\ \langle \boldsymbol{\lambda}, \boldsymbol{\mu}\rangle \le \tau_N\,.$$

Proof. The existence of a separating hyperplane follows from standard convexity (see, e.g., [25, Exercise 2.22]). For us this implies there exist $\boldsymbol{\lambda}' \in \mathbb{R}^{|\mathcal{F}|\times k q}$ and $\tau'_N < \tau'_Y$ such that

$$\forall \boldsymbol{\mu} \in K^Y_\gamma(\mathcal{F}),\ \langle \boldsymbol{\lambda}', \boldsymbol{\mu}\rangle \ge \tau'_Y \text{ and } \forall \boldsymbol{\mu} \in K^N_\beta(\mathcal{F}),\ \langle \boldsymbol{\lambda}', \boldsymbol{\mu}\rangle \le \tau'_N\,.$$

But $\boldsymbol{\lambda}'$ is not necessarily a positive vector. To remedy this, we use the fact that $K^Y_\gamma(\mathcal{F}) \cup K^N_\beta(\mathcal{F})$ is contained in a hyperplane whose coefficients are themselves positive. In particular, we note that for every $\mathcal{D} \in \Delta(\mathcal{F} \times [q]^k)$ we have $\langle \mu(\mathcal{D}), \mathbf{1}\rangle = k$, where $\mathbf{1} \in \mathbb{R}^{|\mathcal{F}|\times kq}$ is the all ones vector, as verified below:

$$\langle \mu(\mathcal{D}), \mathbf{1}\rangle = \sum_{f\in\mathcal{F}, i\in[k], \sigma\in[q]} \mu_{f,i,\sigma} = \sum_{i\in[k]} \left(\sum_{f\in\mathcal{F}, \sigma\in[q]} \mu_{f,i,\sigma} \right) = \sum_{i\in[k]} 1 = k.$$

Let $\lambda'_{\min} = \min_{f,t,\sigma} \lambda'_{f,t,\sigma}$. Now let $\boldsymbol{\lambda}$, τ_Y, and τ_N be given by

$$\lambda_{f,t,\sigma} = \lambda'_{f,t,\sigma} + |\lambda'_{\min}|\,,\ \tau_Y = \tau'_Y + k\cdot|\lambda'_{\min}| \text{ and } \tau_N = \tau'_N + k\cdot|\lambda'_{\min}|\,.$$

Observe that $\boldsymbol{\lambda}$ is a non-negative vector and $\tau_Y > \tau_N$. We also have

$$\forall \boldsymbol{\mu} \in K^Y_\gamma(\mathcal{F}),\ \langle \boldsymbol{\lambda}, \boldsymbol{\mu}\rangle = \langle \boldsymbol{\lambda}', \boldsymbol{\mu}\rangle + |\lambda'_{\min}| \ge \langle \mathbf{1}, \boldsymbol{\mu}\rangle \ge \tau'_Y + k|\lambda'_{\min}| = \tau_Y,$$

as desired. Similarly, we also get $\forall \boldsymbol{\mu} \in K^N_\beta(\mathcal{F}),\ \langle \boldsymbol{\lambda}, \boldsymbol{\mu}\rangle \le \tau_N$, concluding the proof. □

To use the vector $\boldsymbol{\lambda}$ given by Proposition 4.2, we introduce the notion of the bias matrix and the bias of a Max-CSP($\mathcal{F}$) instance Ψ.

Definition 4.3 (Bias (Matrix)). For a non-negative vector $\boldsymbol{\lambda} = (\lambda_{f,i,\sigma})_{f\in\mathcal{F}, i\in[k], \sigma\in[q]} \in \mathbb{R}^{|\mathcal{F}|kq}$ and instance $\Psi = (C_1, \ldots, C_m; w_1, \ldots, w_m)$ of Max-CSP($\mathcal{F}$) where $C_i = (f_i, \mathbf{j}(i))$, where $f_i \in \mathcal{F}$ and $\mathbf{j}(i) \in [n]^k$, we let the *$\boldsymbol{\lambda}$-bias matrix* of Ψ, denoted $\mathrm{bias}_{\boldsymbol{\lambda}}(\Psi)$, be the matrix in $\mathbb{R}^{n\times q}$ given by

$$\mathrm{bias}_{\boldsymbol{\lambda}}(\Psi)_{\ell,\sigma} = \frac{1}{W}\cdot \sum_{i\in[m], t\in[k]:\mathbf{j}(i)_t=\ell} \lambda_{f_i,t,\sigma}\cdot w_i\,,$$

for $\ell \in [n]$ and $\sigma \in [q]$, where $W = \sum_{i\in[m]} w_i$. The $\boldsymbol{\lambda}$-bias of Ψ, denoted $B_{\boldsymbol{\lambda}}(\Psi)$, is defined as $B_{\boldsymbol{\lambda}}(\Psi) = \sum_{\ell=1}^n \max_{\sigma\in[q]} \mathrm{bias}_{\boldsymbol{\lambda}}(\Psi)_{\ell,\sigma}$.

Key to our algorithm for approximating Max-CSP($\mathcal{F}$) is the following algorithm to compute the $\ell_{1,\infty}$ norm of a matrix. Recall that for a matrix $M \in \mathbb{R}^{a\times b}$ the $\ell_{1,\infty}$ norm is the quantity $\|M\|_{1,\infty} = \sum_{i\in[a]}\{\max_{j\in[b]}\{|M_{ij}|\}\}$.

Theorem 4.4 (Implied by [3, Theorem 4.5]). *There exists a constant $c > 0$ such that the $\ell_{1,\infty}$ norm of an $n \times q$ matrix M can be estimated by a linear sketch to within a multiplicative error of $(1+\varepsilon)$ in the turnstile streaming model with $O(\varepsilon^{-c}\cdot q\cdot \log^2 n)$ words (or with $O(\varepsilon^{-c}\cdot q\cdot \log^3 n)$ bits).*

We note that Theorem 4.5 in [3] is much more general. Theorem 4.4 is the special case corresponding to $X = \ell_\infty$ and E_X being simply the identity function. $\alpha(\cdots)$ in this case turns out to be $O(\log n)$, leading to the bounds above [2].

Note that there is a slight distinction between the definitions of $B_{\boldsymbol{\lambda}}(\Psi)$ and $\|\mathrm{bias}_{\boldsymbol{\lambda}}(\Psi)\|_{1,\infty}$, but these quantities are equal since $\mathrm{bias}_{\boldsymbol{\lambda}}$ is a non-negative matrix (which in turn follows from the fact that $\boldsymbol{\lambda}$ is non-negative). We thus get the following corollary.

Corollary 4.5. *There exists a constant c such that for every $k, q, \mathcal{F}$, and $\varepsilon > 0$, there exists a linear sketching streaming algorithm running in space $O(\varepsilon^{-c} \cdot \log^3 n)$ that on input of a stream $\sigma_1, \ldots, \sigma_\ell$ representing a Max-CSP($\mathcal{F}$) instance $\Psi = (C_1, \ldots, C_m; w_1, \ldots, w_m)$ on n variables outputs a $(1 \pm \varepsilon)$ approximation to $B_{\boldsymbol{\lambda}}(\Psi)$.*

We are now ready to describe our algorithm for (γ, β)-Max-CSP($\mathcal{F}$).

ALGORITHM 1: A Streaming Algorithm for (γ, β)-Max-CSP($\mathcal{F}$)

Input: A stream $\sigma_1, \ldots, \sigma_\ell$ representing an instance Ψ of Max-CSP($\mathcal{F}$).

1: Let $\boldsymbol{\lambda} \in \mathbb{R}^{|\mathcal{F}|kq}$, τ_N, and τ_Y be as given by Proposition 4.2 separating $K^Y_\gamma(f)$ and $K^N_\beta(f)$, so $\boldsymbol{\lambda}$ is non-negative and $\tau_N < \tau_Y$.

2: Let $\varepsilon = \frac{\tau_Y - \tau_N}{2(\tau_Y + \tau_N)}$.

3: Using Corollary 4.5, compute a $(1 \pm \varepsilon)$ approximation $\tilde{B}$ to $B_{\boldsymbol{\lambda}}(\Psi)$, i.e.,

$$(1 - \varepsilon)B_{\boldsymbol{\lambda}}(\Psi) \le \tilde{B} \le (1 + \varepsilon)B_{\boldsymbol{\lambda}}(\Psi) \text{ with probability at least } 2/3.$$

4: **if** $\tilde{B} \le \tau_N(1 + \varepsilon)$ **then**
Output: NO.

5: **else**
Output: YES.

Given Corollary 4.5, it follows that the algorithm above uses space $O(\log^3 n)$ on instances on n variables. In what follows we prove that the algorithm correctly solves $(\gamma, \beta) - \mathrm{Max\text{-}CSP}(\mathcal{F})$.

4.1 Analysis of the Correctness of Algorithm 1

Lemma 4.6. *Algorithm 1 correctly solves (γ, β)-Max-CSP($\mathcal{F}$) if $K^Y_\gamma(\mathcal{F})$ and $K^N_\beta(\mathcal{F})$ are disjoint. Specifically, for every Ψ, let $\tau_Y, \tau_N, \varepsilon, \boldsymbol{\lambda}, \tilde{B}$ be as given in Algorithm 1, and we have*

$$\begin{aligned} val_\Psi \ge \gamma \quad &\Rightarrow \quad B_{\boldsymbol{\lambda}}(\Psi) \ge \tau_Y \text{ and } \tilde{B} > \tau_N(1 + \varepsilon), \\ \text{and } val_\Psi \le \beta \quad &\Rightarrow \quad B_{\boldsymbol{\lambda}}(\Psi) \le \tau_N \text{ and } \tilde{B} \le \tau_N(1 + \varepsilon), \end{aligned}$$

provided $(1 - \varepsilon)B_\lambda(\Psi) \le \tilde{B} \le (1 + \varepsilon)B_\lambda(\Psi)$.

In the rest of this section, we will prove Lemma 4.6. The key to our analysis is a distribution $\mathcal{D}(\Psi^{\mathbf{b}}) \in \Delta(\mathcal{F} \times [q]^k)$ that we associate with every instance Ψ and assignment $\mathbf{b} \in [q]^n$ to the variables of Ψ. If Ψ is γ-satisfied by assignment $\mathbf{b}$, we prove that $\boldsymbol{\mu}(\mathcal{D}(\Psi^{\mathbf{b}})) \in K^Y_\gamma(\mathcal{F})$. On the other hand, if Ψ is not β-satisfiable by any assignment, we prove that for every $\mathbf{b}$, $\boldsymbol{\mu}(\mathcal{D}(\Psi^{\mathbf{b}})) \in K^N_\beta(\mathcal{F})$. Finally, we also show that the bias $B_{\boldsymbol{\lambda}}(\Psi)$ relates to $\boldsymbol{\lambda}(\mathcal{D}(\Psi^{\mathbf{b}})) \triangleq \langle \boldsymbol{\mu}(\mathcal{D}(\Psi^{\mathbf{b}})), \boldsymbol{\lambda} \rangle$, where the latter quantity is exactly what needs to be computed (by Proposition 4.2) to distinguish the membership of $\boldsymbol{\mu}(\mathcal{D}(\Psi^{\mathbf{b}}))$ in $K^Y_\gamma(\mathcal{F})$ versus the membership in $K^N_\beta(\mathcal{F})$.

The key step is the definition of these distributions that allows the remaining steps (esp. Lemma 4.9) to be extended, which we present now.

Given an instance $\Psi = (C_1, \ldots, C_m; w_1, \ldots, w_m)$ on n variables with $C_i = (f_i, \mathbf{j}(i))$ and an assignment $\mathbf{b} \in [q]^n$, the distribution $\mathcal{D}(\Psi^{\mathbf{b}}) \in \Delta(\mathcal{F} \times [q]^k)$ is sampled as follows: Sample $i \in [m]$ with probability w_i/W, where $W = \sum_{i\in[m]} w_i$, and output $(f_i, \mathbf{b}\,|_{\mathbf{j}(i)})$.

We start by relating the bias $B_{\boldsymbol{\lambda}(\Psi)}$ to $\mathcal{D}(\Psi)$.

Lemma 4.7. *For every vector $\mathbf{b} \in [q]^n$, we have $\boldsymbol{\lambda}(\mathcal{D}(\Psi^{\mathbf{b}})) = \sum_{\ell=1}^{n} bias_{\boldsymbol{\lambda}}(\Psi)_{\ell,b_\ell}$. Consequently, we have $B_{\boldsymbol{\lambda}}(\Psi) = \sum_{\ell=1}^{n} \max_{\sigma\in[q]} bias_{\boldsymbol{\lambda}}(\Psi)_{\ell,\sigma} = \max_{\mathbf{b}\in[q]^n}\{\boldsymbol{\lambda}(\mathcal{D}(\Psi^{\mathbf{b}}))\}$.*

Proof. We start with the first equality. Fix $b \in [q]^n$. Given $f \in \mathcal{F}$, $t \in [k]$, and $\sigma \in [q]$, we have $\mu(\mathcal{D}(\Psi^{\mathbf{b}}))_{f,t,\sigma} = \frac{1}{W}\sum_{i=1}^{m} w_i \cdot \mathbb{1}[f_i = f, b_{j(i)_t} = \sigma]$. Hence,

$$\begin{aligned}
\boldsymbol{\lambda}(\mathcal{D}(\Psi^{\mathbf{b}})) &= \sum_{f\in\mathcal{F}, t\in[k], \sigma\in[q]} \mu(\mathcal{D}(\Psi^{\mathbf{b}}))_{f,t,\sigma} \cdot \lambda_{f,t,\sigma} \\
&= \frac{1}{W} \sum_{f\in\mathcal{F}, t\in[k], \sigma\in[q]} \sum_{i\in[m]} w_i \cdot \mathbb{1}[f_i = f, b_{j(i)_t} = \sigma] \cdot \lambda_{f,t,\sigma} \\
&= \frac{1}{W} \sum_{i\in[m], t\in[k], \sigma\in[q]: b_{j(i)_t}=\sigma} w_i \cdot \lambda_{f_i,t,\sigma} \\
&= \sum_{\ell=1}^{n} \frac{1}{W} \sum_{i\in[m], t\in[k]: j(i)_t = l} w_i \cdot \lambda_{f_i,t,b_l} \\
&= \sum_{\ell=1}^{n} \mathrm{bias}_{\boldsymbol{\lambda}}(\Psi)_{\ell,b_\ell}.
\end{aligned}$$

For the final equality, observe that

$$B_{\boldsymbol{\lambda}}(\Psi) = \sum_{\ell=1}^{n} \max_{\sigma\in[q]} \mathrm{bias}_{\boldsymbol{\lambda}}(\Psi)_{\ell,\sigma} = \max_{\mathbf{b}\in[q]^n} \sum_{\ell=1}^{n} \mathrm{bias}_{\boldsymbol{\lambda}}(\Psi)_{\ell,b_\ell} = \max_{\mathbf{b}\in[q]^n}\{\boldsymbol{\lambda}(\mathcal{D}(\Psi^{\mathbf{b}}))\}. \qquad \square$$

The following lemmas relate val_Ψ to the properties of $\mathcal{D}(\Psi^{\mathbf{a}})$.

Lemma 4.8. *For every $\Psi \in$ Max-CSP($\mathcal{F}$) and $\mathbf{b} \in [q]^n$, if $val_\Psi(\mathbf{b}) \geq \gamma$, then $\mathcal{D}(\Psi^{\mathbf{b}}) \in S^Y_\gamma(\mathcal{F})$.*

Proof. Follows from the fact that

$$\mathop{\mathbb{E}}_{(f,\mathbf{a})\sim\mathcal{D}(\Psi^{\mathbf{b}})} [C(f,a)(\mathbb{I})] = \frac{1}{W}\sum_{i=1}^{m} w_i \cdot f_i(b\,|_{j(i)}) = \mathrm{val}_\Psi(\mathbf{b}) \geq \gamma,$$

implying $\mathcal{D}(\Psi^{\mathbf{b}}) \in S^Y_\gamma(\mathcal{F})$. $\square$

Lemma 4.9. *For every $\Psi \in$ Max-CSP($\mathcal{F}$), if $val_\Psi \leq \beta$, then for all $\mathbf{b} \in [q]^n$, we have $\mathcal{D}(\Psi^{\mathbf{b}}) \in S^N_\beta(\mathcal{F})$.*

Proof. We prove the contrapositive. We assume that $\exists \mathbf{b} \in [q]^n$ such that $\mathcal{D}(\Psi^{\mathbf{b}}) \notin S^N_\beta(\mathcal{F})$ and show that this implies $\mathrm{val}_\Psi > \beta$. Then there exists $(P_\sigma \in \Delta([q]))_{\sigma\in[q]}$ satisfying the following inequality: $\mathbb{E}_{(f,a)\sim\mathcal{D}(\Psi^{\mathbf{b}})}\left[\mathbb{E}_{\mathbf{c}, c_{i,\sigma}\sim\mathcal{P}_\sigma}[C(f,\mathbf{a})(\mathbf{c})]\right] > \beta$.

We thus have

$$\begin{aligned}
\beta &< \mathop{\mathbb{E}}_{(f,a)\sim\mathcal{D}(\Psi^{\mathbf{b}})} \left[\mathop{\mathbb{E}}_{\mathbf{c}, c_{i,\sigma}\sim\mathcal{P}_\sigma} [C(f,\mathbf{a})(\mathbf{c})]\right] \\
&= \mathop{\mathbb{E}}_{\mathbf{c}, c_{i,\sigma}\sim\mathcal{P}_\sigma} \left[\mathop{\mathbb{E}}_{(f,a)\sim\mathcal{D}(\Psi^{\mathbf{b}})} [C(f,\mathbf{a})(\mathbf{c})]\right]
\end{aligned}$$

$$
\begin{aligned}
&= \mathop{\mathbb{E}}_{\mathbf{c}, c_{i,\sigma}\sim\mathcal{P}_\sigma}\left[\frac{1}{W}\sum_{i=1}^{m} w_i \cdot f_i((c_{t,b_{j(i)_t}})_{t\in[k]})\right] \\
&= \frac{1}{W}\sum_{i=1}^{m} w_i \cdot \mathop{\mathbb{E}}_{\mathbf{c}, c_{i,\sigma}\sim\mathcal{P}_\sigma}\left[f_i((c_{t,b_{j(i)_t}})_{t\in[k]})\right] \\
&= \frac{1}{W}\sum_{i=1}^{m} w_i \cdot \mathop{\mathbb{E}}_{\mathbf{x}, x_\ell\sim\mathcal{P}_{b_\ell}}\left[f_i((x_{j(i)_t})_{t\in[k]})\right] \\
&= \mathop{\mathbb{E}}_{\mathbf{x}, x_\ell\sim\mathcal{P}_{b_\ell}}\left[\frac{1}{W}\sum_{i=1}^{m} w_i \cdot f_i((x_{j(i)_t})_{t\in[k]})\right] \\
&= \mathop{\mathbb{E}}_{\mathbf{x}, x_\ell\sim\mathcal{P}_{b_\ell}}[\mathrm{val}_\Psi(\mathbf{x})] \\
&\le \max_{\mathbf{x}\in[q]^n} \mathrm{val}_\Psi(\mathbf{x}) \\
&= \mathrm{val}_\Psi\,,
\end{aligned}
$$

which contradicts the assumption that $\mathrm{val}_\Psi \le \beta$. This concludes the proof of the claim and hence the lemma. □

The key step above is the one asserting $\frac{1}{W}\sum_{i=1}^{m} w_i \cdot \mathbb{E}_{\mathbf{c}, c_{i,\sigma}\sim\mathcal{P}_\sigma}\left[f_i((c_{t,b_{j(i)_t}})_{t\in[k]})\right] = \frac{1}{W}\sum_{i=1}^{m} w_i \cdot \mathbb{E}_{\mathbf{x}, x_\ell\sim\mathcal{P}_{b_\ell}}\left[f_i((x_{j(i)_t})_{t\in[k]})\right]$, which relies crucially on column symmetry of the distributions used in the definition of $S^N_\beta(\mathcal{F})$ in Definition 3.1. Without this restriction, or even more stringent ones, this step of the rounding would fail. And the reason we can't use a more stringent restriction will become clear in the proof of Theorem 3.10 (and is specifically used in the proof of Lemma 5.8). We also note that this key equality relies on the assumption that the variables in a single constraint are *distinct*. In particular, the left-hand side assumes $c_{i,\sigma}$s are drawn independently, whereas the right side allows this only for the distinct variables x_ℓ in a constraint.

5 SKETCHING AND STREAMING SPACE LOWER BOUNDS FOR MAX-CSP($\mathcal{F}$)

In this section, we prove our two lower bound results, modulo a communication complexity lower bound, which is proved in Sections 6 to 8. We start by restating the results to be proved. Recall (from Definition 3.9) the notion of a padded one-wise pair of distributions: $(\mathcal{D}_1, \mathcal{D}_2)$ is a padded one-wise pair if there exist $\mathcal{D}_0, \mathcal{D}'_1, \mathcal{D}'_2$ and $\tau \in [0,1]$ such that for every $i \in \{1,2\}$, $\mathcal{D}'_i$ is one-wise independent, and $\mathcal{D}_i = \tau\mathcal{D}_0 + (1-\tau)\mathcal{D}'_i$.

The first theorem we prove is the lower bound in the streaming setting for padded one-wise pairs of distributions. We restate the theorem below for convenience.

Theorem 3.10 (Streaming Lower Bound). *For every $q, k \in \mathbb{N}$, every family of functions $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$ and for every $0 \le \beta < \gamma \le 1$, if there exists a padded one-wise pair of distributions $\mathcal{D}_Y \in S^Y_\gamma(\mathcal{F})$ and $\mathcal{D}_N \in S^N_\beta(\mathcal{F})$ then, for every $\varepsilon > 0$, every non-uniform randomized streaming algorithm that solves the $(\gamma-\varepsilon, \beta+\varepsilon)$-Max-CSP($\mathcal{F}$) problem requires $\Omega(\sqrt{n})$ space. Furthermore, if $\gamma = 1$, then $(1, \beta+\varepsilon)$-Max-CSP(f) requires $\Omega(\sqrt{n})$ space.*

We also restate the lower bound against sketching algorithms from Theorem 3.3 as a separate theorem below.

Theorem 5.1 (Lower Bounds against Sketching Algorithms). *For every $q, k \in \mathbb{N}$, every family of functions $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$ and for every $0 \le \beta < \gamma \le 1$, if $K^Y_\gamma(\mathcal{F}) \cap K^N_\beta(\mathcal{F}) \neq \emptyset$,*

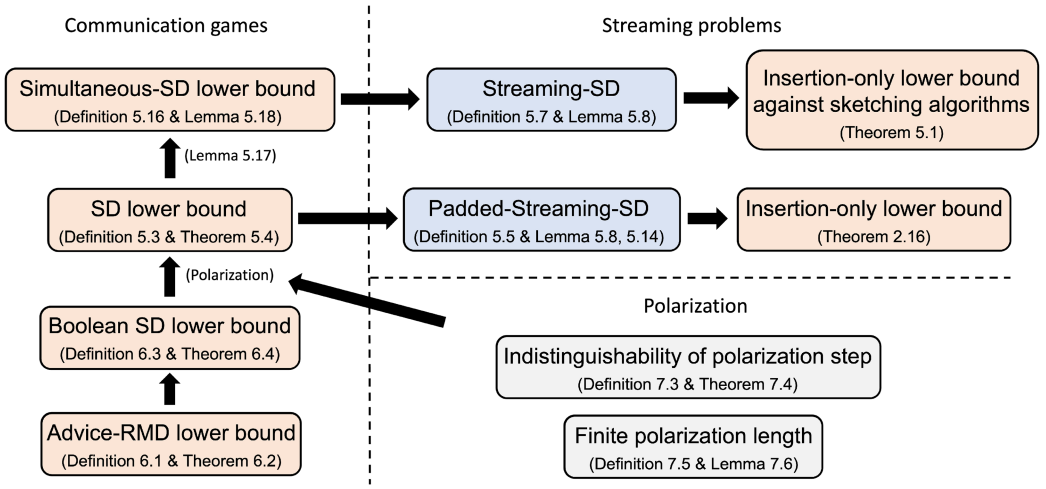


Fig. 2. The roadmap of our lower bounds. The top two rows describe the results of this section, while the remaining rows describe notions and results from Sections 6 to 8.

then for every $\varepsilon > 0$, any sketching algorithm for the $(\gamma - \varepsilon, \beta + \varepsilon)$-Max-CSP($\mathcal{F}$) problem requires $\Omega(\sqrt{n})$ space. Furthermore, if $\gamma = 1$, then any sketching algorithm for $(1, \beta + \varepsilon)$-Max-CSP($\mathcal{F}$) requires $\Omega(\sqrt{n})$ space.

To prove both theorems, we introduce a new communication game we call the *SD* in Section 5.1. In Theorem 5.4 we state a lower bound on the communication complexity of this problem. This lower bound is established in Sections 6 to 8. We then use this lower bound to prove Theorem 3.10 in Section 5.2 and to prove Theorem 5.1 in Section 5.3 (See Figure 2).

5.1 The Signal Detection Problem and Results

In this section we introduce our communication game and state the lower bound for this game. We start with the definition of a general one-way communication game.

Definition 5.2 (One-way Communication Game). Given two distributions $\mathcal{Y}$ and $\mathcal{N}$, an instance of the two-player one-way communication game is a pair (X, Y) drawn either from $\mathcal{Y}$ or from $\mathcal{N}$. Two computationally unbounded parties, Alice and Bob, receive X and Y, respectively. A protocol $\Pi = (\Pi_A, \Pi_B)$ is a pair of functions with $\Pi_A(X) \in \{0,1\}^c$ denoting Alice's message to Bob, and $\Pi_B(\Pi_A(X), Y) \in \{\mathbf{YES}, \mathbf{NO}\}$ denoting the protocol's output. We denote this output by $\Pi(X, Y)$. The complexity of this protocol is the parameter c specifying the maximum length of Alice's message $\Pi_A(X)$. The advantage of the protocol Π is the quantity

$$\left| \Pr_{(X,Y)\sim\mathcal{Y}}[\Pi(X,Y) = \mathbf{YES}] - \Pr_{(X,Y)\sim\mathcal{N}}[\Pi(X,Y) = \mathbf{YES}] \right| .$$

We now define the specific game we are interested in.

Definition 5.3 (Signal Detection (SD) Problem). Let $n, k, q \in \mathbb{N}, \alpha \in (0,1)$, where k, q, and α are constants with respect to n, and αn is an integer less than n/k. Let $\mathcal{F}$ be a finite set. For a pair $\mathcal{D}_Y$ and $\mathcal{D}_N$ of distributions over $\mathcal{F} \times [q]^k$, we consider the following two-player one-way communication problem $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD.

— The generator samples the following objects:
(1) $\mathbf{x}^* \sim \mathsf{Unif}([q]^n)$.
(2) $M \in \{0,1\}^{k\alpha n \times n}$ is chosen uniformly among all matrices with exactly one 1 in each row and at most one 1 in each column. We let $M = (M_1, \ldots, M_{\alpha n})$, where $M_i \in \{0,1\}^{k \times n}$ is the ith block of rows of M, where each block has exactly k rows.
(3) $\mathbf{b} = (\mathbf{b}(1), \ldots, \mathbf{b}(\alpha n))$ is sampled from one of the following distributions:
   – (**YES**) each $\mathbf{b}(i) = (f_i, \widetilde{\mathbf{b}}(i)) \in \mathcal{F} \times [q]^k$ is sampled according to $\mathcal{D}_Y$.
   – (**NO**) each $\mathbf{b}(i) = (f_i, \widetilde{\mathbf{b}}(i)) \in \mathcal{F} \times [q]^k$ is sampled according to $\mathcal{D}_N$.
(4) $\mathbf{z} = (\mathbf{z}(1), \ldots, \mathbf{z}(\alpha n))$ is determined from M, $\mathbf{x}^*$ and $\mathbf{b} = (\mathbf{b}(1), \ldots, \mathbf{b}(\alpha n))$ as follows. Recall that $\mathbf{b}(i) = (f_i, \widetilde{\mathbf{b}}(i))$. We let $\mathbf{z}(i) = (f_i, \widetilde{z}_i) \in \mathcal{F} \times \{0,1\}$, where $\widetilde{z}_i = 1$ iff $M_i \mathbf{x}^* = \widetilde{\mathbf{b}}(i)$.
— Alice receives $\mathbf{x}^*$ as input.
— Bob receives M and $\mathbf{z}$ as input.

In the special case when the set $\mathcal{F}$ contains just one element, $|\mathcal{F}| = 1$, we call the corresponding communication problem $(\mathcal{D}_Y, \mathcal{D}_N)$-SD.

We note that our communication game is slightly different from those in previous works: Specifically, the problem studied in [43, 56] is called the *BHM* problem from [43] and the works [57, 58] study a variant called the *Implicit Hidden Partition* problem. While these problems are similar, they are less expressive than our formulation and specifically do not seem to capture all Max-CSP(f) problems.

There are two main differences between the previous settings and our setting. The first difference is the way to encode the matching matrix M. In all the previous works, each edge (or hyperedge) is encoded by a single row in M where the corresponding columns are assigned to 1, so that $m = \alpha n$. However, it turns out that this encoding hides too much information and hence we do not know how to reduce the problem to general Max-CSP. We unfold the encoding by using k rows to encode a single k-hyperedge (leading to the setting of $m = k\alpha n$ in our case). The second difference is that we allow the masking vector $\mathbf{b}$ to be sampled from a more general distribution. This is also for the purpose of establishing a reduction to general Max-CSP. Due to the above two differences, it is not clear how to derive communication lower bounds for general $\mathcal{D}_Y$ and $\mathcal{D}_N$ by reduction from the previous works.

Theorem 5.4 (Communication Lower Bound for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD). *For every k, q, every finite set $\mathcal{F}$, every pair of distributions $\mathcal{D}_Y, \mathcal{D}_N \in \Delta(\mathcal{F} \times [q]^k)$ with $\boldsymbol{\mu}(\mathcal{D}_Y) = \boldsymbol{\mu}(\mathcal{D}_N)$ there exists $\alpha_0 > 0$ such that for every $0 < \alpha \le \alpha_0$ and $\delta > 0$ there exists $\tau > 0$ such that the following holds: Every protocol for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD achieving advantage δ on instances of length n requires $\tau\sqrt{n}$ bits of communication.*

Sections 6 to 8 are devoted to proving Theorem 5.4. The specific proof can be found in Section 7.3. In the rest of this section we use this theorem to prove Theorems 5.1 and 3.10.

5.2 The Streaming Lower Bound

The hardness of SD suggests a natural path for hardness of Max-CSP($\mathcal{F}$) problems in the streaming setting. Such a reduction would take two distributions $\mathcal{D}_Y \in S^Y_\gamma$ and $\mathcal{D}_N \in S^N_\beta$ with matching marginals, construct distributions $\mathcal{Y}$ and $\mathcal{N}$ of RMD, and then interpret these distributions (in a natural way) as distributions over instances of Max-CSP(f) that are indistinguishable to small-space algorithms. While the exact details of this "interpretation" need to be spelled out, every step in this path can be achieved. Unfortunately, this does not mean any hardness for Max-CSP(f) since the CSPs generated by this reduction would consist of instances that have at most one constraint per variable, and such instances are easy to solve!

To go from the instance suggested by the SD problem to hard CSP instances, we instead pick T samples (somewhat) independently from the distributions $\mathcal{Y}$ and $\mathcal{N}$ suggested by the SD problem and concatenate these. With an appropriate implementation of this notion (see Definition 5.5), it turns out it is possible to use the membership of the underlying distributions in S^Y_γ and S^N_β to argue that the resulting instances Ψ do (almost always) have $\mathrm{val}_\Psi \geq \gamma$ or $\mathrm{val}_\Psi \leq \beta$. (We prove this after appropriate definitions in Lemma 5.8.) But now one needs to connect the streaming problem generated from the T-fold sampled version to the SD problem.

To this end we formalize the T-fold streaming problem, which we call the $(\mathcal{D}_Y, \mathcal{D}_N, T)$-streaming-SD problem, in Definition 5.5. Unfortunately, we are not able to reduce the $(\mathcal{D}_Y, \mathcal{D}_N)$-SD problem to the $(\mathcal{D}_Y, \mathcal{D}_N, T)$-streaming-SD problem for all $\mathcal{D}_Y$ and $\mathcal{D}_N$.[9] But in the setting where $\mathcal{D}_Y$ and $\mathcal{D}_N$ have uniform marginals, we are able to effect the reduction and thus show that the streaming problem requires large space. This is a special case of Lemmas 5.12 and 5.14, which we discuss next.

We are able to extend our reduction from SD to streaming-SD slightly beyond the uniform marginal case, to the case where $\mathcal{D}_Y$ and $\mathcal{D}_N$ form a padded one-wise pair, but both the streaming problem and the analysis of the resulting CSP value need to be altered to deal with this case, as elaborated next. Let $\tau \in [0,1]$ and $\mathcal{D}_0, \mathcal{D}'_Y, \mathcal{D}'_N$ be such that for $i \in \{Y, N\}$ we have $\mathcal{D}_i = \tau\mathcal{D}_0 + (1-\tau)\mathcal{D}'_i$ and $\mathcal{D}'_i$ has uniform marginals. Our padded streaming problem, denoted the $(\mathcal{D}'_Y, \mathcal{D}'_N, T, \mathcal{D}_0, \tau)$-padded-streaming-SD problem, includes an appropriately large number of constraints generated according to $\mathcal{D}_0$, followed by T samples chosen according to the $(\mathcal{D}'_Y, \mathcal{D}'_N, T)$-streaming-SD problem. See Definition 5.5 for a formal definition. In Lemma 5.8 we show that the CSP value of the resulting streaming problem inherits the properties of $\mathcal{D}_Y$ and $\mathcal{D}_N$ (which is not as immediate for padded-streaming-SD as for streaming-SD). We then show effectively that $(\mathcal{D}'_Y, \mathcal{D}'_N)$-SD reduces to $(\mathcal{D}'_Y, \mathcal{D}'_N, T, \mathcal{D}_0, \tau)$-padded-streaming-SD. See Lemmas 5.12 and 5.14. Putting these together leads to a proof of Theorem 3.10.

5.2.1 *The (Padded) Streaming SD Problem.*

Definition 5.5 ($(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T)$-streaming-SD). For $k, q, T \in \mathbb{N}$, $\alpha \in (0, 1/k]$, a finite set $\mathcal{F}$ and distributions $\mathcal{D}_Y, \mathcal{D}_N$ over $\mathcal{F} \times [q]^k$, the streaming problem $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T; \alpha, k, q)$-streaming-SD is the task of distinguishing, for every n, $\boldsymbol{\sigma} \sim \mathcal{Y}_{\mathrm{stream},n}$ from $\boldsymbol{\sigma} \sim \mathcal{N}_{\mathrm{stream},n}$ where for a given length parameter n, the distributions $\mathcal{Y}_{\mathrm{stream}} = \mathcal{Y}_{\mathrm{stream},n}$ and $\mathcal{N}_{\mathrm{stream}} = \mathcal{N}_{\mathrm{stream},n}$ are defined as follows:

- Let $\mathcal{Y}$ be the distribution over instances of length n, i.e., triples $(\mathbf{x}^*, M, \mathbf{z})$, from the definition of $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD. For $\mathbf{x} \in [q]^n$, let $\mathcal{Y}|_{\mathbf{x}}$ denote the distribution $\mathcal{Y}$ conditioned on $\mathbf{x}^* = \mathbf{x}$. The stream $\boldsymbol{\sigma} \sim \mathcal{Y}_{\mathrm{stream}}$ is sampled as follows: Sample $\mathbf{x}^*$ uniformly from $[q]^n$. Let $(M^{(1)}, \mathbf{z}^{(1)}), \ldots, (M^{(T)}, \mathbf{z}^{(T)})$ be sampled independently according to $\mathcal{Y}|_{\mathbf{x}^*}$. Let $\boldsymbol{\sigma}^{(t)}$ be the pair $(M^{(t)}, \mathbf{z}^{(t)})$ presented as a stream of edges with labels in $\mathcal{F} \times \{0,1\}$, i.e., $\mathbf{z}^{(t)} = (f_i, \tilde{z}_i)$. Specifically, for $t \in [T]$ and $i \in [\alpha n]$, let $\boldsymbol{\sigma}^{(t)}(i) = (e^{(t)}(i), \mathbf{z}^{(t)}(i))$, where $e^{(t)}(i)$ is the ith hyperedge of $M^{(t)}$, i.e., $e^{(t)}(i) = (j^{(t)}(k(i-1)+1), \ldots, j^{(t)}(k(i-1)+k)$, and $j^{(t)}(\ell)$ is the unique index j such that $M^{(t)}_{j,\ell} = 1$. Finally, we let $\boldsymbol{\sigma} = \boldsymbol{\sigma}^{(1)} \circ \cdots \circ \boldsymbol{\sigma}^{(T)}$ be the concatenation of the $\boldsymbol{\sigma}^{(t)}$s.
- $\boldsymbol{\sigma} \sim \mathcal{N}_{\mathrm{stream}}$ is sampled similarly except we now sample $(M^{(1)}, \mathbf{z}^{(1)}), \ldots, (M^{(T)}, \mathbf{z}^{(T)})$ independently according to $\mathcal{N}|_{\mathbf{x}^*}$, where $\mathcal{N}|_{\mathbf{x}}$ is the distribution $\mathcal{N}$ condition on $\mathbf{x}^* = \mathbf{x}$.

Again, when α, k, q are clear from context, we suppress them and simply refer to the $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T)$-streaming-SD problem.

[9] Roughly, this problem arises from the fact that the T samples $(\mathbf{x}^*(t), M(t), \mathbf{z}(t))$ are not sampled independently from $\mathcal{Y}$ (or $\mathcal{N}$ for $t \in [T]$). Instead, they are sampled independently conditioned on $\mathbf{x}^*(1) = \cdots = \mathbf{x}^*(T)$. This hidden correlation in *both* the **YES** and the **NO** cases turns out to be a serious problem.

Remark 5.6. We note that when $\mathcal{D}_N = \mathcal{D}_{\mathcal{F}} \times \mathrm{Unif}([q]^k)$ for some $\mathcal{D}_{\mathcal{F}} \in \Delta(\mathcal{F})$, then the distributions $\mathcal{N}|_{\mathbf{x}^*}$ are identical for all $\mathbf{x}^*$ (and the variables $\mathbf{z}^{(t)}(i)$ are distributed as $\mathcal{D}_{\mathcal{F}} \times \mathrm{Bern}(q^{-k})$ independently for every t, i).

For technical reasons, we need the following *padded* version of streaming-SD to extend our lower bound techniques in the streaming setting beyond the setting of one-wise independent distributions.

Definition 5.7 ($(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T, \mathcal{D}_0, \tau)$-padded-streaming-SD). For $k, q, T \in \mathbb{N}$, $\alpha \in (0, 1/k]$, $\tau \in [0, 1)$, a finite set $\mathcal{F}$, and distributions $\mathcal{D}_Y, \mathcal{D}_N, \mathcal{D}_0$ over $\mathcal{F} \times [q]^k$, the streaming problem $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T, \mathcal{D}_0, \tau; \alpha, k, q)$-padded-streaming-SD is the task of distinguishing, for every n, $\boldsymbol{\sigma} \sim \mathcal{Y}_{\text{pad-stream},n}$ from $\boldsymbol{\sigma} \sim \mathcal{N}_{\text{pad-stream},n}$ where for a given length parameter n, the distributions $\mathcal{Y}_{\text{pad-stream}} = \mathcal{Y}_{\text{pad-stream},n}$ and $\mathcal{N}_{\text{pad-stream}} = \mathcal{N}_{\text{pad-stream},n}$ are defined as follows: Sample $\mathbf{x}^*$ from $[q]^n$ uniformly. For each $i \in [\frac{\tau}{1-\tau}\alpha n T]$, uniformly sample a tuple $e^{(0)}(i) = (i_1, \ldots, i_k) \in \binom{[n]}{k}$ and $(f_i, \mathbf{b}^{(0)}(i)) \sim \mathcal{D}_0$, let $\boldsymbol{\sigma}^{(0)}(i) = (e^{(0)}(i), (f_i, \mathbf{1}_{\mathbf{b}^{(0)}(i)=\mathbf{x}^*|_{e^{(0)}(i)}}))$. Next, sample $\boldsymbol{\sigma}^{(1)}, \ldots, \boldsymbol{\sigma}^{(T)}$ according to the Yes and No distribution of $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T)$-streaming-SD, respectively. Finally, let $\boldsymbol{\sigma} = \boldsymbol{\sigma}^{(0)} \circ \cdots \circ \boldsymbol{\sigma}^{(T)}$ be the concatenation of the $\boldsymbol{\sigma}^{(t)}$s.

Again, when α, k, q are clear from context, we suppress them and simply refer to the $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T, \mathcal{D}_0, \tau)$-padded-streaming-SD problem. Note that when $\tau = 0$, $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T, \mathcal{D}_0, \tau)$-padded-streaming-SD is the same as $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T)$-streaming-SD.

5.2.2 CSP Value of padded-streaming-SD. There is a natural way to convert instances of padded-streaming-SD to instances of a Max-CSP($\mathcal{F}$) problem. In this section we make this conversion explicit and show how to use properties of the underlying distributions $\mathcal{D}_0, \mathcal{D}_Y, \mathcal{D}_N$ to get bounds on the value of the instances produced.

Note that an instance $\boldsymbol{\sigma}$ of padded-streaming-SD is simply a sequence $(\sigma(1), \ldots, \sigma(\ell))$, where each $\sigma(i) = (\mathbf{j}(i), \mathbf{z}(i))$ with $\mathbf{j}(i) \in [n]^k$ and $\mathbf{z}(i) = (f_i, \tilde{z}_i) \in \mathcal{F} \times \{0, 1\}$. This sequence is already syntactically very close to the description of a Max-CSP($\mathcal{F}$) instance. Formally, we define an instance $\Psi(\boldsymbol{\sigma})$ of Max-CSP($\mathcal{F}$) as follows. For each $\sigma_i = (\mathbf{j}(i), \mathbf{z}(i))$ with $\mathbf{z}(i) = (f_i, \tilde{z}_i)$, if $\tilde{z}_i = 1$, we add the constraint $f_i(\mathbf{x}|_{\mathbf{j}(i)})$ to $\Psi(\boldsymbol{\sigma})$; otherwise, we do not add any constraint to the formula.

In what follows we show that if $\mathcal{D}_Y \in S^Y_\gamma$, then for all sufficiently large constant T and sufficiently large n, if we draw $\boldsymbol{\sigma} \sim \mathcal{Y}_{\text{pad-stream},n}$, then with high probability, $\Psi(\boldsymbol{\sigma})$ has value at least $\gamma - o(1)$. Conversely, if $\mathcal{D}_N \in S^N_\beta$, then for all sufficiently large n, if we draw $\boldsymbol{\sigma} \sim \mathcal{N}_{\text{pad-stream},n}$, then with high probability $\Psi(\boldsymbol{\sigma})$ has value at most $\beta + o(1)$.

Lemma 5.8 (CSP Value of padded-streaming-SD). *For every $q, k \in \mathbb{N}$, $\mathcal{F} \subseteq \{f : [q]^k \to \{0, 1\}\}$, $0 \le \beta < \gamma \le 1$, $\varepsilon > 0$, $\tau = [0, 1)$, and distributions $\mathcal{D}_Y, \mathcal{D}_N, \mathcal{D}_0 \in \Delta(\{-1, 1\}^k)$ there exists α_0 such that for every $\alpha \in (0, \alpha_0]$ the following hold for every sufficiently large T:*

*(1) If $\tau\mathcal{D}_0 + (1-\tau)\mathcal{D}_Y \in S^Y_\gamma$, then for every sufficiently large n, the $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T, \mathcal{D}_0, \tau)$-padded-streaming-SD **YES** instance $\boldsymbol{\sigma} \sim \mathcal{Y}_{\text{pad-stream},n}$ satisfies $\Pr[\mathit{val}_{\Psi(\boldsymbol{\sigma})} < (\gamma - \varepsilon)] \le \exp(-n)$.*[10]

*(2) If $\tau\mathcal{D}_0 + (1-\tau)\mathcal{D}_N \in S^N_\beta$, then for every sufficiently large n, the $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T, \mathcal{D}_0, \tau)$-padded-streaming-SD **NO** instance $\boldsymbol{\sigma} \sim \mathcal{N}_{\text{pad-stream},n}$ satisfies $\Pr[\mathit{val}_{\Psi(\boldsymbol{\sigma})} > (\beta + \varepsilon)] \le \exp(-n)$.*

Furthermore, if $\gamma = 1$, then $\Pr_{\boldsymbol{\sigma} \sim \mathcal{Y}_{\text{pad-stream},n}}\left[\mathit{val}_{\Psi(\boldsymbol{\sigma})} = 1\right] = 1$.

[10] In this lemma and proof we use $\exp(-n)$ to denote functions of the form c^{-n} for some $c > 1$ that does not depend on n or T, but may depend on all other parameters including q, k, $\mathcal{D}_Y$, $\mathcal{D}_N$, $\mathcal{D}_0$, β, γ, ε.

Proof. We assume $\varepsilon \le 1/2$ (and if not we prove the lemma for $\varepsilon' = \frac{1}{2}$ and this implies the lemma also for ε). We prove the lemma for $\alpha_0 = \frac{\varepsilon}{20kq^k}$ and $T_0 = 1000/(\varepsilon^2\alpha)$. In what follows we set $\eta = \frac{\varepsilon}{20kq^k}$.

In what follows we let $N_0 = \frac{\tau\alpha nT}{1-\tau}$, $N_t = \alpha n$ for $t \in [T]$ and $N = N_0 + TN_1$. Recall that an instance of $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T, \mathcal{D}_0, \tau)$-padded-streaming-SD consists of a stream $\boldsymbol{\sigma} = \boldsymbol{\sigma}^{(0)} \circ \cdots \circ \boldsymbol{\sigma}^{(T)}$, where $\boldsymbol{\sigma}^{(t)} = (\boldsymbol{\sigma}^{(t)}(i) | i \in [N_t])$ and $\boldsymbol{\sigma}^{(t)}(i) = (e^{(t)}(i), (f_i^{(t)}, \tilde{z}^{(t)}(i))$, where $e^{(t)}(i)$ denotes a k-uniform hyperedge on $[n]$ and $f^{(t)}(i) \in \mathcal{F}$ and $\tilde{z}^{(t)}(i) \in \{0,1\}$. Finally, recall that $\boldsymbol{\sigma}^{(t)} \sim \mathcal{Y}|_{\mathbf{x}^*}$ in the **YES** case and $\boldsymbol{\sigma}^{(t)} \sim \mathcal{N}|_{\mathbf{x}^*}$ in the **NO** case independently for each t, where $\mathbf{x}^* \sim \mathsf{Unif}([q]^n)$ is common across all t. We use $\mathcal{I} = (\{0\} \times [T_0]) \cup ([T] \times [T_1])$ to denote the set of legal pairs of indices (t,i). We let m denote the total number of constraints in $\Psi(\boldsymbol{\sigma})$, with m_t denoting the number of constraints from $\boldsymbol{\sigma}^{(t)}$ for $0 \le t \le T$. (Note that m and the m_ts are random variables.)

For $\eta > 0$, define $\mathbf{x}^*$ to be *η-good* if for every $\sigma \in [q]$, we have $|\{i \in [n] \mid \mathbf{x}^*_i = \sigma\}| \in [(1-\eta) \cdot \frac{n}{q}, (1+\eta) \cdot \frac{n}{q}]$. A straightforward application of Chernoff bounds shows that for every $\eta > 0$ the vector $\mathbf{x}^*$ is η-good with probability $1 - \exp(-n)$.

Below we condition on a good $\mathbf{x}^*$ and prove the following: (1) we show the expected value of m is roughly $q^{-k} \cdot N$ and furthermore m is sharply concentrated around its expectation, (2) in the **YES** case we prove that the expected number of constraints satisfied by $\mathbf{x}^*$ is roughly at least $\gamma \cdot q^{-k} \cdot N$ and again this variable is sharply concentrated around its expectation, and (3) in the **NO** case we prove that the expected number of constraints satisfied by any assignment is roughly at most $\beta \cdot q^{-k} \cdot N$ and again this variable is sharply concentrated around its expectation. We note that the sharp concentration part is essentially the same in all cases and it is bounding the expectations that is different in each case. That being said, the analysis of the **NO** case does require sharper concentration since we need to take a union bound over all possible assignments.

Bounding the number of constraints. We start with step (1). Fix an η-good $\mathbf{x}^*$. Note that $m_t = \sum_{i \in [N_t]} \tilde{z}^{(t)}(i)$ for every $0 \le t \le T$. We divide the analysis into two subparts. In step (1a) we bound $\mu \triangleq \mathbb{E}[\tilde{z}^{(t)}(i)]$ (in particular this expectation does not depend on i or t). Note that $m = \sum_{t=0}^{T} \sum_{i \in [N_t]} \tilde{z}^{(t)}(i)$ and so bounding μ bounds $\mathbb{E}[m] = \mu \cdot N$. Then in step (1b) we show that m is concentrated around its expected value.

For step (1a), let p_σ denote the fraction of occurrences of the letter σ in $\mathbf{x}^*$, i.e., $p_\sigma = \frac{1}{n}|\{i \in [n] | \mathbf{x}^*_i = \sigma\}|$. Note that given a sequence $\widetilde{\mathbf{b}}^{(t)}(i) = \mathbf{a} \in [q]^k$, the probability that $\tilde{z}^{(t)}(i) = 1$ over a random choice of $e^{(t)}(i)$ depends on $\mathbf{a}$ as well as the p_σs. (Specifically this probability is $\prod_{j=1}^{k} p_{\mathbf{a}_j} \pm O(k^2/n)$, where the additive correction term accounts for the sampling without replacement in the choice of $e^{(t)}(i)$.) However, if the vector $\mathbf{x}^*$ is good, this dependence has little quantitative effect. In particular, if $\mathbf{x}^*$ is η-good, we have $\mu \in (\frac{1}{q} \pm \eta)^k \pm O(k^2/n)$ and thus we get $q^{-k} - 2k\eta \le \mu \le q^{-k} + 2k\eta$ provided $\eta \le 1/(4kq)$ and n is sufficiently large. This simplifies further to $\mu \in (1 \pm \frac{\varepsilon}{10})q^{-k}$ using $\eta \le q^{-k}\varepsilon/(20k)$. Summing up over $(t,i) \in \mathcal{I}$, we get $\mathbb{E}[m] \in (1 \pm \frac{\varepsilon}{10})q^{-k}N$.

We now turn to step (1b), i.e., proving that m is concentrated around its expectation. (In this part we work a little harder than necessary to prove that the failure probability is $\exp(-nT)$ rather than $\exp(-n)$. This is not necessary but will be needed for the similar step in step (3).) Let $\tilde{Z}$ denote the set of random variables $\{\tilde{z}^{(t)}(i) | (t,i) \in \mathcal{I}\}$, and for $(t,i) \in \mathcal{I}$, let $\tilde{Z}_{-(t,i)} = \tilde{Z} \setminus \{\tilde{z}^{(t)}(i)\}$. We first show that for every $(t,i) \in \mathcal{I}$ we have $\mathbb{E}[\tilde{z}^{(t)}(i) \mid \tilde{Z}_{-(t,i)}] \in (1 \pm \frac{\varepsilon}{10})\mathbb{E}[\tilde{z}^{(t)}(i)]$. Let B_t denote the tth block of variables, i.e., $B_t = \{\tilde{z}^{(t)}(i) | i \in [N_t]\}$. Now note that the only dependence among the $\tilde{z}^{(t)}(i)$s is among the variables within a block, while the blocks themselves are independent. Furthermore, the variables in the block B_0 are independent of each other. Thus, for $i \in [N_0]$ we have $\mathbb{E}[\tilde{z}^{(0)}(i) | Z_{-(0,i)}] = \mathbb{E}[\tilde{z}^{(0)}(i)]$. For $t > 0$ we have that the variables from block B_t may depend

on each other due to the constraint that the underlying set of hyperedges is vertex disjoint. Fix $(t,i) \in \mathcal{I}$ with $t > 0$ and let S be the set of variables touched by the hyperedges from block B_t, excluding $e^{(t)}(i)$. Now consider picking a hyperedge uniformly from $[n]$ and let ψ be the probability that this hyperedge touches S. We clearly have $\psi \le k|S|/n \le k\alpha$. On the other hand, ψ also upper bounds the difference between $\mathbb{E}[\tilde{z}^{(t)}(i) \mid \tilde{Z}_{-(t,i)}]$ and $\mathbb{E}[\tilde{z}^{(t)}(i)]$, so we have

$$|\,\mathbb{E}[\tilde{z}^{(t)}(i) \mid \tilde{Z}_{-(t,i)}] - \mathbb{E}[\tilde{z}^{(t)}(i)]| \le \psi \le k\alpha \le \frac{\varepsilon q^{-k}}{20} \le \frac{\varepsilon}{10}\,\mathbb{E}[\tilde{z}^{(t)}(i)].$$

Applying Lemma 2.8 to the variables of $\tilde{Z}$ (arranged in some arbitrary order), we have $\Pr[m \notin ((q^{-k} \cdot (1 \pm \varepsilon/10)^3] \le \exp(-nT)$. Using $(1 \pm \varepsilon/10)^3 \subseteq (1 \pm \varepsilon/2)$ for $\varepsilon < 1$, we get

$$\Pr[m \notin (1 \pm \varepsilon/2) \cdot q^{-k} N] \le \exp(-nT). \tag{5.9}$$

*Lower bounding the number of satisfied constraints in the **YES** case.* Let $Z^{(t)}(i)$ be the indicator variable for the event that the ith element of $\boldsymbol{\sigma}^{(t)}$ produces a constraint that is satisfied by $\mathbf{x}^*$, i.e., $Z^{(t)}(i) = \tilde{z}^{(t)}(i) \cdot f_i(\mathbf{x}^*|_{\mathbf{j}^{(t)}(i)})$. Note that the number of constraints satisfied by $\mathbf{x}^*$ is $\sum_{(t,i)\in\mathcal{I}} Z^{(t)}(i)$. Note further that $Z^{(0)}(i)$s are identically distributed across $i \in [N_0]$, and $Z^{(t)}(i)$s are also identically distributed across $t \in [T]$ and $i \in [N_0]$. By construction (see Definition 5.7), we have $\mathbb{E}[Z^{(0)}(i)] = \mathbb{E}_{(f,\mathbf{b})\sim\mathcal{D}_0}[f(\mathbf{b}) \cdot \mathbb{E}_{\mathbf{j}}[\mathbb{1}(\mathbf{x}^*|_{\mathbf{j}} = \mathbf{b})]]$. By the η-goodness of $\mathbf{x}^*$, we have that for every $\mathbf{b} \in [q]^k$, $\mathbb{E}_{\mathbf{j}}[\mathbb{1}(\mathbf{x}^*|_{\mathbf{j}} = \mathbf{b})] \ge (1 - \frac{\varepsilon}{10})q^{-k}$. Thus, we get $\mathbb{E}[Z^{(0)}(i)] \ge (1 - \frac{\varepsilon}{10})q^{-k} \cdot \mathbb{E}_{(f,\mathbf{b})\sim\mathcal{D}_0}[f(\mathbf{b})]$. Similarly, for $t > 0$ we have $\mathbb{E}[Z^{(t)}(i)] = \mathbb{E}_{(f,\mathbf{b})\sim\mathcal{D}_Y}[f(\mathbf{b}) \cdot \mathbb{E}_{\mathbf{j}}[\mathbb{1}(\mathbf{x}^*|_{\mathbf{j}} = \mathbf{b})]] \ge (1 - \frac{\varepsilon}{10})q^{-k} \cdot \mathbb{E}_{(f,\mathbf{b})\sim\mathcal{D}_Y}[f(\mathbf{b})]$. Using linearity of expectations, we now get

$$\begin{aligned}
\mathbb{E}\left[\sum_{(t,i)\in\mathcal{I}} Z^{(t)}(i)\right] &= N_0\,\mathbb{E}[Z^{(0)}(1)] + TN_T\,\mathbb{E}[Z^{(1)}(1)] \\
&= N(\tau\,\mathbb{E}[Z^{(0)}(1)] + (1-\tau)\,\mathbb{E}[Z^{(1)}(1)]) \\
&\ge \left(1 - \frac{\varepsilon}{10}\right) q^{-k} N \cdot (\tau \mathop{\mathbb{E}}_{(f,\mathbf{b})\sim\mathcal{D}_0}[f(\mathbf{b})] + (1-\tau) \mathop{\mathbb{E}}_{(f,\mathbf{b})\sim\mathcal{D}_Y}[f(\mathbf{b})]) \\
&= \left(1 - \frac{\varepsilon}{10}\right) q^{-k} N \cdot \mathop{\mathbb{E}}_{(f,\mathbf{b})\sim\tau\mathcal{D}_0+(1-\tau)\mathcal{D}_Y}[f(\mathbf{b})] \\
&\ge \gamma \cdot \left(1 - \frac{\varepsilon}{10}\right) q^{-k} N,
\end{aligned}$$

where the final inequality uses $\tau\mathcal{D}_0 + (1-\tau)\mathcal{D}_Y \in S^Y_\gamma(\mathcal{F})$. The concentration can be analyzed exactly as in step (1b). In particular, if we let Z denote all variables $Z^{(t)}(i)$s, then we have $\mathbb{E}[Z^{(t)}(i)|Z \setminus \{Z^{(t)}(i)\}] \ge \mathbb{E}[Z^{(t)}(i)] - \frac{\varepsilon}{10}q^{-k}$.

$$\Pr\left[\sum_{(t,i)\in\mathcal{I}} Z^{(t)}(i) \le (\gamma - \frac{3\varepsilon}{10}) \cdot q^{-k} N \le \gamma \cdot (1 - \frac{\varepsilon}{10}) q^{-k} N - \frac{\varepsilon}{5} q^{-k} N\right] \le \exp(-nT). \tag{5.10}$$

*Upper bounding the number of satisfiable constraints in the **NO** case.* Fix an assignment $\boldsymbol{\nu} \in [q]^k$ and consider the expected number of constraints satisfied by $\boldsymbol{\nu}$. (We will later take a union bound over all $\boldsymbol{\nu}$.) Let $W^{(t)}(i)$ be the indicator variable for the event that the ith element of $\boldsymbol{\sigma}^{(t)}$ produces a constraint that is satisfied by $\boldsymbol{\nu}$, i.e., $W^{(t)}(i) = \tilde{z}^{(t)}(i) \cdot f_i(\boldsymbol{\nu}|_{\mathbf{j}^{(t)}(i)})$. Note once again that $W^{(0)}(i)$s are identically distributed across i and $W^{(t)}(i)$s are identical across $t > 0$ and i. Let $\mu_0 = \mathbb{E}[W^{(0)}(1)]$ and $\mu_N = \mathbb{E}[W^{(1)}(1)]$. Note that the expected number of satisfied constraints is $\mathbb{E}[\sum_{(t,i)\in\mathcal{I}} W^{(t)}(i)] =$

$N \cdot (\tau\mu_0 + (1-\tau)\mu_N)$, so we bound μ_0 and μ_N. By construction we have

$$\mu_0 = \mathop{\mathbb{E}}_{(f,\mathbf{b})\sim\mathcal{D}_0,\mathbf{j}}[f(\boldsymbol{\nu}|_{\mathbf{j}}) \cdot \mathbb{1}(\mathbf{x}^*|_{\mathbf{j}} = \mathbf{b})] = \mathop{\mathbb{E}}_{(f,\mathbf{b})\sim\mathcal{D}_0,\mathbf{j}}[\mathbb{1}(\mathbf{x}^*|_{\mathbf{j}} = \mathbf{b})] \cdot \mathop{\mathbb{E}}_{(f,\mathbf{b})\sim\mathcal{D}_0,\mathbf{j}}[f(\boldsymbol{\nu}|_{\mathbf{j}}) \mid \mathbb{1}(\mathbf{x}^*|_{\mathbf{j}} = \mathbf{b})],$$

where $\mathbf{j}$ is a uniform random sequence of k distinct elements of $[n]$. As argued earlier, for every $\mathbf{b}$ we have $\mathbb{E}_{\mathbf{j}}[\mathbb{1}(\mathbf{x}^*|_{\mathbf{j}} = b)] \le (1 + \frac{\varepsilon}{10})q^{-k}$ for η-good $\mathbf{x}^*$. So we turn to bounding the second term.

For $\sigma, \rho \in [q]$ let $\mathcal{P}_\sigma(\rho)$ be the fraction of coordinates in $\boldsymbol{\nu}$ that take the value ρ among those coordinates where $\mathbf{x}^*$ is σ, i.e., $\mathcal{P}_\sigma(\rho) = \frac{|\{i\in[n]|\boldsymbol{\nu}_i=\rho \ \&\ \mathbf{x}^*_i=\sigma\}|}{|\{i\in[n]|\mathbf{x}^*_i=\sigma\}|}$. Note that for every σ, $\mathcal{P}_\sigma$ is a probability distribution in $\Delta(q)$. Furthermore, conditioning on $\mathbf{x}^*|_{\mathbf{j}}(\ell) = \mathbf{b}(\ell)$, the distribution of $\boldsymbol{\nu}|_{\mathbf{j}}(\ell)$ is given by $\mathcal{P}_{\mathbf{b}(\ell)}$. Thus, the joint distribution of $\boldsymbol{\nu}|_{\mathbf{j}}$ is $O(k^2/n)$-close in total variation distance to $\mathcal{P}_{\mathbf{b}(1)} \times \cdots \times \mathcal{P}_{\mathbf{b}(k)}$. We thus have

$$\begin{aligned}\mathop{\mathbb{E}}_{(f,\mathbf{b})\sim\mathcal{D}_0,\mathbf{j}}[f(\boldsymbol{\nu}|_{\mathbf{j}}) \mid \mathbb{1}(\mathbf{x}^*|_{\mathbf{j}} = \mathbf{b})] &\le \mathop{\mathbb{E}}_{(f,\mathbf{a})\sim\mathcal{D}_0}\Big[\mathop{\mathbb{E}}_{\mathbf{c},\mathbf{c}_\ell\sim\mathcal{P}_{a_\ell}}[f(\mathbf{c})]\Big] + O(k^2/n)\\ &= \mathop{\mathbb{E}}_{(f,\mathbf{a})\sim\mathcal{D}_0}\Big[\mathop{\mathbb{E}}_{\mathbf{d},d_{\ell,\sigma}\sim\mathcal{P}_\sigma}[C(f,\mathbf{a})(\mathbf{d})]\Big] + O(k^2/n),\end{aligned}$$

where $\mathbf{c} \in [q]^k$ and $\mathbf{d} \in [q]^{k\times q}$. Note that the final expression is simply a change of notation applied to the middle expression above to make the expression syntactically closer to the notation in the definition of $S^N_\beta(\mathcal{F})$. Combining with the bound on $\mathbb{E}_{\mathbf{j}}[\mathbb{1}(\mathbf{x}^*|_{\mathbf{j}} = b)]$ above, we get

$$\mu_0 \le (1 + \frac{\varepsilon}{10})q^{-k} \cdot \Big(\mathop{\mathbb{E}}_{(f,\mathbf{a})\sim\mathcal{D}_0}\Big[\mathop{\mathbb{E}}_{\mathbf{d},d_{\ell,\sigma}\sim\mathcal{P}_\sigma}[C(f,\mathbf{a})(\mathbf{d})]\Big]\Big) + O(k/n).$$

Similarly, we get

$$\mu_N \le (1 + \frac{\varepsilon}{10})q^{-k} \cdot \Big(\mathop{\mathbb{E}}_{(f,\mathbf{a})\sim\mathcal{D}_N}\Big[\mathop{\mathbb{E}}_{\mathbf{d},d_{\ell,\sigma}\sim\mathcal{P}_\sigma}[C(f,\mathbf{a})(\mathbf{d})]\Big]\Big) + O(k/n).$$

Now combining the two conditions above, we get

$$\begin{aligned}(\tau\mu_0 + (1-\tau)\mu_N) &\le (1 + \frac{\varepsilon}{10})q^{-k} \cdot \left(\mathop{\mathbb{E}}_{(f,\mathbf{a})\sim\tau\mathcal{D}_0+(1-\tau)\mathcal{D}_N}\Big[\mathop{\mathbb{E}}_{\mathbf{d},d_{\ell,\sigma}\sim\mathcal{P}_\sigma}[C(f,\mathbf{a})(\mathbf{d})]\Big]\right) + O(k^2/n)\\ &\le \beta \cdot (1 + \frac{\varepsilon}{10})q^{-k} + O(k^2/n)\\ &\le \beta \cdot (1 + \frac{\varepsilon}{9})q^{-k},\end{aligned}$$

where the final inequality uses the fact that n is sufficiently large. We thus conclude the the expected number of constraints satisfied by $\boldsymbol{\nu}$ is at most $\beta \cdot (1 + \frac{\varepsilon}{9})q^{-k}N$. Concentration around the mean is now similar to before. In particular, if we let W denote the set of all $W^{(t)}(i)$s, then we still have $\mathbb{E}[W^{(t)}(i)|W\setminus\{W^{(t)}(i)\}] \le \mathbb{E}[W^{(t)}(i)] + k\alpha \le \mathbb{E}[W^{(t)}(i)] + \frac{\varepsilon}{10}q^{-k}N$, and so by Lemma 2.8 we get

$$\Pr\left[\sum_{(t,i)\in\mathcal{I}} W^{(t)}(i) \ge (\beta + \frac{2\varepsilon}{9}) \cdot q^{-k}N \ge \beta(1 + \varepsilon/9)q^{-k}N + \frac{\varepsilon}{9}q^{-k}N\right] \le \exp(-nT).$$

In particular, by using T sufficiently large, we get that the probability that more than $(\beta + \frac{2\varepsilon}{9}) \cdot q^{-k}N$ constraints are satisfied by $\boldsymbol{\nu}$ is at most c^{-n} for some $c > q$. So by a union bound over all possible $\boldsymbol{\nu}$s we get the following:

$$\Pr\left[\exists\boldsymbol{\nu} \in [q]^k \text{ s.t. } \boldsymbol{\nu} \text{ satisfies more than } (\beta + \frac{2\varepsilon}{9}) \cdot q^{-k}N \text{ constraints}\right] \le \exp(-nT). \tag{5.11}$$

Putting it together. Putting the above together, we get that in the **YES** case with probability $1 - \exp(-n)$ we have that $\mathbf{x}^*$ is good and the number of constraints is at most $(1 + \frac{\varepsilon}{2})q^{-k}N$ (by Equation (5.9)), while the number of satisfied constraints is at least $(\gamma - \frac{3\varepsilon}{10}) \cdot q^{-k}N$ (by Equation (5.10)). Taking ratios, we get

$$\mathsf{val}_{\Psi(\boldsymbol{\sigma})} \geq \frac{\gamma - \frac{3\varepsilon}{10}}{1 + \frac{\varepsilon}{2}} \geq \gamma - \varepsilon.$$

Similarly, in the **NO** case with probability at least $1 - \exp(-n)$ we have that $\mathbf{x}^*$ is good and the number of constraints is at least $(1 - \frac{\varepsilon}{2})q^{-k}N$ (by Equation (5.9)), while the number of satisfied constraints is at most $(\beta + \frac{2\varepsilon}{9}) \cdot q^{-k}N$ (by Equation (5.11)). Taking ratios, we get

$$\mathsf{val}_{\Psi(\boldsymbol{\sigma})} \leq \frac{\beta + \frac{2\varepsilon}{9}}{1 - \frac{\varepsilon}{2}} \leq \beta + \varepsilon.$$

This proves the main part of the lemma.

The furthermore part follows from the fact that if $\gamma = 1$, then every constraint in the **YES** case is satisfied by $\mathbf{x}^*$. □

5.2.3 Reduction from One-way $(\mathcal{D}_Y, \mathcal{D}_N)$-SD to padded-streaming-SD. We start by reducing SD to padded-streaming-SD in the special case where $\mathcal{D}_N$ is "uniform on the variables" in the sense defined next. We say a distribution $\mathcal{D} \in \Delta(\mathcal{D} \times [q]^k)$ is *uniform on the variables* if there exists a distribution $\mathcal{D}_f \in \Delta(\mathcal{F})$ such that $\mathcal{D} = \mathcal{D}_f \times \mathsf{Unif}([q]^k)$. The following lemma implies that in this special case padded-streaming-SD is hard. Since this holds for all one-wise independent distributions $\mathcal{D}_Y$, by applying the lemma twice, we get that padded-streaming-SD is hard for all one-wise independent $\mathcal{D}_Y$ and $\mathcal{D}_N$.

Lemma 5.12. *Let $\mathcal{F}$ be a finite set, $T, q, k \in \mathbb{N}$, $\alpha \in (0, \alpha_0(k)]$, $\tau \in [0, 1)$, and $\mathcal{D}_Y, \mathcal{D}_N, \mathcal{D}_0 \in \Delta(\mathcal{F} \times [q]^k)$ with $\mathcal{D}_Y$ being one-wise independent and $\mathcal{D}_N = \mathcal{D}_f \times \mathsf{Unif}([q]^k)$ for some $\mathcal{D}_f \in \Delta(\mathcal{F})$ and $\boldsymbol{\mu}(\mathcal{D}_Y) = \boldsymbol{\mu}(\mathcal{D}_N)$. Suppose there is a streaming algorithm* **ALG** *that solves $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T, \mathcal{D}_0, \tau)$-padded-streaming-SD on instances of length n with advantage Δ and space s; then there is a one-way protocol for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD on instances of length n using at most sT bits of communication, achieving advantage at least Δ/T.*

The proof of Lemma 5.12 is based on a hybrid argument (e.g., [56, Lemma 6.3]). We provide a proof here based on the proof of [41, Lemma 4.11].

Proof of Lemma 5.12. Note that since we are interested in distributional advantage, we can fix the randomness in **ALG** so that it becomes a deterministic algorithm. By an averaging argument, the randomness can be chosen to ensure the advantage does not decrease. Let Γ denote the evolution of the function of **ALG** as it processes a block of edges. That is, if the algorithm is in state s and receives a stream $\boldsymbol{\sigma}$, then it ends in state $\Gamma(s, \boldsymbol{\sigma})$. Let s_0 denote its initial state.

We consider the following collection of (jointly distributed) random variables: Let $\mathbf{x}^* \sim \mathsf{Unif}(\{-1, 1\}^n)$. Denote $\mathcal{Y} = \mathcal{Y}_{\text{pad-stream},n}$ and $\mathcal{N} = \mathcal{N}_{\text{pad-stream},n}$. Let $(\boldsymbol{\sigma}_Y^{(0)}, \boldsymbol{\sigma}_Y^{(1)}, \ldots, \boldsymbol{\sigma}_Y^{(T)}) \sim \mathcal{Y}|_{\mathbf{x}^*}$. Similarly, let $(\boldsymbol{\sigma}_N^{(0)}, \boldsymbol{\sigma}_N^{(1)}, \ldots, \boldsymbol{\sigma}_N^{(T)}) \sim \mathcal{N}|_{\mathbf{x}^*}$. Recall by Remark 5.6 that since $\mathcal{D}_N = \mathcal{D}_f \times \mathsf{Unif}([q]^k)$, we have that $\mathcal{N}|_{\mathbf{x}^*}$ is independent of $\mathbf{x}^*$, a feature that will be crucial to this proof.

Let S_t^Y denote the state of **ALG** after processing $\boldsymbol{\sigma}_Y^{(0)}, \ldots, \boldsymbol{\sigma}_Y^{(t)}$, i.e., $S_0^Y = \Gamma(s_0, \boldsymbol{\sigma}_Y^{(0)})$ and $S_t^Y = \Gamma(S_{t-1}^Y, \boldsymbol{\sigma}_Y^{(t)})$, where s_0 is the fixed initial state (recall that **ALG** is deterministic). Similarly, let S_t^N denote the state of **ALG** after processing $\boldsymbol{\sigma}_N^{(0)}, \ldots, \boldsymbol{\sigma}_N^{(t)}$. Note that since $\boldsymbol{\sigma}_Y^{(0)}$ has the same distribution (conditioned on the same $\mathbf{x}^*$) as $\boldsymbol{\sigma}_N^{(0)}$ by definition, we have $\|S_0^Y - S_0^N\|_{tvd} = 0$.

Let $S^Y_{a:b}$ denote the sequence of states $(S^Y_a, \ldots, S^Y_b)$ and similarly for $S^N_{a:b}$. Now let $\Delta_t = \|S^Y_{0:t} - S^N_{0:t}\|_{tvd}$. Observe that $\Delta_0 = 0$, while $\Delta_T \geq \Delta$. (The latter is based on the fact that **ALG** distinguishes the two distributions with advantage Δ.) Thus, $\Delta \leq \Delta_T - \Delta_0 = \sum_{t=0}^{T-1}(\Delta_{t+1} - \Delta_t)$ and so there exists $t^* \in \{0, 1, \ldots, T-1\}$ such that

$$\Delta_{t^*+1} - \Delta_{t^*} = \|S^Y_{0:t^*+1} - S^N_{0:t^*+1}\|_{tvd} - \|S^Y_{0:t^*} - S^N_{0:t^*}\|_{tvd} \geq \frac{\Delta}{T}\,.$$

Now consider the random variable $\tilde{S} = \Gamma(S^Y_{t^*}, \boldsymbol{\sigma}_N^{(t^*+1)})$ (so the previous state is from the **YES** distribution and the input is from the **NO** distribution). We claim below that $\|S^Y_{t^*+1} - \tilde{S}\|_{tvd} = \mathbb{E}_{A\sim_d S^Y_{0:t^*}}[\|S^Y_{t^*+1}|_{S^Y_{0:t^*}=A} - \tilde{S}|_{S^Y_{0:t^*}=A}\|_{tvd}] \geq \Delta_{t^*+1} - \Delta_{t^*}$. Once we have the claim, we show how to get a space $T \cdot s$ protocol for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_n)$-SD with advantage $\Delta_{t^*+1} - \Delta_{t^*}$, concluding the proof of the lemma.

Claim 5.13. $\|S^Y_{t^*+1} - \tilde{S}\|_{tvd} \geq \Delta_{t^*+1} - \Delta_{t^*}$.

Proof. First, by triangle inequality for the total variation distance, we have

$$\|S^Y_{t^*+1} - \tilde{S}\|_{tvd} \geq \|S^Y_{t^*+1} - S^N_{t^*+1}\|_{tvd} - \|\tilde{S} - S^N_{t^*+1}\|_{tvd}\,.$$

Recall that $\tilde{S} = \Gamma(S^Y_{t^*}, \boldsymbol{\sigma}_N^{(t^*+1)})$ and $S^N_{t^*+1} = \Gamma(S^N_{t^*}, \boldsymbol{\sigma}_N^{(t^*+1)})$. Also, note that $\boldsymbol{\sigma}_N^{(t^*+1)}$ follows the product distribution $(\mathcal{D}_f \times \mathsf{Bern}(q^{-k}))^{\alpha n}$ and in particular is independent of $S^Y_{t^*}$ and $S^N_{t^*}$. (This is where we rely crucially on the property $\mathcal{D}_N = \mathcal{D}_f \times \mathsf{Unif}([q]^k)$.) Furthermore, Γ is a deterministic function, and so we can apply the data processing inequality (Item (2) of Proposition 2.7 with $X = S^Y_{t^*}$, $Y = S^N_{t^*}$, $W = \boldsymbol{\sigma}_N^{(t^*+1)}$, and $f = \Gamma$) to conclude

$$\|\tilde{S} - S^N_{t^*+1}\|_{tvd} = \|\Gamma(S^Y_{t^*}, \boldsymbol{\sigma}_N^{(t^*+1)}) - \Gamma(S^N_{t^*}, \boldsymbol{\sigma}_N^{(t^*+1)})\|_{tvd} \leq \|S^Y_{t^*} - S^N_{t^*}\|_{tvd}.$$

Combining the two inequalities above, we get

$$\|S^Y_{t^*+1} - \tilde{S}\|_{tvd} \geq \|S^Y_{t^*+1} - S^N_{t^*+1}\|_{tvd} - \|S^Y_{t^*} - S^N_{t^*}\|_{tvd} = \Delta_{t^*+1} - \Delta_{t^*},$$

as desired. □

We now show how a protocol can be designed for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD that achieves advantage at least $\theta = \mathbb{E}_{A\sim_d S^Y_{0:t^*}}[\|S^Y_{t^*+1}|_{S_{0:t^*}=A} - \tilde{S}|_{S_{0:t^*}=A}\|_{tvd}] \geq \Delta_{t^*+1} - \Delta_{t^*}$, concluding the proof of the lemma. The protocol uses the distinguisher $T_A : \{0,1\}^s \to \{0,1\}$ such that $\mathbb{E}_{A, S^Y_{t^*+1}, \tilde{S}}[T_A(S^Y_{t^*+1})] - \mathbb{E}[T_A(\tilde{S})] \geq \theta$, which is guaranteed to exist by the definition of total variation distance.

Our protocol works as follows: Let Alice receive input $\mathbf{x}^*$ and Bob receive inputs $(M, \mathbf{z})$ sampled from either $\mathcal{Y}_{\mathrm{SD}}|_{\mathbf{x}^*}$ or $\mathcal{N}_{\mathrm{SD}}|_{\mathbf{x}^*}$, where $\mathcal{Y}_{\mathrm{SD}}$ and $\mathcal{N}_{\mathrm{SD}}$ are the Yes and No distribution of $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD, respectively.

(1) Alice samples $(\boldsymbol{\sigma}^{(0)}, \boldsymbol{\sigma}^{(1)}, \ldots, \boldsymbol{\sigma}^{(T)}) \sim \mathcal{Y}|_{\mathbf{x}^*}$ and computes $A = S^Y_{0:t^*} \in \{0,1\}^{(t^*+1)s}$ and sends A to Bob.
(2) Bob extracts $S^Y_{t^*}$ from A; computes $\widehat{S} = \Gamma(S^Y_{t^*}, \boldsymbol{\sigma})$, where $\boldsymbol{\sigma}$ is the encoding of $(M, \mathbf{z})$ as a stream; and outputs **YES** if $T_A(\widehat{S}) = 1$ and **NO** otherwise.

Note that if $(M, \mathbf{z}) \sim \mathcal{Y}_{\mathrm{SD}}|_{\mathbf{x}^*}$, then $\widehat{S} \sim_d S^Y_{t^*+1}|_{S^Y_{0:t^*}=A}$, while if $(M, \mathbf{z}) \sim \mathcal{N}_{\mathrm{SD}}|_{\mathbf{x}^*}$, then $\widehat{S} \sim \tilde{S}_{S^Y_{0:t^*}=A}$. It follows that the advantage of the protocol above exactly equals $\mathbb{E}_A[T_A(S^Y_{t^*+1})] - \mathbb{E}_A[T_A(\tilde{S})] \geq \theta \geq \Delta_{t^*+1} - \Delta_{t^*} \geq \Delta/T$. This concludes the proof of the lemma. □

By combining Lemma 5.12 with Theorem 5.4, we immediately have the following consequence.

Lemma 5.14. *For $k \in \mathbb{N}$ let $\alpha_0(k)$ be as given by Theorem 5.4. Let $T \in \mathbb{N}$, $\alpha \in (0, \alpha_0(k)]$, $\tau \in [0,1)$, and $\mathcal{D}_0, \mathcal{D}_Y, \mathcal{D}_N, \in \Delta(\mathcal{F} \times [q]^k)$, where $\mathcal{D}_Y$ and $\mathcal{D}_N$ are one-wise independent distributions with $\boldsymbol{\mu}(\mathcal{D}_Y) = \boldsymbol{\mu}(\mathcal{D}_N)$.*

Then every streaming algorithm **ALG** *solving $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T, \mathcal{D}_0, \tau)$-padded-streaming-SD in the streaming setting with advantage $1/8$ for all lengths n uses space $\Omega(\sqrt{n})$.*

Proof. Let **ALG** be an algorithm using space s solving $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T, \mathcal{D}_0, \tau)$-padded-streaming-SD with advantage $1/8$. For $g \in \mathcal{F}$, let $p_g = \Pr_{(f,\sigma)\sim\mathcal{D}_Y}[f = g]$ and let $\mathcal{D}_f$ be the distribution given by $\mathcal{D}_f(g) = p_g$. Let $\mathcal{D}_M = \mathcal{D}_f \times \mathsf{Unif}([q]^k)$. Note that $\mathcal{D}_M$ is uniform on the variables and satisfies $\boldsymbol{\mu}(\mathcal{D}_M) = \boldsymbol{\mu}(\mathcal{D}_Y) = \boldsymbol{\mu}(\mathcal{D}_N)$. Then, by the triangle inequality, **ALG** solves either the $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_M, T, \mathcal{D}_0, \tau)$-padded-streaming-SD with advantage $1/16$ or the $(\mathcal{F}, \mathcal{D}_N, \mathcal{D}_M, T, \mathcal{D}_0, \tau)$-padded-streaming-SD with advantage $1/16$. Assume without loss of generality it is the former. Then, by Lemma 5.12, there exists a one-way protocol for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_M)$-SD using at most sT bits of communication with advantage at least $1/(16T)$. Applying Theorem 5.4 with $\delta = 1/(16T) > 0$, we now get that $s = \Omega(\sqrt{n})$. □

5.2.4 Proof of the Streaming Lower Bound. We are now ready to prove Theorem 3.10.

Proof of Theorem 3.10. We combine Theorem 5.4, Lemma 5.14, and Lemma 5.8. So in particular, we set our parameters α and T so that the conditions of these statements are satisfied. Specifically, k and $\varepsilon > 0$, let $\alpha_0^{(1)}$ be the constant from Theorem 5.4, and let $\alpha_0^{(2)}$ be the constant from Lemma 5.8. Let $\alpha_0 = \min\{\alpha_0^{(1)}, \alpha_0^{(2)}\}$. Given $\alpha \in (0, \alpha_0)$, let T_0 be the constant from Lemma 5.8 and let $T = T_0$. (Note that these choices allow for both Theorem 5.4 and Lemma 5.8 to hold.)

Suppose there exists a streaming algorithm **ALG** that solves $(\gamma - \varepsilon, \beta + \varepsilon)$-Max-CSP$(\mathcal{F})$. Let $\tau \in [0,1)$ and $\mathcal{D}_Y, \mathcal{D}_N, \mathcal{D}_0$ be distributions such that (i) $\mathcal{D}_Y$ and $\mathcal{D}_N$ are one-wise independent, (ii) $\tau\mathcal{D}_0 + (1-\tau)\mathcal{D}_Y \in S_\gamma^Y(\mathcal{F})$, and (iii) $\tau\mathcal{D}_0 + (1-\tau)\mathcal{D}_N \in S_\beta^N(\mathcal{F})$.

Let n be sufficiently large and let $\mathcal{Y}_{\mathsf{stream},n}$ and $\mathcal{N}_{\mathsf{stream},n}$ denote the distributions of **YES** and **NO** instances of $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T, \mathcal{D}_0, \tau)$-padded-streaming-SD of length n. Since α and T satisfy the conditions of Lemma 5.8, we have for every sufficiently large n

$$\Pr_{\sigma\sim\mathcal{Y}_{\mathsf{stream},n}}\left[\mathrm{val}_{\Psi(\sigma)} < (\gamma-\varepsilon)\right] = o(1) \text{ and } \Pr_{\sigma\sim\mathcal{N}_{\mathsf{stream},n}}\left[\mathrm{val}_{\Psi(\sigma)} > (\beta+\varepsilon)\right] = o(1)\,.$$

We conclude that **ALG** can distinguish **YES** instances of Max-CSP$(\mathcal{F})$ from **NO** instances with advantage at least $1/4 - o(1) \geq 1/8$. However, since $\mathcal{D}_Y, \mathcal{D}_N$, and α satisfy the conditions of Lemma 5.14 (in particular $\mathcal{D}_Y$ and $\mathcal{D}_N$ are one-wise independent and $\alpha \in (0, \alpha_0(k))$), such an algorithm requires space at least $\Omega(\sqrt{n})$. Thus, we conclude that any streaming algorithm that solves $(\gamma-\varepsilon, \beta+\varepsilon)$-Max-CSP$(\mathcal{F})$ requires $\Omega(\sqrt{n})$ space.

Finally, note that if $\gamma = 1$, then in Lemma 5.8, we have $\mathrm{val}_\Psi = 1$ with probability one. Repeating the above reasoning with this information shows that $(1, \beta+\varepsilon)-$Max-CSP$(\mathcal{F})$ requires $\Omega(\sqrt{n})$-space. □

5.3 The Lower Bound against Sketching Algorithms

In the absence of a reduction from SD to streaming-SD for general $\mathcal{D}_Y$ and $\mathcal{D}_N$, we turn to other means of using the hardness of SD. In particular, we use lower bounds on the communication complexity of a T-player communication game in the *simultaneous communication* setting—one that is significantly easier to obtain lower bounds for than the one-way setting. Below we describe a family of T-player simultaneous communication games, which we call $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T)$-simultaneous-SD. (See Definition 5.15.) We then show a simple reduction from $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD

to $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T)$-simultaneous-SD. Combining this reduction with our lower bounds on SD and the reduction from simultaneous-SD to streaming complexity leads to the proof of Theorem 5.1.

5.3.1 T-Player Simultaneous Version of SD. In this section, we consider the complexity of *T-player number-in-hand simultaneous message-passing communication games* (abbrev. T-player simultaneous communication games). Such games are described by two distributions $\mathcal{Y}$ and $\mathcal{N}$. An instance of the game is a T-tuple $(X^{(1)}, \ldots, X^{(T)})$ drawn either from $\mathcal{Y}$ or from $\mathcal{N}$, and $X^{(t)}$ is given as input to the tth player. A (simultaneous communication) protocol $\Pi = (\Pi^{(1)}, \ldots, \Pi^{(T)}, \Pi_{\text{ref}})$ is a $(T+1)$-tuple of functions with $\Pi^{(t)}(X^{(t)}) \in \{0,1\}^c$ denoting the tth player's message to the *referee*, and $\Pi_{\text{ref}}(\Pi^{(1)}(X^{(1)}), \ldots, \Pi^{(T)}(X^{(T)})) \in \{\mathbf{YES}, \mathbf{NO}\}$ denoting the protocol's output. We denote this output by $\Pi(X^{(1)}, \ldots, X^{(T)})$. The complexity of this protocol is the parameter c specifying the maximum length of $\Pi^{(1)}(X^{(1)}), \ldots, \Pi^{(T)}(X^{(T)})$ (maximized over all X). The advantage of the protocol Π is the quantity

$$\left| \Pr_{(X^{(1)}, \ldots, X^{(T)}) \sim \mathcal{Y}} [\Pi(X^{(1)}, \ldots, X^{(T)}) = \mathbf{YES}] - \Pr_{(X^{(1)}, \ldots, X^{(T)}) \sim \mathcal{N}} [\Pi(X^{(1)}, \ldots, X^{(T)}) = \mathbf{YES}] \right|.$$

Definition 5.15 ($(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T)$-simultaneous-SD). For $k, T \in \mathbb{N}$, $\alpha \in (0, 1/k]$, a finite set $\mathcal{F}$, distributions $\mathcal{D}_Y, \mathcal{D}_N$ over $\mathcal{F} \times [q]^k$, the $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T)$-simultaneous-SD is a T-player communication game given by a family of instances $(\mathcal{Y}_{\text{simul},n}, \mathcal{N}_{\text{simul},n})_{n \in \mathbb{N}, n \geq 1/\alpha}$, where for a given n, $\mathcal{Y} = \mathcal{Y}_{\text{simul},n}$ and $\mathcal{N} = \mathcal{N}_{\text{simul},n}$ are as follows: Both $\mathcal{Y}$ and $\mathcal{N}$ are supported on tuples $(\mathbf{x}^*, M^{(1)}, \ldots, M^{(T)}, \mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(T)})$, where $\mathbf{x}^* \in [q]^n$, $M^{(t)} \in \{0,1\}^{k\alpha n \times n}$, and $\mathbf{z}^{(t)} \in (\mathcal{F} \times \{0,1\})^{k\alpha n}$, where the pair $(M^{(t)}, \mathbf{z}^{(t)})$ is the tth player's inputs for all $t \in [T]$. We now specify the distributions of $\mathbf{x}^*$, $M^{(t)}$, and $\mathbf{z}^{(t)}$ in $\mathcal{Y}$ and $\mathcal{N}$:

- In both $\mathcal{Y}$ and $\mathcal{N}$, $\mathbf{x}^*$ is distributed uniformly over $[q]^n$.
- In both $\mathcal{Y}$ and $\mathcal{N}$, the matrix $M^{(t)} \in \{0,1\}^{\alpha k n \times n}$ is chosen uniformly (and independently of $\mathbf{x}^*$) among matrices with exactly one 1 per row and at most one 1 per column.
- The vector $\mathbf{z}^{(t)}$ is determined from $M^{(t)}$ and $\mathbf{x}^*$ as follows. Sample a random vector $\mathbf{b}^{(t)} \in (\mathcal{F} \times [q]^k)^{\alpha k n}$ whose distribution differs in $\mathcal{Y}$ and $\mathcal{N}$. Specifically, let $\mathbf{b}^{(t)} = (\mathbf{b}^{(t)}(1), \ldots, \mathbf{b}^{(t)}(\alpha n))$ be sampled from one of the following distributions (independent of $\mathbf{x}^*$ and M):
  - $\mathcal{Y}$: Each $\mathbf{b}^{(t)}(i) = (f_i, \tilde{\mathbf{b}}(i)) \in \mathcal{F} \times [q]^k$ is sampled independently according to $\mathcal{D}_Y$.
  - $\mathcal{N}$: Each $\mathbf{b}^{(t)}(i) = (f_i, \tilde{\mathbf{b}}(i)) \in \mathcal{F} \times [q]^k$ is sampled independently according to $\mathcal{D}_N$.

  We now set $\mathbf{z}^{(t)} = (f_i, \tilde{z}_i)$, where $\tilde{z}_i = 1$ iff $(M^{(t)}\mathbf{x}^*) = \tilde{\mathbf{b}}^{(t)}(i)$.

If $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$, then given an instance $\boldsymbol{\sigma} = (\mathbf{x}^*, M^{(1)}, \ldots, M^{(T)}, \mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(T)})$, we will let $\Psi(\boldsymbol{\sigma})$ represent the associated instance of Max-CSP($\mathcal{F}$) as described in Section 5.2.2.

Note that the instance $\Psi(\boldsymbol{\sigma})$ obtained in the **YES** and **NO** cases of $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T)$-simultaneous-SD is distributed exactly according to instances derived in the **YES** and **NO** cases of $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T, \mathcal{D}_0, \tau = 0)$-padded-streaming-SD and thus Lemma 5.8 can still be applied to conclude that **YES** instances usually satisfy $\text{val}_{\Psi(\boldsymbol{\sigma})} \geq \gamma - o(1)$ and **NO** instances usually satisfy $\text{val}_{\Psi(\boldsymbol{\sigma})} \leq \beta - o(1)$. We will use this property when proving Theorem 5.1.

We start by showing that the simultaneous-SD problems above do not have low-communication protocols when the marginals of $\mathcal{D}_Y$ and $\mathcal{D}_N$ match.

Lemma 5.16. *Let $\mathcal{F}$ be a finite set, $k, q, T \in \mathbb{N}$, $\mathcal{D}_Y, \mathcal{D}_N \in \Delta(\mathcal{F} \times [q]^k)$, and let $\alpha \in (0, 1/k]$. Suppose there is a protocol Π that solves $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T)$-simultaneous-SD on instances of length n with advantage Δ and space s; then there is a one-way protocol for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD on instances of length n using at most $s(T-1)$ bits of communication and achieving advantage at least Δ/T.*

Proof. Let us first fix the randomness in Π so that it becomes a deterministic protocol. Note that by an averaging argument the advantage of Π does not decrease. Recall that $\mathcal{Y}$ and $\mathcal{N}$ are Yes

and No input distributions of $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T)$-simultaneous-SD, and we have

$$\Pr_{X\sim\mathcal{Y}}[\Pi(X) = \mathbf{YES}] - \Pr_{X\sim\mathcal{N}}[\Pi(X) = \mathbf{YES}] \geq \Delta\,.$$

Now, we define the following distributions $\mathcal{D}_0, \ldots, \mathcal{D}_T$. Let $\mathcal{D}_0 = \mathcal{Y}$ and $\mathcal{D}_T = \mathcal{N}$. For each $t \in [T-1]$, we define $\mathcal{D}_t$ to be the distribution of input instances of $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T)$-simultaneous-SD by sampling $\mathbf{b}^{(t')}(i)$ independently according to $\mathcal{D}_Y$ (resp. $\mathcal{D}_N$) for all $t' \leq t$ (resp. $t' > t$) and i (see Definition 5.15 to recall the definition). Next, for each $t \in [T]$, let

$$\Delta_t = \Pr_{X\sim\mathcal{D}_t}[\Pi(X) = \mathbf{YES}] - \Pr_{X\sim\mathcal{D}_{t-1}}[\Pi(X) = \mathbf{YES}]\,.$$

Observe that $\sum_{t\in[T]} \Delta_t = \Delta$ and hence there exists $t^* \in [T]$ such that $\Delta_{t^*} \geq \Delta/T$.

Now we describe a protocol Π' for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD as follows. On input $(\mathbf{x}^*, M, \mathbf{z})$, Alice receives $\mathbf{x}^*$ and Bob receives $(M, \mathbf{z})$. Alice first samples matrices $M^{(1)}, \ldots, M^{(t^*-1)}, M^{(t^*+1)}, \ldots, M^{(T)}$ as the second item in Definition 5.15. Next, Alice samples $\mathbf{b}^{(t')}(i) = (f_i, \tilde{\mathbf{b}}^{(t')}(i))$ according to $\mathcal{D}_Y$ (resp. $\mathcal{D}_N$) for all $t' < t^*$ (resp. $t' > t^*$) and $i \in [\alpha n T]$ and sets $\mathbf{z}^{(t')}(i) = (f_i, \tilde{z}_i)$ as the third item in Definition 5.15. Note that Alice can do this because she possesses $\mathbf{x}^*$. Finally, Alice sends $\{\Pi^{(t')}(M^{(t')}, \mathbf{z}^{(t')})\}_{t'\in[T]\setminus\{t^*\}}$ to Bob. After receiving Alice's message $(X^{(1)}, \ldots, X^{(t^*-1)}, X^{(t^*+1)}, \ldots, X^{(T)})$, Bob computes $\Pi^{(t^*)}(M, \mathbf{z})$ and outputs $\Pi'(M, \mathbf{z}) = \Pi_{\mathrm{ref}}(X^{(1)}, \ldots, X^{(t^*-1)}, \Pi^{(t^*)}(M, \mathbf{z}), X^{(t^*+1)}, \ldots, X^{(T)})$.

It is clear from the construction that the protocol Π' uses at most $s(T-1)$ bits of communication. To see Π' has advantage at least Δ/T, note that if $(\mathbf{x}^*, M, \mathbf{z})$ is sampled from the Yes distribution $\mathcal{Y}_{\mathrm{SD}}$ of $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD, then $((M^{(1)}, \mathbf{z}^{(1)}), \ldots, (M^{(t^*-1)}, \mathbf{z}^{(t^*-1)}), (M, \mathbf{z}), (M^{(t^*+1)}, \mathbf{z}^{(t^*+1)}), \ldots, (M^{(T)}, \mathbf{z}^{(T)}))$ follows the distribution $\mathcal{D}_{t^*}$. Similarly, if $(\mathbf{x}^*, M, \mathbf{z})$ is sampled from the No distribution $\mathcal{N}_{\mathrm{SD}}$ of $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD, then $((M^{(1)}, \mathbf{z}^{(1)}), \ldots, (M^{(t^*-1)}, \mathbf{z}^{(t^*-1)}), (M, \mathbf{z}), (M^{(t^*+1)}, \mathbf{z}^{(t^*+1)}), \ldots, (M^{(T)}, \mathbf{z}^{(T)}))$ follows the distribution $\mathcal{D}_{t^*-1}$. Thus, the advantage of Π' is at least

$$\begin{aligned}&\Pr_{(M,\mathbf{z})\sim\mathcal{Y}_{\mathrm{SD}},\Pi'}[\Pi'(M, \mathbf{z}) = \mathbf{YES}] - \Pr_{(M,\mathbf{z})\sim\mathcal{N}_{\mathrm{SD}},\Pi'}[\Pi'(M, \mathbf{z}) = \mathbf{YES}]\\ &= \Pr_{X\sim\mathcal{D}_{t^*}}[\Pi(X) = \mathbf{YES}] - \Pr_{X\sim\mathcal{D}_{t^*-1}}[\Pi(X) = \mathbf{YES}] = \Delta_{t^*} \geq \Delta/T\,.\end{aligned}$$

We conclude that there is a one-way protocol for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD using at most $s(T-1)$ bits of communication achieving advantage at least Δ/T. □

As an immediate consequence of Theorem 5.4 and Lemma 5.16, we get that $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T)$-simultaneous-SD requires $\Omega(\sqrt{n})$ bits of communication when the marginals of $\mathcal{D}_Y$ and $\mathcal{D}_N$ match.

Lemma 5.17. *For every $k, q \in \mathbb{N}$, there exists $\alpha_0 > 0$ such that for every $\alpha \in (0, \alpha_0)$ and $\delta > 0$ the following holds: For every finite set $\mathcal{F}$ and $T \in \mathbb{N}$ and every pair of distributions $\mathcal{D}_Y, \mathcal{D}_N \in \Delta(\mathcal{F}\times[q]^k)$ with $\boldsymbol{\mu}(\mathcal{D}_Y) = \boldsymbol{\mu}(\mathcal{D}_N)$, there exists $\tau > 0$ and n_0 such that for every $n \geq n_0$, every protocol for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T)$-simultaneous-SD achieving advantage δ on instances of length n requires $\tau\sqrt{n}$ bits of communication.*

We are now ready to prove Theorem 5.1.

5.3.2 *Proof of Theorem 5.1.*

Proof of Theorem 5.1. The proof is a straightforward combination of Lemma 5.8 and Lemma 5.17 and so we pick parameters so that all these are applicable. Given ε and k, let $\alpha_0^{(1)}$ be as given by Lemma 5.8 and let $\alpha_0^{(2)}$ be as given by Lemma 5.17. Let $\alpha = \min\{\alpha_0^{(1)}, \alpha_0^{(2)}\}$. Given this choice of α, let T_0 be as given by Lemma 5.8. We set $T = T_0$ below. Let n be sufficiently large.

Throughout this proof we will be considering integer weighted instances of Max-CSP($\mathcal{F}$) on n variables with constraints. Note that such an instance Ψ can be viewed as a vector in $\mathbb{Z}^N$, where $N = O(|\mathcal{F}| \times n^k)$ represents the number of possibly distinct constraint applications on n variables. Let $\Gamma = \{\Psi | \mathsf{val}_\Psi \geq \gamma - \varepsilon\}$. Let $B = \{\Psi | \mathsf{val}_\Psi \leq \beta + \varepsilon\}$. Suppose there exists a sketching algorithm $\mathbf{ALG}_1$ that solves $(\gamma - \varepsilon, \beta + \varepsilon)$-Max-CSP($\mathcal{F}$) using at most $s(n)$ bits of space. Note that $\mathbf{ALG}_1$ must achieve advantage at least $1/3$ on the problem (Γ, B). By running several independent copies of $\mathbf{ALG}_1$ and thresholding appropriately, we can get an algorithm $\mathbf{ALG}_2$ with space $O(s)$ and advantage $1 - \frac{1}{100}$ solving (Γ, B).

Now, let SKETCH and COMB be the compression and combination functions as given by this sketching algorithm (see Definition 2.3). We use these to design a protocol for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T)$-simultaneous-SD as follows.

Let $(M^{(t)}, \mathbf{z}^{(t)})$ denote the input to the tth player in $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T)$-simultaneous-SD. Each player turns his/her inputs into $\Psi^{(t)} = (C_1^{(t)}, \ldots, C_{m_t}^{(t)})$, where $C_i^{(t)}$ corresponds to the constraint $(\mathbf{j}^{(t)}(i), f_i^{(t)})$, with $\mathbf{j}_i^{(t)} \in [n]^k$ the indicator vector for the ith hyperedge of $M^{(t)}$. Next, the players use shared randomness to compute the sketch of their input $\mathsf{SKETCH}(\Psi^{(t)})$ and send it to the referee. Finally, the referee computes the sketch for all streams $\mathsf{COMB}(\mathsf{SKETCH}(\Psi^{(1)}), \ldots, \mathsf{SKETCH}(\Psi^{(T)}))$ and outputs the corresponding answer.

To analyze the above, note that the communication is $O(s)$. Next, by the advantage of the sketching algorithm, we have that

$$\min_{\Psi \in \Gamma}[\mathbf{ALG}_2(\Psi) = 1] - \max_{\Psi \in B}[\mathbf{ALG}_2(\Psi) = 1] \geq 1 - 12/100. \tag{5.18}$$

Now we consider what happens when $\Psi \sim \mathcal{Y}_{\mathrm{simul},n}$ and $\Psi \sim \mathcal{N}_{\mathrm{simul},n}$. By Lemma 5.8, we have that $\Pr_{\Psi \sim \mathcal{Y}_{\mathrm{simul},n}}[\Psi \in \Gamma] \geq 1 - o(1)$ and $\Pr_{\Psi \sim \mathcal{N}_{\mathrm{simul},n}}[\Psi \in B] \geq 1 - o(1)$. Combining with Equation (5.18), we thus get

$$\Pr_{\Psi \sim \mathcal{Y}_{\mathrm{simul},n}}[\mathbf{ALG}_2(\Psi) = 1] - \Pr_{\Psi \sim \mathcal{N}_{\mathrm{simul},n}}[\mathbf{ALG}_2(\Psi) = 1] \geq 1 - 12/100 - o(1) \geq 1/2.$$

We thus get an $O(s)$ simultaneous communication protocol for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N, T)$-simultaneous-SD with advantage at least $1/2$.

Now we conclude by applying Lemma 5.17 with $\delta = 1/2$ to get that $s = \Omega(\sqrt{n})/T = \Omega(\sqrt{n})$, thus yielding the theorem. □

6 HARDNESS OF ADVICE SIGNAL DETECTION WITH UNIFORM MARGINALS

The goal of this section is to prove a variant of Theorem 5.4 that will be used in Section 7 and Section 8 for a proof of the general case of Theorem 5.4. Recall that in the $(\mathcal{D}_Y, \mathcal{D}_N)$-SD problem, $|\mathcal{F}| = 1$, so we omit $\mathcal{F}$. The main result of this section, presented in Theorem 6.4, gives an $\Omega(\sqrt{n})$ lower bound on the communication complexity of $(\mathcal{D}_Y, \mathcal{D}_N)$-SD for distributions with matching marginals $\boldsymbol{\mu}(\mathcal{D}_Y) = \boldsymbol{\mu}(\mathcal{D}_N)$ for the case when (i) the alphabet is Boolean $\{-1, 1\}$,[11] (ii) the marginals are uniform $\boldsymbol{\mu}(\mathcal{D}_Y) = \boldsymbol{\mu}(\mathcal{D}_N) = 0^k$, but (iii) both players also receive a specific advice vector $\mathbf{a}$. We define the corresponding Advice-SD communication game below.

In order to prove the hardness of Advice-SD, we first define the Advice-RMD communication game and prove an $\Omega(\sqrt{n})$ lower bound on the communication complexity of this game in Theorem 6.2. The proof of the main result of this section, Theorem 6.4, will then follow from the corresponding lower bounds for Advice-RMD in Theorem 6.2.

[11] Throughout this section we use $\{-1, 1\}$ to denote the Boolean domain.

6.1 Hardness of Advice-RMD

In this section we state a theorem that establishes the hardness of RMD in the Boolean setting and with uniform marginals while allowing for advice. The proof of this theorem is postponed to Section 6.3. First we define the Advice-RMD one-way communication game.

Definition 6.1 (Advice-RMD). Let $n, k \in \mathbb{N}, \alpha \in (0,1)$, where k and α are constants with respect to n, and αn is an integer less than n/k. For a pair $\mathcal{D}_Y$ and $\mathcal{D}_N$ of distributions over $\{-1,1\}^k$, we consider the following two-player one-way communication problem $(\mathcal{D}_Y, \mathcal{D}_N)$-Advice-RMD.

— The generator samples the following objects:
(1) $\mathbf{x}^* \sim \mathsf{Unif}(\{-1,1\}^n)$.
(2) $\Gamma \in S_n$ is chosen uniformly among all permutations of n elements.
(3) We let $M \in \{0,1\}^{k\alpha n \times n}$ be a partial permutation matrix capturing $\Gamma^{-1}(j)$ for $j \in [k\alpha n]$. Specifically, $M_{ij} = 1$ if and only if $j = \Gamma(i)$. We view $M = (M_1, \ldots, M_{\alpha n})$, where each $M_i \in \{0,1\}^{k \times n}$ is a block of k successive rows of M.
(4) $\mathbf{b} = (\mathbf{b}(1), \ldots, \mathbf{b}(\alpha n))$ is sampled from one of the following distributions:
  – (**YES**) each $\mathbf{b}(i) \in \{-1,1\}^k$ is sampled according to $\mathcal{D}_Y$.
  – (**NO**) each $\mathbf{b}(i) \in \{-1,1\}^k$ is sampled according to $\mathcal{D}_N$.
(5) $\mathbf{z} = M\mathbf{x}^* \odot \mathbf{b}$, where $\odot$ denotes the coordinate-wise product of the elements.
(6) Define a vector $\mathbf{a} \in [k]^n$ as $a_j = i$, where $i = \Gamma^{-1}(j) \pmod k$ for every $j \in [n]$.
— Alice receives $\mathbf{x}^*$ and $\mathbf{a}$ as input.
— Bob receives M, $\mathbf{z}$, and $\mathbf{a}$ as input.

We follow the approach of [43] to prove the following theorem showing a $\Omega(\sqrt{n})$ communication lower bound for Boolean Advice-RMD. We postpone the proof to Section 6.3.

Theorem 6.2 (Communication Lower Bound for Boolean Advice-RMD). *For every $k \in \mathbb{N}$ and every pair of distributions $\mathcal{D}_Y, \mathcal{D}_N \in \Delta(\{-1,1\}^k)$ with uniform marginals $\boldsymbol{\mu}(\mathcal{D}_Y) = \boldsymbol{\mu}(\mathcal{D}_N) = 0^k$ there exists $\alpha_0 > 0$ such that for every $\alpha \le \alpha_0$ and $\delta > 0$ there exists $\tau > 0$ such that every protocol for $(\mathcal{D}_Y, \mathcal{D}_N)$-Advice-RMD achieving advantage δ requires $\tau\sqrt{n}$ bits of communication on instances of length n.*

6.2 Hardness of Advice-SD

Let us first extend the definition of the SD problem to the following Advice-SD one-way communication game.

Definition 6.3 (Advice-SD). Let $n, k, q \in \mathbb{N}, \alpha \in (0,1)$, where k, q, and α are constants with respect to n, and $\alpha n/k$ is an integer less than n. For a pair $\mathcal{D}_Y$ and $\mathcal{D}_N$ of distributions over $[q]^k$, we consider the following two-player one-way communication problem $(\mathcal{D}_Y, \mathcal{D}_N)$-Advice-SD.

— The generator samples the following objects:
(1) $\mathbf{x}^* \sim \mathsf{Unif}([q]^n)$.
(2) $\Gamma \in S_n$ is chosen uniformly among all permutations of n elements.
(3) We let $M \in \{0,1\}^{k\alpha n \times n}$ be a partial permutation matrix capturing $\Gamma^{-1}(j)$ for $j \in [k\alpha n]$. Specifically, $M_{ij} = 1$ if and only if $j = \Gamma(i)$. We view $M = (M_1, \ldots, M_{\alpha n})$, where each $M_i \in \{0,1\}^{k \times n}$ is a block of k successive rows of M.
(4) $\mathbf{b} = (\mathbf{b}(1), \ldots, \mathbf{b}(\alpha n))$ is sampled from one of the following distributions:
  – (**YES**) each $\mathbf{b}(i) \in [q]^k$ is sampled according to $\mathcal{D}_Y$.
  – (**NO**) each $\mathbf{b}(i) \in [q]^k$ is sampled according to $\mathcal{D}_N$.
(5) $\mathbf{z} = (z_1, \ldots, z_{\alpha n}) \in \{0,1\}^{\alpha n}$ is determined from M, $\mathbf{x}^*$, and $\mathbf{b}$ as follows. We let $z_i = 1$ if $M_i\mathbf{x}^* = \mathbf{b}(i)$, and $z_i = 0$ otherwise.

(6) Define a vector $\mathbf{a} \in [k]^n$ as $a_j = i$, where $i = \Gamma^{-1}(j) \pmod k$ for every $j \in [n]$.
— Alice receives $\mathbf{x}^*$ and $\mathbf{a}$ as input.
— Bob receives M, $\mathbf{z}$, and $\mathbf{a}$ as input.

Almost immediately we get the following corollary for the Advice-SD problem from Theorem 6.2.

Theorem 6.4 (Communication Lower Bound for Boolean Advice-SD). *For every $k \in \mathbb{N}$ and every pair of distributions $\mathcal{D}_Y, \mathcal{D}_N \in \Delta(\{-1,1\}^k)$ with uniform marginals $\boldsymbol{\mu}(\mathcal{D}_Y) = \boldsymbol{\mu}(\mathcal{D}_N) = 0^k$ there exists $\alpha_0 > 0$ such that for every $0 < \alpha \le \alpha_0$ and $\delta > 0$ there exists $\tau > 0$, such that every protocol for $(\mathcal{D}_Y, \mathcal{D}_N)$-advice-SD achieving advantage δ requires $\tau\sqrt{n}$ bits of communication on instances of length n.*

Proof. We show that a protocol achieving advantage δ in the $(\mathcal{D}_Y, \mathcal{D}_N)$-Advice-SD game with s bits of communication implies a protocol achieving advantage δ for the $(\mathcal{D}_Y, \mathcal{D}_N)$-Advice-RMD game with s bits of communication. Then the lower bounds of Theorem 6.2 for distributions with matching marginals will finish the proof.

Assume that there exists Bob's algorithm $\mathcal{B}(M, \mathbf{z}, \mathbf{a}, \text{Alice's message})$ that distinguishes $\mathbf{b}_i \sim \mathcal{D}_Y$ and $\mathbf{b}_i \sim \mathcal{D}_N$ with advantage δ in the Advice-SD game. For the Advice-RMD game, we keep the same algorithm for Alice and modify Bob's algorithm as follows. Bob receives $M \in \{0,1\}^{k\alpha n \times n}$, $\mathbf{z} \in \{-1,1\}^{k\alpha n}$, $\mathbf{a}$, and Alice's message, and partitions $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_{\alpha n})$, where $\mathbf{z}_i \in \{-1,1\}^k$. For each $i \in [\alpha n]$, Bob computes $\widetilde{z}_i \in \{0,1\}$ as follows: $\widetilde{z}_i = 1$ if and only if $\mathbf{z}_i = 1^k$. Now Bob sets $\mathbf{z}' = (\widetilde{z}_1, \ldots, \widetilde{z}_{\alpha n}) \in \{0,1\}^{\alpha n}$ and outputs $\mathcal{B}(M, \mathbf{z}', \mathbf{a}, \text{Alice's message})$. It is easy to see that in both **YES** and **NO** cases, the distribution of the vectors $\mathbf{z}'$ computed by Bob is the distribution of vectors $\mathbf{z}$ sampled in the $(\mathcal{D}_Y, \mathcal{D}_N)$-Advice-SD game. Thus, the protocol achieves advantage δ for the $(\mathcal{D}_Y, \mathcal{D}_N)$-Advice-SD game using s bits of communication as desired. □

6.3 Proof of Theorem 6.2

Our proof of Theorem 6.2 follows the methodology of [43] with some modifications as required by the Advice-RMD formulation. Their proof uses Fourier analysis to reduce the task of proving a communication lower bound to that of proving some combinatorial identities about randomly chosen matchings. We follow the same approach, and this leads us to different conditions about randomly chosen hypermatchings, which require a fresh analysis in Lemma 6.9.

Without loss of generality, in the following we assume that n is a multiple of k. A vector $\mathbf{a} \in [k]^n$ is called an advice vector if for every $i \in [k]$, $|\{j \colon a_j = i\}| = n/k$. For an advice vector $\mathbf{a} \in [k]^n$, we say that a partial permutation matrix $M \in \{0,1\}^{k\alpha n \times n}$ of a permutation Γ is $\mathbf{a}$-respecting if for every $i \in [k\alpha n]$ and $j \in [n]$, $M_{ij} = 1$ if and only if $a_j = i \pmod k$. Intuitively, $\mathbf{a}$ is the advice vector that tells you which congruence class $\Gamma(j)$ lies in.

For each advice vector $\mathbf{a} \in [k]^n$, each $\mathbf{a}$-respecting partial permutation matrix $M \in \{0,1\}^{k\alpha n \times n}$, distribution $\mathcal{D}$ over $\{-1,1\}^k$, and a fixed Alice's message, the posterior distribution function $p_{M,\mathcal{D},\mathbf{a}} : \{-1,1\}^{k\alpha n} \to [0,1]$ is defined as follows. For each $\mathbf{z} \in \{-1,1\}^{k\alpha n}$, let

$$p_{M,\mathcal{D},\mathbf{a}}(\mathbf{z}) := \Pr_{\substack{\mathbf{x}^* \in \{-1,1\}^n \\ \mathbf{b} \sim \mathcal{D}^{\alpha n}}} [\mathbf{z} = (M\mathbf{x}^*) \odot \mathbf{b} \mid M,\ \mathbf{a},\ \text{Alice's message}] = \mathop{\mathbb{E}}_{\mathbf{x}^* \in A} \mathop{\mathbb{E}}_{\mathbf{b} \sim \mathcal{D}^{\alpha n}} [\mathbf{1}_{\mathbf{z} = (M\mathbf{x}^*) \odot \mathbf{b}}],$$

where $A \subset \{-1,1\}^n$ is the set of Alice's inputs that correspond to the message.

Lemma 6.5. *Let $\mathbf{a} \in [k]^n$, $A \subseteq \{-1,1\}^n$, and $f : \{-1,1\}^n \to \{0,1\}$ be the indicator function of A. Let $k \in \mathbb{N}$ and $\alpha \in (0, 1/100k)$. Let $\mathcal{D}$ be a distribution over $\{-1,1\}^k$ such that $\mathbb{E}_{\mathbf{a} \sim \mathcal{D}}[a_j] = 0$ for all*

$j \in [k]$.

$$\mathop{\mathbb{E}}_{\substack{M \\ M \text{ is } \mathbf{a}\text{-resp.}}} [\|p_{M,\mathcal{D},\mathbf{a}} - U\|_{tvd}^2] \le \frac{2^{2n}}{|A|^2} \sum_{\ell \ge 2}^{k\alpha n} h(\ell) \cdot \sum_{\substack{\mathbf{v} \in \{0,1\}^n \\ |\mathbf{v}| = \ell}} \widehat{f}(\mathbf{v})^2 ,$$

where $U \sim \mathsf{Unif}(\{-1,1\}^{k\alpha n})$ *and for each* $\ell \in [n]$,

$$h(\ell) = \max_{\substack{\mathbf{v}_\ell \in \{0,1\}^n \\ |\mathbf{v}_\ell| = \ell}} \Pr_{\substack{M \\ M \text{ is } \mathbf{a}\text{-resp.}}} \left[\exists \mathbf{s} \in \{0,1\}^{k\alpha n} \backslash \{0^{k\alpha n}\},\ |\mathbf{s}(i)| \neq 1\ \forall i,\ M^\top \mathbf{s} = \mathbf{v}_\ell \right] .$$

Here, for a vector $\mathbf{s} \in \{0,1\}^{k\alpha n}$ *and integer* $i \in [\alpha n]$, $\mathbf{s}(i) \in \{0,1\}^k$ *denotes the* i*th group of* k *coordinates of* $\mathbf{s}$.

Proof. Observe that

$$\|p_{M,\mathcal{D},\mathbf{a}} - U\|_2^2 = \sum_{\mathbf{s} \in \{0,1\}^{k\alpha n}} \left(\widehat{p}_{M,\mathcal{D},\mathbf{a}}(\mathbf{s}) - \widehat{U}(\mathbf{s}) \right)^2 = \sum_{\mathbf{s} \in \{0,1\}^{k\alpha n} \backslash \{0^{k\alpha n}\}} \widehat{p}_{M,\mathcal{D},\mathbf{a}}(\mathbf{s})^2 .$$

Now by the Cauchy–Schwarz inequality, we have that

$$\begin{aligned} \mathop{\mathbb{E}}_{\substack{M \\ M \text{ is } \mathbf{a}\text{-resp.}}} \left[\|p_{M,\mathcal{D},\mathbf{a}} - U\|_{tvd}^2 \right] &\le 2^{2k\alpha n} \mathop{\mathbb{E}}_{\substack{M \\ M \text{ is } \mathbf{a}\text{-resp.}}} \left[\|p_{M,\mathcal{D},\mathbf{a}} - U\|_2^2 \right] \\ &= 2^{2k\alpha n} \mathop{\mathbb{E}}_{\substack{M \\ M \text{ is } \mathbf{a}\text{-resp.}}} \left[\sum_{\mathbf{s} \in \{0,1\}^{k\alpha n} \backslash \{0^{k\alpha n}\}} \widehat{p_{M,\mathcal{D},\mathbf{a}}}(\mathbf{s})^2 \right] . \end{aligned} \tag{6.6}$$

The following claim shows that the Fourier coefficients of the posterior distribution $p_{M,\mathcal{D},\mathbf{a}}$ can be bounded from above by a certain Fourier coefficient of the indicator function f. Let's define $\mathrm{GOOD} := \{\mathbf{s} \in \{0,1\}^{k\alpha n} \mid |\mathbf{s}(i)| \neq 1\ \forall i\}$.

Claim 6.7.

$$\mathop{\mathbb{E}}_{\substack{M \\ M \text{ is } \mathbf{a}\text{-resp.}}} [\|p_{M,\mathcal{D},\mathbf{a}} - U\|_{tvd}^2] \le \frac{2^{2n}}{|A|^2} \sum_{\mathbf{s} \in GOOD \backslash \{0^{k\alpha n}\}} \mathop{\mathbb{E}}_{\substack{M \\ M \text{ is } \mathbf{a}\text{-resp.}}} \left[\widehat{f}(M^\top \mathbf{s})^2 \right] .$$

Proof. Observe that

$$\widehat{p_{M,\mathcal{D},\mathbf{a}}}(\mathbf{s}) = \frac{1}{2^{k\alpha n}} \sum_{\mathbf{z} \in \{-1,1\}^{k\alpha n}} p_{M,\mathcal{D},\mathbf{a}}(\mathbf{z}) \prod_{\substack{i \in [\alpha n], j \in [k] \\ s(i)_j = 1}} z(i)_j .$$

Recall that $p_{M,\mathcal{D},\mathbf{a}}(\mathbf{z}) = \mathbb{E}_{\mathbf{x}^* \in A} \mathbb{E}_{\mathbf{b} \sim \mathcal{D}^{\alpha n}} [\mathbf{1}_{\mathbf{z} = M\mathbf{x}^* \odot \mathbf{b}}]$; the equation becomes

$$= \frac{1}{2^{k\alpha n}} \cdot \mathop{\mathbb{E}}_{\mathbf{x}^* \in A} \left[\prod_{\substack{i \in [\alpha n], j \in [k] \\ s(i)_j = 1}} (M\mathbf{x}^*)_{i,j} \right] \mathop{\mathbb{E}}_{\mathbf{b} \sim \mathcal{D}^{\alpha n}} \left[\prod_{\substack{i \in [\alpha n], j \in [k] \\ s(i)_j = 1}} b(i)_j \right] .$$

Since $\mathbb{E}_{\mathbf{a} \sim \mathcal{D}}[a_j] = 0$ for all $j \in [k]$, the right-most sum is 0 if there exists i such that $|\mathbf{s}(i)| = 1$. This equation becomes

$$\le \frac{1}{2^{k\alpha n}} \cdot \left| \mathop{\mathbb{E}}_{\mathbf{x}^* \in A} \left[\prod_{\substack{i \in [\alpha n], j \in [k] \\ s(i)_j = 1}} (M\mathbf{x}^*)_{i,j} \right] \right| \cdot \mathbf{1}_{\mathbf{s} \in \mathrm{GOOD}} .$$

Note that as each row and column of M has at most 1 non-zero entry, we have

$$= \frac{1}{2^{k\alpha n}} \cdot \left| \mathop{\mathbb{E}}_{\mathbf{x}^* \in A} \left[\prod_{\substack{i \in [n] \\ (M^\top \mathbf{s})_i = 1}} \mathbf{x}^*_i \right] \right| \cdot \mathbf{1}_{\mathbf{s} \in \mathrm{GOOD}}.$$

Now we relate the above quantity to the Fourier coefficients of f. Recall that f is the indicator function of the set A and hence for each $\mathbf{v} \in \{0,1\}^n$, we have

$$\widehat{f}(\mathbf{v}) = \frac{1}{2^n} \sum_{\mathbf{x}^*} f(\mathbf{x}^*) \prod_{i \in [n] : v_i = 1} \mathbf{x}^*_i = \frac{1}{2^n} \sum_{\mathbf{x}^* \in A} \prod_{i \in [n] : v_i = 1} \mathbf{x}^*_i .$$

Thus, the Fourier coefficient of p_M can be bounded as follows:

$$\widehat{p_{M,\mathcal{D},\mathbf{a}}}(\mathbf{s}) \le \frac{1}{2^{\alpha k n}} \cdot \frac{2^n}{|A|} \left| \widehat{f}(M^\top \mathbf{s}) \right| \cdot \mathbf{1}_{\mathbf{s} \in \mathrm{GOOD}} . \tag{6.8}$$

By plugging Equation (6.8) into Equation (6.6), we have the desired bound and complete the proof of Claim 6.7. □

Next, by Claim 6.7, we have

$$\mathop{\mathbb{E}}_{\substack{M \\ M \text{ is a-resp.}}} [\|p_{M,\mathcal{D},\mathbf{a}} - U\|^2_{tvd}] \le \frac{2^{2n}}{|A|^2} \sum_{\mathbf{s} \in \mathrm{GOOD} \setminus \{0^{\alpha k n}\}} \mathop{\mathbb{E}}_{\substack{M \\ M \text{ is a-resp.}}} \left[\widehat{f}(M^\top \mathbf{s})^2 \right] .$$

Since for a fixed M the map $M^\top$ is injective, the right-hand side of the above inequality has the following combinatorial form:

$$= \frac{2^{2n}}{|A|^2} \sum_{\mathbf{v} \in \{0,1\}^n \setminus \{0^n\}} \Pr_{\substack{M \\ M \text{ is a-resp.}}} \left[\exists \mathbf{s} \in \mathrm{GOOD} \setminus \{0^{k\alpha n}\},\ M^\top \mathbf{s} = \mathbf{v} \right] \widehat{f}(\mathbf{v})^2 .$$

By symmetry, the above probability term will be the same for $\mathbf{v}$ and $\mathbf{v}'$ having the same Hamming weight. Recall that

$$h(\ell) = \max_{\substack{\mathbf{v}_\ell \in \{0,1\}^n \\ |\mathbf{v}_\ell| = \ell}} \Pr_{\substack{M \\ M \text{ is a-resp.}}} \left[\exists \mathbf{s} \in \mathrm{GOOD} \setminus \{0^{k\alpha n}\},\ M^\top \mathbf{s} = \mathbf{v}_\ell \right] .$$

This equation becomes

$$\le \frac{2^{2n}}{|A|^2} \sum_{\ell \ge 1}^{n} h(\ell) \cdot \sum_{\substack{\mathbf{v} \in \{0,1\}^n \\ |\mathbf{v}| = \ell}} \widehat{f}(\mathbf{v})^2 .$$

Note that for $\ell = 1$ and every $\ell > \alpha k n$, $h(\ell) = 0$ by definition. Thus, this expression simplifies to the following:

$$= \frac{2^{2n}}{|A|^2} \sum_{\ell \ge 2}^{\alpha k n} h(\ell) \cdot \sum_{\substack{\mathbf{v} \in \{0,1\}^n \\ |\mathbf{v}| = \ell}} \widehat{f}(\mathbf{v})^2 .$$

This completes the proof of Lemma 6.5. □

Now we bound from above the combinatorial quantity $h(\ell)$ from Lemma 6.5.

Lemma 6.9. *For every $0 < \alpha \in (0, 1/100k^2)$ and $\ell \in [k\alpha n]$, we have*

$$h(\ell) = \max_{\substack{\mathbf{v}_\ell \in \{0,1\}^n \\ |\mathbf{v}_\ell| = \ell}} \Pr_{\substack{M \\ M \text{ is } \mathbf{a}\text{-resp.}}} \left[\exists \mathbf{s} \neq 0,\ |\mathbf{s}(i)| \neq 1\ \forall i,\ M^\top \mathbf{s} = \mathbf{v}_\ell\right] \leq \left(\frac{\ell}{n}\right)^{\ell/2} (e^3 \alpha k^5)^{\ell/2}.$$

Proof. By symmetry, without loss of generality we can fix the advice vector $\mathbf{a} = (1^{n/k} 2^{n/k} \dots k^{n/k})$. For non-negative integers $\ell_1, \dots, \ell_k$, we say that $\mathbf{v}_\ell \in \{0,1\}^n$ is an $(\ell_1, \dots, \ell_k)$-vector if for every $i \in [k]$, $\mathbf{v}$ has exactly ℓ_i entries equal to 1 in the ith group of n/k coordinates. For fixed values of ℓ_i, let us define

$$h(\ell_1, \dots, \ell_k) = \Pr_{\substack{M \\ M \text{ is } \mathbf{a}\text{-resp.}}} \left[\exists \mathbf{s} \neq 0,\ |\mathbf{s}(i)| \neq 1\ \forall i,\ M^\top \mathbf{s} \text{ is a } (\ell_1, \dots, \ell_k)\text{-vector}\right].$$

We note that

$$h(\ell) = \max_{\substack{\ell_1, \dots, \ell_k \geq 0 \\ \sum_i \ell_i = \ell}} h(\ell_1, \dots, \ell_k). \tag{6.10}$$

An equivalent way to compute the probability $h(\ell_1, \dots, \ell_k)$ is to fix the matching $M = \{(i, n/k + i, \dots, (k-1)n/k + i) | i \in [\alpha n]\}$ and to let $\mathbf{v}$ be a random $(\ell_1, \dots, \ell_k)$-vector . Then

$$h(\ell_1, \dots, \ell_k) = \Pr_{\substack{\mathbf{v} \\ \mathbf{v} \text{ is } (\ell_1, \dots, \ell_k)}} \left[\exists \mathbf{s} \neq 0,\ |\mathbf{s}(i)| \neq 1\ \forall i,\ M^\top \mathbf{s} = \mathbf{v}\right] = \frac{|U|}{|V|}, \tag{6.11}$$

where $V \subseteq \{0,1\}^n$ is the set of all $(\ell_1, \dots, \ell_k)$-vectors, and $U = \{\mathbf{u} \in V : \exists \mathbf{s} \neq 0,\ |\mathbf{s}(i)| \neq 1\ \forall i,\ M^\top \mathbf{s} = \mathbf{u}\}$. From $\ell_1 + \dots + \ell_k = \ell$, the number of $(\ell_1, \dots, \ell_k)$-vectors is

$$|V| = \prod_{i=1}^{k} \binom{n/k}{\ell_i} \geq \binom{n/k}{\sum_{i=1}^k \ell_i} = \binom{n/k}{\ell} \geq \left(\frac{n}{k\ell}\right)^\ell, \tag{6.12}$$

where the first inequality uses that $n/k \geq k\alpha n \geq \ell$ for $\alpha \leq 1/k^2$.

For a vector $\mathbf{s} \in \{0,1\}^{k\alpha n}$, let $T_\mathbf{s} = \{i : |\mathbf{s}(i)| > 0\}$ be the set of indices of non-zero blocks of $\mathbf{s}$. In order to give an upper bound on the size of U, first we pick a set $T_\mathbf{s}$, and then we choose a vector $\mathbf{u}$ such that $M^\top \mathbf{s} = \mathbf{u}$ for some $\mathbf{s}$ corresponding to the set $T_\mathbf{s}$. Note that since for each $i \in T$, $\mathbf{s}(i) > 0$ and $\mathbf{s}(i) \neq 1$, by the definition of $h(\ell)$, the size of $t = |T| \leq k/2$. For every t, the number of ways to choose $T_\mathbf{s}$ is $\binom{\alpha n}{t}$. For a fixed $T_\mathbf{s}$, it remains to choose the ℓ coordinates of $\mathbf{u}$ among at most kt non-zero coordinates of $\mathbf{s}$. For a vector $\mathbf{s} \in \{0,1\}^{k\alpha n}$, let $T_\mathbf{s} = \{i \in [\alpha n] : |\mathbf{s}(i)| > 0\}$ be the set of indices of non-zero blocks of $\mathbf{s}$. In order to give an upper bound on the size of U, first we pick a set T, and then we choose a vector $\mathbf{u}$ such that $M^\top \mathbf{s} = \mathbf{u}$ for some $\mathbf{s}$ with (i) $|\mathbf{s}(i) \neq 1|$ for all i and (ii) $T_\mathbf{s} = T$. Note that since for each $i \in T$, $\mathbf{s}(i) > 0$ and $|\mathbf{s}(i)| \neq 1$, the size of $t = |T| \leq \ell/2$. For every t, the number of ways to choose T is $\binom{\alpha n}{t}$. For a fixed T, it remains to choose the ℓ coordinates of $\mathbf{u}$ among at most kt non-zero coordinates of $\mathbf{s}$. This gives us the following upper bound on the size of $|U|$:

$$|U| \leq \max_{t \leq \ell/2} \binom{\alpha n}{t} \binom{kt}{\ell}. \tag{6.13}$$

The second term of the upper bound in Equation (6.13) can be bounded from above by

$$\binom{kt}{\ell} \leq \left(\frac{ekt}{\ell}\right)^\ell \leq \left(\frac{ek\ell/2}{\ell}\right)^\ell = \left(\frac{ek}{2}\right)^\ell.$$

Now we'll show that the first term of the upper bound in Equation (6.13) can be bounded from above by $\left(\frac{2ek\alpha n}{\ell}\right)^{\ell/2}$. If $\ell \geq 2\alpha n$, then

$$\binom{\alpha n}{t} \leq 2^{\alpha n} \leq 2^{\ell/2} \leq \left(\frac{2ek\alpha n}{\ell}\right)^{\ell/2},$$

where in the last inequality we use $\ell \leq k\alpha n$. If $\ell < 2\alpha n$, then $t \leq \ell/2 < \alpha n$, and

$$\binom{\alpha n}{t} \leq \left(\frac{e\alpha n}{t}\right)^{t} \leq \left(\frac{2e\alpha n}{\ell}\right)^{\ell/2} < \left(\frac{2ek\alpha n}{\ell}\right)^{\ell/2}.$$

The above implies that

$$|U| \leq \max_{t \leq \min\{\alpha n, \ell/2\}} \binom{\alpha n}{t}\binom{kt}{\ell} \leq \left(\frac{ek}{2}\right)^{\ell} \left(\frac{2ek\alpha n}{\ell}\right)^{\ell/2} \leq \left(\frac{n}{\ell}\right)^{\ell/2} (e^3\alpha k^3)^{\ell/2}. \tag{6.14}$$

Finally, from Equations (6.10) to (6.12) and (6.14),

$$h(\ell) = \max_{\substack{\ell_1,\ldots,\ell_k \geq 0 \\ \sum_i \ell_i = \ell}} h(\ell_1, \ldots, \ell_k) = \frac{|U|}{|V|} \leq \left(\frac{k\ell}{n}\right)^{\ell} \cdot \left(\frac{n}{\ell}\right)^{\ell/2} (e^3\alpha k^3)^{\ell/2} \leq \left(\frac{\ell}{n}\right)^{\ell/2} (e^3\alpha k^5)^{\ell/2}. \qquad \square$$

In Lemma 6.15 below we give the final ingredient needed for the proof of Theorem 6.2. If U is the uniform distribution over $\{-1,1\}^k$, then we show that for every large set $A \subseteq \{0,1\}^n$ of inputs x corresponding to a fixed Alice's message (and a fixed advice $\mathbf{a}$), $\mathbb{E}_{M,\, M \text{ is a-resp.}}[\|p_{M,\mathcal{D},\mathbf{a}} - U\|_{tvd}^2]$ is small.

Lemma 6.15. *For every $k \in \mathbb{N}$ there exists $\alpha_0 > 0$ such that for every $0 < \alpha \leq \alpha_0, \delta \in (0,1)$, and $c \leq \frac{\delta\sqrt{n}}{100\sqrt{\alpha k^5}}$ the following holds for all large enough n. If $\mathcal{D}$ is a distribution over $\{-1,1\}^k$ such that for all $j \in [k]$, $\mathbb{E}_{\mathbf{a}\sim\mathcal{D}}[a_j] = 0$, and $A \subseteq \{-1,1\}^n$ is of size $|A| \geq 2^{n-c}$, then*

$$\mathop{\mathbb{E}}_{\substack{M \\ M \text{ is a-resp.}}} [\|p_{M,\mathcal{D},\mathbf{a}} - U\|_{tvd}^2] \leq \frac{\delta^2}{16},$$

where $U \sim \mathrm{Unif}(\{-1,1\}^{k\alpha n})$.

Proof. Lemma 6.5 and Lemma 6.9 imply that for every A of size $|A| \geq 2^{n-c}$,

$$\mathop{\mathbb{E}}_{\substack{M \\ M \text{ is a-resp.}}} [\|p_{M,\mathcal{D},\mathbf{a}} - U\|_{tvd}^2] \leq \frac{2^{2n}}{|A|^2} \cdot \sum_{\ell \geq 2}^{k\alpha n} \left(\frac{\ell}{n}\right)^{\ell/2} (e^3\alpha k^5)^{\ell/2} \sum_{\substack{\mathbf{v}\in\{0,1\}^n \\ |\mathbf{v}|=\ell}} \widehat{f}(\mathbf{v})^2.$$

For every $\ell \in [4c]$, Lemma 2.11 implies that

$$\frac{2^{2n}}{|A|^2} \sum_{\substack{\mathbf{v}\in\{0,1\}^n \\ |\mathbf{v}|=\ell}} \widehat{f}(\mathbf{v})^2 \leq \left(\frac{4\sqrt{2}c}{\ell}\right)^{\ell}.$$

By the Parseval identity, $\sum_{\mathbf{v}} \widehat{f}(\mathbf{v})^2 \leq 1$. This gives us that

$$\mathop{\mathbb{E}}_{\substack{M \\ M \text{ is a-resp.}}} [\|p_{M,\mathcal{D},\mathbf{a}} - U\|_{tvd}^2] \leq \sum_{\ell \geq 2}^{4c} \left(\frac{\ell}{n}\right)^{\ell/2} (e^3\alpha k^5)^{\ell/2} \cdot \left(\frac{4\sqrt{2}c}{\ell}\right)^{\ell} + \frac{2^{2n}}{|A|^2} \cdot \max_{4c < \ell \leq k\alpha n} \left\{\left(\frac{\ell}{n}\right)^{\ell/2} (e^3\alpha k^5)^{\ell/2}\right\}.$$

Recall that $c \le \frac{\delta\sqrt{n}}{100\sqrt{\alpha k^5}}$. Let $\alpha_0 = \frac{1}{2e^3k^5}$. Then for every $\alpha \le \alpha_0$, the max term on the right-hand side is maximized by $\ell = 4c + 1$ for all large enough n:

$$\begin{aligned}
&\le \sum_{\ell \ge 2}^{4c} \left(\frac{32e^3\alpha k^5 c^2}{n\ell}\right)^{\ell/2} + \left(\frac{8e^3 c\alpha k^5}{n}\right)^{2c} \\
&\le \sum_{\ell \ge 2}^{4c} \left(\frac{\delta^2}{30}\right)^{\ell/2} + \left(\frac{8e^3\delta\sqrt{\alpha}}{100\sqrt{k^3}\sqrt{n}}\right)^{2c} \\
&< \frac{\delta^2}{16}.
\end{aligned}$$

□

We are ready to finish the proof of Theorem 6.2.

Proof of Theorem 6.2. Let us set $\tau = \frac{\delta}{200\sqrt{\alpha k^5}}$, and let α_0 be as set in Lemma 6.15. Suppose that there exists a one-way communication protocol for $(\mathcal{D}_Y, \mathcal{D}_N)$-Advice-RMD that uses $s = \tau\sqrt{n}$ bits of communication and has advantage at least δ. By the triangle inequality, there must exist a protocol with advantage $\delta/2$ and s bits of communication for either the $(\mathcal{D}_Y, \mathcal{D}_{unif})$-Advice-RMD or the $(\mathcal{D}_N, \mathcal{D}_{unif})$-Advice-RMD problem. Without loss of generality, we assume that $(\mathcal{D}_Y, \mathcal{D}_{unif})$-Advice-RMD can be solved with advantage $\delta/2$. Then,

$$\|p_{M,\mathcal{D}_Y,\mathbf{a}} - p_{M,\mathcal{D}_{unif},\mathbf{a}}\|_{tvd} \ge \frac{\delta}{2}.$$

Without loss of generality, we can assume that Alice's protocol is deterministic. In other words, for every $\mathbf{a}$, Alice's s-bit communication protocol partitions the set of $\{-1,1\}^n$ of inputs x into 2^s sets $A_1, \ldots, A_{2^s} \subseteq \{-1,1\}^n$ according to the message sent by Alice. Therefore, at least $(1-\delta/4)$-fraction of inputs $x \in \{-1,1\}^n$ belongs to sets A_i of size $|A_i| \ge \frac{\delta}{4} \cdot 2^{n-s} \ge 2^{n-c}$ for $c = s + 1 - \log\delta$. By Lemma 6.15, for every A_i of size $|A_i| \ge 2^{n-c}$,

$$\|p_{M,\mathcal{D}_Y,\mathbf{a}} - p_{M,\mathcal{D}_{unif},\mathbf{a}}\|_{tvd}|_{\mathbf{x}^*\in A_i} = \mathop{\mathbb{E}}_{\substack{M \\ M \text{is } \mathbf{a}\text{-resp.}}} [\|p_{M,\mathcal{D},\mathbf{a}} - U\|_{tvd}|_{\mathbf{x}^*\in A_i}] \le \delta/4.$$

Finally,

$$\begin{aligned}
\|p_{M,\mathcal{D}_Y,\mathbf{a}} - p_{M,\mathcal{D}_{unif},\mathbf{a}}\|_{tvd} &\le \Pr[x \in A_i : |A_i| < 2^{n-c}] \\
&\quad + \Pr[x \in A_i : |A_i| \ge 2^{n-c}] \cdot \|p_{M,\mathcal{D}_Y,\mathbf{a}} - p_{M,\mathcal{D}_{unif},\mathbf{a}}\|_{tvd}|_{\mathbf{x}^*\in A_i} \\
&\le \delta/4 + (1-\delta/4)\cdot\delta/4 \\
&< \delta/2.
\end{aligned}$$

□

7 HARDNESS OF SIGNAL DETECTION

In this section we extend the hardness result of the SD problems for the special distributions described in Section 6 to the fully general setting, thus proving the following theorem.

Theorem 5.4 (Communication Lower Bound for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD). *For every k, q, every finite set $\mathcal{F}$, every pair of distributions $\mathcal{D}_Y, \mathcal{D}_N \in \Delta(\mathcal{F} \times [q]^k)$ with $\boldsymbol{\mu}(\mathcal{D}_Y) = \boldsymbol{\mu}(\mathcal{D}_N)$ there exists $\alpha_0 > 0$ such that for every $0 < \alpha \le \alpha_0$ and $\delta > 0$ there exists $\tau > 0$ such that the following holds: Every protocol for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD achieving advantage δ on instances of length n requires $\tau\sqrt{n}$ bits of communication.*

The bulk of this section is devoted to proving that for every pair of distributions $\mathcal{D}_Y$ and $\mathcal{D}_N$, we can find a path (a sequence) of intermediate distributions $\mathcal{D}_Y = \mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_L = \mathcal{D}_N$ such that

adjacent pairs in this sequence are indistinguishable by a "basic" argument, where a basic argument is a combination of an indistinguishability result from Theorem 7.4 and a shifting argument.

Our proof comes in the following steps:

(1) For every marginal vector $\boldsymbol{\mu}$, we identify a *canonical* distribution $\mathcal{D}_{\boldsymbol{\mu}}$ that we use as the endpoint of the path. So it suffices to prove that for all $\mathcal{D}$, $\mathcal{D}$ is indistinguishable from $\mathcal{D}_{\boldsymbol{\mu}(\mathcal{D})}$; i.e., there is a path of finite length from $\mathcal{D}$ to $\mathcal{D}_{\boldsymbol{\mu}(\mathcal{D})}$.
(2) We give a combinatorial proof that there is a path of finite length (some function of k) that takes us from an arbitrary distribution to the canonical one.

Putting these ingredients together along with a proof that a "basic step" is indistinguishable gives us the final theorem.

Let $\mathcal{Q} = [q_1] \times \cdots [q_k]$, where $\forall i, q_i \in \mathbb{N}$. We start with the definition of the chain and the canonical distribution. For a distribution $\mathcal{D} \in \Delta(\mathcal{Q})$, its support is the set $\mathrm{supp}(\mathcal{D}) = \{\mathbf{a} \in \mathcal{Q} \mid \mathcal{D}(\mathbf{a}) > 0\}$. For $\mathcal{D} \in \mathcal{Q}$, we define the marginal vector $\boldsymbol{\mu}(\mathcal{D}) = (\mu_{i,\sigma})_{i\in[k],\sigma\in[q_i]}$ as $\mu_{i,\sigma} = \Pr_{\mathbf{a}\sim\mathcal{D}}[a_i = \sigma]$. Next, we consider the following partial order on $\mathcal{Q}$. For vectors $\mathbf{a}, \mathbf{b} \in \mathcal{Q}$ we use the notation $\mathbf{a} \le \mathbf{b}$ if $a_i \le b_i$ for every $i \in [k]$. Further, we use $\mathbf{a} < \mathbf{b}$ if $\mathbf{a} \le \mathbf{b}$ and $\mathbf{a} \ne \mathbf{b}$.

Definition 7.1 (Chain). We refer to a sequence $\mathbf{a}(0) < \mathbf{a}(1) < \cdots < \mathbf{a}(\ell)$, $\mathbf{a}(i) \in \mathcal{Q}$ for every $i \in \{0, \ldots, \ell\}$, as a *chain* of length ℓ. Note that chains in $\mathcal{Q}$ have length at most $\sum_{i=1}^{k}(q_i - 1)$.

Lemma 7.2 (Canonical Distribution). *Given a vector of marginals $\boldsymbol{\mu} = (\mu_{i,\sigma})_{i\in[k],\sigma\in[q_i]}$, there exists a unique distribution $\mathcal{D}$ with matching marginals ($\boldsymbol{\mu}(\mathcal{D}) = \boldsymbol{\mu}$) such that the support of $\mathcal{D}$ is a chain. We call this the* canonical distribution *$\mathcal{D}_{\mu}$ associated with $\boldsymbol{\mu}$.*

Proof. We will prove the proposition by applying induction on $\sum_{i=1}^{k} q_i$. In the base case when $\sum_{i=1}^{k} q_i = k$, there is only one point in the support of the distribution and the claim holds trivially. For $\sum_{i=1}^{k} q_i > k$, define $h = \arg\min_{i\in[k]} \mu_{i,q_i}$ and $\tau = \mu_{h,q_h}$. Let $\tilde{q}_h = q_h - 1$ and $\tilde{q}_i = q_i$, for $i \ne h$. Define a vector of marginals $\tilde{\boldsymbol{\mu}} = (\tilde{\mu}_{i,\sigma})_{i\in[k],\sigma\in[\tilde{q}_i]}$ as follows: $\tilde{\mu}_{i,\sigma} = (\mu_{i,\sigma} - \tau)/(1-\tau)$ if $i \ne h$ and $\sigma = q_i$, and $\tilde{\mu}_{i,\sigma} = \mu_{i,\sigma}/(1-\tau)$ otherwise. By the induction hypothesis, there exists a unique distribution $\tilde{\mathcal{D}}$ supported on a chain such that $\boldsymbol{\mu}(\tilde{\mathcal{D}}) = \tilde{\boldsymbol{\mu}}$. Observe that the distribution $\mathcal{D} = (1-\tau)\tilde{\mathcal{D}} + \tau\{(q_1, \ldots, q_k)\}$ has marginal μ and is supported on a chain. We will now show that $\mathcal{D}$ is the unique distribution with these properties. For a distribution $\mathcal{D} \in \Delta([q_1] \times \cdots \times [q_k])$ and $\mathbf{v} \in [q_1] \times \cdots \times [q_k]$, we define $\mathcal{D}(\mathbf{v}) = \Pr_{\mathbf{c}\sim\mathcal{D}}[\mathbf{c} = \mathbf{v}]$. Note that it suffices to prove that if $\mathcal{D}' \in \Delta([q_1] \times \cdots \times [q_k])$ is supported on a chain and $\boldsymbol{\mu}(\mathcal{D}') = \boldsymbol{\mu}$, then $\mathcal{D}'(q_1, \ldots, q_k) = \tau$. Clearly $\mathcal{D}'(q_1, \ldots, q_k) \le \tau$. Let $\mathbf{u}$ be lexicographically the largest vector smaller than $(q_1, \ldots, q_k)$ in the support of $\mathcal{D}'$. Let r be an index where $\mathbf{u}_r < q_r$. Since $\mathcal{D}'$ is supported on a chain, $\mathcal{D}'(\mathbf{v}) = 0$ for $\mathbf{v} \in [q_1] \times \cdots \times [q_k]$ such that $\mathbf{v}_r = q_r$ and $\mathbf{v} \ne (q_1, \ldots, q_k)$. Hence, $\mu_{r,q_r} = \mathcal{D}'(q_1, \ldots, q_k)$. Since $\tau = \min_{i\in[k]} \mu_{i,q_i}$, we have $\tau \le \mu_{r,q_r} = \mathcal{D}'(q_1, \ldots, q_k)$. □

For $\mathbf{u}, \mathbf{v} \in \mathcal{Q}$, let $\mathbf{u}' = \min\{\mathbf{u}, \mathbf{v}\} \triangleq (\min\{u_1, v_1\}, \ldots, \min\{u_k, v_k\})$ and let $\mathbf{v}' = \max\{\mathbf{u}, \mathbf{v}\} \triangleq (\max\{u_1, v_1\}, \ldots, \max\{u_k, v_k\})$. We say $\mathbf{u}$ and $\mathbf{v}$ are incomparable if $\mathbf{u} \not\le \mathbf{v}$ and $\mathbf{v} \not\le \mathbf{u}$. Note that if $\mathbf{u}$ and $\mathbf{v}$ are incomparable, then $\{\mathbf{u}, \mathbf{v}\}$ and $\{\mathbf{u}', \mathbf{v}'\}$ are disjoint.[12]

Definition 7.3 (Polarization (Update) Operator). Given a distribution $\mathcal{D} \in \Delta(\mathcal{Q})$ and incomparable elements $\mathbf{u}, \mathbf{v} \in \mathcal{Q}$, we define the $(\mathbf{u}, \mathbf{v})$-*polarization* of $\mathcal{D}$, denoted $\mathcal{D}_{\mathbf{u},\mathbf{v}}$, to be the distribution as

[12]To see this, suppose $\mathbf{u} = \mathbf{u}'$, and then we have $u_j = \min\{u_j, v_j\}$ for all $j \in [k]$ and hence $\mathbf{u} \le \mathbf{v}$, which is a contradiction. The same analysis works for the other cases.

given below. Let $\varepsilon = \min\{\mathcal{D}(\mathbf{u}), \mathcal{D}(\mathbf{v})\}$.

$$\mathcal{D}_{\mathbf{u},\mathbf{v}}(\mathbf{b}) = \begin{cases} \mathcal{D}(\mathbf{b}) - \varepsilon & , \mathbf{b} \in \{\mathbf{u}, \mathbf{v}\} \\ \mathcal{D}(\mathbf{b}) + \varepsilon & , \mathbf{b} \in \{\mathbf{u}', \mathbf{v}'\} \\ \mathcal{D}(\mathbf{b}) & , \text{otherwise.} \end{cases}$$

We refer to $\varepsilon(\mathcal{D}, \mathbf{u}, \mathbf{v}) = \min\{\mathcal{D}(\mathbf{u}), \mathcal{D}(\mathbf{v})\}$ as the polarization amount.

It can be verified that the polarization operator preserves the marginals, i.e., $\boldsymbol{\mu}(\mathcal{D}) = \boldsymbol{\mu}(\mathcal{D}_{\mathbf{u},\mathbf{v}})$. Note also that this operator is non-trivial, i.e., $\mathcal{D}_{\mathbf{u},\mathbf{v}} = \mathcal{D}$, if $\{\mathbf{u}, \mathbf{v}\} \nsubseteq \mathrm{supp}(\mathcal{D})$.

Theorem 7.4 (Indistinguishability of the Polarization Step). *Let $n, k, q \in \mathbb{N}$, $\alpha \in (0, 1)$, where k, q, α are constants with respect to n and αn is an integer less than n/k. For a distribution $\mathcal{D} \in \Delta([q]^k)$, incomparable vectors $\mathbf{u}, \mathbf{v} \in [q]^k$, and $\delta > 0$, there exists $\tau > 0$ such that every protocol for $(\mathcal{D}, \mathcal{D}_{\mathbf{u},\mathbf{v}})$-SD achieving advantage δ requires $\tau\sqrt{n}$ bits of communication.*

We defer the proof of this theorem to Section 8.2 and focus instead on the number of steps.

7.1 Finite Upper Bound on the Number of Polarization Steps

In this section we prove that there is a finite upper bound on the number of polarization steps needed to move from a distribution $\mathcal{D} \in \Delta(\mathcal{Q})$ to the canonical distribution with marginal $\boldsymbol{\mu}(\mathcal{D})$, i.e., $\mathcal{D}_{\boldsymbol{\mu}(\mathcal{D})}$. Together with the indistinguishability result from Theorem 7.4 this allows us to complete the proof of Theorem 5.4 by going from $\mathcal{D}_Y$ to $\mathcal{D}_{\boldsymbol{\mu}(\mathcal{D}_Y)} = \mathcal{D}_{\boldsymbol{\mu}(\mathcal{D}_N)}$ and then to $\mathcal{D}_N$ by using the triangle inequality for indistinguishability.

In this section we extend our considerations to functions $A : \mathcal{Q} \to \mathbb{R}^{\geq 0}$. Let $\mathcal{F}(\mathcal{Q}) = \{A : \mathcal{Q} \to \mathbb{R}^{\geq 0}\}$. For $A \in \mathcal{F}(\mathcal{Q})$ and $i \in [k]$, let $\mu_0(A) = \sum_{\mathbf{a}\in\mathcal{Q}} A(\mathbf{a})$. Note $\Delta(\mathcal{Q}) \subseteq \mathcal{F}(\mathcal{Q})$ and $A \in \Delta(\mathcal{Q})$ if and only if $A \in \mathcal{F}(\mathcal{Q})$ and $\mu_0(A) = \sum_{\mathbf{a}\in\mathcal{Q}} A(\mathbf{a}) = 1$. We extend the definition of marginals, support, canonical distribution, and polarization operators to $\mathcal{F}(\mathcal{Q})$. In particular, we let $\boldsymbol{\mu}(A) = (\mu_0, (\mu_{i,\sigma})_{i\in[k],\sigma\in[q_i]})$, where $\mu_{i,\sigma} = \sum_{\mathbf{a}\in\mathcal{Q}:\mathbf{a}_i=\sigma} A(\mathbf{a})$. We also define canonical function and polarization operators so as to preserve $\boldsymbol{\mu}(A)$. So given arbitrary A, let $\mathcal{D} = \frac{1}{\mu_0(A)} \cdot A$. Note $\mathcal{D} \in \Delta(\mathcal{Q})$. For $\boldsymbol{\mu} = (\mu_0, (\mu_{i,\sigma})_{i\in[k],\sigma\in[q_i]})$, where $\forall i, \sum_{\sigma\in[q_i]} \mu_{i,\sigma} = \mu_0$, we define $A_{\boldsymbol{\mu}} = \mu_0 \cdot \mathcal{D}_{\boldsymbol{\mu}'}$, where $\boldsymbol{\mu}' = (\mu_{i,\sigma}/\mu_0)_{i\in[k],\sigma\in[q_i]}$ is the canonical function associated with $\boldsymbol{\mu}$.

Definition 7.5 (Polarization Length). For distribution $A \in \mathcal{F}(\mathcal{Q})$, where $\mathcal{Q} = [q_1] \times \cdots \times [q_k]$, let $N(A)$ be the smallest t such that there exists a sequence $\mathbf{A} = A_0, A_1, \ldots, A_t$ such that $A_0 = A$, $A_t = A_{\boldsymbol{\mu}(A)}$ is canonical and for every $i \in [t]$ it holds that there exists incomparable $\mathbf{u}_i, \mathbf{v}_i \in \mathrm{supp}(A_{i-1})$ such that $A_i = (A_{i-1})_{\mathbf{u}_i,\mathbf{v}_i}$. If no such finite sequence exists, then let $N(A)$ be infinite. Let $N(k, q_1, \ldots, q_k) = \sup_{A\in\mathcal{F}(\mathcal{Q})}\{N(A)\}$, and $\tilde{N}(Q) = \max_{k,q_1,\ldots,q_k \mid \sum_i q_i = Q} N(k, q_1, \ldots, q_k)$. Again, if $N(A) = \infty$ for some A or if no finite upper bound exists, $\tilde{N}(Q)$ is defined to be ∞.

Note that if $\mathcal{D} \in \Delta(\mathcal{Q})$, so is every element in the sequence, so the polarization length bound below applies also to distributions. Our main lemma in this subsection is the following:

Lemma 7.6 (A Finite Upper Bound on $\tilde{N}(Q)$). *$\tilde{N}(Q)$ is finite for every finite Q. Specifically, $\tilde{N}(Q) \leq (Q^2 + 3)\tilde{N}(Q - 1)$. Consequently, for every $k, q_1, \ldots, q_k$, $N(q_1, \ldots, q_k)$ is finite as well.*

We prove Lemma 7.6 constructively in the following four steps.

Step 1: The algorithm Polarize. Let us start with some notations. For $A \in \mathcal{F}([q_1] \times \cdots \times [q_k])$ we let $A|_{x_\ell=q_\ell}$ denote the function A restricted to the domain $[q_1] \times \cdots \times [q_{\ell-1}] \times \{q_\ell\} \times [q_{\ell+1}] \times \cdots \times [q_k]$. Note that $A|_{x_\ell=q_\ell}$ is effectively a $(k-1)$-dimensional function. We also define $A|_{x_\ell<q_\ell}$ as the restriction of A to the domain $[q_1] \times \cdots \times [q_{\ell-1}] \times [q_\ell - 1] \times [q_{\ell+1}] \times \cdots \times [q_k]$.

The goal of the rest of the proof is to show that Algorithm 2 terminates after a finite number of steps and outputs $A_{\boldsymbol{\mu}(A)}$.

ALGORITHM 2: Polarize(·)

Input: $A \in \mathcal{F}([q_1] \times \cdots \times [q_k])$.

if k=1 OR $\nexists i : q_i \geq 2$ **then**
  Output: A.
WLOG, let $q_k \geq 2$.
$t \leftarrow 0$; $Q^- \leftarrow \sum_{i=1}^{k}(q_i - 1) - 1$; $Q^+ \leftarrow \sum_{i=1}^{k-1}(q_i - 1)$
$(A_0)|_{x_k < q_k} \leftarrow$ Polarize$(A|_{x_k < q_k})$; $(A_0)|_{x_k = q_k} \leftarrow$ Polarize$(A|_{x_k = q_k})$
Let $(1)^k = \mathbf{a}_t(0) < \cdots < \mathbf{a}_t(Q^-) = (q_1, \ldots, q_{k-1}, q_k - 1)$ be a chain supporting $(A_t)|_{x_k < q_k}$.
Let $((1)^{k-1}, q_k) = \mathbf{b}_t(0) < \cdots < \mathbf{b}_t(Q^+) = (q_1, \ldots, q_k)$ be a chain supporting $(A_t)|_{x_k = q_k}$.
while $\exists (i, j)$ with $j < Q^+$ s.t. $\max\{\mathbf{a}_t(i), \mathbf{b}_t(j)\} = (q_1, \ldots, q_k)$ and $A_t(\mathbf{a}_t(i)), A_t(\mathbf{b}_t(j)) > 0$ **do**
  Let (i_t, j_t) be the lexicographically smallest such pair (i, j).
  $B_t \leftarrow (A_t)_{\mathbf{a}_t(i_t), \mathbf{b}_t(j_t)}$.
  $(A_{t+1})|_{x_k < q_k} \leftarrow$ Polarize$(B_t|_{x_k < q_k})$; $(A_{t+1})|_{x_k = q_k} \leftarrow (B_t)|_{x_k = q_k}$.
  $t \leftarrow t + 1$.
  Let $(1)^k = \mathbf{a}_t(0) < \cdots < \mathbf{a}_t(Q^-) = (q_1, \ldots, q_k - 1)$ be a chain supporting $(A_t)|_{x_k < q_k}$.
  Let $((1)^{k-1}, q_k) = \mathbf{b}_t(0) < \cdots < \mathbf{b}_t(Q^+) = (q_1, \ldots, q_k)$ be a chain supporting $(A_t)|_{x_k = q_k}$.
Let $\ell \in [k]$ be such that for every $\mathbf{a} \in [q_1] \times \cdots \times [q_k] \setminus \{(q_1, \ldots, q_k)\}$ we have $A_t(\mathbf{a}) > 0 \Rightarrow a_\ell < q_\ell$.
$(A_{t+1})|_{x_\ell < q_\ell} \leftarrow$ Polarize$(A_t)|_{x_\ell < q_\ell}$; $(A_{t+1})|_{x_\ell = q_\ell} \leftarrow (A_t)|_{x_\ell = q_\ell}$.
Output: A_{t+1}.

Step 2: Correctness assuming Polarize *terminates.*

Claim 7.7 (Correctness Condition of Polarize). *For every $A \in \mathcal{F}([q_1] \times \cdots \times [q_k])$, if* Polarize *terminates, then* Polarize$(A) = A_{\boldsymbol{\mu}(A)}$. *In particular,* Polarize(A) *has the same marginals as A and is supported on a chain.*

Proof. First, by the definition of the polarization operator (Definition 7.3), the marginals of A_t are the same for every t. So in the rest of the proof, we focus on inductively showing that if Polarize terminates, then Polarize(A) is supported on a chain.

The base case where $k = 1$ is trivially supported on a chain as desired.

When $k > 1$, note that when the algorithm enters the Clean-up stage, if we let m and n denote the largest indices such that $A_t(\mathbf{a}_t(m)), A_t(\mathbf{b}_t(n)) > 0$ and $A_t(\mathbf{b}_t(n)) \neq (q_1, \ldots, q_k)$, then the condition that $\max\{\mathbf{a}_t(m), \mathbf{b}_t(n)\} \neq (q_1, \ldots, q_k)$ implies that there is a coordinate ℓ such that $\mathbf{a}_t(m)_\ell < q_\ell$ and $\mathbf{b}_t(n)_\ell < q_\ell$. Since every $\mathbf{c}$ such that $A_t(\mathbf{c}) > 0$ and $c_k < q_k$ satisfies $\mathbf{c} \leq \mathbf{a}_t(m)$, we have that $A_t(\mathbf{c}) > 0$ implies $c_\ell < q_\ell$. Similarly, for every $\mathbf{c} \neq (q_1, \ldots, q_k)$ such that $c_k = q_k$, we have that $A_t(\mathbf{c}) > 0$ implies $c_\ell < q_\ell$. We conclude that A_t is supported on $\{(q_1, \ldots, q_k)\} \cup \{\mathbf{c} \mid c_\ell < q_\ell\}$. Thus, by the induction hypothesis, after polarizing $(A_t)|_{x_\ell < q_\ell}$ and leaving $(A_t)|_{x_\ell = q_\ell}$ unchanged, we get that the resulting function A_{t+1} is supported on a chain as desired and complete the induction. We conclude that if Polarize terminates, we have Polarize$(A) = A_{\boldsymbol{\mu}(A)}$. □

Step 3: Invariant in Polarize. Now, in the rest of the proof of Lemma 7.6, the goal is to show that for every input A, the number of iterations of the while loop in Algorithm 2 is finite. The key claim (Claim 7.11) here asserts that the sequence of pairs (i_t, j_t) is monotonically increasing in lexicographic order. Once we establish this claim, it follows that there are at most $Q^- \cdot Q^+$ iterations of the while loop and so $\tilde{N}(Q) \leq (Q^2 + 3)\tilde{N}(Q - 1)$, proving Lemma 7.6. Before proving Claim 7.11, we establish the following properties that remain invariant after every iteration of the while loop.

Claim 7.8. *For every $t \geq 0$, we have that $(A_t)|_{x_k = q_k}$ and $(A_t)|_{x_k < q_k}$ are both supported on chains.*

Proof. For $(A_t)|_{x_k<q_k}$, the claim follows from the correctness of the recursive call to Polarize. For $(A_t)|_{x_k=q_k}$, we claim by induction on t that the supporting chain $\mathbf{b}_t(0) < \cdots < \mathbf{b}_t(Q^+)$ never changes (with t). To see this, note that $\mathbf{b}_t(k-1) = (q_1, \ldots, a_k)$ is the only point in the support of $(A_t)|_{x_k=q_k}$ that increases in value, and this is already in the supporting chain. Thus, $\mathbf{b}_t(0) < \cdots < \mathbf{b}_t(Q^+)$ continues to be a supporting chain for $(A_{t+1})|_{x_k=q_k}$. □

For $\mathbf{c} \in [q_1] \times \cdots \times [q_k]$, we say that a function $A : [q_1] \times \cdots \times [q_k] \to \mathbb{R}^{\geq 0}$ is $\mathbf{c}$-*respecting* if for every $\mathbf{c}'$ such that $A(\mathbf{c}') > 0$, we have $\mathbf{c}' \geq \mathbf{c}$ or $\mathbf{c}' \leq \mathbf{c}$. We say that A is $\mathbf{c}$-*downward-respecting* if A is $\mathbf{c}$-respecting and the points in the support of A above $\mathbf{c}$ form a partial chain; specifically, if $\mathbf{u}, \mathbf{v} > \mathbf{c}$ have $A(\mathbf{u}), A(\mathbf{v}) > 0$, then either $\mathbf{u} \geq \mathbf{v}$ or $\mathbf{v} \geq \mathbf{u}$.

Note that if A is supported on a chain, then A is $\mathbf{c}$-respecting for every point $\mathbf{c}$ in the chain. Conversely, if A is supported on a chain and A is $\mathbf{c}$-respecting, then A is supported on a chain that includes $\mathbf{c}$.

Claim 7.9. *Let A be a $\mathbf{c}$-respecting function and let $\tilde{A}$ be obtained from A by a finite sequence of polarization updates, as in Definition 7.3. Then $\tilde{A}$ is also $\mathbf{c}$-respecting. Furthermore, if A is $\mathbf{c}$-downward-respecting and $\mathbf{w} > \mathbf{c}$, then $\tilde{A}$ is also $\mathbf{c}$-downward-respecting and $A(\mathbf{w}) = \tilde{A}(\mathbf{w})$.*

Proof. Note that it suffices to prove the claim for a single update by a polarization operator since the rest follows by induction. So let $\tilde{A} = A_{\mathbf{u},\mathbf{v}}$ for incomparable $\mathbf{u}, \mathbf{v} \in \mathrm{supp}(A)$. Since A is $\mathbf{c}$-respecting and $\mathbf{u}, \mathbf{v}$ are incomparable, either $\mathbf{u} \leq \mathbf{c}, \mathbf{v} \leq \mathbf{c}$ or $\mathbf{u} \geq \mathbf{c}, \mathbf{v} \geq \mathbf{c}$. Suppose the former is true, then $\max\{\mathbf{u}, \mathbf{v}\} \leq \mathbf{c}$ and $\min\{\mathbf{u}, \mathbf{v}\} \leq \mathbf{c}$, and hence, $\tilde{A}$ is $\mathbf{c}$-respecting. Similarly, in the case when $\mathbf{u} \geq \mathbf{c}, \mathbf{v} \geq \mathbf{c}$, we can show that $\tilde{A}$ is $\mathbf{c}$-respecting. The furthermore part follows by noticing that for $\mathbf{u}$ and $\mathbf{v}$ to be incomparable if A is $\mathbf{c}$-downward-respecting and $A(\mathbf{u}), A(\mathbf{v}) > 0$, then $\mathbf{u}, \mathbf{v} \leq \mathbf{c}$, and so the update changes A only at points below $\mathbf{c}$. □

The following claim asserts that in every iteration of the while loop, by the lexicographically minimal choice of (i_t, j_t), there exists a coordinate $h \in [k-1]$ such that every vector $c < a_t(i_t)$ in the support of A_t, B_t, or A_{t+1} has $c_h < q_h$, and every vector $c \neq (q_1, \ldots, q_k)$ in the support of $(A_t)|_{x_k=q_k}$ has $c_h < q_h$.

Claim 7.10. *For every $t \geq 0$, $\exists h \in [k-1]$ such that $\forall \mathbf{c} \in [q_1] \times \cdots \times [q_k]$, if $\mathbf{c} \in \mathit{supp}(A_t) \cup \mathit{supp}(B_t) \cup \mathit{supp}(A_{t+1})$, then the following hold:*

— *If $\mathbf{c} < \mathbf{a}_t(i_t)$, then $c_h < q_h$.*
— *If $c_k = q_k$ and $\mathbf{c} \neq (q_1, \ldots, q_k)$, then $c_h < q_h$.*

Proof. Since (i_t, j_t) is lexicographically the smallest incomparable pair in the support of A_t, for $i < i_t$, $j < Q^+$, and $A_t(\mathbf{a}(i)), A_t(\mathbf{b}(j)) > 0$, we have $\max\{\mathbf{a}(i), \mathbf{b}(j)\} \neq (q_1, \ldots, q_k)$. Let m be the largest index smaller than i_t such that $A_t(\mathbf{a}_t(m)) > 0$. Similarly, let $n < Q^+$ be the largest index such that $A_t(\mathbf{b}_t(n)) > 0$. Then the fact that $\max\{\mathbf{a}_t(m), \mathbf{b}_t(n)\} \neq (q_1, \ldots, q_k)$ implies that there exists $h \in [k-1]$ such that $\mathbf{a}_t(m)_h < q_h$ and $\mathbf{b}_t(n)_h < q_h$. Now, using the fact (from Claim 7.8) that $(A_t)|_{x_k<q_k}$ is supported on a chain, we conclude that for every $\mathbf{c} < \mathbf{a}_t(i_t)$, $A_t(\mathbf{c}) > 0$ implies that $\mathbf{c} \leq \mathbf{a}_t(m)$ and hence, $c_h < q_h$. Similarly, for every vector $\mathbf{c} \neq (q_1, \ldots, q_k)$ in the support of $(A_t)|_{x_k=q_k}$, by the maximality of n, we have $c_h < q_h$.

We now assert that the same holds for B_t. First, recall that since $B_t = (A_t)_{\mathbf{a}_t(i_t), \mathbf{b}_t(j_t)}$, we have that $\mathrm{supp}(B_t) \subset \mathrm{supp}(A_t) \cup \{(q_1, \ldots, q_k), \min\{\mathbf{a}_t(i_t), \mathbf{b}_t(j_t)\}\}$. Next, note that the only point (other than $(q_1, \ldots, q_k)$) where B_t is larger than A_t is $\min\{\mathbf{a}_t(i_t), \mathbf{b}_t(j_t)\}$. It suffices to show that $\min\{\mathbf{a}_t(i_t), \mathbf{b}_t(j_t)\}_h < q_h$. We have $\min\{\mathbf{a}_t(i_t), \mathbf{b}_t(j_t)\} \leq \mathbf{b}_t(j_t) \leq \mathbf{b}_t(n)$ and hence $\min\{\mathbf{a}_t(i_t), \mathbf{b}_t(j_t)\}_h < q_h$.

Finally, we assert that the same holds also for A_{t+1}. Since $A_{t+1}|_{x_k=q_k} = B_t|_{x_k=q_k}$, the second item in the claim follows trivially. To prove the first item, let us consider $\mathbf{a}' \in [q_1] \times \cdots \times [q_k]$ defined

as follows: $\mathbf{a}'_h = q_h - 1$ and $\mathbf{a}'_r = \mathbf{a}_t(i_t)_r$ for $r \neq h$. Note that $B_t|_{x_k<q_k}$ is $\mathbf{a}_t(i_t)$-respecting since potentially the only new point in its support (compared to $A_t|_{x_k<q_k}$) is $\min\{\mathbf{a}_t(i_t), \mathbf{b}_t(j_t)\} \leq \mathbf{a}_t(i_t)$. From the previous paragraph we also have that if $B_t(\mathbf{c}) > 0$ and $\mathbf{c} < \mathbf{a}_t(i_t)$, then $c_h < q_h$ and hence, $\mathbf{c} \leq \mathbf{a}'$. On the other hand, if $B_t(\mathbf{c}) > 0$ and $\mathbf{c} \geq \mathbf{a}_t(i_t)$, then $\mathbf{c} \geq a'$. Therefore, $B_t|_{x_k<q_k}$ is $\mathbf{a}'$-respecting. By applying Claim 7.9, we conclude that $(A_{t+1})|_{x_k<q_k}$ is also $\mathbf{a}'$-respecting. It follows that if $\mathbf{c} < \mathbf{a}(i_t)$ and $A_{t+1}(\mathbf{c}) > 0$, then $\mathbf{c} \leq \mathbf{a}'$ and so $c_h < q_h$. □

Step 4: Proof of Lemma 7.6. The following claim establishes that the while loop in the Polarize algorithm terminates after a finite number of iterations.

Claim 7.11. *For every $t \geq 0$, $(i_t, j_t) < (i_{t+1}, j_{t+1})$ in lexicographic ordering.*

Proof. Consider the chain $\mathbf{a}_{t+1}(0) < \cdots < \mathbf{a}_{t+1}(Q^-)$ supporting $A_{t+1}|_{x_k<q_k}$. Note that for $i \geq i_t$, $A_{t+1}|_{x_k<q_k}$ is $\mathbf{a}_t(i)$-respecting (since $A_t|_{x_k<q_k}$ and $B_t|_{x_k<q_k}$ were also so). In particular, $A_t|_{x_k<q_k}$ is $\mathbf{a}_t(i)$-respecting because it is supported on a chain containing $a_t(i)$. Next $B_t|_{x_k<q_k}$ is $\mathbf{a}_t(i)$-respecting since potentially the only new point in its support is $\min\{\mathbf{a}_t(i_t), \mathbf{b}_t(j_t)\} \leq \mathbf{a}_t(i)$. Finally, $A_{t+1}|_{x_k<q_k}$ is also $\mathbf{a}_t(i)$-respecting using Claim 7.9. Thus, we can build a chain containing $\mathbf{a}_t(i)$ that supports $A_{t+1}|_{x_k<q_k}$. It follows that we can use $\mathbf{a}_{t+1}(i) = \mathbf{a}_t(i)$ for $i \geq i_t$. Now consider $i < i_t$. We must have $\mathbf{a}_{t+1}(i) < \mathbf{a}_{t+1}(i_t) = \mathbf{a}_t(i_t)$. By Claim 7.10, there exists $h \in [k-1]$ such that for $i < i_t$, $\mathbf{a}_{t+1}(i)_h < q_h$.

We now turn to analyzing (i_{t+1}, j_{t+1}). By definition, $A_{t+1}(\mathbf{a}_{t+1}(i_{t+1})) > 0$ and $A_{t+1}(\mathbf{b}_{t+1}(b_{t+1})) > 0$. First, let us show that $i_t \leq i_{t+1}$. On the contrary, let us assume that $i_{t+1} < i_t$. It follows from the above paragraph that $\mathbf{a}_{t+1}(i_{t+1})_h < q_h$. Also, for every $\mathbf{b}_{t+1}(j)$ with $j < Q^+$ and $A_{t+1}(\mathbf{b}_{t+1}(j)) > 0$, we have $\mathbf{b}_{t+1}(j)_h < q_h$. Therefore, $\max\{\mathbf{a}(i_{t+1}), \mathbf{b}(j_{t+1})\} \neq (q_1, \ldots, q_k)$ (in particular $\max\{\mathbf{a}(i_{t+1}), \mathbf{b}(j_{t+1})\}_h < q_h$), which is a contradiction.

Next, we show that if $i_{t+1} = i_t$, then $j_{t+1} \geq j_t$. By the minimality of (i_t, j_t) in the tth round, for $j < j_t$ such that $A_t(b_t(j)) > 0$, we have $\max\{a_t(i_t), b_t(j)\} \neq (q_1, \ldots, q_k)$. Since $i_{t+1} = i_t$, $a_{t+1}(i_{t+1}) = a_{t+1}(i_t) = a_t(i_t)$. We already noted in the proof of Claim 7.8 that $\mathbf{b}_t(0) < \cdots < \mathbf{b}_t(Q^+)$ is also a supporting chain for $(A_{t+1})|_{x_k=q_k}$. The only point where the function $A_{t+1}|_{x_k=q_k}$ has greater value than $A_t|_{x_k=q_k}$ is $(q_1, \ldots, q_k)$. Therefore, for $j < j_t$ such that $A_{t+1}(b_{t+1}(j)) > 0$, we have $\max\{a_{t+1}(i_{t+1}), b_{t+1}(j)\} \neq (q_1, \ldots, q_k)$ and hence, $j_{t+1} \geq j_t$.

So far, we have established that $(i_{t+1}, j_{t+1}) \geq (i_t, j_t)$ in lexicographic ordering. Finally, we will show that $(i_{t+1}, j_{t+1}) \neq (i_t, j_t)$ by proving that at least one of $A_{t+1}(\mathbf{a}_{t+1}(i_t))$ and $A_{t+1}(\mathbf{b}_{t+1}(j_t))$ is zero. The polarization update ensures that at least one of $B_t(\mathbf{a}_t(i_t))$ and $B_t(\mathbf{b}_t(j_t))$ is zero. If $B_t(\mathbf{b}_t(j_t)) = 0$, then by definition, we have $A_{t+1}(\mathbf{b}_{t+1}(j_t)) = A_{t+1}(\mathbf{b}_t(j_t)) = 0$. Finally, to handle the case $B_t(\mathbf{a}_t(i_t)) = 0$, let us again define $\mathbf{a}'$ as: $\mathbf{a}'_h = q_h - 1$ and $\mathbf{a}'_r = \mathbf{a}_t(i_t)_r$ for $r \neq h$, where h is as given by Claim 7.10. We assert that $B_t|_{x_k<q_k}$ is $\mathbf{a}'$-downward-respecting. As shown in the proof of Claim 7.10, we have that $B_t|_{x_k<q_k}$ is $\mathbf{a}'$-respecting. The support of $B_t|_{x_k<q_k}$ is contained in $\{\mathbf{a}_t(0), \ldots, \mathbf{a}_t(Q^-)\} \cup \{\min\{\mathbf{a}_t(i_t), \mathbf{b}_t(j_t)\}\}$ and $\min\{\mathbf{a}_t(i_t), \mathbf{b}_t(j_t)\} < \mathbf{a}_t(i_t)$, and by Claim 7.10, $\min\{\mathbf{a}_t(i_t), \mathbf{b}_t(j_t)\} \leq \mathbf{a}'$. It follows that $B_t|_{x_k<q_k}$ is $\mathbf{a}'$-downward-respecting. Finally, by the furthermore part of Claim 7.9 applied to $B_t|_{x_k<q_k}$ and $\mathbf{w} = \mathbf{a}_t(i_t)$, we get that $A_{t+1}(\mathbf{a}_{t+1}(i_t)) = A_{t+1}(\mathbf{a}_t(i_t)) = B_t(\mathbf{a}_t(i_t)) = 0$. It follows that $(i_{t+1}, j_{t+1}) \neq (i_t, j_t)$. □

Proof of Lemma 7.6. By Claim 7.7, we know that if Algorithm 2 terminates, then we have $\text{Polarize}(A) = A_{\boldsymbol{\mu}(A)}$. Hence, the maximum number of polarization updates used in Polarize (on input from $\mathcal{F}([q_1] \times \cdots \times [q_k])$) serves as an upper bound for $\tilde{N}(Q)$, for $Q = \sum_{i=1}^{k} q_k$. By Claim 7.11, we know that there are at most Q^2 iterations of the while loop and so $\tilde{N}(Q) \leq (Q^2 + 3)\tilde{N}(Q-1)$ as desired. □

7.2 Reduction from Single Function to a Family of Functions

In this subsection, we prove the following lemma that reduces an SD problem for a single function to an SD problem for a family of functions.

Lemma 7.12. *Suppose there exists $\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N$, $\delta > 0$ with $\boldsymbol{\mu}(\mathcal{D}_Y) = \boldsymbol{\mu}(\mathcal{D}_N)$ and a $c = c(n)$-communication protocol achieving advantage δ solving $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD on instances of length n for every $n \geq n_0$. Then there exist $\mathcal{D}_1, \mathcal{D}_2 \in \Delta([q]^k)$ with $\boldsymbol{\mu}(\mathcal{D}_1) = \boldsymbol{\mu}(\mathcal{D}_2)$, $\delta' > 0$, n_0', and a c-communication protocol achieving advantage δ' solving $(\mathcal{D}_1, \mathcal{D}_2)$-SD on instances of length $n \geq n_0'$ using $O(s)$ bits of communication.*

We prove the lemma by a hybrid argument, where we slowly change the distribution $\mathcal{D}_Y$ to $\mathcal{D}_N$ by considering one function from $\mathcal{F}$ at a time. The crux of the lemma is in showing that two adjacent steps in this sequence are at least as hard as some single-function SD problem, which follows from the following lemma.

Lemma 7.13. *Let $n, k, q \in \mathbb{N}$, $\alpha \in (0,1)$, where k, q, α are constants with respect to n and αn is an integer less than n/k. Let $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$. For every $\varepsilon, \delta \in (0,1]$, there exist $n' = \Omega(n)$ and constants $\alpha', \delta' \in (0,1)$ such that the following holds. For every distribution $\mathcal{D}_Y, \mathcal{D}_N, \mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2 \in \Delta(\mathcal{F} \times [q]^k)$ such that $\mathcal{D}_Y = (1-\varepsilon)\mathcal{D}_0 + \varepsilon\mathcal{D}_1$ and $\mathcal{D}_N = (1-\varepsilon)\mathcal{D}_0 + \varepsilon\mathcal{D}_2$ and for every $c \in \mathbb{N}$, suppose there exists a protocol for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD with parameters n and α using c bits of communication with advantage δ; then there exists a protocol for $(\mathcal{F}, \mathcal{D}_1, \mathcal{D}_2)$-SD with parameters n' and α' using c bits of communication with advantage δ'.*

The proof idea of Lemma 7.13 is very similar to that of Theorem 7.4. We defer the proof to Section 8.2 and turn to showing how Lemma 7.12 follows.

Proof of Lemma 7.12. Let $\mathbf{ALG}(\mathbf{x}^*; M, \mathbf{z})$ be the c-bit protocol for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD achieving advantage δ guaranteed to exist by the theorem statement. Let $\mathcal{F} = \{f_1, \ldots, f_\ell\}$. Since $\boldsymbol{\mu}(\mathcal{D}_Y) = \boldsymbol{\mu}(\mathcal{D}_N)$ for each $i \in [m]$, we have that $\Pr[f = f_i \colon (f, \mathbf{b}) \sim \mathcal{D}_Y] = \Pr[f = f_i \colon (f, \mathbf{b}) \sim \mathcal{D}_N]$. Let us denote this probability by $w^{(i)}$, $w^{(i)} = \Pr[f = f_i \colon (f, \mathbf{b}) \sim \mathcal{D}_Y] = \Pr[f = f_i \colon (f, \mathbf{b}) \sim \mathcal{D}_N]$ for each $i \in [\ell]$. For each $i \in [\ell]$, let $\mathcal{D}_Y^{(i)}$ be the distribution of a random variable $\mathbf{b} \in [q]^k$ that is sampled from $(f, \mathbf{b}) \sim \mathcal{D}_Y$ conditioned on $f = f_i$. Similarly, for each $i \in [\ell]$, let $\mathcal{D}_N^{(i)}$ be the distribution of $\mathbf{b} \in [q]^k$ from $(f, \mathbf{b}) \sim \mathcal{D}_N$ conditioned on $f = f_i$. This way we have that $\mathcal{D}_Y$ and $\mathcal{D}_N$ are the mixture distributions: $\mathcal{D}_Y = \sum_{i \in [\ell]} w^{(i)} \cdot \mathcal{D}_Y^{(i)}$ and $\mathcal{D}_N = \sum_{i \in [\ell]} w^{(i)} \cdot \mathcal{D}_N^{(i)}$.

For every $i \in \{0, \ldots, \ell\}$, we define a distribution $\mathcal{D}^{(i)}$ as the following mixture distribution:

$$\mathcal{D}^{(i)} = \sum_{j \in \{1, \ldots, i\}} w^{(j)} \cdot \mathcal{D}_N^{(j)} + \sum_{j \in \{i+1, \ldots, \ell\}} w^{(j)} \cdot \mathcal{D}_Y^{(j)} .$$

Let $p_i = \Pr[\mathbf{ALG}(\mathbf{x}^*; M, \mathbf{z}) = \mathbf{YES} \colon (f, \mathbf{b}) \sim \mathcal{D}^{(i)}]$ for every $i \in \{0, \ldots, \ell\}$. Observe that $p_0 = \Pr[\mathbf{ALG}(\mathbf{x}^*; M, \mathbf{z}) = \mathbf{YES} \colon (f, \mathbf{b}) \sim \mathcal{D}_Y]$ and $p_\ell = \Pr[\mathbf{ALG}(\mathbf{x}^*; M, \mathbf{z}) = \mathbf{YES} \colon (f, \mathbf{b}) \sim \mathcal{D}_N]$. Since the advantage of $\mathbf{ALG}$ in distinguishing $\mathcal{D}_Y$ and $\mathcal{D}_N$ is at least δ, we have that

$$\delta = |p_0 - p_\ell| = \left| \sum_{i \in \{0, \ldots, \ell-1\}} (p_i - p_{i+1}) \right| \leq \sum_{i \in \{0, \ldots, \ell-1\}} |p_i - p_{i+1}| .$$

Let $\delta' = \delta/\ell$. We have that at least one term of this sum is $|p_i - p_{i+1}| \geq \delta'$. From this we conclude that for some $i \in \{0, \ldots, \ell - 1\}$, $\mathbf{ALG}$ achieves advantage at least δ' for $(\mathcal{F}, \mathcal{D}^{(i)}, \mathcal{D}^{(i+1)})$-SD.

It remains to show that if one can distinguish $\mathcal{D}^{(i)}$ and $\mathcal{D}^{(i+1)}$ that differ only for $(f, \mathbf{b})$ with $f = f_{i+1}$, then one can also distinguish $\mathcal{D}_1 = \mathcal{D}_Y^{(i+1)}$ and $\mathcal{D}_2 = \mathcal{D}_N^{(i+1)}$. Since $\boldsymbol{\mu}(\mathcal{D}_1) = \boldsymbol{\mu}(\mathcal{D}_2)$, this will finish the proof. We show that $\mathcal{D}_1$ and $\mathcal{D}_2$ are distinguishable using Lemma 7.13.

Let us define $\varepsilon = w_{i+1}$, $\mathcal{D} = \frac{1}{1-\varepsilon}\left(\sum_{j\in\{1,\ldots,i\}} w^{(j)} \cdot \mathcal{D}_N^{(j)} + \sum_{j\in\{i+2,\ldots,\ell\}} w^{(j)} \cdot \mathcal{D}_Y^{(j)}\right)$. Now observe that $\mathcal{D}^{(i)} = (1-\varepsilon)\mathcal{D} + \varepsilon\mathcal{D}_1$ and $\mathcal{D}^{(i+1)} = (1-\varepsilon)\mathcal{D} + \varepsilon\mathcal{D}_2$. Now by Lemma 7.13, a protocol that distinguishes $\mathcal{D}^{(i)}$ and $\mathcal{D}^{(i+1)}$ implies a protocol for $(\mathcal{D}_1, \mathcal{D}_2)$-SD with advantage $\delta'' > 0$ and communication complexity $O(s)$. □

7.3 Putting It Together

We now have the ingredients in place to prove Theorem 5.4, which we recall below for convenience.

Theorem 5.4 (Communication Lower Bound for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD). *For every k, q, every finite set $\mathcal{F}$, every pair of distributions $\mathcal{D}_Y, \mathcal{D}_N \in \Delta(\mathcal{F} \times [q]^k)$ with $\boldsymbol{\mu}(\mathcal{D}_Y) = \boldsymbol{\mu}(\mathcal{D}_N)$ there exists $\alpha_0 > 0$ such that for every $0 < \alpha \le \alpha_0$ and $\delta > 0$ there exists $\tau > 0$ such that the following holds: Every protocol for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD achieving advantage δ on instances of length n requires $\tau\sqrt{n}$ bits of communication.*

Proof of Theorem 5.4. Fix $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$ and distributions $\mathcal{D}_Y, \mathcal{D}_N \in \Delta(\mathcal{F} \times [q]^k)$ with $\boldsymbol{\mu} = \boldsymbol{\mu}(\mathcal{D}_Y) = \boldsymbol{\mu}(\mathcal{D}_N)$. Lemma 7.12, applied to $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$, gives us n_0, δ', and distributions $\mathcal{D}'_Y, \mathcal{D}'_N \in \Delta([q]^k)$ with $\boldsymbol{\mu}' = \boldsymbol{\mu}(\mathcal{D}'_Y) = \boldsymbol{\mu}(\mathcal{D}'_N)$ such that any c-communication protocol for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD with advantage δ implies a c-communication protocol for $(\mathcal{D}'_Y, \mathcal{D}'_N)$-SD with advantage δ' for all $n \ge n_0$. Now we'll focus on proving a lower bound for the problem $(\mathcal{D}'_Y, \mathcal{D}'_N)$-SD.

Lemma 7.6, applied to $\mathcal{D}'_Y$, gives us $\mathcal{D}_0 = \mathcal{D}'_Y, \mathcal{D}_1, \ldots, \mathcal{D}_t = \mathcal{D}_{\boldsymbol{\mu}'}$ such that $\mathcal{D}_{i+1} = (\mathcal{D}_i)_{\mathbf{u}(i),\mathbf{v}(i)}$; i.e., $\mathcal{D}_i$ is an update of $\mathcal{D}_i$, with $t \le \tilde{N}(Q) < \infty$, for $Q = \sum_{i=1}^k q_k$. Similarly, Lemma 7.6, applied to $\mathcal{D}'_N$, gives us $\mathcal{D}'_0 = \mathcal{D}'_N, \mathcal{D}'_1, \ldots, \mathcal{D}'_{t'} = \mathcal{D}_{\boldsymbol{\mu}'}$ such that $\mathcal{D}'_{i+1} = (\mathcal{D}'_i)_{\mathbf{u}'(i),\mathbf{v}'(i)}$ with $t' \le \tilde{N}(Q) < \infty$.

Applying Theorem 7.4 with $\delta'' = \delta'/(2\tilde{N}(Q))$ to the pairs $\mathcal{D}_i$ and $\mathcal{D}_{i+1}$, we get that there exists τ_i such that every protocol for $(\mathcal{D}_i, \mathcal{D}_{i+1})$-SD requires $\tau_i\sqrt{n}$ bits of communication to achieve advantage δ''. Similarly applying Theorem 7.4 again with $\delta'' = \delta'/(2\tilde{N}(Q))$ to the pairs $\mathcal{D}'_i$ and $\mathcal{D}'_{i+1}$, we get that there exists τ'_i such that every protocol for $(\mathcal{D}'_i, \mathcal{D}'_{i+1})$-SD requires $\tau'_i\sqrt{n}$ bits of communication to achieve advantage δ''.

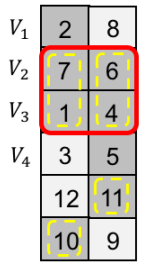
Letting $\tau' = \min\left\{\min_{i\in[t]}\{\tau_i\}, \min_{i\in[t']}\{\tau'_i\}\right\}$, we get, using the triangle inequality for indistinguishability, that every protocol Π' for $(\mathcal{D}'_Y, \mathcal{D}'_N)$-SD achieving advantage $(t+t')\delta'' \le \delta'$ requires $\tau'\sqrt{n}$ bits of communication. Finally, by Lemma 7.12, every protocol Π for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD achieving advantage δ requires $\tau'\sqrt{n}$ bits of communication. □

8 INDISTINGUISHABILITY OF THE POLARIZATION STEP

Recall that in Definition 7.3 we define a polarization operator that polarizes a distribution $\mathcal{D} \in \Delta([q]^k)$ to $\mathcal{D}_{\mathbf{u},\mathbf{v}} \in \Delta([q]^k)$ for every incomparable pair $(\mathbf{u}, \mathbf{v})$. In this section, we show that $(\mathcal{D}, \mathcal{D}_{\mathbf{u},\mathbf{v}})$-SD requires $\Omega(\sqrt{n})$ communication.

Theorem 7.4 (Indistinguishability of the Polarization Step). *Let $n, k, q \in \mathbb{N}$, $\alpha \in (0,1)$, where k, q, α are constants with respect to n and αn is an integer less than n/k. For a distribution $\mathcal{D} \in \Delta([q]^k)$, incomparable vectors $\mathbf{u}, \mathbf{v} \in [q]^k$, and $\delta > 0$, there exists $\tau > 0$ such that every protocol for $(\mathcal{D}, \mathcal{D}_{\mathbf{u},\mathbf{v}})$-SD achieving advantage δ requires $\tau\sqrt{n}$ bits of communication.*

Let $\mathbf{u} \vee \mathbf{v}, \mathbf{u} \wedge \mathbf{v} \in [q]^k$ be given by $u_i \vee v_i = \max\{u_i, v_i\}$ and $u_i \wedge v_i = \min\{u_i, v_i\}$. Let $\mathcal{A}_Y = \mathrm{Unif}(\{\mathbf{u}, \mathbf{v}\})$ and $\mathcal{A}_N = \mathrm{Unif}(\{\mathbf{u} \vee \mathbf{v}, \mathbf{u} \wedge \mathbf{v}\})$. We prove Theorem 7.4 in two steps. First, we use the Boolean hardness in Theorem 6.4 to show in Lemma 8.1 that the hardness holds for the special case $(\mathcal{A}_Y, \mathcal{A}_N)$-SD. Next, we reduce $(\mathcal{A}_Y, \mathcal{A}_N)$-advice-SD to $(\mathcal{D}, \mathcal{D}_{\mathbf{u},\mathbf{v}})$-SD for arbitrary distribution $\mathcal{D} \in \Delta([q]^k)$.



$(\mathbf{u}_1, \mathbf{v}_1) = (1,3);\ (\mathbf{u}_2, \mathbf{v}_2) = (3,2)$

$(\mathbf{x}_R)_1$	$(\mathbf{x}_R)_2$	$(\mathbf{x}_R)_3$	$(\mathbf{x}_R)_4$	$(\mathbf{x}_R)_5$	$(\mathbf{x}_R)_6$	$(\mathbf{x}_R)_7$	$(\mathbf{x}_R)_8$	$(\mathbf{x}_R)_9$	$(\mathbf{x}_R)_{10}$	$(\mathbf{x}_R)_{11}$	$(\mathbf{x}_R)_{12}$
1	3	2	3	1	2	3	1	1	3	2	2

$T_1 = \{1,2,7,10\}$ $X_1 = \{1,7\}$ $W_1 = \{2,10\}$ $S_1 = \{1,7,10\}$

$T_2 = \{4,5,6,11\}$ $X_2 = \{4,6\}$ $W_2 = \{5,11\}$ $S_2 = \{4,6,11\}$

$U = \{2,3\}$

Fig. 3. An example of shared randomness used in Lemma 8.1. Here $n = 12$, $k = 2$, $q = 3$, and $\alpha = 1/3$. The value of $\mathbf{x}_R \in [q]^n$ is listed in a table. Consider $(u_1, v_1) = (1,3)$ and $(u_2, v_2) = (3,2)$. The variables in sets T_1, T_2 are marked grey. The variables corresponding to the set U are circled with red lines and the variables corresponding to sets S_1, S_2 are circled with yellow dashed lines.

8.1 Reduce a Boolean SD Problem to a Non-Boolean SD Problem

In this subsection, we consider a special case of $\mathbf{u}, \mathbf{v} \in [q]^k$ where $u_i \neq v_i$ for every $i \in [k]$. The following key lemma of this subsection establishes the hardness of $(\mathcal{A}_Y, \mathcal{A}_N)$-SD via a reduction from a Boolean SD problem to a non-Boolean version.

Lemma 8.1. *Let $n, k, q \in \mathbb{N}$, $\alpha \in (0,1)$, where k, q, α are constants with respect to n and αn is an integer less than n/k. For $\mathbf{u}, \mathbf{v} \in [q]^k$ satisfying $u_i \neq v_i$ for all $i \in [k]$ and $\delta > 0$, there exists $\tau > 0$ such that every protocol for $(\mathcal{A}_Y, \mathcal{A}_N)$-SD achieving advantage δ requires $\tau\sqrt{n}$ bits of communication.*

We prove Lemma 8.1 by a reduction. For such $\mathbf{u}, \mathbf{v}$, let $\bar{\mathbf{u}}, \bar{\mathbf{v}} \in \{0,1\}^k$ be the Boolean version given by $(\bar{u}_i, \bar{v}_i) = (0,1)$ if $u_i < v_i$ and $(\bar{u}_i, \bar{v}_i) = (1,0)$ if $u_i > v_i$. Let $\bar{\mathcal{A}}_Y = \mathsf{Unif}(\{\bar{\mathbf{u}}, \bar{\mathbf{v}}\})$ and $\bar{\mathcal{A}}_N = \mathsf{Unif}(\{\bar{\mathbf{u}} \vee \bar{\mathbf{v}}, \bar{\mathbf{u}} \wedge \bar{\mathbf{v}}\})$. Note that both $\bar{\mathcal{A}}_Y$ and $\bar{\mathcal{A}}_N$ are distributions on the Boolean domain with uniform marginals. Thus, Theorem 6.4 shows that any protocol for $(\bar{\mathcal{A}}_Y, \bar{\mathcal{A}}_N)$-advice-SD requires $\Omega(\sqrt{n})$ bits of communication. In the rest of this subsection, we reduce $(\bar{\mathcal{A}}_Y, \bar{\mathcal{A}}_N)$-advice-SD to $(\mathcal{A}_Y, \mathcal{A}_N)$-SD.

For every $\bar{n}, k, \bar{\alpha}, q, \delta$, let $n = 2q\bar{n}$ and $\alpha = q^{k-1}2^{-(k+2)}\bar{\alpha}$. Let $\bar{I} = (\bar{\mathbf{x}}, \bar{\Gamma}, \bar{\mathbf{b}}, \bar{M}, \bar{\mathbf{z}}, \bar{\mathbf{a}})$ denote an instance of $(\bar{\mathcal{A}}_Y, \bar{\mathcal{A}}_N)$-advice-SD of length $\bar{n}$ with parameter $\bar{\alpha}$. We show below how Alice and Bob can use their inputs and shared randomness to generate an instance $I = (\mathbf{x}, \Gamma, \mathbf{b}, M, \mathbf{z}, \mathbf{a})$ of $(\mathcal{A}_Y, \mathcal{A}_N)$-advice-SD of length n with parameter α "locally" and "nearly" according to the correct distributions. Namely, we show that with high probability if $\bar{I}$ is a Yes (resp. No) instance of $(\bar{\mathcal{A}}_Y, \bar{\mathcal{A}}_N)$-advice-SD, then I will be a Yes (resp. No) instance of $(\mathcal{A}_Y, \mathcal{A}_N)$-SD.

Step 1: Specify the shared randomness. The common randomness between Alice and Bob is an instance $I_R = (\mathbf{x}_R, \Gamma_R, \mathbf{b}_R, M_R, \mathbf{z}_R, \mathbf{a}_R)$ drawn according to the Yes[13] distribution of $(\mathcal{A}_Y, \mathcal{A}_N)$-advice-SD of length n with parameter α. For $j \in [\alpha n]$, let V_j denote the set of variables in the jth constraint, i.e., $V_j = \{\ell \in [n] \mid \Gamma_R(\ell) \in \{k(j-1)+1, \ldots, k(j-1)+k\}\}$. For $i \in [k]$, let T_i be the set of variables that are in the ith partition and take on values in $\{u_i, v_i\}$, i.e., $T_i = \{j \in [n] \mid a_j = i \,\&\, (\mathbf{x}_R)_i \in \{u_i, v_i\}\}$. Let $U \subseteq [\alpha n]$ be the set of constraints that work on variables in T_i, i.e., $U = \{j \in [\alpha n] \mid V_j \subseteq \cup_i T_i\}$. See Figure 3 for an example.

If $|U| \geq \bar{\alpha}\bar{n}$, we say an error of type (1) has occurred. For $i \in [k]$, let $X_i \subseteq T_i$ be the set of variables that operate on constraints in U, i.e., $X_i = T_i \cap (\cup_{j \in U} V_j)$. Let $W_i \subseteq T_i$ be a set of variables that do not participate in any constraint, i.e., $W_i = T_i \setminus (\cup_{j \in [\alpha n]} V_j)$. Finally, let S_i be any set satisfying

[13]The reduction also works if we used No distribution. However, the mapping between Yes and No instances would get flipped. Namely, if $\bar{I}$ is a Yes (resp. No) instance of $(\bar{\mathcal{A}}_Y, \bar{\mathcal{A}}_N)$-advice-SD, then I will be a No (resp. Yes) instance of $(\mathcal{A}_Y, \mathcal{A}_N)$-SD.

$|S_i| = \bar{n}/k$ with $X_i \subseteq S_i \subseteq X_i \cup W_i$ if such a set exists. If no such set exists, we say an error of type (2) has occurred.

Step 2: Specify the reduction. If there is an error, we simply set $I = I_R$. If no errors have occurred, our reduction will embed $\bar{I}$ into I_R by replacing the constraints in U and the variables in $\cup_i S_i$ as described next. Note that we have to specify variables $(\mathbf{x}, \Gamma, \mathbf{b}, M, \mathbf{z})$. In particular, we want the private inputs to be computed locally. We verify the local property of the reduction in Claim 8.2 and prove the correctness of the reduction in Claim 8.3.

— $\mathbf{x}$: Let $\rho : [\bar{n}] \to \cup_i S_i$ be a bijection satisfying $\bar{a}_j = i \Rightarrow \rho(j) \in S_i$. We now define $\mathbf{x} \in [q]^n$ as follows:

$$x_j = \begin{cases} (\mathbf{x}_R)_j & \text{if } j \notin \cup_{i\in[k]} S_i \\ u_i & j \in S_i \text{ for some } i \in [k] \text{ and } u_i < v_i \text{ and } \bar{x}_j = 0 \\ u_i & j \in S_i \text{ for some } i \in [k] \text{ and } u_i > v_i \text{ and } \bar{x}_j = 1 \\ v_i & j \in S_i \text{ otherwise.} \end{cases}$$

— Γ and M: Let $V = \{V(1), \ldots, V(\bar{n})\}$ with $V(j) < V(j+1)$ be such that $V = \{j \in [n] | \Gamma_R(j) \in \cup_{i\in[k]} S_i\}$. For $j \in [n]$ we let

$$\Gamma(j) = \begin{cases} \Gamma_R(j) & \text{if } j \notin V \\ \rho(\bar{\Gamma}(\bar{j})) & \text{if } j = V(\bar{j}). \end{cases}$$

It may be verified that Γ is a permutation and furthermore the constraints in Γ corresponding to $j \in U$ are derived from constraints of $\bar{I}$. M is then defined as the partial permutation matrix capturing $\Gamma^{-1}(j)$ for $j \in [k\alpha n]$.

— $\mathbf{b}$: Since $\mathbf{b}$ is a hidden variable and won't be given to Alice and Bob, we postpone the specification of $\mathbf{b}$ to the proof of Claim 8.3.

— $\mathbf{z}$: Let $\mathbf{z}(j) = \bar{\mathbf{z}}(V(j))$ if $j \in U$ and $\mathbf{z}(j) = \mathbf{z}_R(j)$ otherwise.

Step 3: Correctness of the reduction assuming no error occurs.

Claim 8.2 (The Reduction Can Be Computed Locally). *Let $\bar{I} = (\bar{\mathbf{x}}, \bar{\Gamma}, \bar{\mathbf{b}}, \bar{M}, \bar{\mathbf{z}}, \bar{\mathbf{a}})$ be an instance of $(\bar{\mathcal{A}}_Y, \bar{\mathcal{A}}_N)$-advice-SD and $I_R = (\mathbf{x}_R, \Gamma_R, \mathbf{b}_R, M_R, \mathbf{z}_R, \mathbf{a}_R)$ be the shared randomness of Alice and Bob. The above reduction satisfies the following local properties:*

— *Alice can compute $\mathbf{x}$ using I_R and $(\bar{\mathbf{x}}, \bar{\mathbf{a}})$.*

— *Bob can compute $(M, \mathbf{z})$ using I_R and $(\bar{M}, \bar{\mathbf{z}}, \bar{\mathbf{a}})$.*

Proof.

— Note that from the construction, it suffices to have $\{S_i\}, \bar{\mathbf{a}}, \bar{\mathbf{x}}$ to compute $\mathbf{x}$. Since $\{S_i\}$ can be obtained from I_R, we conclude that Alice can compute $\mathbf{x}$ using I_R and $(\bar{\mathbf{x}}, \bar{\mathbf{a}})$.

— Note that from the construction, it suffices to have $\{S_i\}, \Gamma_R, \bar{\Gamma}(j)$, where $j \in [k\alpha n]$, to compute M. Since $\bar{\Gamma}(j)$ is encoded in $\bar{M}$ for every $j \le k\alpha n$, and the other information can be obtained from I_R, we know that M can be computed from I_R and $\bar{M}$. Finally, since $\mathbf{z} = \mathbf{z}'$, $\mathbf{z}$ can also be computed from I_R. We conclude that Bob can compute $(M, \mathbf{z})$ using I_R and $(\bar{M}, \bar{\mathbf{z}}, \bar{\mathbf{a}})$. □

Claim 8.3 (The Distribution of I). *Let $\bar{I} = (\bar{\mathbf{x}}, \bar{\Gamma}, \bar{\mathbf{b}}, \bar{M}, \bar{\mathbf{z}}, \bar{\mathbf{a}})$ be an instance drawn from either the Yes or No distribution of $(\bar{\mathcal{A}}_Y, \bar{\mathcal{A}}_N)$-advice-SD and $I_R = (\mathbf{x}_R, \Gamma_R, \mathbf{b}_R, M_R, \mathbf{z}_R, \mathbf{a}_R)$ be a instance drawn from the Yes distribution of $(\mathcal{A}_Y, \mathcal{A}_N)$-advice-SD. Let $I = (\mathbf{x}, \Gamma, \mathbf{b}, M, \mathbf{z}, \mathbf{a})$ be the result of applying the above reduction on $\bar{I}$ and I_R. Then the following hold:*

— *$\mathbf{x} \sim \mathsf{Unif}([q]^n)$.*

— *M is a uniformly random partial permutation matrix as required in item 3 of Definition 6.3.*

— *Suppose there is no error happening in the reduction.*

– *If $\bar{I}$ is a Yes instance, then* $\Pr[\mathbf{z}(j) = 1] = \Pr_{\mathbf{b}(j)\sim\mathcal{A}}[(M\mathbf{x})(j) = \mathbf{b}(j)]$ *for every* $j \in [\alpha n]$.
– *If $\bar{I}$ is a No instance, then* $\Pr[\mathbf{z}(j) = 1] = \Pr_{\mathbf{b}(j)\sim\mathcal{A}'}[(M\mathbf{x})(j) = \mathbf{b}(j)]$ *for every* $j \in [\alpha n]$.

Namely, if $\bar{I}$ is a Yes (resp. No) instance of $(\bar{\mathcal{A}}_Y, \bar{\mathcal{A}}_N)$-advice-SD, then I is a Yes (resp. No) instance of $(\mathcal{A}_Y, \mathcal{A}_N)$-SD.

Proof.

— To prove $\mathbf{x} \sim \mathsf{Unif}([q]^n)$, observe that $\mathbf{x}$ is obtained from $\mathbf{x}_R$ by flipping some of the u_i to v_i (and vice versa). In particular, (i) $\mathbf{x}_R \sim \mathsf{Unif}([q]^n)$ and (ii) the flipping is decided by $\bar{\mathbf{x}}$, which is uniformly sampled from $\{0,1\}^{\bar{n}}$ and is independent to $\mathbf{x}_R$. Note that for a fixed $\mathbf{x}_R$, S_i, and $j \in S_i$, the probability of x_j being set to u_i is the same as being set to v_i. As a result, by symmetry of u_i and v_i, we conclude that $\mathbf{x} \sim \mathsf{Unif}([q]^n)$.
— By the symmetry of the n variables, M is a uniformly random partial permutation matrix as required in item 3 of Definition 6.3.
— Suppose there is no error happening in the reduction. We consider the following two cases: (i) $j \in [\alpha n]\backslash U$ and (ii) $j \in U$.

(i) For each $j \in [\alpha n]\backslash U$, by the construction we have $\mathbf{z}(j) = \mathbf{z}_R(j)$, and hence when fixing $\mathbf{x}_R, M_R$, we have $\Pr[\mathbf{z}(j) = 1] = \Pr[\mathbf{z}_R(j) = 1] = \Pr_{\mathbf{b}_R(j)\sim\mathcal{A}}[(M_R\mathbf{x}_R)(j) = \mathbf{b}_R(j)]$. We set $\mathbf{b}(j) = \mathbf{b}_R(j)$ and note that $\mathbf{b}(j) \sim \mathcal{A}_Y$ (resp. $\mathbf{b}(j) \sim \mathcal{A}_N$) if $\bar{\mathbf{b}}(j) \sim \bar{\mathcal{A}}_Y$ (resp. $\bar{\mathbf{b}}(j) \sim \bar{\mathcal{A}}_N$) for every $j \in U$. Finally, since $j \notin U$, there exists $i \in [k]$ such that $(M_R\mathbf{x}_R(j))_i = (M\mathbf{x}(j))_i \notin \{u_i, v_i\}$ and hence $\Pr_{\mathbf{b}_R(j)\sim\mathcal{A}_Y}[(M_R\mathbf{x}_R)(j) = \mathbf{b}_R(j)] = \Pr[(M\mathbf{x})(j) = \mathbf{b}(j)] = 0$. So we have $\Pr[\mathbf{z}(j) = 1] = \Pr_{\mathbf{b}(j)\sim\mathcal{A}_Y}[(M\mathbf{x})(j) = \mathbf{b}(j)]$ (resp. $\Pr[\mathbf{z}(j) = 1] = \Pr_{\mathbf{b}(j)\sim\mathcal{A}_N}[(M\mathbf{x})(j) = \mathbf{b}(j)]$) if $\bar{I}$ is a Yes (resp. No) instance as desired.

(ii) For each $j \in U$, by construction we have $\mathbf{z}(j) = \bar{\mathbf{z}}(V(j))$. We set

$$\mathbf{b}(j)_i = \begin{cases} u_i & \text{if } u_i < v_i \text{ and } \bar{\mathbf{b}}(V(j))_i = 0 \\ u_i & \text{if } u_i > v_i \text{ and } \bar{\mathbf{b}}(V(j))_i = 1 \\ v_i & \text{otherwise.} \end{cases}$$

First, observe that $\mathbf{z}(j) = 1$ iff $(M\mathbf{x})(j) = \mathbf{b}(j)$. To see this, note that

$$\begin{aligned} \mathbf{z}(j) = 1 &\Leftrightarrow \bar{\mathbf{z}}(V(j)) = 1 \\ &\Leftrightarrow (\bar{M}\bar{\mathbf{x}})(V(j)) = \bar{\mathbf{b}}(V(j)) \,. \end{aligned}$$

For each $i \in [k]$, if $u_i < v_i$ and $\bar{\mathbf{b}}(V(j))_i = (\bar{M}\bar{\mathbf{x}})(V(j))_i = 0$, we have $\mathbf{b}(j)_i = (M\mathbf{x})(j)_i = u_i$. Similarly, for all the other situations we have $\mathbf{b}(j)_i = (M\mathbf{x})(j)$ and hence the equation becomes

$$\Leftrightarrow (M\mathbf{x})(j) = \mathbf{b}(j),$$

as desired.

Next, observe that if $\bar{I}$ is a Yes (resp. No) instance, then $\mathbf{b}(j) \sim \mathcal{A}_Y$ (resp. $\mathbf{b}(j) \sim \mathcal{A}_N$). We analyze the two cases as follows:

– If $\bar{I}$ is a Yes instance, we have $\bar{\mathbf{b}}(V(j)) \sim \bar{\mathcal{A}}_Y = \mathsf{Unif}(\{\bar{\mathbf{u}}, \bar{\mathbf{v}}\})$. Recall that $(\bar{u}_i, \bar{v}_i) = (0,1)$ if $u_i < v_i$ and $(\bar{u}_i, \bar{v}_i) = (1,0)$ otherwise. Now observe that, by the above choice of $\mathbf{b}(j)$, we have $\bar{\mathbf{b}}(V(j)) = \bar{\mathbf{u}}$ iff $\mathbf{b}(j) = \mathbf{u}$ (resp. $\bar{\mathbf{b}}(V(j)) = \bar{\mathbf{v}}$ iff $\mathbf{b}(j) = \mathbf{v}$). Thus, we have $\mathbf{b}(j) \sim \mathcal{A}_Y$, as desired.
– If $\bar{I}$ is a No instance, we have $\bar{\mathbf{b}}(V(j)) \sim \bar{\mathcal{A}}_N = \mathsf{Unif}(\{\bar{\mathbf{u}} \vee \bar{\mathbf{v}}, \bar{\mathbf{u}} \wedge \bar{\mathbf{v}}\})$. Recall that for each $i \in [k]$, $u_i \vee v_i = \max\{u_i, v_i\}$ and $u_i \wedge v_i = \min\{u_i, v_i\}$. Now observe that, by the above choice of $\mathbf{b}(j)$, we have $\bar{\mathbf{b}}(V(j)) = \bar{\mathbf{u}} \vee \bar{\mathbf{v}}$ iff $\mathbf{b}(j) = \mathbf{u} \vee \mathbf{v}$ (resp. $\bar{\mathbf{b}}(V(j)) = \bar{\mathbf{u}} \wedge \bar{\mathbf{v}}$ iff $\mathbf{b}(j) = \mathbf{u} \wedge \mathbf{v}$). Thus, we have $\mathbf{b}(j) \sim \mathcal{A}_N$, as desired.

To sum up, for each $j \in U$, we have $\Pr[\mathbf{z}(j) = 1] = \Pr_{\mathbf{b}(j)\sim\mathcal{A}_Y}[(M\mathbf{x})(j) = \mathbf{b}(j)]$ (resp. $\Pr[\mathbf{z}(j) = 1] = \Pr_{\mathbf{b}(j)\sim\mathcal{A}_N}[(M\mathbf{x})(j) = \mathbf{b}(j)]$) if $\bar{I}$ is a Yes (resp. No) instance, as desired. □

Step 4: An error occurs with low probability.

Claim 8.4. *When n is sufficiently large, the probability of an error happening in the reduction is at most $2^{-\Omega((2/q)^{2k}\alpha n)}$.*

Proof. Recall that for given $\bar{n}, k, \bar{\alpha}, q, \delta$, we let $n = 2q\bar{n}$ and $\alpha = q^{k-1}2^{-(k+2)}\bar{\alpha}$.

Note that U is a sum of αn i.i.d. $\mathrm{Bern}((2/q)^k)$. So by concentration inequality, we have $\Pr[|U| > 2(2/q)^k\alpha n] < 2^{-\Omega((2/q)^{2k}\alpha n)}$. By the choice of parameters, we have $2(2/q)^k\alpha n \le \bar{\alpha}\bar{n}$. Thus, type (1) error happens with probability at most $2^{-\Omega((2/q)^{2k}\alpha n)}$.

Note that by the choice of parameters, we have $|X_i| = |U| \le \bar{n}/k$ and hence type (2) error happens only when $|U| + |W_i| < \bar{n}/k$ for some $i \in [k]$. For each $i \in [k]$, note that $|W_i|$ is a sum of $n/k - \alpha n$ i.i.d. $\mathrm{Bern}(2/q)$. So by concentration inequality, we have $\Pr[|W_i| < (n/k - \alpha n)/q] < 2^{-\Omega((1/q)^2(n/k-\alpha n))}$. By the choice of parameters, we have $(n/k - \alpha n)/q \ge \bar{n}/k$. Thus, type (2) error happens with probability at most $2^{-\Omega((1/q)^2(n/k-\alpha n))} \le 2^{-\Omega((2/q)^{2k}\alpha n)}$. □

Step 5: Proof of Lemma 8.1.

Proof of Lemma 8.1. For every $\bar{n}, k, \bar{\alpha}, q, \delta$, we let $n = 2q\bar{n}$ and $\alpha = q^{k-1}2^{-(k+2)}\bar{\alpha}$. Suppose there is a protocol for $(\mathcal{A}_Y, \mathcal{A}_N)$-SD using $C(n)$ bits of communication and achieving advantage δ. We show how to get a protocol $\bar{\Pi}$ for $(\bar{\mathcal{A}}_Y, \bar{\mathcal{A}}_N)$-advice-SD with parameters $(\bar{n}, \bar{\alpha})$ using $C(n)$ bits of communication and achieving advantage $\delta/2$.

Let $\bar{I} = (\bar{\mathbf{x}}, \bar{\Gamma}, \bar{\mathbf{b}}, \bar{M}, \bar{\mathbf{z}}, \bar{\mathbf{a}})$ be an instance drawn from either the Yes or No distribution of $(\bar{\mathcal{A}}_Y, \bar{\mathcal{A}}_N)$-advice-SD where $(\bar{\mathbf{x}}, \bar{\mathbf{a}})$ is Alice's private input and $(\bar{M}, \bar{\mathbf{z}}, \bar{\mathbf{a}})$ is Bob's private input. The protocol $\bar{\Pi}$ works as follows. Alice and Bob first use their private input and the shared randomness to compute $\mathbf{x}$ and $(M, \mathbf{z})$, respectively. This can be done due to Claim 8.2. Next, Alice and Bob simply invoke the protocol Π on the new instance $\mathbf{x}$ and $(M, \mathbf{z})$ and output the result accordingly.

It is immediate to see that $\bar{\Pi}$ only uses $C(n)$ bits of communication. To show that $\bar{\Pi}$ has advantage at least $\delta/2$, we first show that the joint distribution of $(\mathbf{x}, M, \mathbf{z})$ is the same as that from an instance of $(\mathcal{A}_Y, \mathcal{A}_N)$-SD if there is no error in the reduction. By Claim 8.3, $\mathbf{x} \sim \mathrm{Unif}([q]^n)$ and M follows the distribution as required in item 3 of Definition 6.3.

When there is no error in the reduction and $\bar{I}$ is sampled from the Yes (resp. No) distribution of $(\bar{\mathcal{A}}_Y, \bar{\mathcal{A}}_N)$-advice-SD, Claim 8.3 implies that $\mathbf{z}$ follows the conditional distribution (conditioned on $\mathbf{x}$ and M) of a Yes (resp. No) instance of $(\mathcal{A}_Y, \mathcal{A}_N)$-SD as required in item 5 of Definition 6.3. Next, Claim 8.4 shows that the probability of an error happening in the reduction is at most $\delta/2$. Finally, by triangle inequality, we conclude that $\bar{\Pi}$ has advantage at least $\delta/2$ in solving $(\bar{\mathcal{A}}_Y, \bar{\mathcal{A}}_N)$-advice-SD.

To conclude, by Theorem 6.4, any protocol for $(\bar{\mathcal{A}}_Y, \bar{\mathcal{A}}_N)$-advice-SD with advantage $\delta/2$ requires $\bar{\tau}\sqrt{\bar{n}}$ bits of communication. Thus, we have $C(n) \ge \bar{\tau}\sqrt{\bar{n}} \ge \tau\sqrt{n}$ for some constant $\tau > 0$. □

8.2 Indistinguishability of Shifting Distributions

In this subsection, we prove the following lemma, which was used in Section 7.2 for reducing a single-function SD to a multi-function SD, and will be used in Section 8.3 for reductions between various SD problems.

Lemma 7.13. *Let $n, k, q \in \mathbb{N}$, $\alpha \in (0, 1)$, where k, q, α are constants with respect to n and αn is an integer less than n/k. Let $\mathcal{F} \subseteq \{f : [q]^k \to \{0, 1\}\}$. For every $\varepsilon, \delta \in (0, 1]$, there exist $n' = \Omega(n)$ and constants $\alpha', \delta' \in (0, 1)$ such that the following holds. For every distribution $\mathcal{D}_Y, \mathcal{D}_N, \mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2 \in$*

$\Delta(\mathcal{F}\times[q]^k)$ *such that* $\mathcal{D}_Y=(1-\varepsilon)\mathcal{D}_0+\varepsilon\mathcal{D}_1$ *and* $\mathcal{D}_N=(1-\varepsilon)\mathcal{D}_0+\varepsilon\mathcal{D}_2$ *and for every* $c\in\mathbb{N}$*, suppose there exists a protocol for* $(\mathcal{F},\mathcal{D}_Y,\mathcal{D}_N)$*-SD with parameters* n *and* α *using* c *bits of communication with advantage* δ*; then there exists a protocol for* $(\mathcal{F},\mathcal{D}_1,\mathcal{D}_2)$*-SD with parameters* n' *and* α' *using* c *bits of communication with advantage* δ'*.*

Proof. Given the parameters n,α, and $\varepsilon\in(0,1)$, define $n'=\varepsilon n$ and $\alpha'=2\alpha$.

Let $(\mathbf{x}',M',\mathbf{b}',\mathbf{z}')$ be an instance of the $(\mathcal{F},\mathcal{D}_1,\mathcal{D}_2)$-SD problem where $\mathbf{x}'\in[q]^{n'}$, $M'\in\{0,1\}^{k\alpha' n'\times n'}$, $\mathbf{b}'\in[q]^{k\alpha' n'}$, $\mathbf{z}'\in\{0,1\}^{\alpha' n'}$. Let R' be the shared randomness defined later. We specify the map $(\mathbf{x}',M',\mathbf{b}',\mathbf{z}',R')\mapsto(\mathbf{x},M,\mathbf{b},\mathbf{z})$, where $\mathbf{x}\in[q]^n$, $M\in\{0,1\}^{k\alpha n\times n}$, $\mathbf{b}\in[q]^{k\alpha n}$, $\mathbf{z}\in\{0,1\}^{\alpha n}$.

A reduction from $(\mathcal{F},\mathcal{D}_1,\mathcal{D}_2)$-SD to $(\mathcal{F},\mathcal{D}_Y,\mathcal{D}_N)$-SD

Let $\mathbf{y}\sim\mathsf{Unif}([q]^{n-n'})$, $\mathbf{w}\sim\mathsf{Bern}(2\varepsilon)^{\alpha n}$. Let $\Gamma\in\{0,1\}^{n\times n}$ be a uniform permutation matrix. Let $\mathbf{c}=(\mathbf{c}(1),\dots,\mathbf{c}((n-n')/k))$, where $\mathbf{c}(i)\sim\mathcal{D}$ are chosen independently.

— Let $R'=(\mathbf{y},\mathbf{w},\Gamma,\mathbf{c})$ be the shared randomness.

Let $\#_w(i)=|\{j\in[i]\mid w_j=1\}|$ denote the number 1s among the first i coordinates of $\mathbf{w}$. If $\#_w(\alpha n)\geq\alpha' n'$ or if $\alpha n-\#_w(\alpha n)\geq(n-n')/k$, we declare an error. Note $\mathbb{E}[\#_w(n)]=\alpha' n'/2$, so the probability of error is negligible (specifically it is $\exp(-n)$).

Given $(\mathbf{x}',M',\mathbf{b}',\mathbf{z}',R')$, we now define $(\mathbf{x},M,\mathbf{b},\mathbf{z})$ as follows:

— Let $\mathbf{x}=\Gamma(\mathbf{x}',\mathbf{y})$, so $\mathbf{x}$ is a random permutation of the concatenation of $\mathbf{x}'$ and $\mathbf{y}$.

— Let $M'=(M'_1,\dots,M'_{\alpha' n'})$, where $M'_i\in\{0,1\}^{k\times n'}$. We extend M'_i to $N_i\in\{0,1\}^{k\times n}$ by adding all-zero columns to the right. For $i\in\{1,\dots,(n-n')/k\}$, let $P_i\in\{0,1\}^{k\times n}$ be given by $(P_i)_{j\ell}=1$ if and only if $\ell=n'+(i-1)k+j$. Next we define a matrix $\tilde{M}\in\{0,1\}^{k\alpha n\times n}=(\tilde{M}_1,\dots,\tilde{M}_{\alpha n})$, where $\tilde{M}_i\in\{0,1\}^{k\times n}$ is defined as follows: If $w_i=1$, then we let $\tilde{M}_i=N_{\#_w(i)}$ or else we let $\tilde{M}_i=P_{i-\#_w(i)}$. Finally, we let $M=\tilde{M}\cdot\Gamma^{-1}$.

— Let $\mathbf{b}=(\mathbf{b}(1),\dots,\mathbf{b}(\alpha n))$, where $\mathbf{b}(i)=\mathbf{b}'(\#_w(i))$ if $w_i=1$; otherwise $\mathbf{b}(i)=\mathbf{c}(i-\#_w(i))$.

— Let $z_i=1$ if and only if $M_i\mathbf{x}=\mathbf{b}(i)$ for every $i\in[\alpha n]$.

Now, we verify that the reduction satisfies the following success conditions.

Success conditions for the reduction

(1) **The reduction is locally well defined.** Namely, there exist random strings R' so that (i) Alice can get $\mathbf{x}$ through a map $(\mathbf{x}',R')\mapsto\mathbf{x}$ while Bob can get $(M,\mathbf{z})$ through a map $(M',\mathbf{z}',R')\mapsto(M,\mathbf{z})$.

(2) **The reduction is sound and complete.** Namely, (i) $z_i=1$ if and only if $M_i\mathbf{x}=\mathbf{b}(i)$ for all $i\in[\alpha n]$. (ii) If $\mathbf{b}'\sim\mathcal{D}_1^{\alpha' n'}$, then $\mathbf{b}\sim\mathcal{D}_Y^{\alpha n}$. Similarly, if $\mathbf{b}'\sim\mathcal{D}_2^{\alpha' n'}$, then $\mathbf{b}\sim\mathcal{D}_N^{\alpha n}$. (iii) $\mathbf{x}\sim\mathsf{Unif}([q]^n)$ and M is a uniformly random matrix conditioned on having exactly one "1" per row and at most one "1" per column.

Claim 8.5. *If* $\#_w(\alpha n)\leq\alpha' n'$ *and* $\alpha n-\#_w(\alpha n)\leq(n-n')/k$*, then the second map in the reduction is locally well defined, sound, and complete. In particular, the error event happens with probability at most* $\exp(-\Omega(n))$ *over the randomness of* R'*.*

Proof. To see the reduction is locally well defined, first note that Alice can compute $\mathbf{x}=\Gamma(\mathbf{x}',\mathbf{y})$ from $\mathbf{x}'$ and the shared randomness R' locally. As for Bob, note that the maximum index needed

for N and $\mathbf{b}'$ (resp. P and $\mathbf{c}$) is at most $\#_w(\alpha n)$ (resp. $\alpha n - \#_w(i)$). Namely, if $\#_w(\alpha n) \le \alpha' n'$ and $\alpha n - \#_w(\alpha n) \le (n - n')/k$, then M and $\mathbf{b}$ are well defined. Note that this happens with probability at least $1 - 2^{-\Omega(n)}$. Also, one can verify from the construction that M and $\mathbf{b}$ can be locally computed by M', $\mathbf{b}'$, and the shared randomness R'.

To see the reduction is sound and complete, (i) $z_i = 1$ if and only if $M_i\mathbf{x} = \mathbf{b}(i)$ for every $i \in [\alpha n]$ directly follows from the construction, and as for (ii), if $\mathbf{b}' \sim \mathcal{D}_1^{\alpha' n'}$. Now, for each $i \in [\alpha n]$, $\mathbf{b}(i) = \mathbf{b}'(\#_w(i))$ with probability ε and $\mathbf{b}(i) = \mathbf{c}(i - \#_w(i))$ with probability $1 - \varepsilon$. As $\mathbf{b}'(i') \sim \mathcal{D}_1$ for every $i' \in [\alpha' n']$ and $\mathbf{c}(i') \sim \mathcal{D}_0$ for every $i' \in [(n - n')/k]$, we have $\mathbf{b}(i) \sim (1-\varepsilon)\mathcal{D}_0 + \varepsilon\mathcal{D}_1 = \mathcal{D}_Y$ as desired. Similarly, one can show that if $\mathbf{b}' \sim \mathcal{D}_2^{\alpha' n'}$, then for every $i' \in [\alpha' n']$ we have $\mathbf{b}(i') \sim \mathcal{D}_N$. Finally, we have $\mathbf{x} \sim \mathsf{Unif}([q]^n)$ and M is a uniformly random matrix with exactly one "1" per row and at most one "1" per column (due to the application of a random permutation Γ) by construction.

This completes the proof of the success conditions (1) and (2) for the reduction. □

To wrap up the proof of Lemma 7.13, suppose there is a protocol Π for $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD with parameters n and α using c bits of communication with advantage δ. We describe a protocol Π' for $(\mathcal{F}, \mathcal{D}_1, \mathcal{D}_2)$-SD with parameters n' and α' using c bits of communication with advantage at least $\delta - 2^{-\Omega(n)}$.

Let $(\mathbf{x}', M', \mathbf{b}', \mathbf{z}')$ be an instance of the $(\mathcal{F}, \mathcal{D}_1, \mathcal{D}_2)$-SD problem where $\mathbf{x}' \in [q]^{n'}$, $M' \in \{0,1\}^{k\alpha' n' \times n'}$, $\mathbf{b}' \in [q]^{k\alpha' n'}$, $\mathbf{z}' \in \{0,1\}^{\alpha' n'}$. Let R' be the shared randomness defined above. In the new protocol Π', Alice and Bob compute their private inputs $\mathbf{x}$ and $(M, \mathbf{z})$, respectively. By Claim 8.5, the computation can be done locally with their original private inputs and the shared randomness. Also, with probability at least $1 - 2^{-\Omega(n)}$, the Yes (resp. No) instance of $(\mathcal{F}, \mathcal{D}_1, \mathcal{D}_2)$-SD is mapped to the Yes (resp. No) instance of $(\mathcal{F}, \mathcal{D}_Y, \mathcal{D}_N)$-SD. Namely, by directly applying Π on the new inputs, Alice and Bob can achieve $\delta - 2^{-\Omega(n)}$ advantage on $(\mathcal{F}, \mathcal{D}_1, \mathcal{D}_2)$ using the same amount of communication as desired. □

8.3 Proof of Theorem 7.4

Let $\mathbf{u}, \mathbf{v}$ be incomparable, let $S = \{i \in [k] \mid u_i \neq v_i\}$, and let $k'' = |S|$.

Step 1: Specify the auxiliary distributions:

- Let $\mathcal{A}_Y = \mathsf{Unif}(\{\mathbf{u}|_S, \mathbf{v}|_S\})$ and $\mathcal{A}_N = \mathsf{Unif}(\{(\mathbf{u}|_S) \vee (\mathbf{v}|_S), (\mathbf{u}|_S) \wedge (\mathbf{v}|_S)\})$. By Lemma 8.1, $(\mathcal{A}_Y, \mathcal{A}_N)$-SD requires $\tau\sqrt{n}$ space.
- Let $\mathcal{D}_1 = \mathsf{Unif}(\{\mathbf{u}, \mathbf{v}\})$ and $\mathcal{D}_2 = \mathsf{Unif}(\{\mathbf{u} \vee \mathbf{v}, \mathbf{u} \wedge \mathbf{v}\})$.
- Finally, there exists $\mathcal{D}_0$ such that we have $\mathcal{D} = (1 - 2\varepsilon)\mathcal{D}_0 + 2\varepsilon\mathcal{D}_1$ and $\mathcal{D}_{\mathbf{u},\mathbf{v}} = (1 - 2\varepsilon)\mathcal{D}_0 + 2\varepsilon\mathcal{D}_2$.

In the following, we are going to describe reduction from $(\mathcal{A}_Y, \mathcal{A}_N)$-SD with parameters (n'', α'', k'') to $(\mathcal{D}_1, \mathcal{D}_2)$-SD with parameters (n', α', k). And by Lemma 7.13, there exists a reduction from $(\mathcal{D}_1, \mathcal{D}_2)$-SD with parameters (n', α', k) to $(\mathcal{D}, \mathcal{D}_{\mathbf{u},\mathbf{v}})$-SD with parameters (n, α, k).

Step 2: Overview of the reduction from $(\mathcal{A}_Y, \mathcal{A}_N)$-SD to $(\mathcal{D}, \mathcal{D}_{\mathbf{u},\mathbf{v}})$-SD. Let Π be a protocol for $(\mathcal{D}, \mathcal{D}_{\mathbf{u},\mathbf{v}})$-SD with parameter $\alpha \le 1/(200k)$ using $C(n)$ communication bits to achieve advantage δ on instances of length n. We let $n'' = (k''\varepsilon/k)n$, $\alpha'' = (2k/k'')\alpha$ and design a protocol Π'' for $(\mathcal{A}_Y, \mathcal{A}_N)$-SD with parameter α'' achieving advantage at least $\delta/2$ on instances of length n'' using $C''(n'') = C(n)$ communication. Thus, by Lemma 8.1, there exists a constant $\tau'' > 0$ such that $C(n) = C''(n'') \ge \tau''\sqrt{n''} = \tau''\sqrt{(k''\varepsilon/k)}\sqrt{n}$, as desired.

To construct such reduction, we first reduce the above instance of $(\mathcal{A}_Y, \mathcal{A}_N)$-SD to an instance of $(\mathcal{D}_1, \mathcal{D}_2)$-SD with parameters $n' = kn''/k''$ and $\alpha' = \alpha'' n''/n'$. Next, we invoke Lemma 7.13 to get a protocol Π' (from Π), which achieves $\delta - 2^{-\Omega(n)}$ advantage on $(\mathcal{D}_1, \mathcal{D}_2)$-SD using $C(n)$ communication.

Without loss of generality, we assume Π' is deterministic and our new protocol Π'' for $(\mathcal{A}_Y, \mathcal{A}_N)$-SD uses shared randomness between Alice and Bob. The protocol Π'' is a map: $(\mathbf{x}'', M'', \mathbf{b}'', \mathbf{z}'', R'') \mapsto (\mathbf{x}', M', \mathbf{b}', \mathbf{z}')$.

Before describing the map, let us first state the desired conditions.

Success conditions for the reduction

(1) **The reduction is locally well defined.** Namely, there exists a random string R'' so that (i) Alice can get $\mathbf{x}'$ through the maps $(\mathbf{x}'', R'') \mapsto \mathbf{x}'$ while Bob can get $(M', \mathbf{z}')$ through the map $(M'', \mathbf{z}'', R'') \mapsto (M', \mathbf{z}')$.
(2) **The reduction is sound and complete.** Namely, (i) $z'_i = 1$ if and only if $M'_i \mathbf{x}' = \mathbf{b}'(i)$ for all $i \in [\alpha' n']$. (ii) If $\mathbf{b}'' \sim \mathcal{A}_Y^{\alpha'' n''}$, then $\mathbf{b}' \sim \mathcal{D}_1^{\alpha' n'}$. Similarly, if $\mathbf{b}'' \sim \mathcal{A}_N^{\alpha'' n''}$, then $\mathbf{b}' \sim \mathcal{D}_2^{\alpha' n'}$. (iii) $\mathbf{x}' \sim \mathsf{Unif}([q]^{n'})$ and M' is a uniformly random matrix conditioned on having exactly one "1" per row and at most one "1" per column.

Step 3: Specify and analyze the reduction from $(\mathcal{A}_Y, \mathcal{A}_N)$-SD to $(\mathcal{D}_1, \mathcal{D}_2)$-SD. We now specify the first map mentioned above and prove that it satisfies conditions (1) and (2).

A reduction from $(\mathcal{A}_Y, \mathcal{A}_N)$-SD to $(\mathcal{D}_1, \mathcal{D}_2)$-SD

— Let $R'' \sim \mathsf{Unif}([q]^{n'-n''})$ be the shared randomness.

Given $(\mathbf{x}'', M'', \mathbf{b}'', \mathbf{z}'', R'')$, we define $(\mathbf{x}', M', \mathbf{b}', \mathbf{z}')$ as follows. To get M', $\mathbf{z}'$, and $\mathbf{b}'$ we need some more notations. First, note that $\alpha'' n'' = \alpha' n'$ due to the choice of parameters.

— Let $\mathbf{x}' = (\mathbf{x}'', R'')$.
— M' can be viewed as the stacking of matrices $M''_1, \ldots, M''_{\alpha'' n''} \in \{0,1\}^{k'' \times n''}$. We first extend M''_i by adding all-zero columns at the end to get $N'_i \in \{0,1\}^{k'' \times n'}$. We then stack N'_i on top of $P'_i \in \{0,1\}^{(k-k'') \times n'}$ to get M'_i, where $(P'_i)_{j\ell} = 1$ if and only if $\ell = n'' + (i-1)k + j$. We let M' be the stacking of $M'_1, \ldots, M'_{\alpha' n'}$.
— Let $\mathbf{b}'' = (\mathbf{b}''(1), \ldots, \mathbf{b}''(\alpha' n'))$. Let $\tilde{\mathbf{u}} = (u_{k''+1}, \ldots, u_k)$ denote the common parts of $\mathbf{u}$ and $\mathbf{v}$. We let $\mathbf{b}'(i) = (\mathbf{b}''(i), \tilde{\mathbf{u}})$ and $\mathbf{b}' = (\mathbf{b}'(1), \ldots, \mathbf{b}'(\alpha' n'))$.
— Let $z'_i = 1$ if and only if $M'_i \mathbf{x}' = \mathbf{b}'(i)$ for all $i \in [\alpha' n']$ as required.

Claim 8.6. *The above reduction is locally well defined, sound, and complete.*

Proof. To see the map is locally well defined, note that Alice can compute $\mathbf{x}' = (\mathbf{x}'', R'')$ locally. Similarly, Bob can compute M' locally by construction. As for $\mathbf{z}'$, note that for every $i \in [\alpha' n']$, $z'_i = 1$ if and only if $z''_i = 1$ and $P'_i \mathbf{x}' = \tilde{\mathbf{u}}$. Since Bob has $\mathbf{z}'$ and can locally compute $P'_i \mathbf{x}'$ for every i, he can also compute $\mathbf{z}'$ locally.

To see the map is sound and complete, (i) $z'_i = 1$ if and only if $M'_i \mathbf{x}' = \mathbf{b}'(i)$ follows from the construction. As for (ii), for each $i \in [\alpha' n'] = [\alpha'' n'']$, if $\mathbf{b}''_i \sim \mathcal{A}_Y = \mathsf{Unif}(\{\mathbf{u}|_S, \mathbf{v}|_S\})$, then $\mathbf{b}'_i \sim \mathsf{Unif}(\{(\mathbf{u}|_S, \tilde{\mathbf{u}}), (\mathbf{v}|_S, \tilde{\mathbf{u}})\}) = \mathsf{Unif}(\{\mathbf{u}, \mathbf{v}\}) = \mathcal{D}_1$, as desired. Similarly, one can show that if $\mathbf{b}''_i \sim \mathcal{A}_N$, then $\mathbf{b}'_i \sim \mathcal{D}_1$. Finally, we have $\mathbf{x}' \sim \mathsf{Unif}([q]^{n'})$ by construction and hence (iii) holds.

This completes the proof of conditions (1) and (2) for the reduction. □

Step 4: Proof of Theorem 7.4.

Proof of Theorem 7.4. Let us start with setting up the parameters. Given $k \in (0, 1/(200k)), \alpha, n, \mathcal{D}$, and incomparable pair $(\mathbf{u}, \mathbf{v}) \in \mathrm{supp}(\mathcal{D})$ and polarization amount

$\varepsilon = \varepsilon(\mathcal{D}, \mathbf{u}, \mathbf{v})$, let $k'' = |\{i \in [k] \,|\, u_i \neq v_i\}|$, $n'' = (k''\varepsilon/k)n$, $\alpha'' = (2k/k'')\alpha$, $n' = kn''/k''$, $\alpha' = \alpha'' n'/n'$, and $\delta'' = \delta/2$.

Now, for the sake of contradiction, we assume that there exists a protocol $\Pi = (\Pi_A, \Pi_B)$ for $(\mathcal{D}, \mathcal{D}_{\mathbf{u},\mathbf{v}})$-SD with advantage δ and at most $\tau\sqrt{n}$ bits of communication.

First, by Claim 8.6, if $(\mathbf{x}'', M'', \mathbf{z}'')$ is a Yes (resp. No) instance of $(\mathcal{A}_Y, \mathcal{A}_N)$-SD, then the output of the reduction, i.e., $(\mathbf{x}', M', \mathbf{z}')$, is a Yes (resp. No) instance of $(\mathcal{D}_1, \mathcal{D}_2)$-SD. Next, Alice and Bob run the protocol Π' from Lemma 7.13 on $(\mathbf{x}', M', \mathbf{z}')$. By the correctness of the reduction as well as the protocol Π', we know that Alice and Bob have advantage at least $\delta - \exp(-\Omega(n)) \geq \delta/2 = \delta''$ in solving $(\mathcal{A}_Y, \mathcal{A}_N)$-SD with at most $\tau\sqrt{n} = \tau\sqrt{(k/(k''\varepsilon))n''}$ bits of communication.

Finally, by Lemma 8.1, we know that there exists a constant $\tau_0 > 0$ such that any protocol for $(\mathcal{A}_Y, \mathcal{A}_N)$-SD with advantage δ'' requires at least $\tau_0\sqrt{n''}$ bits of communication. This implies that $\tau \geq \tau_0\sqrt{k''\varepsilon/k}$. We conclude that any protocol for $(\mathcal{D}, \mathcal{D}_{\mathbf{u},\mathbf{v}})$-SD with advantage δ requires at least $\tau\sqrt{n}$ bits of communication. □

9 DICHOTOMY FOR EXACT COMPUTATION

In this section we prove Theorem 3.16. For this, we will use tight bounds on the randomized communication complexity of the **Disjointness (Disj)** and **Gap Hamming Distance (GHD)** problems.

Definition 9.1 (Disjointness (Disj)). In the Disj_n problem, Alice and Bob receive binary strings $x, y \in \{0,1\}^n$ of Hamming weight $\Delta(x) = \Delta(y) = n/4$, respectively. If the Hamming distance $\Delta(x, y) = n/2$, the players must output 1; if $\Delta(x, y) < n/2$, they must output 0.

Definition 9.2 (Gap Hamming Distance (GHD)). In the $\mathrm{GHD}_{n,t,g}$ problem, Alice and Bob receive binary strings $x, y \in \{0,1\}^n$, respectively. If the Hamming distance $\Delta(x, y) \geq t + g$, the players must output 1; if $\Delta(x, y) \leq t - g$, they must output 0; otherwise, they may output either 0 and 1.

The following results give tight bounds on the randomized communication complexity of Disj and GHD.

Theorem 9.3 ([52, 68]). *For all large enough n, any randomized protocol solving Disj_n with probability $2/3$ must use $\Omega(n)$ bits of communication.*

Theorem 9.4 ([29, 72, 75]). *For every $a \in (0, 1/2]$ and every $g \geq 1$ and all large enough n the following holds. If $t \in [an, (1-a)n]$, then any randomized protocol solving $\mathrm{GHD}_{n,t,g}$ with probability $2/3$ must use $\Omega(\min\{n, n^2/g^2\})$ bits of communication.*

Equipped with these results, we are ready to prove Theorem 3.16.

Theorem 3.16. *For every $q, k \in \mathbb{N}$, and every family of functions $\mathcal{F} \subseteq \{f : [q]^k \to \{0,1\}\}$, the following hold:*

(1) If $\mathcal{F}$ is constant satisfiable, then there exists a deterministic linear sketching algorithm that uses $O(\log n)$ space and solves Max-CSP($\mathcal{F}$) exactly optimally.
(2) If $\mathcal{F}$ is not constant satisfiable, then the following hold in the streaming setting:
  (a) Every probabilistic algorithm solving Max-CSP($\mathcal{F}$) exactly requires $\Omega(n)$ space.
  (b) For every $\varepsilon = \varepsilon(n) > 0$, $(1, 1-\varepsilon)$-Max-CSP($\mathcal{F}$) requires $\Omega(\min\{n, \varepsilon^{-1}\})$-space[14] on sufficiently large inputs.
  (c) For $\rho_{\min}(\mathcal{F})$ defined in Definition 3.5, for every $\rho_{\min}(\mathcal{F}) < \gamma < 1$ and every $\varepsilon = \varepsilon(n) > 0$, $(\gamma, \gamma - \varepsilon)$-Max-CSP($\mathcal{F}$) requires $\Omega(\min\{n, \varepsilon^{-2}\})$-space[8] on sufficiently large inputs.

[14] The constant hidden in the Ω depends on $\mathcal{F}$, but (obviously) not on ε.

While this theorem doesn't give tight bounds on the space complexity of Max-CSP($\mathcal{F}$) in terms of n, the dependence on ε is tight. For every family of functions $\mathcal{F}$, if we sample $O(n/\varepsilon^2)$ random constraints, then by the Chernoff bound we preserve the values of all assignments within a factor of $1 \pm \varepsilon$.

Proof. For the first item of the theorem, we note that the maximum number of simultaneously satisfiable constraints in a σ-satisfiable formula is the number of non-zero constraints $f \in \mathcal{F} \setminus \{\mathbf{0}\}$ in it. This can be computed in space $O(\log n)$.

Now we turn to the proof of the second item of the theorem in the streaming setting. To this end, first we prove that there exists an unsatisfiable instance I of Max-CSP($\mathcal{F} \setminus \{\mathbf{0}\}$). Let I be an instance on kq variables that has every constraint from $\mathcal{F} \setminus \{\mathbf{0}\}$ applied to every (unordered) k-tuple of distinct variables. Any assignment $\boldsymbol{\nu} \in [q]^{kq}$ has at least k equal coordinates. That is, there exists $\sigma \in [q]$ such that $\Sigma = \{i \colon \boldsymbol{\nu}_i = \sigma\}$ has size $|\Sigma| \geq k$. Since $\mathcal{F}$ is not σ-satisfiable, there exists a function $f \in \mathcal{F} \setminus \{\mathbf{0}\}$ that $f(\sigma^k) \neq 1$. Thus, the corresponding constraint of I is not satisfied by $\boldsymbol{\nu}$.

Now we pick a minimal unsatisfied formula J on kq variables with constraints from $\mathcal{F} \setminus \{\mathbf{0}\}$, which is a formula such that all proper subsets of the constraints of J can be simultaneously satisfied. Since J doesn't have zero-constraints, J must have at least two constraints. We partition J into two arbitrary non-empty subsets of constraints $J = J_A \sqcup J_B$. Note that by minimality of J, J_A and J_B are both satisfiable.

Observe that item 2(a) of the theorem follows from 2(b) by setting $\varepsilon = \Theta(1/n)$. In order to prove the item 2(b), we reduce Disj_m for $m = |J|^{-1}\varepsilon^{-1}$ to Max-CSP($\mathcal{F}$) on n variables. We can assume that $\varepsilon \geq \frac{kq}{n|J|}$, as for smaller ε the optimal lower bound of $\Omega(n)$ is implied by this setting. We partition the n variables of Max-CSP($\mathcal{F}$) into at least m groups of size kq. Let $x, y \in \{0,1\}^m$ be the inputs of Alice and Bob in the Disj_m problem. If $x_i = 1$, then Alice applies the constraints J_A to the ith block of kq variables of the formula. Similarly, if $y_i = 1$, then Bob applies the constraints J_B to the ith block of kq variables. Let C_A and C_B be the sets of constraints produced by Alice and Bob, respectively, and let $\Psi = C_A \cup C_B$. Since $\Delta(x) = \Delta(y) = m/4$, the total number of constraints in the formula $|\Psi| = |J|m/4$. Note that Ψ is satisfiable if and only if $\mathrm{Disj}(x, y) = 1$. Therefore, if x and y are disjoint, then $\mathrm{val}(C_A \cup C_B) = 1$; otherwise,

$$\mathrm{val}(\Psi) \leq 1 - \frac{4}{|J|m} < 1 - \varepsilon\,.$$

Any streaming algorithm that receives constraints C_A and C_B and solves $(1, 1-\varepsilon)$-Max-CSP($\mathcal{F}$) with probability $2/3$ also solves the Disj_m problem. Therefore, by Theorem 9.3, such an algorithm must use space $\Omega(m) = \Omega(1/\varepsilon)$.

In order to prove item 2(c), we reduce the $\mathrm{GHD}_{n,t,g}$ problem to Max-CSP($\mathcal{F}$) on $nkq + O(1) = O(n)$ variables, where $t = n(1-\gamma)$ and $g \geq 1$ will be determined later. We will create two groups of constraints: The first group of constraints $C_A \cup C_B$ will have value $1 - O(\Delta(x, y)/n)$, and the second group of constraints will have value close to $\rho_{\min}$. By taking a weighted combination of these two groups, we will get a formula whose value is less than $\gamma - \varepsilon$ for $\Delta(x, y) \geq t + g$, and whose value is at least γ for $\Delta(x, y) \leq t$.

Again, we start with a minimal unsatisfiable formula on kq variables. If $|J| = 2d$ is even, then we arbitrarily partition J into two sets of d constraints J_A and J_B. If $|J|$ is odd, then we add one constraint to $|J|$ as follows. By minimality, there is an assignment that satisfies $|J| - 1$ constraints of J; let c be one of these constraints. We add another copy of c to J and partition J into two sets of d constraints J_A and J_B. Note that while J_A and J_B are satisfiable, only $2d - 1$ constraints of $J_A \cup J_B$ can be satisfied simultaneously.

Let $x, y \in \{0,1\}^n$ be the inputs of Alice and Bob in the $\mathrm{GHD}_{n,t,g}$ problem. If $x_i = 1$, then Alice applies the constraints J_A to the ith block of kq variables of the formula; otherwise Alice applies

the constraint J_B to these variables. Similarly, if $y_i = 1$ or $y_i = 0$, then Bob applies the constraints J_A or J_B to the ith block of kq variables. Let C_A and C_B be the sets of constraints produced by Alice and Bob, respectively. Observe that $|C_A| = |C_B| = nd$. The set of constraints added by Alice and Bob when processing their ith coordinates is satisfiable if and only if $x_i = y_i$. When $x_i \neq y_i$, then by the construction of J, exactly $2d - 1$ constraints are satisfiable. Therefore,

$$\mathsf{val}(C_A \cup C_B) = 1 - \frac{\Delta(x, y)}{2dn}.$$

Let $\gamma' = (\gamma + \rho_{\min})/2 < \gamma$. By the definition of $\rho_{\min}(\mathcal{F})$, there exists n_0 and a formula Φ' of Max-CSP(f) such that $\mathsf{val}(\Phi') = \gamma'$. By taking several copies of Φ' on the same n_0 variables, we get an instance Φ with $D = |\Phi| = \frac{n(2d-1)(1-\gamma)}{\gamma-\gamma'} = \Theta(n)$ constraints and value $\mathsf{val}(\Phi) = \gamma'$.

Now we output an instance Ψ of Max-CSP($\mathcal{F}$) on $nkq + n_0$ variables that is simply a union of $C_A \cup C_B$ and Φ on disjoint sets of variables. By construction,

$$\mathsf{val}(\Psi) = \frac{(2dn - \Delta(x, y)) + \gamma' D}{2dn + D}.$$

In the case when $\Delta(x, y) \leq t = (1 - \gamma)n$, we have

$$\mathsf{val}(\Psi) \geq \frac{2dn - (1 - \gamma)n + \gamma' D}{2dn + D} = \gamma.$$

And for the case of $\Delta(x, y) \geq t + g = (1 - \gamma)n + g$, we have that

$$\mathsf{val}(\Psi) \leq \frac{(2dn - (1 - \gamma)n - g) + \gamma' D}{2dn + D} = \gamma - \frac{g}{2dn + D} = \gamma - \varepsilon$$

for $g = \varepsilon(2dn + D) = \Theta(n\varepsilon)$.

Therefore, any streaming algorithm for $(\gamma, \gamma - \varepsilon)$-Max-CSP($\mathcal{F}$) will imply a protocol for the $\mathsf{GHD}_{n,t,g}$ problem. By Theorem 9.4, such a streaming algorithm must use at least $\Omega(\min\{n, n^2/g^2\}) = \Omega(\min\{n, \varepsilon^{-2}\})$ bits of communication. □

ACKNOWLEDGMENTS

We are grateful to Lijie Chen, Gillat Kol, Dmitry Paramonov, Raghuvansh Saxena, Zhao Song, and Huacheng Yu for detecting a fatal error in an earlier version of this article [36] and then for pinpointing the location of the error. As a result, the main theorem of the current article is significantly different than the theorem claimed in the previous version.

Thanks to Johan Håstad for many pointers to the work on approximation resistance and answers to many queries. Thanks to Dmitry Gavinsky, Julia Kempe, and Ronald de Wolf for prompt and detailed answers to our queries on outdated versions of their work [43]. Thanks to Prasad Raghavendra for answering our questions about the approximation resistance dichotomy from his work [67]. Thanks to Saugata Basu for the pointers to the algorithms for quantified theory of the reals. Thanks to Jelani Nelson for pointers to ℓ_1 norm estimation algorithms used in the earlier version of this article. Thanks to Alex Andoni for pointers to $\ell_{1,\infty}$ norm estimation algorithms. Thanks to anonymous referees of many versions of this work for their valuable comments. In particular, we thank the referees for clarifying the gap between linear sketching algorithms and dynamic streaming algorithms.

REFERENCES

[1] Yuqing Ai, Wei Hu, Yi Li, and David P. Woodruff. 2016. New characterizations in turnstile streams with applications. In *31st Conference on Computational Complexity (CCC'16)*. LIPIcs, 20:1–20:22.

[2] Alexandr Andoni. 2020. Personal Communication. (December 24, 2020).

[3] Alexandr Andoni, Robert Krauthgamer, and Krzysztof Onak. 2011. Streaming algorithms via precision sampling. In *IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS'11)*. IEEE, 363–372. https://doi.org/10.1109/FOCS.2011.82

[4] Sepehr Assadi. 2022. A two-pass (conditional) lower bound for semi-streaming maximum matching. In *ACM-SIAM Symposium on Discrete Algorithms (SODA'22)*. SIAM, 708–742.

[5] Sepehr Assadi and Soheil Behnezhad. 2021. Beating two-thirds for random-order streaming matching. In *48th International Colloquium on Automata, Languages, and Programming (ICALP'21)*. LIPIcs, 19:1–19:13.

[6] Sepehr Assadi, Soheil Behnezhad, Sanjeev Khanna, and Huan Li. 2023. On regularity lemma and barriers in streaming and dynamic matching. In *55th Annual ACM Symposium on Theory of Computing (STOC 2023)*. ACM, 131–144.

[7] Sepehr Assadi, Andrew Chen, and Glenn Sun. 2022. Deterministic graph coloring in the streaming model. In *54th Annual ACM SIGACT Symposium on Theory of Computing (STOC'22)*. ACM, 261–274.

[8] Sepehr Assadi and Aditi Dudeja. 2021. Ruling sets in random order and adversarial streams. In *35th International Symposium on Distributed Computing (DISC'21)*. LIPIcs, 6:1–6:18.

[9] Sepehr Assadi, Arun Jambulapati, Yujia Jin, Aaron Sidford, and Kevin Tian. 2022. Semi-streaming bipartite matching in fewer passes and optimal space. In *ACM-SIAM Symposium on Discrete Algorithms (SODA'22)*. SIAM, 627–669.

[10] Sepehr Assadi, Sanjeev Khanna, and Yang Li. 2017. On estimating maximum matching size in graph streams. In *28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'17)*. SIAM, 1723–1742.

[11] Sepehr Assadi, Sanjeev Khanna, and Yang Li. 2021. Tight bounds for single-pass streaming complexity of the set cover problem. *SIAM J. Comput.* 50, 3 (2021), 341–376. https://doi.org/10.1137/16M1095482

[12] Sepehr Assadi, Sanjeev Khanna, Yang Li, and Grigory Yaroslavtsev. 2016. Maximum matchings in dynamic graph streams and the simultaneous communication model. In *27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'16)*. SIAM, 1345–1364.

[13] Sepehr Assadi, Gillat Kol, Raghuvansh R. Saxena, and Huacheng Yu. 2020. Multi-pass graph streaming lower bounds for cycle counting, MAX-CUT, matching size, and other problems. In *61st IEEE Annual Symposium on Foundations of Computer Science (FOCS'20)*. IEEE, 354–364.

[14] Sepehr Assadi, Gillat Kol, and Zhijun Zhang. 2022. Rounds vs communication tradeoffs for maximal independent sets. In *63rd IEEE Annual Symposium on Foundations of Computer Science (FOCS'22)*. IEEE, 1193–1204.

[15] Sepehr Assadi, Pankaj Kumar, and Parth Mittal. 2022. Brooks' theorem in graph streams: A single-pass semi-streaming algorithm for coloring. In *54th Annual ACM SIGACT Symposium on Theory of Computing (STOC'22)*. ACM, 234–247.

[16] Sepehr Assadi and Vishvajeet N. 2021. Graph streaming lower bounds for parameter estimation and property testing via a streaming XOR lemma. In *53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC'21)*. ACM, 612–625.

[17] Sepehr Assadi and Ran Raz. 2020. Near-quadratic lower bounds for two-pass graph streaming algorithms. In *61st IEEE Annual Symposium on Foundations of Computer Science (FOCS'20)*. IEEE, 342–353.

[18] Sepehr Assadi and Vihan Shah. 2022. An asymptotically optimal algorithm for maximum matching in dynamic streams. In *13th Innovations in Theoretical Computer Science Conference (ITCS'22)*. LIPIcs, 9:1–9:23.

[19] Sepehr Assadi and Janani Sundaresan. 2023. (Noisy) gap cycle counting strikes back: Random order streaming lower bounds for connected components and beyond. In *55th Annual ACM Symposium on Theory of Computing (STOC'23)*. ACM, 183–195.

[20] Sepehr Assadi and Chen Wang. 2022. Sublinear time and space algorithms for correlation clustering via sparse-dense decompositions. In *13th Innovations in Theoretical Computer Science Conference (ITCS'22)*. LIPIcs, 10:1–10:20.

[21] Per Austrin and Elchanan Mossel. 2009. Approximation resistant predicates from pairwise independence. *Comput. Complex.* 18, 2 (2009), 249–271. https://doi.org/10.1007/s00037-009-0272-6

[22] Libor Barto and Marcin Kozik. 2012. Robust satisfiability of constraint satisfaction problems. In *Proceedings of the 44th Symposium on Theory of Computing Conference (STOC'12)*. ACM, 931–940. https://doi.org/10.1145/2213977.2214061

[23] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. 2006. *Algorithms in Real Algebraic Geometry*. Springer.

[24] Soheil Behnezhad. 2023. Dynamic algorithms for maximum matching size. In *ACM-SIAM Symposium on Discrete Algorithms (SODA'23)*. SIAM, 129–162.

[25] Stephen P. Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

[26] Joanna Boyland, Michael Hwang, Tarun Prasad, Noah Singer, and Santhoshini Velusamy. 2022. On sketching approximations for symmetric Boolean CSPs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX'22)*. LIPIcs, 38:1–38:23.

[27] Andrei A. Bulatov. 2017. A dichotomy theorem for nonuniform CSPs. In *58th IEEE Annual Symposium on Foundations of Computer Science (FOCS'17)*. IEEE, 319–330.

[28] Amit Chakrabarti. 2020. Data stream algorithms. *Lecture Notes on Data Stream Algorithms* (2020), 94. https://www.cs.dartmouth.edu/~ac/Teach/data-streams-lecnotes.pdf

[29] Amit Chakrabarti and Oded Regev. 2012. An optimal lower bound on the communication complexity of gap-Hamming-distance. *SIAM J. Comput.* 41, 5 (2012), 1299–1317.

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

[31] Lijie Chen, Gillat Kol, Dmitry Paramonov, Raghuvansh R. Saxena, Zhao Song, and Huacheng Yu. 2021. Near-optimal two-pass streaming algorithm for sampling random walks over directed graphs. In *48th International Colloquium on Automata, Languages, and Programming (ICALP'21)*. LIPIcs, 52:1–52:19.

[32] Lijie Chen, Gillat Kol, Dmitry Paramonov, Raghuvansh R. Saxena, Zhao Song, and Huacheng Yu. 2023. Towards multi-pass streaming lower bounds for optimal approximation of max-cut. In *ACM-SIAM Symposium on Discrete Algorithms (SODA'23)*. SIAM, 878–924.

[33] Yu Chen, Sanjeev Khanna, and Zihan Tan. 2023. Sublinear algorithms and lower bounds for estimating MST and TSP cost in general metrics. In *50th International Colloquium on Automata, Languages, and Programming (ICALP'23)*. LIPIcs, 37:1–37:16.

[34] Ashish Chiplunkar, John Kallaugher, Michael Kapralov, and Eric Price. 2022. Factorial lower bounds for (almost) random order streams. In *63rd IEEE Annual Symposium on Foundations of Computer Science (FOCS'22)*. IEEE, 486–497.

[35] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, and Santhoshini Velusamy. 2022. Approximability of all finite CSPs with linear sketches. In *62nd IEEE Annual Symposium on Foundations of Computer Science*. IEEE, 1197–1208.

[36] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, and Santhoshini Velusamy. 2021. Classification of the streaming approximability of Boolean CSPs. *CoRR* abs/2102.12351v1 (February 24, 2021), 1–49. arXiv:2102.12351 https://arxiv.org/abs/2102.12351v1

[37] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, and Santhoshini Velusamy. 2021. Approximability of all Boolean CSPs with linear sketches. *CoRR* abs/2102.12351v3 (April 14, 2021), 1–60. arXiv:2102.12351 https://arxiv.org/abs/2102.12351v3

[38] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, and Santhoshini Velusamy. 2021. Approximability of all finite CSPs with linear sketches. *CoRR* abs/2105.01161 (May 3, 2021), 1–75. arXiv:2105.01161 https://arxiv.org/abs/2105.01161

[39] Chi-Ning Chou, Alexander Golovnev, Amirbehshad Shahrasbi, Madhu Sudan, and Santhoshini Velusamy. 2022. Sketching approximability of (weak) monarchy predicates. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX'22)*. LIPIcs, 35:1–35:17.

[40] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, Ameya Velingker, and Santhoshini Velusamy. 2022. Linear space streaming lower bounds for approximating CSPs. In *54th Annual ACM SIGACT Symposium on Theory of Computing (STOC'22)*. ACM, 275–288.

[41] Chi-Ning Chou, Alexander Golovnev, and Santhoshini Velusamy. 2020. Optimal streaming approximations for all Boolean Max-2CSPs and Max-kSAT. In *61st IEEE Annual Symposium on Foundations of Computer Science (FOCS'20)*. IEEE, 330–341.

[42] Víctor Dalmau and Andrei A. Krokhin. 2013. Robust satisfiability for CSPs: Hardness and algorithmic results. *ACM Trans. Comput. Theory* 5, 4 (2013), 15:1–15:25. https://doi.org/10.1145/2540090

[43] Dmitry Gavinsky, Julia Kempe, Iordanis Kerenidis, Ran Raz, and Ronald de Wolf. 2009. Exponential separation for one-way quantum communication complexity, with applications to cryptography. *SIAM J. Comput.* 38, 5 (2009), 1695–1708.

[44] Ashish Goel, Michael Kapralov, and Sanjeev Khanna. 2012. On the communication and streaming complexity of maximum bipartite matching. In *23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'12)*. SIAM, 468–485.

[45] Venkatesan Guruswami, Johan Håstad, Rajsekar Manokaran, Prasad Raghavendra, and Moses Charikar. 2011. Beating the random ordering is hard: Every ordering CSP is approximation resistant. *SIAM J. Comput.* 40, 3 (2011), 878–914.

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

[47] Venkatesan Guruswami, Ameya Velingker, and Santhoshini Velusamy. 2017. Streaming complexity of approximating max 2CSP and max acyclic subgraph. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX'17)*. LIPIcs, 8:1–8:19.

[48] Johan Håstad. 2001. Some optimal inapproximability results. *J. ACM* 48, 4 (2001), 798–859.

[49] Zengfeng Huang, Bozidar Radunovic, Milan Vojnovic, and Qin Zhang. 2015. Communication complexity of approximate matching in distributed graphs. In *32nd International Symposium on Theoretical Aspects of Computer Science (STACS'15)*. LIPIcs, 460–473.

[50] Jeff Kahn, Gil Kalai, and Nathan Linial. 1988. The influence of variables on Boolean functions. In *29th Annual Symposium on Foundations of Computer Science (FOCS'88)*. IEEE, 68–80.

[51] John Kallaugher and Eric Price. 2020. Separations and equivalences between turnstile streaming and linear sketching. In *52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC'20)*. ACM, 1223–1236.

[52] Bala Kalyanasundaram and Georg Schintger. 1992. The probabilistic communication complexity of set intersection. *SIAM J. Discrete Math.* 5, 4 (1992), 545–557.

[53] Michael Kapralov. 2013. Better bounds for matchings in the streaming model. In *SODA 2013*. SIAM, 1679–1697.

[54] Michael Kapralov. 2021. Space lower bounds for approximating maximum matching in the edge arrival model. In *ACM-SIAM Symposium on Discrete Algorithms (SODA'21)*. SIAM, 1874–1893.
[55] Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. 2014. Approximating matching size from random streams. In *25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'14)*. SIAM, 734–751.
[56] Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. 2015. Streaming lower bounds for approximating MAX-CUT. In *26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'15)*. SIAM, 1263–1282.
[57] Michael Kapralov, Sanjeev Khanna, Madhu Sudan, and Ameya Velingker. 2017. $(1+\Omega(1))$-approximation to MAX-CUT requires linear space. In *28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'17)*. SIAM, 1703–1722.
[58] Michael Kapralov and Dmitry Krachun. 2019. An optimal space lower bound for approximating MAX-CUT. In *51st Annual ACM SIGACT Symposium on Theory of Computing (STOC'19)*. ACM, 277–288.
[59] Subhash Khot. 2002. On the power of unique 2-prover 1-round games. In *34th Annual ACM Symposium on Theory of Computing (STOC'02)*. ACM, 767–775.
[60] Subhash Khot, Madhur Tulsiani, and Pratik Worah. 2014. A characterization of strong approximation resistance. In *Symposium on Theory of Computing (STOC'14)*. ACM, 634–643.
[61] Gillat Kol, Dmitry Paramonov, Raghuvansh R. Saxena, and Huacheng Yu. 2023. Characterizing the multi-pass streaming complexity for solving Boolean CSPs exactly. In *14th Innovations in Theoretical Computer Science Conference (ITCS'23)*. LIPIcs, 80:1–80:15.
[62] Christian Konrad. 2015. Maximum matching in turnstile streams. In *Algorithms - ESA 2015 - 23rd Annual European Symposium (ESA'15)*. Springer, 840–852.
[63] Gábor Kun, Ryan O'Donnell, Suguru Tamaki, Yuichi Yoshida, and Yuan Zhou. 2012. Linear programming, width-1 CSPs, and robust satisfaction. In *Innovations in Theoretical Computer Science 2012 (ITCS'12)*. ACM, 484–495.
[64] Yi Li, Huy L. Nguyen, and David P. Woodruff. 2014. Turnstile streaming algorithms might as well be linear sketches. In *Symposium on Theory of Computing (STOC'14)*. ACM, 174–183.
[65] Ryan O'Donnell. 2014. *Analysis of Boolean Functions*. Cambridge University Press.
[66] Aaron Potechin. 2019. On the approximation resistance of balanced linear threshold functions. In *51st Annual ACM SIGACT Symposium on Theory of Computing (STOC'19)*. ACM, 430–441.
[67] Prasad Raghavendra. 2008. Optimal algorithms and inapproximability results for every CSP? In *40th Annual ACM Symposium on Theory of Computing (STOC'08)*. ACM, 245–254.
[68] Alexander A. Razborov. 1990. On the distributional complexity of disjointness. In *Automata, Languages and Programming, 17th International Colloquium (ICALP'90)*. Springer, 249–253.
[69] Raghuvansh R. Saxena, Noah Singer, Madhu Sudan, and Santhoshini Velusamy. 2023. Streaming complexity of CSPs with randomly ordered constraints. In *ACM-SIAM Symposium on Discrete Algorithms (SODA'23)*. SIAM, 4083–4103.
[70] Raghuvansh R. Saxena, Noah G. Singer, Madhu Sudan, and Santhoshini Velusamy. 2023. Improved streaming algorithms for maximum directed cut via smoothed snapshots. In *64th IEEE Annual Symposium on Foundations of Computer Science (FOCS'23)*. IEEE.
[71] Thomas J. Schaefer. 1978. The complexity of satisfiability problems. In *10th Annual ACM Symposium on Theory of Computing (STOC'78)*. ACM, 216–226.
[72] Alexander A. Sherstov. 2012. The communication complexity of gap Hamming distance. *Theory Comput.* 8, 1 (2012), 197–208.
[73] Noah Singer, Madhu Sudan, and Santhoshini Velusamy. 2021. Streaming approximation resistance of every ordering CSP. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX'21)*. LIPIcs, 17:1–17:19.
[74] Noah G. Singer. 2023. Oblivious algorithms for the Max-kAND problem. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX'23)*.
[75] Thomas Vidick. 2012. A concentration inequality for the overlap of a vector on a large set, with application to the communication complexity of the gap-Hamming-distance problem. *Chicago J. Theor. Comput. Sci.* 18, 1 (2012), 1–12.
[76] Dmitriy Zhuk. 2017. A proof of CSP dichotomy conjecture. In *58th IEEE Annual Symposium on Foundations of Computer Science (FOCS'17)*. IEEE, 331–342.

Received 14 March 2022; revised 13 September 2023; accepted 21 February 2024