ABSTRACT

For a constraint satisfaction problem (CSP), a robust satisfaction algorithm is one that outputs an assignment satisfying most of the constraints on instances that are near-satisfiable. It is known that the CSPs that admit efficient robust satisfaction algorithms are precisely those of bounded width, i.e., CSPs whose satisfiability can be checked by a simple local consistency algorithm (e.g., 2-SAT or Horn-SAT in the Boolean case). While the exact satisfiability of a bounded width CSP can be checked by combinatorial algorithms, the robust algorithm is based on rounding a canonical Semi Definite Programming (SDP) relaxation.

In this work, we initiate the study of robust satisfaction algorithms for promise CSPs, which are a vast generalization of CSPs that have received much attention recently. The motivation is to extend the theory beyond CSPs, as well as to better understand the power of SDPs. We present robust SDP rounding algorithms under some general conditions, namely the existence of majority or alternating threshold polymorphisms. On the hardness front, we prove that the lack of such polymorphisms makes the PCSP hard for all pairs of symmetric Boolean predicates. Our method involves a novel method to argue SDP gaps via the absence of certain colorings of the sphere, with connections to sphere Ramsey theory.

We conjecture that PCSPs with robust satisfaction algorithms are precisely those for which the feasibility of the canonical SDP implies (exact) satisfiability. We also give a precise algebraic condition, known as a minion characterization, of which PCSPs have the latter property.

CCS CONCEPTS
• Theory of computation → Problems, reductions and completeness: Rounding techniques.

KEYWORDS

Approximation algorithms, Semidefinite programming, Promise constraint satisfaction problems, Sphere Ramsey theory

ACM Reference Format:
Joshua Brakensiek, Venkatesan Guruswami, and Sai Sandeep. 2023. SDPs and Robust Satisfiability of Promise CSP. In Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC ’23), June 20–23, 2023, Orlando, FL, USA. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3564246.3585180

This work is licensed under a Creative Commons Attribution 4.0 International License.

1 INTRODUCTION

Horn-SAT and 2-SAT are Boolean constraint satisfaction problems (CSPs) that admit simple combinatorial algorithms for satisfiability. They are both examples of bounded width CSPs: the existence of locally consistent assignments (which satisfy all local constraints involving some bounded number of variables, and which are consistent on the intersections) implies the existence of a global satisfying assignment.

While the simple local propagation algorithms for Horn-SAT and 2-SAT work when the instance is perfectly satisfiable, they are not robust to errors—if the given instance is almost satisfiable, the local consistency based algorithms do not guarantee solutions that satisfy almost all the constraints. In a beautiful work, Zwick [43] initiated the study of finding ‘robust’ algorithms for CSPs, namely algorithms that output solutions satisfying \( 1 - f(\epsilon) \) fraction of the constraints when the instance is promised to be \( 1 - \epsilon \) satisfiable, where \( f(\epsilon) \to 0 \) as \( \epsilon \to 0 \). Zwick obtained robust algorithms for 2-SAT using SDP rounding and for Horn-SAT based on LP rounding. The PCP theorem together with Schaefer’s reductions [36] shows that Boolean CSPs that are NP-Hard are also APX-hard with perfect completeness, which in particular means that they do not admit robust satisfiability algorithms. The only other interesting Boolean CSP besides Horn-SAT and 2-SAT for which satisfiability is polynomial-time decidable is Linear Equations modulo 2. Håstad [30] in his seminal work showed that even for 3-LIN (when all equations involve just three variables), for every \( \epsilon, \delta > 0 \), it is NP-Hard to output a solution satisfying \( \frac{1}{2} + \delta \) fraction of the constraints even when the instance is guaranteed to have a solution satisfying \( 1 - \epsilon \) fraction of the constraints.

Unlike Horn-SAT or 2-SAT, the satisfiability algorithm for 3-LIN is not local, and 3-LIN does not have bounded width. Together with Schaefer’s dichotomy theorem [36], this yields that for Boolean CSPs, bounded width characterizes robust satisfiability. For CSPs over general domains, a landmark result in the algebraic approach to CSP due to Barto and Kozik [7] showed that CSPs that are not bounded width can express linear equations. A reduction from Håstad’s result then shows that CSPs that are not bounded width do not admit robust algorithms. Guruswami and Zhou [29] conjectured the converse—namely that all bounded width CSPs, over any domain, admit robust algorithms. Another work by Barto and Kozik [8] resolved this conjecture in the affirmative, thus giving a full characterization of CSPs that have robust algorithms.

In this work, we study robust algorithms for the class of Promise Constraint Satisfaction Problems (PCSPs). PCSPs are a generalization of CSPs where each constraint has a strong form and a weak form. Given an instance that is promised to have a solution satisfying the stronger form of the constraints, the objective is to find a solution satisfying the weaker form of the constraints. A classic example of PCSPs is (1-in-3-SAT, NAE-3-SAT). While both...
the underlying CSPs are NP-Hard, the resulting PCSP does have a polynomial-time algorithm: given an instance of 1-in-3-SAT that is promised to be satisfiable, we can find an assignment to the variables in polynomial time that satisfies each constraint as an NAE-3-SAT instance. PCSPs are a vast generalization of CSPs and capture key problems such as approximate graph and hypergraph coloring.

Since their formal introduction in [4] and subsequent detailed study in [13] and [6], there has been a flurry of recent works on PCSPs [1–3, 5, 9, 11, 12, 15, 16, 18, 24]. These have led to a rich and still developing theory aimed at classifying the complexity of PCSPs, by tying their (in)tractability to the symmetries associated with their defining relations, and understanding the power and limitations of various algorithmic approaches influential for CSPs in the context of PCSPs.

Against this backdrop, we initiate the study of robust algorithms for PCSPs. The motivation is two-fold. First, as algorithms resilient to a small noise in the input, robust algorithms are important in their own right. Second, in the CSP world, the existence of efficient robust algorithms is equivalent to having bounded width and being decided by \( O(1) \) levels of Sherali-Adams for CSPs [39]. It is also proven [40] to be equivalent to being decided by the basic semi-definite programming (SDP) relaxation [39]. Here, we say that the basic SDP decides a CSP if for every instance \( \Phi \) of the CSP, \( \Phi \) has an assignment satisfying all the constraints if and only if there is a vector solution satisfying all the constraints in the SDP relaxation. For CSPs, therefore, the study of robust algorithms sheds light on, and in fact, precisely captures, the power of the most popular algorithmic approaches. Robust algorithms for PCSPs provide a rich context to understand how well these algorithmic tools generalize beyond CSPs.

The main question that we are interested in this work is the following.

**Question 1.1. Which PCSPs admit polynomial time robust algorithms?**

As is the case with CSPs, a natural approach to characterize which PCSPs have robust algorithms is via the bounded width of PCSPs. However, it turns out that bounded width for PCSPs is weaker than having robust algorithms. Concretely, Atserias and Dalmau [2] have proved recently that the PCSP (1-in-3-SAT, NAE-3-SAT) does not have bounded width. Our work implies there is a robust algorithm for this PCSP. Atserias and Dalmau also proved that this PCSP is decided by \( O(1) \) levels of Sherali-Adams, and as we shall prove later, it is also decided by the basic SDP.\(^1\) On the other hand, Raghavendra’s framework of converting integrality gaps of CSPs to the hardness of approximation applies to PCPS as well [35]. In particular, his result implies that every PCSP that is not decided by the basic SDP does not admit a polynomial-time robust algorithm, assuming the Unique Games Conjecture [33]. This gives a powerful tool to show the absence of a polynomial time robust algorithm for a PCSP \( \Gamma \) (albeit under the Unique Games Conjecture): showing an integrality gap for the basic SDP relaxation of \( \Gamma \). With this connection, Question 1.1 naturally leads to the following question.

**Question 1.2. Which PCSPs are decided by the basic SDP relaxation?**

We make progress on Question 1.1 and Question 1.2 by studying the polymorphisms of PCSPs. Polymorphisms are closure properties of satisfying solutions to (Promise) CSPs. As a concrete example, consider the 2-SAT CSP: given an instance \( \Phi \) of 2-SAT over \( n \) variables \( x_1, x_2, \ldots, x_n \), suppose that \( u, v, w \) are three assignments to these variables satisfying all the constraints in \( \Phi \), then the assignment \( z = \text{MAJ}(u, v, w) \) for every \( i \in [n] \), also satisfies all the constraints in \( \Phi \). This shows that the Majority function on three variables is a polymorphism of the 2-SAT CSP. More generally, the Majority function on any odd number of variables is a polymorphism of the 2-SAT CSP. Similarly, the Parity function on any odd number of variables is a polymorphism of 3-LIN. On the other hand, there are no non-trivial polymorphisms for 3-SAT. Polymorphisms are the central objects in the Universal algebraic approach to CSPs [10, 19, 20, 31, 32, 42], which has then been extended to PCSPs [6, 13].

At a high level, the existence of non-trivial polymorphisms implies algorithms, and vice-versa. The key challenge is to precisely characterize which polymorphisms lead to algorithms. It is known that the polymorphism family of a PCSP fully captures its computational complexity, i.e., if there are PCSPs \( \Gamma, \Gamma' \) such that the polymorphism family of \( \Gamma \), \( \text{Pol}(\Gamma) \) is contained in \( \text{Pol}(\Gamma') \), then \( \Gamma' \) is formally easier than \( \Gamma \), i.e., there is a gadget reduction from \( \Gamma \) to \( \Gamma' \). It turns out that this gadget reduction preserves the existence of robust algorithms as well. Thus, Question 1.1 and Question 1.2 can be rephrased as Which polymorphisms lead to robust algorithms for PCSPs? Which polymorphisms lead to being decided by the basic SDP relaxation?

We make progress on these questions on two fronts: first, for a large class of Boolean symmetric\(^2\) PCSPs where we allow negation of variables, we characterize the polymorphisms that lead to robust algorithms. Our algorithms are based on novel rounding schemes for the basic SDP relaxation, and our hardness results are proved using integrity gaps for the basic SDP relaxation. Second, towards understanding the power of basic SDP for promise CSPs, we introduce a minon \( M \) and show that a PCSP \( \Gamma \) can be decided by basic SDP if and only if there is a minon homomorphism from \( M \) to the minon of polymorphisms of \( \Gamma \).

### 1.1 Our Results: Robust Algorithms and Hardness

As is the case with CSPs, if a PCSP is NP-Hard, then it does not admit polynomial time robust algorithms, assuming \( P \neq \text{NP} \). Thus, Question 1.1 is only relevant for PCSPs that can be solved in polynomial time. A large class of PCSPs for which polynomial time solvability has been fully characterized is the Boolean symmetric PCPS. In [13], the authors showed that a Boolean symmetric PCSP with folding (i.e., we allow negating the variables) can be solved in

---

\(^1\)We say that the basic SDP decides a PCSP (formally defined in Section 2) if for every instance \( \Phi \) of the PCSP, if there is a vector solution satisfying all the strong constraints in \( \Phi \), then \( \Phi \) has an assignment satisfying all the weak constraints.

\(^2\)A predicate \( P \) is symmetric if for every satisfying assignment \( (x_1, \ldots, x_k) \) to \( P \), any permutation of that assignment also satisfies \( P \). For a Boolean predicate whether an assignment satisfies a predicate depends only on the Hamming weight. A PCSP is said to be symmetric if all the predicates in the template are symmetric.
polynomial time if and only if it contains at least one of Alternating-Threshold (AT), Majority (MAJ) or Parity polymorphisms of all odd arities.3

Robust Algorithms. Our main algorithmic result shows that in two of these cases when the PCSP has MAJ or AT polymorphisms of all odd arities, the PCSP admits a robust algorithm.

Theorem 1.3. Every Boolean folded PCSP \(\Gamma\) that contains AT or MAJ polymorphisms of all odd arities admits a polynomial time robust algorithm. In particular,

1. If \(\Gamma\) contains MAJ polymorphisms of all odd arities, then for every \(\varepsilon > 0\), there exists a polynomial time algorithm that given an instance of \(\Gamma\) that is promised to have a solution satisfying \(1 - \varepsilon\) fraction of the constraints, outputs a solution satisfying \(1 - O(\varepsilon \log \frac{1}{\varepsilon})\) fraction of the constraints.4

2. If \(\Gamma\) contains AT polymorphisms of all odd arities, then for every \(\varepsilon > 0\), there exists a polynomial time algorithm that outputs a solution satisfying \(1 - \frac{\log \log \frac{1}{\varepsilon}}{\log \frac{1}{\varepsilon}}\) fraction of the constraints on an instance promised to have a solution satisfying \(1 - \varepsilon\) fraction of the constraints.

Similar to the robust algorithms for CSPs [8, 21, 43], our robust algorithms for PCSPs with MAJ and AT polymorphisms are based on rounding the basic SDP relaxation. The main challenge here is to obtain robust algorithms for a large class of PCSPs without access to the predicates and just using the properties of their polymorphisms. We achieve this using a combination of polymorphic tools where we use the fact that the PCSP contains MAJ or AT polymorphisms to deduce structural properties of the underlying predicate pairs, and SDP rounding tools where we then use these structural properties to get a robust algorithm.

For MAJ polymorphisms, we first reduce the problem to the case when every weak predicate is of the form \(x_1 \land \cdots \land x_n\). We then show that the robust algorithm of Charikar, Makarychev, and Makarychev [21] for the 2-SAT CSP generalizes to these classes of PCSPs. While the analysis of [21] is tailored towards 2-SAT, we give a completely different analysis that is not based on predicates and instead uses the existence of MAJ polymorphisms as a black box. Similarly, for the AT polymorphisms, we first use the properties of AT polymorphisms to reduce to the weighted hyperplane PCSP that generalizes the (1-in-3-SAT, NAE-3-SAT) PCSP. We then give a robust algorithm for the weighted hyperplane PCSP based on a random threshold rounding technique. A detailed overview of our algorithmic ideas appears in [14].

Hardness Results. Unlike our robust algorithms, which work for general Boolean PCSP with the said polymorphisms, in our hardness results, we rely on the symmetry of the predicates defining the PCSP. Furthermore, we assume that the PCSP contains a single predicate pair \(\Gamma = (P, Q)\) that does not admit AT or MAJ polymorphisms of all odd arities. We show that for such Boolean symmetric folded PCSPs, the basic SDP relaxation has an integrality gap with perfect completeness, i.e., there is a finite instance on which the SDP relaxation satisfies all the strong constraints with zero error but the instance is not satisfiable even using the weak constraints. By

Raghavendra’s framework connecting SDP gaps and Unique-Games hardness [35], the integrality gap rules out robust satisfaction algorithms (under the Unique Games conjecture (UGC) [33]).

Theorem 1.4. Let \(\Gamma = (P, Q)\) be a pair of symmetric Boolean predicates such that \(AT_{L_1}, \text{MAJ}_{L_2} \not\in \text{Pol}(\Gamma)\) for some odd integers \(L_1, L_2\). Then, under the UGC, unless \(P = \text{NP}\), there is no polynomial time robust algorithm for the PCSP associated with \(\Gamma\), where we allow negating variables and setting constants in the constraints.

Similar to our algorithmic result, we first use the properties of the polymorphisms to reduce to a small set of fixed template PCSPs. We obtain integrality gaps for these PCSPs for the basic SDP relaxation, which then implies robust hardness under the UGC. For PCSPs, strong integrality gaps [37, 38, 41] are known for the basic SDP relaxation and its strengthenings such as the Lasserre hierarchy, almost all of them being random constructions. For the case of PCSPs, analyzing the random constructions is trickier since we need to sample the constraints with a precise density such that there is a vector solution to the strong constraints, but the weak constraints are not satisfied. Instead, we take the opposite approach where we first construct the vector solution and then add all the constraints that the vector solution satisfies. This is similar in spirit to Feige and Schechtman’s integrality gap [26] for MAX-CUT where they first sampled \(n\) uniformly random points on a \(d\)-dimensional sphere and then added edges between every pair of points whose distance falls within a preset range.

In particular, we first construct an infinite integrality gap instance where the vertex set corresponds to the \(n\)-dimensional sphere \(S^n\) for a large integer \(n\) and there are constraints for every tuple of vertices whose corresponding vectors satisfy the SDP constraints. For the set of fixed template PCSPs that we study, we show that this instance is not satisfiable, even using weak constraints. A compactness argument then implies the existence of a finite integrality gap instance. As we shall see later, by using our minion characterization result, showing that the infinite instance has no satisfiable assignment is a necessary step to obtain a finite integrality gap instance. Toward showing that the infinite instance does not have an assignment satisfying all the weak constraints, we study colorings of the sphere \(f : S^n \rightarrow \{-1, +1\}\) and use a result of Matoušek and Rödl [34] from sphere Ramsey theory where the existence of monochromatic configurations in colorings of the sphere are studied. While their result directly applies to some PCSPs, for others, we combine their result with new techniques to prove the existence of structured configurations in sphere colorings. A more detailed overview appears in [14].

The Power of SDPs and Robust PCSP Algorithms. Both our algorithmic and hardness results crucially use the basic SDP relaxation. As our algorithms for the AT and MAJ polymorphisms are based on rounding the basic SDP, we get that every Boolean folded PCSP that contains AT or MAJ polymorphisms is decided by the basic SDP. On the hardness front, Theorem 1.4, shows that a vast majority of Boolean symmetric folded PCSPs without AT or MAJ polymorphisms cannot be decided by the basic SDP. This suggests a more general relation between the basic SDP and robust algorithms for PCSPs. At an intuitive level, for both the existence of robust algorithms and being decided by the basic SDP, the underlying requirement seems to be the existence of polymorphism

---

3Later an analogous result was shown without the folding restriction in [27].

4Here, \(O\) hides multiplicative poly logarithmic factors.
families that are robust to noise. While our results show that this is true for the PCSPs that we study in this paper (noise stability is one crucial aspect that distinguishes MAJ and AT from Parity.), we believe this is a more general phenomenon and motivates us to make the following conjecture.

**Conjecture 1.5.** A PCSP $\Gamma$ has a polynomial time robust algorithm if $\Gamma$ is decided by the basic SDP relaxation. Else, there is no polynomial time robust algorithm for $\Gamma$, unless $P = NP$.

As mentioned earlier, if there is an integrality gap for $\Gamma$ with respect to the basic SDP relaxation, then by Raghavendra’s [35] result, we get that $\Gamma$ does not have a polynomial time robust algorithm, assuming the Unique Games Conjecture. This already proves one direction of Conjecture 1.5. The other direction is more interesting: can we obtain robust algorithms for PCSPs just using the fact that basic SDP decides them? We remind the reader that the conjecture is already proven for CSPs, where the existence of robust algorithms [8] and decidability by basic SDP [40] are both shown to be equivalent to having bounded width.

### 1.2 Minion Characterization of Basic SDP

In addition to our concrete characterization of robust algorithms for a subfamily of PCSPs, we also present a novel algebraic characterization of which PCSPs can be decided via basic SDP. Originally, in the study of CSPs, such algebraic characterizations were structured as follows (e.g., [19, 42]).

- “Algorithm $A$ solves CSP($\Gamma$), if and only if there is a polymorphism $f \in \text{Pol}(\Gamma)$ with specific properties.”

Since the early days of PCSPs, it has been known that a single polymorphism cannot dictate hardness (c.f., [13]), and thus one must instead consider a sequence of polymorphisms (e.g., [16]):

- “Algorithm $A$ solves CSP($\Gamma$), if and only if there is an infinite sequence of polymorphisms $f_1, f_2, \ldots \in \text{Pol}(\Gamma)$ with specific properties.”

However, in many cases, such a characterization is unfeasible or unwieldy. Instead, a more general approach, pioneered by [6], captures the structure of polymorphism via a *minion*. A key property of the polymorphisms of a PCSP $\Gamma$ is that the function family $\text{Pol}(\Gamma)$ is closed under taking minors\(^5\). A minion is an abstraction based on this: it is a collection of objects each with an arity, and for every object $a$ of arity $m$, and a mapping $\pi : [m] \rightarrow [n]$, there is a unique object $b$ of arity $n$ that is said to be a minor of $a$ w.r.t. $\pi$. A *minion homomorphism* is a mapping between minions that preserves the minor operation. A powerful way to capture the limits of algorithms for PCSPs is via minion homomorphisms:

- “Algorithm $A$ solves CSP($\Gamma$), if and only if there is a minion homomorphism from $M_A$ to $\text{Pol}(\Gamma)$.”

Many recent papers [16, 23, 24] have proven such characterizations in various contexts. Our contribution to this line of work is showing that the basic SDP can be captured by a minion, which we call $M_{\text{SDP}}$.

**Theorem 1.6.** The basic SDP decides a PCSP $\Gamma$ if and only if there is a minion homomorphism from $M_{\text{SDP}}$ to $\text{Pol}(\Gamma)$.

\(^5\)A function $f : D^n_a \rightarrow D_b$ of arity $n$ is said to be a minor of another function $g : D^m_a \rightarrow D_b$ of arity $m$ with respect to a mapping $\pi : [m] \rightarrow [n]$ such that $f(x_1, x_2, \ldots, x_n) = g(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(m)})$ for every $x \in D^m_a$.

We note that a similar minion was concurrently and independently discovered by Ciardo-Zivny [25]. The theorem applies equally to Boolean and non-Boolean PCSPs.

The construction of the $M_{\text{SDP}}$ minion is inspired by the vector interpretation of solutions to the basic SDP. Each object in the minion is a collection of orthogonal vectors which sum to a reference vector $v_0$. The minors involve adding groups of vectors together. Having a minion homomorphism from $M_{\text{SDP}}$ to $\text{Pol}(\Gamma)$ implies that there are polymorphisms of $\Gamma$ whose minors behave exactly like combining orthogonal vectors.

Proving Theorem 1.6 has a few technical hurdles. One challenge is that SDP solutions may require vectors of an arbitrarily large dimension. For these arbitrarily-large dimensional relationships to be captured in our minion, we have that the families of vectors making up $M_{\text{SDP}}$ reside in a (countably) infinite-dimensional vector space. Similar techniques have been used in other minion constructions [23, 24].

Another challenge that appears specifically unique to this paper is that a Basic SDP solution gives a vector corresponding to each variable, but for the proof to go through additional vectors are needed which correspond to the constraints. (The variable vectors are "projections" of the constraint vectors.) Obtaining such constraints would typically be done via Sum-of-Squares or a related routine, but we prove that including such vector constraints are without loss of generality. That is, any basic SDP solution can be extended to a solution that includes constraint vectors without modifying the original variable vectors. This gives us enough vector structure to prove that the minion homomorphism corresponds to the basic SDP solution.

### Relation with Sphere Colorings

By a result of [6], there is a minion homomorphism $M_{\text{SDP}} \rightarrow \text{Pol}(\Gamma)$ if and only if there is an assignment satisfying all the constraints in a "universal" instance of PCSP($\Gamma$) known as a *free structure*. In the case that $\Gamma$ is a Boolean folded PCSP, this free structure for $M_{\text{SDP}}$ turns out to be an instance where every possible unit vector is a variable. The constraints correspond to collections of vectors that satisfy the corresponding basic SDP constraints. This is precisely the same infinite instance that we use to show integrality gaps. Thus, the result of [6] translates to the Boolean folded PCSPs world as stating that a Boolean folded PCSP $\Gamma$ is decided by the basic SDP if and only if there is an assignment satisfying all the constraints in the infinite integrality gap instance. For the general theory of approximation of basic SDPs, similar constructs with sphere coloring being a ‘universal’ gap have appeared in the literature (e.g. in [17]).

### Organization of the Paper

We first start by introducing formal definitions and some general observations in Section 2 (experts familiar with SDPs and CSPs can skip or just skim this section). We provide our algorithmic results (Theorem 1.3) in Section 3 and prove the hardness results (Theorem 1.4) in Section 4. We defer the study of the basic SDP minion to the full version [14] where we also give technical overview of our results and describe several open problems raised by our work.
2 PRELIMINARIES

Notations. We use \([n]\) to denote the set \([1, 2, \ldots, n]\). A predicate or a relation over a domain \(D\) of arity \(k\) is a subset of \(D^k\). For a relation \(P \subseteq [q]^k\) of arity \(k\), we abuse the notation and use \(P\) both as a subset of \([q]^k\), and also as a function \(P : [q]^k \rightarrow \{0, 1\}\). We use boldface letters to denote vectors and roman letters to denote their elements, e.g., \(x = (x_1, x_2, \ldots, x_k)\). We have \(S^x := \{v \in \mathbb{R}^n : ||v||_1 = 1\}\). For a vector \(v \in D^k\) and a function \(f : D^k \rightarrow D^k\) to denote \((f(v_1), f(v_2), \ldots, f(v_k))\).

For a vector \(x \in \{-1, 1\}^k\), we use \(hw(x)\) to denote the number of \(+1\)s in \(x\), i.e., \(hw(x) = \frac{\sum_{i=1}^{k} x_i}{k}\). For \(S \subseteq \{0, 1, \ldots, k\}\), we use \(Ham(S)\) to denote \(\{x \in \{-1, +1\}^k : hw(x) \in S\}\). We use \(NAE_k\) to denote the set \(Ham_1(2, \ldots, k-1)\), and \(k\)-SAT to denote the set \(Ham(1, 2, \ldots, k)\). For vectors \(x, y \in \mathbb{R}^n\), we use \(x \cdot y\) and \((x, y)\) interchangeably to denote \(\sum_{i} x_iy_i\).

2.1 PCSPs and Polymorphisms

We first define Constraint Satisfaction Problems (CSP).

**Definition 2.1.** (CSP) Let \(\Gamma = \{P_1, P_2, \ldots, P_l\}\) be a finite set of predicates over a domain \(D\), where \(P_i \subseteq D^{k_i}\). In an instance \(\Phi = (V, C)\) of CSP(\(\Gamma\)), the Constraint Satisfaction Problem (CSP) associated with the predicate set \(\Gamma\), we have a set of \(n\) variables \(V = \{u_1, u_2, \ldots, u_n\}\) that are to be assigned values from \(D\). There are \(m\) constraints \(C = \{C_1, C_2, \ldots, C_m\}\) each consisting of a tuple of variables \(C_j = (u_{j,1}, u_{j,2}, \ldots, u_{j,l_j})\) which are \(V^{l_j}\) and an associated predicate \(p^{(j)} \in \Gamma\) of the same arity \(l_j\). An assignment \(\sigma : V \rightarrow D\) is said to satisfy the constraint \(C_j\) if \(\sigma(C_j) \in p^{(j)}\). There are two computational problems associated with CSP(\(\Gamma\)).

1. In the decision version of CSP(\(\Gamma\)), the objective is to decide if there is an assignment that satisfies all the constraints.

2. In the search version of CSP(\(\Gamma\)), the objective is to find an assignment that satisfies all the constraints.

We next define Promise Constraint Satisfaction Problems (PCSP).

**Definition 2.2.** *(PCSP)* In a Promise Constraint Satisfaction Problem \(PCSP(\Gamma)\) over a pair of domains \(D_1, D_2\), we have a finite set of \(n\) variables \(V = \{u_1, u_2, \ldots, u_n\}\) and \(m\) constraints \(C = \{C_1, C_2, \ldots, C_m\}\) each consisting of a tuple of variables \(C_j = (u_{j,1}, u_{j,2}, \ldots, u_{j,l_j})\) which are \(V^{l_j}\) and an associated predicate pair \((p^{(j)}, Q^{(j)})\) of \(\Gamma\) of the same arity \(l_j\). An assignment \(\sigma_1 : V \rightarrow D_1\) is said to strongly satisfy the constraint \(C_j\) if \(\sigma_1(C_j) \in p^{(j)}\), and an assignment \(\sigma_2 : V \rightarrow D_2\) is said to weakly satisfy the constraint \(C_j\) if \(\sigma_2(C_j) \in Q^{(j)}\). The following are computational problems associated with PCSP(\(\Gamma\)).

1. In the decision version of PCSP(\(\Gamma\)), given an input instance \(\Phi = (V, C)\) of PCSP(\(\Gamma\)), the objective is to distinguish between the two cases.

   a. There is an assignment \(\sigma_1 : V \rightarrow D_1\) that strongly satisfies all the constraints.

   b. There is no assignment \(\sigma_2 : V \rightarrow D_2\) that weakly satisfies all the constraints.

   (2) In the search version of PCSP(\(\Gamma\)), given an input instance \(\Phi = (V, C)\) of PCSP(\(\Gamma\)) with the promise that there is an assignment \(\sigma_1 : V \rightarrow D_1\) that strongly satisfies all the constraints, the objective is to find an assignment \(\sigma_2 : V \rightarrow D_2\) that weakly satisfies all the constraints.

   (3) In the robust version of PCSP(\(\Gamma\)), given an input instance \(\Phi = (V, C)\) of PCSP(\(\Gamma\)) with the promise that there is an assignment \(\sigma_1 : V \rightarrow D_1\) that strongly satisfies \(1 - \epsilon\) fraction of the constraints, the objective is to find an assignment \(\sigma_2 : V \rightarrow D_2\) that weakly satisfies at least \(1 - f(\epsilon)\) fraction of the constraints for some monotone, nonnegative function \(f\) that satisfies \(f(\epsilon) \rightarrow 0\) as \(\epsilon \rightarrow 0\).

In this paper, we restrict ourselves to Boolean PCSPs where both the domains are equal to \([-1, +1]\). Following the robust algorithms literature of CSPs, we allow the constraints to use the negation of variables and refer to such PCSPs as Boolean folded PCSPs.

**Definition 2.3.** *(Boolean folded PCSPs)* In a Boolean folded PCSP \(\Gamma\), we have a set of \(n\) variables \(V = \{u_1, u_2, \ldots, u_n\}\) and an associated predicate pair \((p^{(j)}, Q^{(j)})\) of \(\Gamma\) of the same arity \(l_j\) such that for every \(i \in [l]\), \(p^{(i)}\) is a subset of \([0, 1]^k\) with the promise that there is an assignment \(\sigma_1 : V \rightarrow [-1, +1]\) that strongly satisfies \(1 - \epsilon\) fraction of the constraints, the objective is to find an assignment \(\sigma_2 : V \rightarrow [-1, +1]\) that weakly satisfies at least \(1 - f(\epsilon)\) fraction of the constraints for some function \(f\) that satisfies \(f(\epsilon) \rightarrow 0\) as \(\epsilon \rightarrow 0\). For simplicity, we use "robust algorithm for \(\Gamma\)" to refer to an algorithm that solves the robust version of PCSP(\(\Gamma\)).

Similar to the general PCSPs, for a Boolean folded PCSP \(\Gamma\), in the robust version of PCSP(\(\Gamma\)), given an input instance \(\Phi = (V, C)\) of PCSP(\(\Gamma\)) with the promise that there is an assignment \(\sigma_1 : V \rightarrow [-1, +1]\) that strongly satisfies \(1 - \epsilon\) fraction of the constraints, the objective is to find an assignment \(\sigma_2 : V \rightarrow [-1, +1]\) that weakly satisfies at least \(1 - f(\epsilon)\) fraction of the constraints for some function \(f\) that satisfies \(f(\epsilon) \rightarrow 0\) as \(\epsilon \rightarrow 0\). For simplicity, we use "robust algorithm for \(\Gamma\)" to refer to an algorithm that solves the robust version of PCSP(\(\Gamma\)).

In our hardness results, we study Boolean folded PCSPs that are symmetric and idempotent. We say that a predicate \(P \subseteq [-1, +1]^k\) is symmetric if for every \(x, y \in [-1, +1]^k\) such that \(hw(x) = hw(y)\), we have \(x \in P\) if and only if \(y \in P\). A Boolean folded symmetric idempotent PCSP \(\Gamma\) is a Boolean folded PCSP in which every predicate involved is symmetric and we also allow the constraints to use constants. We give a formal definition below.

**Definition 2.4.** *(Boolean folded symmetric idempotent PCSPs)* A Boolean folded symmetric idempotent PCSP \(\Gamma = \{(P_1, Q_1), \ldots, (P_l, Q_l)\}\) where \(P_i \subseteq Q_i \subseteq [-1, +1]^k\) is referred to as symmetric and idempotent if the following hold.

1. *(Symmetric)* \(P_i, Q_i\) are symmetric for every \(i \in [l]\).

2. *(Idempotent)* We now allow the constraints to use \(+1\) and \(-1\) along with the literals \(V, \bar{V}\), i.e., each constraint \(C_j\) satisfies
Consider an assignment \( \sigma : V \mapsto \{-1,+1\} \), and let \( \sigma' : V \mapsto \{-1,+1\} \) be defined as \( \sigma'(u_i) = \sigma(u_i) \) and \( \sigma'(\overline{u}) = -\sigma(u) \) for every \( i \in [n] \), and \( \sigma'(b \cdot b') = b \cdot b' \in \{-1,+1\} \). The assignment \( \sigma' \) is said to strongly (and resp. weakly) satisfy the constraint \( C_j \) in \( \Phi \) if \( \sigma'(C_j) \in P^{(j)} \) (and resp. \( \sigma'(C_j) \in Q_j^{(j)} \)).

Associated with every PCSP, there are polymorphisms that capture the closure properties of the satisfying solutions to the PCSP. More formally, we can define the polymorphisms of a PCSP as follows.

**Definition 2.5.** (Polymorphisms of PCSPs) For PCSP(\( \Gamma \)) with \( \Gamma = \{(P_1,Q_1), (P_2,Q_2), \ldots , (P_t,Q_t)\} \) such that for every \( i \in [t] \), \( P_i \subseteq D_{i_1} \times D_{i_2} \times \cdots \times D_{i_k} \) is a polymorphism of arity \( k \) is a function \( f : D^n_{i_1} \times D^n_{i_2} \times \cdots \times D^n_{i_k} \rightarrow D_{i_1} \times D_{i_2} \times \cdots \times D_{i_k} \) that satisfies the below property for every \( i \in [t] \).

For all \( v_1, v_2, \ldots , v_k \in D_{i_1} \) satisfying \((v_1)_j, (v_2)_j, \ldots , (v_k)_j \) \( \in P_j \) for each \( j \in [n] \), we have \( f((v_1)_j, (v_2)_j, \ldots , (v_k)_j) \in Q_j \).

For a Boolean folded PCSP \( \Gamma \), we require that \( f : \{-1,+1\}^n \rightarrow \{-1,+1\} \) satisfies an additional property that \( f \) is folded, i.e., \( f(-v) = -f(v) \forall v \in \{-1,+1\} \). Similarly, for Boolean folded idempotent PCSPs, we require that \( f \) is folded and idempotent, i.e., \( f(1,1,1,\ldots,1) = 1 \) and \( f(0,0,\ldots,0) = 0 \). We use \( \text{Pol}(\Gamma) \) to denote the family of all the polymorphisms of PCSP(\( \Gamma \)).

We extensively study Alternate-Threshold (AT) and Majority (MAJ) polymorphisms in this paper:

1. For an odd integer \( L \geq 1 \) and \( x \in \{-1,+1\}^L \), we have \( \text{AT}_L(x) = +1 \) if \( x_1 \oplus x_2 \oplus x_3 \oplus \cdots \oplus x_L > 0 \), and \( -1 \), otherwise.
2. For an odd integer \( L \geq 1 \) and \( x \in \{-1,+1\}^L \), we have \( \text{MAJ}_L(x) = +1 \) if \( x_1 \oplus x_2 \oplus x_3 \oplus \cdots \oplus x_L = 0 \), and \( -1 \), otherwise.

We also use \( \text{AT}_L(x_1, x_2, \ldots , x_L) \) for \( x \in \{-1,+1\}^L \) (similarly for MAJ) when applying \( \text{AT}_L \) coordinatewise. For a predicate \( P \subseteq \{-1,+1\}^L \), we use \( \text{AT}_L(P) \) to denote the set \( \bigcup_{x_1, x_2, \ldots , x_L \in P} \text{AT}_L(x_1, x_2, \ldots , x_L) \).

We say that \( T \subseteq \text{Pol}(\Gamma) \) (and resp. \( \text{MAJ}(\Gamma) \)) is in \( \text{Pol}(\Gamma) \) for every odd integer \( L \geq 1 \). For a predicate \( P \subseteq \{-1,+1\}^L \), we use \( O_{\text{AT}}(P) \) (and similarly \( O_{\text{MAJ}}(P) \)) to denote the set \( \bigcup_{L \geq 1} \text{AT}_L(P) \).

**Relaxations of PCSPs.** We say that a PCSP \( \Gamma' \) is a relaxation of another PCSP \( \Gamma \) if \( \text{Pol}(\Gamma') \subseteq \text{Pol}(\Gamma) \). If \( \Gamma' \) is a relaxation of \( \Gamma \), then there is a gadget reduction from \( \Gamma' \cup \{(z,=)\} \) to \( \Gamma \cup \{(z,=)\} \), where \( = \) denotes the equality predicate in the relevant domain. More formally, it is referred to as \( \Gamma' \cup \{(z,=)\} \) is positive primitive promise (ppp)-definable from \( \Gamma \cup \{(z,=)\} \).

**Definition 2.6.** (PPP-definability of PCSPs [13, 22]) We say that a PCSP \( \Gamma' = (P',Q') \) containing a single pair of predicates of arity \( k \) is ppp-definable from a PCSP \( \Gamma \) over the same domain pair if there exists a fixed constant \( l \) and an instance \( \Phi \) of PCSP(\( \Gamma' \)) over \( k + l \) variables \( u_1, u_2, \ldots , u_k, x_1, x_2, \ldots , x_l \) such that

1. If \( (x_1, x_2, \ldots , x_k) \in P' \), then there exist \( y_1, y_2, \ldots , y_l \) such that the assignment \((x_1, x_2, \ldots , x_k, y_1, \ldots , y_l)\) strongly satisfies all the constraints in \( \Phi \).
2. If there is an assignment \((z_1, \ldots , z_{k+l})\) weakly satisfying all the constraints in \( \Phi \), then \((z_1, z_2, \ldots , z_l) \in Q' \).

More generally, we say that \( \Gamma' \) is ppp-definable from \( \Gamma \) if every predicate pair in \( \Gamma' \) is ppp-definable from \( \Gamma \).

Note that if \( \Gamma' \) is ppp-definable from \( \Gamma \), then the decision version of PCSP(\( \Gamma' \)) can be reduced to PCSP(\( \Gamma \)) in polynomial time. We now observe that the same holds for the robust version as well. More formally, we have the following proposition.

**Proposition 2.7.** Suppose that the PCSP(\( \Gamma' \)) over a pair of domains \( D_1, D_2 \) is ppp-definable from \( \Gamma \) over the same domain pair. If \( \Gamma' \) has a polynomial time robust algorithm that finds an assignment weakly satisfying \( 1 - f(e) \) fraction of the constraints on instances promised to have an assignment strongly satisfying \( 1 - e \) fraction of the constraints, then \( \Gamma' \) has a polynomial time robust algorithm as well, i.e., there is a polynomial time algorithm that finds an assignment weakly satisfying \( 1 - O_{\text{AT}}(f(e)) \) fraction of the constraints on instances promised to have an assignment strongly satisfying \( 1 - e \) fraction of the constraints.

We defer the proof of Proposition 2.7 to [14].

### 2.2 The Basic SDP

We now describe the Basic SDP relaxation of an instance of a Boolean folded PCSP, similar to how it is presented in [35]. Consider an instance \( \Phi = (V, C) \) of a Boolean folded PCSP \( \Gamma \) where \( V = \{u_1, u_2, \ldots , u_n\} \) and \( C = \{C_1, C_2, \ldots , C_m\} \) with the constraint \( C_j \) using the predicate pair \((p^{(j)}, q^{(j)})\) for \( j \in [m] \). We have a variable \( v_i \) associated with each variable \( u_i \in V \). If a constraint \( C_j \) uses a negated literal \( \overline{v}_i \), we use the vector \(-v_i \) in the first moment and second moment equations of \( C_j \). Towards this, for a literal \( x \in \{u_1, u_2, \ldots , u_n, \overline{u}_1, \overline{u}_2, \ldots , \overline{u}_n\} \), we define \( v(x) = v_i \) if \( x = u_i \), and \( v(x) = -v_i \) if \( x = \overline{u}_i \).

\[
\begin{align*}
\text{minimize:} & \sum_{j=1}^{m} \epsilon_j \\
\text{subject to:} & \epsilon_j \geq 0 \forall j \in [m] \\
& \lambda_j(f) \geq 0 \forall j \in [m], \ f : C_j \rightarrow \{-1,+1\} \\
& \sum_{f \in C_j \rightarrow \{-1,+1\}} \lambda_j(f) = 1 - \epsilon_j \forall j \in [m] \\
& \sum_{f \in C_j \rightarrow \{-1,+1\}, f(C_j) \in P^{(j)}} \lambda_j(f) = 1 - \epsilon_j \forall j \in [m] \\
& \|v_i\|_2^2 = 1 \forall i \in \{0,1, \ldots , n\}
\end{align*}
\]

First moments.

\[
\mathbf{v}(x) \cdot \mathbf{v}(x) = \sum_{f \in C_j \rightarrow \{-1,+1\}} \lambda_j(f) f(x) f(x') \forall j \in [m], \ x, x' \in C_j
\]

Second moments.

\[
\mathbf{v}(x) \cdot \mathbf{v}(x') = \sum_{f \in C_j \rightarrow \{-1,+1\}} \lambda_j(f) f(x) f(x') \forall j \in [m], \ x, x' \in C_j
\]

We say that basic SDP is feasible on \( \Phi \) if the above objective function is zero on \( \Phi \). We show that the SDP is feasible if there is an assignment that strongly satisfies all the constraints of \( \Phi \).
**Proposition 2.8.** Suppose that \( \Phi = (V, C) \) is an instance of a Boolean folded PCSP such that there is an assignment \( \sigma : V \to \{-1, +1\} \) that strongly satisfies all the constraints. Then, the basic SDP is feasible on \( \Phi \).

**Proof.** We set \( v_0 = 1 \), and \( v_i := \sigma(u_i) \in \mathbb{R} \) for every \( i \in [n] \). Let \( \sigma' : V \cup \overline{V} \to \{-1, +1\} \) be defined as \( \sigma'(u_i) = \sigma(u_i) \) and \( \sigma'(\overline{u}_i) = -\sigma(u_i) \) for every \( i \in [n] \). For a \( j \in [m] \) and \( f : C_j \to \{-1, +1\} \), we set \( \lambda_j(f) = 1 \) if \( f(x) = \sigma'(x) \) for every \( x \in C_j \), and we set \( \lambda_j(f) = 0 \) otherwise. These variables satisfy all the constraints in the basic SDP relaxation with \( \epsilon_j = 0 \) for all \( j \in [m] \).

More generally, we get that if there is an assignment that strongly satisfies \( 1 - \epsilon \) fraction of the constraints in \( \Phi \), the objective value of the above relaxation is at most \( \epsilon m \), for every \( \epsilon \geq 0 \). On the other hand, if the basic SDP is feasible for an instance \( \Phi \) of a PCSP, it doesn’t necessarily imply that \( \Phi \) has an assignment weakly satisfying all the constraints. For some PCSPs however, this is indeed the case, and we say that such PCSPs are decided by the basic SDP.

**Definition 2.9.** We say that the basic SDP decides the PCSP \( \Phi \) if for every instance \( \Phi \) such that the basic SDP is feasible on \( \Phi \), there is an assignment to \( \Phi \) that weakly satisfies all the constraints.

We remark that polynomial-time SDP solving algorithms can only compute the objective to within \( 1/\text{poly}(n) \) accuracy [28]. For the sake of robust algorithms, this issue is not relevant: if an instance \( \Phi \) has a solution strongly satisfying \( 1 - \epsilon \) fraction of the constraints, we can find a vector solution to the basic SDP with error at most \( (\epsilon + C)m \) in polynomial time, for arbitrarily small constant \( C > 0 \).

### 2.3 Elementary Properties of Gaussians

We prove a couple of elementary properties of Gaussian distribution that we use later. First, we prove the following anti-concentration inequality for the standard Gaussian random variable.

**Proposition 2.10.** Suppose that \( X \sim N(0, 1) \) has the standard Gaussian distribution. Then, for every \( \epsilon \geq 0 \),
\[
\Pr(|X| \leq \epsilon) \leq \epsilon.
\]

**Proof.** We have
\[
\Pr(|X| \leq \epsilon) = \int_{-\epsilon}^{\epsilon} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx \leq \int_{-\epsilon}^{\epsilon} \frac{1}{\sqrt{2\pi}} dx \leq \epsilon. \quad \Box
\]

We also need the following concentration inequality.

**Proposition 2.11.** Suppose that \( X \sim N(0, \sigma^2) \) has Gaussian distribution with variance \( \sigma^2 \). Then, for every \( t \geq 0 \),
\[
\Pr(X \geq t) \leq e^{-\frac{t^2}{2\sigma^2}}.
\]

### 3 ROBUST ALGORITHMS

#### 3.1 CMM is a Robust Algorithm when Majority is a Polymorphism

We restate Theorem 1.3 for the case of MAJ polymorphisms.

**Theorem 3.1.** Let \( \Gamma = \{(P_i, Q_i) : i \in [d]\} \) be a Boolean folded PCSP with MAJ \( \subseteq \text{Pol}(\Gamma) \). For every \( \epsilon > 0 \), there is a randomized polynomial time algorithm that given an instance \( \Phi \) of PCSP(\( \Gamma \)) that is promised to have an assignment satisfying \( 1 - \epsilon \) fraction of the constraints, finds an assignment to \( \Phi \) that satisfies \( 1 - \tilde{O}(\epsilon^{1/3}) \) fraction of the constraints in expectation.

We now prove Theorem 3.1. Our strategy is to reduce the problem into a special case when every predicate pair in the PCSP \( \Gamma \) is of the form \( \{P, Q\} : \Gamma = \{-1, +1\}^k \setminus \{-1, -1, \ldots, -1\} \} \), and then use the algorithm of Charikar, Makarychev, and Makarychev [21].

For ease of notation, we use \( O(\cdot) \) instead of \( \tilde{O}(\cdot) \) when \( \Gamma \) is clear from the context. We first get rid of all the constraints that use a predicate pair \( (P, Q) \) where \( P \subset Q = \{-1, +1\}^k \) for some integer \( k \) since these constraints are trivially satisfied by any assignment. Suppose that there are \( m \) constraints in \( \Phi \), and \( m' = \alpha m \) of them use predicates of the form \( (P, Q) \) where \( P \subset Q = \{-1, +1\}^k \). We consider the instance \( \Phi' \) containing \( m - m' \) constraints obtained by deleting the constraints that use predicates of the form \( (P, Q) \) where \( P \subset Q = \{-1, +1\}^k \). In the instance \( \Phi' \), we are promised that there is a solution satisfying \( m - m' - \epsilon m \) constraints, i.e., \( 1 - \frac{\epsilon}{\alpha} \) fraction of the constraints. We use the algorithm that we will present later in the subsection on the instance \( \Phi' \) to get an assignment weakly violating at most \( \tilde{O}(m - m') \) constraints. The same assignment weakly violates at most
\[
\tilde{O}\left(\frac{m - m'}{\epsilon^{1/3}}\right) = \tilde{O}\left(\left(\frac{\epsilon}{1 - \alpha}\right)^{1/3}\right) (1 - \alpha)m \leq \tilde{O}(\epsilon^{1/3})m
\]
constraints in \( \Phi \). Thus, it suffices to study Boolean folded PCSPs where no predicate pair is of the form \( (P, \{-1, +1\}^k) \).

We further transform the instance into one in which every predicate pair is of the form \( (P, \{-1, +1\}^k) \setminus \{-1, -1, \ldots, -1\} \).

**Lemma 3.2.** Fix \( \epsilon > 0 \), and consider a Boolean folded PCSP \( \Gamma = \{(P_i, Q_i) : i \in [d]\} \) where \( P_i \subset Q_i \subseteq \{-1, +1\}^k \) for every \( i \in [d] \). Given an instance \( \Phi \) of \( \Gamma \), there is a polynomial time algorithm that outputs an instance \( \Phi' \) of a Boolean folded PCSP \( \Gamma' = \{(P_i', Q_i') : i \in [d]\} \) over the same variable set \( V \) such that the following hold.

1. (Completeness) If an assignment \( \sigma : V \to \{-1, +1\} \) strongly satisfies \( 1 - \epsilon \) fraction of the constraints in \( \Phi \), then \( \sigma \) strongly satisfies at least \( 1 - O(\epsilon) \) fraction of the constraints in \( \Phi' \) as well.
2. (Soundness) If an assignment \( \sigma : V \to \{-1, +1\} \) weakly satisfies \( 1 - \epsilon \) fraction of the constraints in \( \Phi \), then \( \sigma \) weakly satisfies at least \( 1 - O(\epsilon) \) fraction of constraints in \( \Phi \).
3. The resulting PCSP \( \Gamma' \) satisfies the two properties:
   a. For every \( i \in [d], \ Q_i' = \{-1, +1\}^k \setminus \{(\epsilon + 1), \ldots, -1\} \) for some positive integer \( k_i \).
   b. If \( \text{MAJ} \subseteq \text{Pol}(\Gamma) \), then \( \text{MAJ} \subseteq \text{Pol}(\Gamma') \).

We present the proof of Lemma 3.2 in the full version [14]. Given an instance \( \Phi \) of an \( \Gamma \), we transform it to an instance \( \Phi' \) of a PCSP \( \Gamma' \) using Lemma 3.2. If \( \Phi \) is promised to have an assignment strongly satisfying at least \( 1 - \epsilon \) fraction of the constraints, then \( \Phi' \) has an assignment strongly satisfying \( 1 - O(\epsilon) \) fraction of the constraints as well. If there is a polynomial time robust algorithm that outputs an assignment weakly satisfying \( 1 - f(\epsilon) \) fraction of the constraints, then we can use this assignment to obtain a robust algorithm.

---

We use \( \Theta \) to denote a hidden constant which depends on the specific template \( \Gamma \).
algorithm for $\Gamma$ as well. Thus, a polynomial time robust algorithm for $\Gamma'$ gives a polynomial time robust algorithm for $\Gamma$ as well.

For such a Boolean folded PCSP $\Gamma$ where every predicate pair is of the form $(P, \{\{\{1,1\}, \ldots, \{1,1\}\})$ with MAJ $\subseteq \text{Pol}(\Gamma)$, we show that the robust algorithm of Charikar, Makarychev, and Makarychev [21] for 2-SAT generalizes and gives a robust algorithm for Gamma as well. First, state their algorithm.

\begin{enumerate}
\item Given an instance $\Phi$ of $\Gamma$ containing $n$ variables $u_1, u_2, \ldots, u_n$, solve the basic SDP and obtain a set of vectors $v_0, v_1, \ldots, v_n$. Let $\mu \in \mathbb{R}^n$ denote the first moments and $\Sigma \in \mathbb{R}^{n \times n}$ be the gram matrix of these vectors.
\item $\mu_i = v_i \cdot v_0 \quad \forall i \in [n]$ \quad (1)
\item Sample an $n$ dimensional Gaussian $\xi \sim \mathcal{N}(0, \Sigma)$. (Note that $\Sigma$ is positive semidefinite.)
\item Set $\gamma = \epsilon^2$. \quad (2)
\item For each $i \in [n]$, round as follows
\[ \sigma(u_i) = \begin{cases} +1 & \xi_i \geq -\mu_i/\gamma, \\ -1 & \text{otherwise.} \end{cases} \quad (3) \]
\end{enumerate}

We shall prove the following guarantee about the algorithm.

**Theorem 3.3.** Let $\Gamma = \{(P_1, Q_1), \ldots, (P_1, Q_1)\}$ be a Boolean folded PCSP such that $\text{MAJ} \subseteq \text{Pol}(\Gamma)$ where $P_1 \subseteq Q_1 = \{\{\{1,1\}, \ldots, \{1,1\}\} \setminus \{\{1,1,\ldots,1\}\}$ for every $i \in [1]$. Let $\Phi$ be an instance of $\text{PCSP}(\Gamma)$ over $n$ variables and using $m$ constraints for which the basic SDP relaxation has a solution with error value at most $em$. Then, the assignment $\sigma$ output by the above CMM algorithm weakly satisfies $1 - \tilde{O}(\epsilon^{1/3})$ fraction of the constraints in expectation.

We analyze the probability that the output assignment weakly satisfies each constraint separately. Fix a constraint $C_j$ using the predicate pair $(P, Q)$ of arity $k$ with $P \subseteq Q$ and $Q = \{\{\{1,1\}, \ldots, \{1,1\}\} \setminus \{\{1,1,\ldots,1\}\}$ for every $i \in [1]$. Suppose that the basic SDP solution has error equal to $c$ on this constraint, i.e., $\epsilon_j = c$. Our goal is to upper bound the probability that the rounded solution violates the constraint $Q$ by a function of $c$ and $\epsilon$. Using the fact that the expected value of $c$ over all the constraints is at most $\epsilon$, we get our required robustness guarantee. More formally, we prove the following.

**Lemma 3.4.** Fix $j \in [m]$, and suppose that the basic SDP solution has an error value equal to $c$ on $C_j$, i.e., $\epsilon_j = c$. Then, the probability that $\sigma$ does not weakly satisfy $C_j$ is at most

\[ O\left(\sqrt{c} + \sqrt{\log \frac{1}{\epsilon} + 2c} \cdot \frac{1}{1 - \frac{c}{\epsilon}} \sqrt{\frac{1 + \epsilon}{2}} \right). \]

By summing over all the constraints, and using linearity of expectation, the expected number of constraints that are not weakly satisfied by $\sigma$ is at most

\[ O\left(\sqrt{cm} + \sum_{j \in [m]} \left(\sqrt{\log \frac{1}{\epsilon} + 2\epsilon_j} \cdot \frac{1}{1 - \frac{\epsilon_j}{\epsilon}} \right) \right). \]

As the basic SDP has a total error at most $em$, the average value of $\epsilon_j$ over $j \in [m]$ is at most $\epsilon$. Also note that the expression in Equation (1) is a concave function of $\epsilon_j$. Thus, using Jensen’s inequality, we get that the expected number of constraints that are not weakly satisfied by $\sigma$ is at most

\[ O\left(\sqrt{cm} + m \cdot \left(\sqrt{\log \frac{1}{\epsilon} + 2\epsilon} \cdot \frac{1}{1 - \frac{\epsilon}{\epsilon}} \right) \right) \leq O\left(m \epsilon^{1/3}\right) \]

This completes the proof of Theorem 3.3. In the rest of the subsection, we prove Lemma 3.4.

Let $P$ be the convex hull of $P$, where the tuples are viewed as vectors in $\mathbb{R}^k$. Let $P := \left\{ \sum_{a \in P} \lambda_a a : \lambda_a \geq 0 \forall a \in P, \sum_{a \in P} \lambda_a = 1 \right\}$.

We show the following lemma about $P$ using the fact that the PCSP $(P, Q)$ has Majority of all odd arities as polymorphisms. We recall that $\text{OMAJ}(P)$ denotes the set for a predicate $P \subseteq \{-1, +1\}^k$. The proof of the below lemma is in the full version [14].

**Lemma 3.5.** Let $P \subseteq \{-1, +1\}^k$ be a predicate such that $\{-1, -1, \ldots, -1\} \not\subseteq \text{OMAJ}(P)$.

Then, there is a hyperplane separating $P$ from the origin: there exists $w \in \mathbb{R}^k$, $w \geq 0$ and $\|w\|_1 = 1$ such that for every $a \in P$, $\langle a, w \rangle \geq 0$.

Now we use this lemma to complete the proof of Lemma 3.4. Recall that our goal is to lower bound the probability that the assignment $\sigma$ output by the above algorithm weakly satisfies the constraint $C_j$. Suppose that the constraint $C_j$ is on a tuple of literals $S_j = (x_{j,1}, x_{j,2}, \ldots, x_{j,k})$, and uses the predicate pair $(P, Q)$ where $P \subseteq Q = \{-1, +1\}^k \setminus \{(1,1,\ldots,1)\}$. We first simplify the notation a bit. Let $K = 2^k$. We order all the tuples in $\{-1, +1\}^k$ as $a_1, a_2, \ldots, a_k$.

\[ \{a_1, a_2, \ldots, a_k\} := \{-1, +1\}^k. \]

We can also view the tuple $a_i$ as a function $f_i : S_j \rightarrow \{-1, +1\}$ where $f_i(x_{j,p}) = a_{i,p}$. We use $p_1, p_2, \ldots, p_K$ to denote the probabilities assigned by the SDP solution corresponding to the $K$ local assignments $a_1, a_2, \ldots, a_K$.

\[ p_i := \lambda_i(f_i). \]

We have that each $p_i \geq 0$, $\sum_{i \in [K]} p_i = 1$ and $\sum_{i \in [K]} p_i = 1 - c$. We use Lemma 3.5, we get $w \in \mathbb{R}^k$ with $w \geq 0$ and $\|w\|_1 = 1$ such that $w^T a_i \geq 0$ for all $a_i \in P$. Combining this with the above properties of the basic SDP solution, we get the following.

\[ (1) \text{ (First moment)}: \quad \text{We have} \quad w^T \mu = \sum_{i \in [K]} p_i w^T a_i \geq -c \quad (\text{Using Lemma 3.5 and } -1 \leq w^T a_i \leq 1 \forall i \in [K]) \]
by proving

(2) (Second moment.) We have

$$w^T \Sigma w = \sum_{i \in k} p_i (w^T a_i)^2 \leq \sum_{i \in k, a_i \in p} p_i (w^T a_i)^2 + c$$

$$\leq \sum_{i \in k, a_i \in p} p_i w^T a_i + c \leq w^T \mu + 2c.$$ 

We do casework on the value of $w^T \mu$. First, consider the case that $w^T \mu \geq \kappa = \sqrt{\frac{\log \frac{1}{\epsilon}}{2}}.$ As $\|w\|_1=1$, and $w \geq 0$, there exists $i \in [k]$ such that $\mu_i \geq \kappa.$ As $\xi_i \sim N(0, 1),$ using Proposition 2.11, with probability at least $1 - \sqrt{e}$, we have $\xi_i \geq -\frac{d}{\sqrt{e}}.$ Thus, with probability at least $1 - \sqrt{e}$, the rounded solution satisfies $Q$.

Henceforth, we assume $w^T \mu < \kappa$. For notational convenience let $t = -\mu/\sqrt{e}$. We have

$$w^T t \leq \frac{c}{y} \quad (2)$$

and $\|w^T \Sigma w \| \leq \kappa + 2c.$

Note that $w^T \xi \sim N(0, w^T \Sigma w)$. Thus, using Proposition 2.11, with probability at least $1 - \sqrt{e}$, we have that

$$\|w^T \xi\| \leq O(\sqrt{(\kappa + 2c) \log \frac{1}{\epsilon}}) \quad (3)$$

Note that the rounded solution does not satisfy $Q$ only if $t \geq \xi$. We now upper bound the probability that this can occur. Together with Equation (2) and Equation (3), $t \geq \xi$ implies that

$$0 \leq w^T (t - \xi) \leq O\left(\sqrt{(\kappa + 2c) \log \frac{1}{\epsilon}} + \frac{c}{y}\right).$$

Take some coordinate with $w_i \geq 1/k$ and note that

$$t_i - \xi_i \in \left[0, O\left(\sqrt{(\kappa + 2c) \log \frac{1}{\epsilon}} + \frac{c}{y}\right)\right],$$

but this can only happen with probability $O\left(\sqrt{\kappa + c \log \frac{1}{\epsilon}} + \frac{c}{y}\right)$ using Proposition 2.10. Thus, the probability that the assignment $\sigma$ does not satisfy $Q$ is at most

$$O\left(\sqrt{e} + \sqrt{\kappa + c \log \frac{1}{\epsilon}} + \frac{c}{y}\right).$$

This completes the proof of Lemma 3.4.

3.2 Algorithm for AT

We now show how to combine the ideas in the previous two subsections to obtain a polynomial time robust algorithm for every Boolean folded PCSP $\Gamma \equiv \{P_1, Q_1\} \ldots \{P_k, Q_k\}$ with $\Gamma \in \text{Pol}(\Gamma)$. Similar to the MAJ polymorphisms case, we can without loss of generality assume that $Q_i \neq (-1, +1)^{k_i}$ for every $i \in [l]$. We first reduce an arbitrary PCSP $\Gamma$ with $\Gamma \in \text{Pol}(\Gamma)$ to a specific PCSP that we will work with. For a vector $w \in \mathbb{R}^k$, define $\text{sgn}(w)_i$ to be $1$ if $w_i \leq 0$ and $-1$ otherwise. Define $\Gamma_{AT}$ to be the following family of weighted hyperplane predicates:

$$\Gamma_{AT} := \{P_{w,b} := \{x \in \{-1, +1\}^k : w \cdot x = b\}, \quad Q_{w,b} := \{-1, +1\}^k \setminus \{\text{sgn}(w), -\text{sgn}(w)\} : b \in \mathbb{Q}, \quad w \in [0, 1]^k, \quad \text{sgn}(w) > b, \quad -w, \text{sgn}(w) < b\}$$

We observe that these predicates indeed have AT of all odd arities as polymorphisms. The proof of the claim appears in [14].

Claim 3.6. AT $\subseteq \text{Pol}(\Gamma_{AT})$

Let $\Gamma_{const}$ be the PCSP where constants can be specified. That is

$$\{((-1), (-1)), ((+1), (+1))\}.$$ We show that an arbitrary Boolean PCSP with AT polymorphisms can be reduced to the union of the weighted hyperplane and the constant predicates.

Lemma 3.7. Let $\Gamma = \{P_1, Q_1\} \ldots \{P_k, Q_k\}$ be any Boolean folded PCSP with $P_i \subseteq Q_i \subseteq \{-1, +1\}^k$ and $\Gamma \in \text{Pol}(\Gamma)$. Then, $\Gamma$ is pp-definable from a Boolean folded PCSP $\Gamma'$ with $\Gamma' \subseteq \Gamma_{AT} \cup \Gamma_{const}$.

We prove Lemma 3.7 in [14].

Recall that if a PCSP $\Gamma$ is pp-definable from another PCSP $\Gamma'$, if $\Gamma'$ has a polynomial time robust algorithm, then $\Gamma$ has a polynomial time robust algorithm as well (up to losing constant factors in the error parameter). In the rest of this section, we obtain a robust algorithm for a Boolean folded PCSP $\Gamma$ with $\Gamma \subseteq \Gamma_{AT} \cup \Gamma_{const}$, thereby obtaining a robust algorithm for every Boolean folded $\Gamma$ with $\Gamma \subseteq \text{Pol}(\Gamma)$. We state our algorithm below.

(1) Given an instance $\Phi$ of $\Gamma$ containing $n$ variables $V = \{u_1, u_2, \ldots, u_n\}$, solve the basic SDP and obtain a set of vectors $v_0, v_1, \ldots, v_n$.

(2) Sample a vector $\xi \in \mathbb{R}^n$ by choosing each coordinate independently from $N(0, 1)$.

(3) Choose $\delta$ uniformly at random from $\{p, rp, \ldots, r^n p\}$ where $p = e^{c_1}$, and $r = k = \Theta\left(\frac{\log \frac{1}{\epsilon}}{\log \log \frac{1}{\epsilon}}\right)$ such that $r^n p = e^{c_2}$. Here, $c_1$ and $c_2$ are two arbitrary real constants satisfying $0 < c_2 < c_1 < 0.25$.

(4) For every $i \in [n]$, let $v_i = a_i v_0 + v_{i'}$, where $v_{i'}$ is orthogonal to $v_0$. We output an assignment $\sigma$ as follows.

$$\sigma(u_i) = \begin{cases} -1, & \text{if } \langle \xi, v_i \rangle \geq 2\delta a_i |\langle \xi, v_0 \rangle|, \\ +1, & \text{otherwise}. \end{cases}$$

We now analyze the algorithm.

Theorem 3.8. Let $\Gamma$ be a Boolean folded PCSP such that $\Gamma \subseteq \Gamma_{AT} \cup \Gamma_{const}$. Let $\Phi$ be an instance of $\text{PCSP}(\Gamma)$ for which there is a basic SDP solution with average error at most $\epsilon$. Then, the assignment output by the above algorithm satisfies at least $1 - O\left(\frac{\log \frac{1}{\epsilon}}{\log \log \frac{1}{\epsilon}}\right)$ fraction of constraints of $\Phi$ in expectation.

Theorem 3.8 together with Lemma 3.2, Theorem 3.3 complete the proof of Theorem 1.3.

For ease of notation, we just use $O()$ instead of $O_{\epsilon}()$ when $\Gamma$ is clear from the context. As before, we prove Theorem 3.8 by proving a lower bound on the probability that a particular constraint is satisfied. Consider the constraint $C$ over the tuple $S = (x_1, x_2, \ldots, x_k)$ using the predicate pair $(P, Q)$ where $P \subseteq Q \subseteq \{-1, +1\}^k$, and let $c$ denote the error of the SDP solution on constraint $C$. As the average value of $c$ over all the constraints is at most $\epsilon$, using Markov’s inequality, at least $1 - \sqrt{e}$ fraction of the constraints have SDP error at most $\sqrt{e}$. We restrict our analysis to these constraints with SDP error $c \leq \sqrt{e}$ and show that the rounded solution violates the predicates $Q$ with probability at most $O\left(\frac{\log \log \frac{1}{\epsilon}}{\log \log \frac{1}{\epsilon}}\right)$, thereby proving Theorem 3.8.

We first consider the case when $P = Q$ and $P \subseteq \text{Ham}_k\{0, k\}$.
Lemma 3.9. Let $\Gamma$ be a Boolean folded PCSP with $\mathcal{T} \subseteq \text{Pol}(\Gamma)$, and let $\Phi$ be an instance of PCSP(\$). In the basic SDP solution of $\Phi$, suppose that the constraint $C$ using the predicate pair $(P, Q)$ has error at most $\sqrt{e}$, and $P = Q, P \subseteq \text{Ham}_k[0, k]$. Then, the assignment $\sigma$ output by the above algorithm does not weakly satisfy $C$ with probability at most $O\left(\frac{\log \log \frac{1}{\delta}}{\log \frac{1}{\epsilon}}\right)$.

We defer the proof of Lemma 3.9 to [14].

We now consider the case when $P = P_{w, k}$ and $Q = Q_{w, k}$. For ease of notation, we let $v_1, v_2, \ldots, v_k$ denote the vectors assigned to the literals in the constraint $C$. Let $K = 2^k$. As in the analysis of the CMM algorithm, we order all the tuples in $\{-1, +1\}^k$ as $a_1, a_2, \ldots, a_K$.

$$\{a_1, a_2, \ldots, a_K\} = \{-1, +1\}^k.$$ We also view the tuple $a_i$ as a function $f_i : S \rightarrow \{-1, +1\}$ where $f_i(x) = (a_i)$. We use $p_1, p_2, \ldots, p_k$ to denote the probabilities assigned by the SDP solution corresponding to the $K$ local assignments $a_1, a_2, \ldots, a_k$.

We have $p_i = \lambda_j(f_i)$. We then have that $p_i \geq 0$, $\sum_{i \in [k]} p_i = 1$ and $\sum_{i \in [k], a_i \notin P} p_i = 1 - \sqrt{e}$.

Let $v_i = \sum_{a_i \in [k]} w_i a_i v_i$, and let the component of $v_i$ along $v_0$ be $\alpha v_0$, and the component normal to $v_0$ be $v'_i$.

$$v_i = \alpha v_0 + v'_i, (v_0, v'_i, v'_j) = 0.$$ We use the first and second moments properties of the vectors $v_1, v_2, \ldots, v_k$ to get the following.

1. (First moments). We have

$$\alpha = \sum_{i \in [k]} w_i (v_i, v_0) = b + \kappa$$

where $\kappa = \sum_{j \in [K], a_j \notin P} p_j (1 - (w, a_j))$.

We have $|\kappa| = O(\sqrt{e})$.

2. (Second moments). We have

$$\|v_i\|^2 = \sum_{i \in [k]} w_i w_j (v_i, v_j) = \sum_{i \in [k]} p_j (1 - (w, a_j))^2 = b^2 + \kappa'$$

where $|\kappa'| = O(\sqrt{e})$.

Thus, we get $\|v_i\|^2 = \|v_i\|^2 - \alpha^2 = (b^2 + \kappa') - (b + \kappa)^2$ which is at most $O(\sqrt{e})$.

We are now ready to analyze the algorithm. We consider two cases separately:

Case 1. Suppose that there exists $i \in [k]$ such that $\|v'_i\|^2 \geq k \delta^2$. We claim that in this case, the rounded solution satisfies $Q$ with probability at least $1 - O(\frac{1}{\sqrt{e}})$.

Note that $(\zeta, \nu') \sim \mathcal{N}(0, \|v'_i\|^2)$ for every $j \in [k]$. Suppose that we have $\|v'_i\|^2 \geq \delta r^2$ for some $j \in [k]$. Using Proposition 2.10, this implies that $|\langle \zeta, \nu' \rangle| \geq \delta r$ with probability at least $1 - \frac{1}{r}$.

Furthermore, as $(\zeta, \nu) \sim \mathcal{N}(0, 1)$, using Proposition 2.11, we get that $|\langle \zeta, \nu \rangle| \leq r$ with probability at least $1 - O(\epsilon)$. Thus, with probability at least $1 - O(\frac{1}{\sqrt{e}})$, $x_j$ is set to be equal to $+1$ if $|\langle \zeta, \nu' \rangle| > 0$, and $-1$ otherwise.

Hence, to show that the rounded solution satisfies $Q$, it suffices to show that there exist $i_1, i_2 \in [k]$ such that $|\langle \zeta, \nu' \rangle| \geq \delta r$, $|\langle \zeta, \nu'_1 \rangle| \geq \delta r$, and $|\langle \zeta, \nu'_2 \rangle| \geq \delta r$ have opposite signs. As $\|v'_i\|^2 \geq k \delta^2$, with probability at least $1 - O(\frac{1}{\sqrt{e}})$, we have that $|\langle \zeta, \nu' \rangle| \geq k \delta r$. Recall that $\|v'_i\|^2 \leq \frac{\epsilon^2}{1 - \sqrt{e}} \leq 0.25 \delta^2$. Thus, $|\langle \zeta, \nu'_1 \rangle| \leq \delta r$ with probability at least $O(\frac{1}{\sqrt{e}})$. As $|\langle \zeta, \nu'_1 \rangle| \geq k \delta r$, there exists $i' \in [k], i' \neq i$ such that $|\langle \zeta, \nu'_1 \rangle| \geq k \delta r$, and $|\langle \zeta, \nu'_1 \rangle| \leq \delta r$ have opposite signs. Thus, with probability at least $1 - O(\frac{1}{\sqrt{e}})$, $i$ and $i'$ are rounded to different values, which implies that the rounded solution satisfies $Q$.

Case 2. Suppose that for every $i \in [k]$, we have $\|v'_i\|^2 \leq \frac{\epsilon^2}{2r^2}$.

As $(\zeta, \nu) \sim \mathcal{N}(0, \|v'_i\|^2)$, using Proposition 2.11, we get that with probability at least $1 - O(\frac{1}{\sqrt{e}})$, for every $i \in [k]$, $|\langle \zeta, \nu' \rangle| \leq \frac{\epsilon^2}{2r^2}$. On the other hand, using Proposition 2.10, we have that $|\langle \zeta, \nu \rangle| \leq \frac{1}{2} r$ with probability at least $1 - \frac{1}{r}$. Furthermore, as $\alpha^2 + \|v'_i\|^2 = 1$ for every $i \in [k]$, we get that $|\alpha| \geq 1 - \frac{1}{2} \geq \frac{1}{4}$ for every $i \in [k]$. Thus, with probability at least $1 - O(\frac{1}{\sqrt{e}})$, for every $i \in [k]$, $x_j$ is set to be $+1$ if $\alpha_i \geq 0$, and $-1$ otherwise. Combining this with the fact that $\sum_i w_i a_i = b + O(\sqrt{e})$, and $\sum_i w_i > b$ and $\sum_i w_i < -b$, for small enough $\epsilon$, we get the rounded solution has variables assigned $+1$ and $-1$.

Completing the Proof. We finish the proof by showing that with probability at least $1 - O(\frac{1}{\sqrt{e}})$, at least one of the above two cases holds. None of the above two cases hold if for some $i \in [k]$, we have

$$\frac{\delta}{2r^2} < \|v'_i\|^2 < k \delta r^2$$

Or equivalently,

$$\frac{\delta}{k r^2} < \|v'_i\|^2 = \frac{\delta}{2r^2} \frac{k}{\epsilon^2}$$

This holds with probability at most $O(\frac{1}{\sqrt{e}})$ for every value of $\|v'_i\|$ as we are picking $\delta$ from $\{p, rp, \ldots, r^k p\}$ uniformly at random with $\kappa = r$.

4 UNIQUE GAMES BASED HARDNESS

In this section, we prove Theorem 1.4.

First, we use the analysis of AT and MAJ polynomials for symmetric PCSPs with folding and idempotence in [13] to show that we can relax $\Gamma$ into one of five candidate PCSP types.

Lemma 4.1. Let $\Gamma = (P, Q)$ be a Boolean folded symmetric idempotent PCSP such that $\text{MAJ}_{l_1, \ldots, \text{AT}_{l_2}} \not\subseteq \text{Pol}(\Gamma)$ for some odd integers $l_1, l_2$. Then, there exists a Boolean folded PCSP $\Gamma' = (P, Q)$ that is $\text{ppp}$-definable from $\Gamma$ that is equal to either of the following:

1. $k$ is even, $b \in \{1, k - 1\}$, and $\Gamma_1 = (P, Q), P = \text{Ham}_k(\frac{1}{2}), Q = \text{Ham}_k(0, 1, \ldots, k) \setminus \{b\}$.
2. $k$ is odd, $l \in \{0, 1, \ldots, \frac{k - 1}{2}\}$, $\Gamma_2 = (P, Q), P = \text{Ham}_k[l, k l], Q = \text{Ham}_k(0, 1, \ldots, k - 1)$.
3. $\Gamma_3 = (P, Q), P = \text{Ham}_k(l, k), Q = \text{Ham}_k(1, 2, \ldots, k), l \neq 0, l \leq \frac{k - 1}{2}$.
(4) $\Gamma_l = (P, Q), P = \text{Ham}_k(l), Q = \text{Ham}_k(0, 1, \ldots, k) \setminus \{0, k - 1\}$, where $l \in \{1, 2, \ldots, k - 1\}, l \leq \frac{k - 1}{2}$.

(5) $\Gamma_b = (P, Q), P = \text{Ham}_k(1, k), Q = \text{Ham}_k(0, 1, \ldots, k) \setminus \{b\}$ for arbitrary $b \in \{0, 1, \ldots, k\}$.

We defer the proof of Lemma 4.1 to the full version [14].

Recall that if a PCSP $\Gamma'$ is ppp-definable from another PCSP $\Gamma$, if $\Gamma$ has a polynomial time robust algorithm, then $\Gamma'$ has a polynomial time robust algorithm as well (Proposition 2.7). Thus, to show Theorem 1.4, it suffices to show Unique Games hardness of obtaining robust algorithms for the PCSPs $\Gamma_1$–$\Gamma_5$. We achieve this by showing integrality gaps for the basic SDP relaxation of them. Raghavendra’s result for CSPs [35] shows that integrality gaps for the basic SDP relaxation can be translated to Unique Games Conjecture (UGC) [33] based inapproximability results. In fact, his result is verbatim applicable to Promise CSPs as well.

**Theorem 4.2 (Special Case of [35] for Boolean folded PCSPs when the SDP is feasible).** Suppose that for a Boolean folded PCSP $\Gamma$, there is a finite integrality gap for the basic SDP relaxation, i.e., there is a finite instance $I$ of PCSP($\Gamma$) on which the basic SDP relaxation is feasible but there is no assignment that weakly satisfies $I$. Then, there exists a constant $s < 1$ that is a function of $\Gamma, I$ such that the following decision problem is NP-hard for sufficiently small $\epsilon, \delta > 0$, assuming UGC. Given an instance $\Phi$ of $\Gamma$, distinguish between the two cases:

1. (Completeness.) There exists an assignment that strongly satisfies $1 - \epsilon$ fraction of the constraints in $\Phi$.
2. (Soundness.) No assignment weakly satisfies $s + \delta$ fraction of the constraints in $\Phi$.

Thus, to show Theorem 1.4, our goal is to show the existence of finite integrality gaps for the basic SDP relaxations of the Boolean folded PCSPs in Lemma 4.1. To obtain such an integrality gap for the basic SDP relaxation of a PCSP, we study colorings of the $n$ dimensional sphere $S^n$ that satisfy certain properties. We start by defining a few notations that we need.

**Definition 4.3.** Fix a predicate $P \subseteq \{-1, +1\}^k$. We say that a tuple of vectors $V = (v_1, v_2, \ldots, v_k), v_i \in S^n \forall i \in [k]$ are a $P$-configuration with respect to another vector $v_0 \in S^n$ if the tuple of vectors can be assigned to a set of literals in a constraint by the basic SDP relaxation of an instance of PCSP($P, Q$) with zero error, for some $Q \supseteq P$. In other words, there exists a probability distribution $\{\lambda(a) : a \in P\}$ supported on $P$ that satisfies the following properties.

1. $0 \leq \lambda(a) \leq 1$ for all $a \in P$, and $\sum_{a \in P} \lambda(a) = 1$.
2. First moments: $v_i \cdot v_0 = \sum_{a \in P} \lambda(a)a_i \quad \forall i \in [k]$.
3. Second moments: $v_i \cdot v_j = \sum_{a \in P} \lambda(a)a_ia_j \quad \forall i, i' \in [k]$. We now define the notion of a function respecting a Boolean folded PCSP. We refer to functions $f : S^n \rightarrow \{-1, +1\}$ as colorings of the sphere.

**Definition 4.4.** Fix a predicate $P$ and we say that a coloring of the sphere $f : S^n \rightarrow \{-1, +1\}$ that is folded, i.e., $f(-v) = -f(v)$ for every $v \in S^n$ respects the Boolean folded PCSP $(P, Q)$ with respect to a vector $v_0 \in S^n$ if the following condition holds. For every $P$-configuration $V = (v_1, v_2, \ldots, v_k)$ with respect to $v_0$, we have that the colors of the vectors satisfy $Q$, i.e.,

$$(f(v_1), f(v_2), \ldots, f(v_k)) \in Q$$

More generally, we say that a coloring $f : S^n \rightarrow \{-1, +1\}$ respects a Boolean folded PCSP $\Gamma$ with respect to a vector $v_0$ if it respects every predicate pair in $\Gamma$ with respect to $v_0$.

Our key observation is that the absence of such a sphere coloring respecting $\Gamma$ for some finite $n$ gives an integrality gap for the basic SDP relaxation of $\Gamma$.

**Lemma 4.5.** For every Boolean folded PCSP $\Gamma$, the basic SDP decides PCSP($\Gamma$) if and only if for every integer $n \geq 1$, there exists a coloring $f^{(n)} : S^n \rightarrow \{-1, +1\}$ that respects $\Gamma$ with respect to a vector $v_0^{(n)}$.

We defer the proof of Lemma 4.5 to the full version [14]. As a corollary, we get the following.

**Corollary 4.6.** Let $\Gamma$ be a Boolean folded PCSP. Then, there is a finite integrality gap for the basic SDP relaxation of $\Gamma$ if and only if for some positive integer $n$, there exists no folding coloring $f : S^n \rightarrow \{-1, +1\}$ that respects $\Gamma$.

Theorem 4.2 together with Corollary 4.6 shows that if for a Boolean folded PCSP $\Gamma$ does not admit a sphere coloring $f : S^n \rightarrow \{-1, +1\}$ that respects $\Gamma$ for some positive integer $n$, then, $\Gamma$ does not admit a polynomial time robust algorithm, assuming the Unique Games Conjecture. Thus, our goal is to show that the PCSPs mentioned in Lemma 4.1 do not admit sphere coloring that respects them, and use Corollary 4.6 to prove Theorem 1.4.

In the rest of this section, we first prove a couple of lemmas regarding sphere Ramsey theory. Then, we show that the earlier mentioned PCSPs $\Gamma_{1-5}$ do not have sphere coloring respecting them using the sphere Ramsey results.

### 4.1 Sphere Ramsey Theory

We start with a few notations.

For a tuple of vectors $S = (v_1, v_2, \ldots, v_k)$ with $v_i \in S^d$, we use $\rho(S)$ to denote the sphere of the smallest radius that contains $S$ as a subset.

$$\rho(S) := \min\{r : \exists \epsilon \in \mathbb{R}^d, ||c - v_i||_2 = r \forall i \in [k]\}.$$ 

Let $S_1 = (u_1, u_2, \ldots, u_k), S_2 = (v_1, v_2, \ldots, v_k)$ with $u_i \in S^d, v_i \in S^d$ be two tuples with the same arity. We say that $S_1$ and $S_2$ are congruent if they have the same pairwise inner products, i.e., $u_i \cdot u_j = v_i \cdot v_j$ for all $i, j \in [k]$. Matoušek and Rödl [34] proved the following:

**Theorem 4.7 ([34]).** Let $S = (u_1, u_2, \ldots, u_k)$ be a tuple of affinely independent vectors with $\rho(S) < 1$. Then, for every positive integer $r \geq 2$, there exists an integer $n_0 := n_0(S, r)$ such that for every $n \geq n_0$, for every partition $f : S^n \rightarrow [r]$, there exists a tuple of vectors $S' = (v_1, v_2, \ldots, v_k), v_i \in S^n \forall i \in [k]$ that is congruent to $S$, and is monochromatic, i.e., $f(v_i) = f(v_j)$ for every $i, j \in [k]$.
We will use this to show the following lemma regarding sphere colorings.

**Lemma 4.8.** Fix an integer \( k \geq 3 \) and \( r \geq 2 \). There exists \( n_0 := n_0(k) \) such that for every \( n \geq n_0 \) and coloring \( f : \mathbb{S}^n \to [r] \) and \( \gamma \in \mathbb{R} \) with \( \frac{1}{r-1} < \gamma \leq 1 \), there exists a monochromatic set of vectors \( V = \{ v_1, v_2, \ldots, v_k \} \subseteq \mathbb{S}^n \) such that \( v_i \cdot v_j = \gamma \) for every \( i \neq j \).

Proof. Consider an arbitrary set \( S = \{ u_1, u_2, \ldots, u_k \} \) of \( k \) unit vectors in \( \mathbb{S}^n \) such that \( u_i \cdot u_j = \gamma \) for every \( i \neq j \). Such a set \( S \) is guaranteed to exist when \( n \) is large enough. We show that the vectors are affinely independent: suppose for contradiction that there exist real numbers \( c_1, c_2, \ldots, c_k \) not all zero, \( \sum_i c_i u_i = 0 \). We have

\[
0 = u_1 \cdot \left( \sum_i c_i u_i \right) = c_1 + \gamma (c_2 + \ldots + c_k) = c_1 + \gamma (-c_1)
\]

implying that \( c_1 = 0 \). The same argument shows that \( c_i = 0 \) for all \( i \in [k] \), a contradiction.

The set of vectors can be embedded on a sphere of radius strictly smaller than 1: let \( \alpha \in \mathbb{R} \) such that \( 0 < \alpha < \frac{2}{k} \), and let \( u_k \in \sum_i \{ u_i, c = e \cdot u \} \) such that \( \| u_i \|^2 = \sum_i \| u_i \|^2 + 2 \sum_{i \neq j} u_i \cdot u_j = k + \frac{k(k-1)}{\gamma} \). Note that

\[
\| u_j - c \|^2 = 1 - k(1 + (k-1)\gamma)\alpha \left( \alpha - \frac{2}{k} \right)
\]

which is strictly smaller than 1 when \( 0 < \alpha < \frac{2}{k} \). Thus, all the vectors are on a sphere centered at \( c \) and radius strictly smaller than 1, implying that \( \rho(S) < 1 \). Now, we can use Theorem 4.7 on \( S \) and \( f \) to obtain the required set of vectors \( V \). \( \square \)

While Theorem 4.7 is applicable to a wide range of sets \( S \), sometimes we need to apply it to sets \( S \) that do not form a simplex or have \( \rho(S) = 1 \). Towards this, we use the "Spreads" based idea in [34] to obtain a version of Theorem 4.7 directly for certain sets \( S \) where Theorem 4.7 is not applicable.

We use the following notion of Spread vectors from [34]. For an integer \( n \), a vector \( a \in \mathbb{R}^k \), and a set \( J \subseteq [n] \) of cardinality \( k \) with \( J = \{ j_1, j_2, \ldots, j_k \} \), we let

\[
\text{Spread}_n(a, J) = \sum_{i=1}^k a_i e_{j_i}
\]

where \( e_1, e_2, \ldots, e_n \) is an orthonormal basis of \( \mathbb{R}^n \). For a set \( I \subseteq [n] \), let

\[
\text{Spread}_n(a, I) = \{ \text{Spread}_n(a, J) : J \subseteq I, |J| = k \}
\]

We get the following as a direct application of the hypergraph Ramsey theorem.

**Lemma 4.9.** ([34]) For every \( a \in \mathbb{R}^k \), \( n, k \), there exists \( N \) such that in any coloring \( f : \text{Spread}_N(a, [N]) \to [r] \), there exists \( I \) with \( |I| = n \) such that \( \text{Spread}_N(a, I) \) is monochromatic with respect to \( f \), i.e., \( \exists p \in [r] \) such that \( f(v) = p \) for all \( v \in \text{Spread}_N(a, I) \).

Lemma 4.9 implies the following immediately.

**Corollary 4.10.** Let \( U = \{ u_1, u_2, \ldots, u_k \} \) be a set of \( k \) unit vectors such that \( u_i \in \text{Spread}_N(a, [N]) \) for all \( i \in [k] \) for an integer \( N \), and a vector \( a \in \mathbb{R}^N \) with \( \| a \|_2 = 1 \). Then there exists \( n_0 := n_0(U, a, N) \) such that for every \( n \geq n_0 \), for every coloring \( f : \mathbb{S}^n \to [r] \), there exists a set of \( k \) vectors \( V = \{ v_1, v_2, \ldots, v_k \} \) that are all colored the same, and \( v_i \cdot v_j = a_i \cdot a_j \) for every \( i, j \in [k] \).

We use Corollary 4.10 to prove a lemma regarding sphere colorings. For ease of notation, we call a set of \( k \) unit vectors \( V = \{ v_1, v_2, \ldots, v_k \} \) to be \( k \)-regular if \( v_i \cdot v_j = -\frac{1}{k-1} \) for every \( i \neq j \).

**Lemma 4.11.** Fix an integer \( k \geq 2 \). There exists \( n_0 := n_0(k) \) such that for every \( n \geq n_0 \) and folded coloring \( f : \mathbb{S}^n \to \{-1, +1\} \), there exist a \( k \)-regular set of vectors \( V = \{ v_1, v_2, \ldots, v_k \} \subseteq \mathbb{S}^n \) such that exactly \( k-1 \) vectors in \( V \) are colored \(-1 \).

We defer the proof of Lemma 4.11 to the full version [14].

### 4.2 Absence of Sphere Coloring via Sphere Ramsey Theory

First, we show the absence of sphere coloring respecting \( \Gamma_1 \) using Lemma 4.11.

**Lemma 4.12.** Fix an even integer \( k \geq 4 \). There exists an integer \( n_0 \) such that for every \( n \geq n_0 \), there is no folded coloring \( f : \mathbb{S}^n \to \{-1, +1\} \) that respects \( \Gamma_1 = (P, Q), P = \text{Ham}_k \{ \frac{1}{2} \}, Q = \text{Ham}_k \{ 0, 1, \ldots, k \} \setminus \{ b \} \) where \( b \in \{ 1, k-1 \} \).

Proof. Consider a large integer \( n \) and suppose for the sake of contradiction that there is a folded function \( f : \mathbb{S}^n \to \{-1, +1\} \) that respects \( \Gamma_1 \) with respect to a vector \( v_0 \in \mathbb{S}^n \). We get the \( P \)-configuration of vectors \( v_1, v_2, \ldots, v_k \) where we set \( \lambda(a) = \frac{1}{|I|} \) for every \( a \in P \) in Definition 4.3. The vectors satisfy the following properties.

1. (First moments.) \( v_i \cdot v_0 = 0 \) for every \( i \in [k] \).
2. (Second moments.) \( v_i \cdot v_j = \frac{2(k-1)}{(k)} = -\frac{1}{k-1} \).

Our goal is to show that there is a \( P \)-configuration of such vectors such that exactly \( b \) of them are colored \(+1\) according to \( f \). Consider the set of vectors

\[
v_0^* := \{ u \in \mathbb{S}^n : u \cdot v_0 = 0 \}
\]

Using Lemma 4.11, we can obtain a set of \( k \) vectors \( u_1, u_2, \ldots, u_k \in v_0^* \) such that \( u_i \cdot u_j = \frac{1}{k-1} \) and exactly \( k-1 \) of \( \{ u_1, u_2, \ldots, u_k \} \) are colored \(-1 \).

**Lemma 4.13.** Fix an odd integer \( k \geq 3 \) and integer \( l : 0 \leq l \leq \frac{k+1}{2} \). There exists an integer \( n_0 \) such that for every \( n \geq n_0 \), there is no folded coloring \( f : \mathbb{S}^n \to \{-1, +1\} \) that respects \( \Gamma_2 = (P, Q), P = \text{Ham}_k \{ l, \frac{k+1}{2} \}, Q = \text{Ham}_k \{ 0, 1, 2, \ldots, k \} \setminus \{ k-1 \} \).

We show the absence of sphere coloring respecting \( \Gamma_2, \Gamma_3 \), and \( \Gamma_4 \) using Lemma 4.8.
Proof. Consider a large integer $n$ and suppose for contradiction that there is a folded function $f : \mathbb{S}^n \to \{-1, +1\}$ that respects $\Gamma_2$ with respect to a vector $v_0 \in \mathbb{S}^n$. The $P$-configuration that we consider is a set of vectors $v_1, v_2, \ldots, v_k$ that are obtained by setting $\lambda(a)$ in Definition 4.3 as follows. We first sample an integer $t \in \{l, k+1\}$ as below.

$$t = \begin{cases} l, & \text{with probability } \frac{1}{r_1} - \frac{1}{r_2}, \\ k+1, & \text{with probability } \frac{1}{r_2}, \end{cases}$$

where $s = l - (k - l) < 0$. The probability distribution $\lambda$ is obtained by sampling a uniform element of $\text{Ham}_k \{l\}$. In other words, we have

$$\lambda(a) = \begin{cases} \frac{1}{(1-s)(\frac{k}{2})}, & \text{if } a \in \text{Ham}_k \{l\}, \\ \frac{1}{(1-s)(\frac{k}{2})}, & \text{else if } a \in \text{Ham}_k \{k+1\}, \\ 0, & \text{otherwise}. \end{cases}$$

We obtain the following properties:

1. (First moments). $v_i \cdot v_0 = \frac{1}{(1-s)(\frac{k}{2})} (l - (k - l) + \frac{1}{r_2}) = 0$ for every $i \in [k]$.
2. (Second moments). By symmetry of variables, we get that $v_i \cdot v_j = 0$ for every $i \neq j$, for some $y = y(k, l)$. We have

$$k + k(k-1)y = \sum_{i \in P} v_i = \sum_{a \in P} \lambda(a) \|a\|^2 > 0.$$ 

Thus, we get that $\frac{1}{k} < y < 1$.

Now, restricting ourselves to the vectors in $\mathbb{S}^n$ that are orthogonal to $v_0$, and using Lemma 4.8, we get that there exists a $P$-configuration of vectors that are all colored the same. By taking the negation of these vectors if needed, we get our required claim. \hfill $\square$

Lemma 4.14. Fix integers $k, l$ such that $0 < l \leq \frac{k+1}{2}$. Then, there exists an integer $n_0$ such that for every $n \geq n_0$, there is no folded $f : \mathbb{S}^n \to \{-1, +1\}$ that respects $\Gamma_3 = (P, Q)$, $P = \text{Ham}_k \{l, k\}$, $Q = \text{Ham}_k \{1, 2, \ldots, k\}$, where $l \neq 0, l \leq \frac{k+1}{2}$.

Lemma 4.15. Fix integers $k \geq 3, l \in \{1, \ldots, k-1\}, l \leq \frac{k+1}{2}$. There exists integer $n_0$ such that for every $n \geq n_0$, there does not exist a coloring $f : \mathbb{S}^n \to \{0, 1\}$ that is folded and respects the PCSP $\Gamma_4 = (P, Q), P = \text{Ham}_k \{l\}, Q = \text{Ham}_k \{0, 1, \ldots, k\} \setminus \{0, k - 1\}$.

We defer the proofs of Lemma 4.14 and Lemma 4.15 to the full version [14].

4.3 Absence of Sphere Coloring via Connectivity of Configurations

Finally, we show the absence of sphere coloring for $\Gamma_5$ using a connectivity lemma.

Lemma 4.16. Fix integers $k \geq 3, b \in \{0, 1, \ldots, k\} \setminus \{1, k\}$. There exists an integer $n_0$ such that for every $n \geq n_0$, there does not exist a coloring $f : \mathbb{S}^n \to \{0, 1\}$ that is folded and respects the PCSP $\Gamma_5 = (P, Q), P = \text{Ham}_k \{1, k\}, Q = \text{Ham}_k \{0, 1, \ldots, k\} \setminus \{b\}$.

We dedicate the rest of the section to proving Lemma 4.16.

We pick the configuration of vectors along the same lines as in Lemma 4.13. Fix $v_0 \in \mathbb{S}^n$. The $P$-configuration that we study is a set of vectors $v_1, v_2, \ldots, v_k$ that is obtained by first sampling $t \in \{1, k\}$ such that

$$t = \begin{cases} 1, & \text{with probability } \frac{k}{k+1}, \\ k, & \text{with probability } \frac{k}{k+1}. \end{cases}$$

Then, we sample a uniform element from $\text{Ham}_k \{t\}$. We get the following properties:

1. (First moments). $v_i \cdot v_0 = \left(\frac{k}{k+1}\right) - \left(\frac{1}{k+1}\right) = 0$ for every $i \in [k]$.
2. (Second moments). For every $i \neq j, v_i \cdot v_j = \left(\frac{k}{k+1}\right) - \left(\frac{1}{k+1}\right) \frac{1}{k+1} = \frac{k-1}{k+1}$.

Thus, $\frac{1}{k+1} < y < 1$.

For ease of notation, we let $\alpha = \frac{k-1}{k+1}$. Furthermore, by restricting ourselves to vectors in $\mathbb{S}^n$ that are orthogonal to $v_0$, we just focus on $P$-configurations that are a set of $k$ unit vectors all of whose pairwise inner product is equal to $\alpha$. We refer to these sets of vectors, i.e., a set $V$ of $k$ unit vectors $v_1, v_2, \ldots, v_k \in \mathbb{S}^n$ an $\alpha$-configuration if the inner product of every pair of them is equal to $\alpha$. Given the folded sphere coloring $f$, our goal is to show that there is an $\alpha$-configuration of vectors $V$ among which exactly $b$ of them are assigned $+1$.

To show that there are $\alpha$-configurations that have exactly $b$ vectors that are colored $+1$, we show a connectivity lemma (Lemma 4.17) where we prove that between any two $\alpha$-configurations, there exists a path using $O_k, (1)$ $\alpha$-configurations where we change a single vector at each step in the path.

Lemma 4.17. Fix an integer $k \geq 2$ and $0 < \alpha < 1$. Suppose that $U$ and $V$ are two $\alpha$-configurations in $\mathbb{S}^n$. Then, there exist $n_0 = n_0(k, \alpha)$, and $L = L(k, \alpha)$ such that as long as $n \geq n_0$, there exist $\alpha$-configurations $V_j, V_2, \ldots, V_l$ such that

1. The endpoints are $U$ and $V$, i.e., $U = V_1, V = V_l$.
2. Any two consecutive configurations differ in exactly one vector i.e., $|V_j \cap V_{j+1}| = k - 1$ for every $j \in [L - 1]$.

As there is an $\alpha$-configuration where all are $k$ vectors are colored $+1$, and the $\alpha$-configuration obtained by negating these vectors where all the vectors are colored $-1$, the connectivity lemma then shows that for every $b \in \{0, 1, \ldots, k\}$, there exists an $\alpha$-configuration that has exactly $b$ vectors that are colored $+1$.

We are now ready to prove Lemma 4.16.

Proof. Suppose for contradiction that there exists a coloring $f : \mathbb{S}^n \to \{0, 1\}$ that is folded and respects the PCSP $\Gamma_5$. Consider an arbitrary set of vectors $v_1, v_2, \ldots, v_{2k+1}$ that are all orthogonal to $v_0$, and have pairwise inner product $\alpha$. Such a set is guaranteed to exist as $\alpha \geq 0$. There exists a set of $k$ vectors among these that are all assigned the same color in $f$. Let these form the configuration $U$, and the set of negations of these vectors be the configuration $V$. Using Lemma 4.17, there exists a path from $U$ to $V$ where we change a single vector in each step. Note that the endpoints of the path have $0$ and $k$ vectors assigned $+1$ respectively. Since we change at most one vector at a time, there exists a configuration where we have exactly $b$ is $1$, a contradiction. \hfill $\square$
