

Rainbow coloring hardness via low sensitivity polymorphisms*

Venkatesan Guruswami Sai Sandeep

Computer Science Department
 Carnegie Mellon University
 Pittsburgh, USA.
 {venkatg, spallerl}@cs.cmu.edu

Abstract

A k -uniform hypergraph is said to be r -rainbow colorable if there is an r -coloring of its vertices such that every hyperedge intersects all r color classes. Given as input such a hypergraph, finding a r -rainbow coloring of it is NP-hard for all $k \geq 3$ and $r \geq 2$. Therefore, one settles for finding a rainbow coloring with fewer colors (which is an easier task). When $r = k$ (the maximum possible value), i.e., the hypergraph is k -partite, one can efficiently 2-rainbow color the hypergraph, i.e., 2-color its vertices so that there are no monochromatic edges. In this work we consider the next smaller value of $r = k - 1$, and prove that in this case it is NP-hard to rainbow color the hypergraph with $q := \lceil \frac{k-2}{2} \rceil$ colors. In particular, for $k \leq 6$, it is NP-hard to 2-color $(k - 1)$ -rainbow colorable k -uniform hypergraphs.

Our proof follows the algebraic approach to promise constraint satisfaction problems. It proceeds by characterizing the polymorphisms associated with the approximate rainbow coloring problem, which are rainbow colorings of some product hypergraphs on vertex set $[r]^n$. We prove that any such polymorphism $f : [r]^n \rightarrow [q]$ must be C -fixing, i.e., there is a small subset S of C coordinates and a setting $a \in [q]^S$ such that fixing $x_{|S} = a$ determines the value of $f(x)$. The key step in our proof is bounding the sensitivity of certain rainbow colorings, thereby arguing that they must be juntas. Armed with the C -fixing characterization, our NP-hardness is obtained via a reduction from smooth Label Cover.

1 Introduction

Graph and hypergraph coloring are one of the most studied problems in Graph Theory and Theoretical Computer Science. Even though there is a simple algorithm to check if a given graph is 2-colorable or not, checking if a 3-uniform hypergraph can be colored with two colors so that no hyperedge is monochromatic is one of the classic NP-hard problems. This raises the question of identifying if 2-coloring is easy on special hypergraphs of interest. For example, if a k -uniform hypergraph is k -partite, i.e., the vertices can be partitioned into k parts so that every hyperedge intersects each part, then there are simple algorithms to properly color the hypergraph with two colors. Suppose we know that a k -uniform hypergraph is promised to be $k - 1$ -partite, can we color it with two colors?

An equivalent way to formulate this question is in terms of rainbow coloring. A k -uniform hypergraph is said to be *r -rainbow colorable* if there is a coloring of vertices with r colors such

*Research supported in part by NSF grants CCF-1422045, CCF-1526092, and CCF-1814603.

that all the r colors appear in every edge. Unlike usual coloring, rainbow coloring becomes harder as we have more colors. Note that r -partiteness is the same thing as r -rainbow colorability. As mentioned above, a k -uniform hypergraph that is promised to be k -rainbow colorable can be efficiently colored with two colors. One big hammer approach for this is to use semidefinite programming and find a unit vector for each vertex such that the sum of the vectors in each edge sum to zero, and then use random hyperplane rounding. But the 2-coloring can also be performed by a simple random walk algorithm — start with an arbitrary coloring, and as long as there is a monochromatic edge, pick an arbitrary one and flip the color of a random vertex in it. This process will converge to a 2-coloring in a quadratic number of iterations with high probability [McD93].

If we relax the k -rainbow colorability assumption slightly to that of $(k - 1)$ -rainbow colorability, there are no known efficient algorithms for 2-coloring. It is tempting to conjecture that in fact this task is hard (in fact, even if we are allowed c colors for any constant c ; this was shown assuming the V Label Cover conjecture in [BG17]). If we relax the rainbow colorability assumption further, Austrin, Bhangale, and Potukuchi proved that it is NP-hard to 2-color a k -uniform hypergraph when it is promised to be $(k - 2\sqrt{k})$ -rainbow colorable [ABP18]. They also showed that it is NP-hard to 2-color a 4-uniform hypergraph even if it is 3-rainbow colorable. In this work, we focus on hardness results for the $(k - 1)$ -rainbow colorable case, as this promise is the closest to k -partiteness which makes 2-coloring easy. While we can't show hardness of 2-coloring, we show that rainbow coloring with $\lceil \frac{k-2}{2} \rceil$ colors is hard. Formally, our main result is the following.

Theorem 1. *Fix an integer $k \geq 4$. Given a k -uniform hypergraph that is promised to be $(k - 1)$ -rainbow colorable, it is NP-hard to rainbow color it with $\lceil \frac{k-2}{2} \rceil$ colors.*

As a corollary, we also get the following, which extends the similar result of [ABP18] for the $k = 4$ case (their techniques did not generalize beyond the 4-uniform case).

Theorem 2. *For $k \leq 6$, given a k -uniform hypergraph that is promised to be $(k - 1)$ -rainbow colorable, it is NP hard to 2-color it.*

1.1 Techniques

There have been broadly three lines of attack on proving hardness for graph and hypergraph coloring problems.

1. The first line of work gives reductions from Label Cover analyzed using Fourier-analytic techniques of the sort originally pioneered by Håstad [Hås01]. Early applications of this method showed strong hardness results for coloring 2-colorable hypergraphs of low uniformity with any constant number of colors [GHS02, Hol02, Kho02, Sak14]. This approach, augmented with the invariance principle of [MOO10] and some of its extensions such as [DMR09, Mos10, Wen13], was used to prove further hardness results for hypergraph coloring [BK10, GL18] and strong conditional hardness results for graph coloring [DMR09]. These methods usually also prove a stronger statement about finding independent sets in the graphs or hypergraphs. For rainbow coloring, it is proved in [GL18] by combining together many of these techniques that a $(k/2)$ -rainbow colorable k -uniform hypergraph cannot be colored with any constant number of colors in polynomial time unless $P = NP$.

2. A less extensive line of work proceeds via combinatorial gadgets that are analyzed using ideas based on the chromatic number of Kneser graphs and similar results. The first exemplar of this approach was the hardness of $O(1)$ -coloring 2-colorable 3-uniform hypergraphs shown in [DRS05]. Unlike the analytic results for 4-uniform hypergraphs mentioned above, this result does not show hardness of finding large independent sets. (This was later shown in [KS14] using the analytic approach, albeit under the d -to-1 conjecture.) A few recent results have revived this combinatorial approach, by re-deriving and improving some previous hardness results for hypergraph coloring using simpler proofs [Bha18, ABP19].
3. The third and most recent line of work adapts the universal algebraic method behind the complexity classification of constraint satisfaction problems that culminated in the resolution of the Feder-Vardi CSP dichotomy conjecture [Bul17, Zhu17]. Here, the coloring problem is viewed as a Promise Constraint Satisfaction Problem (PCSP), and its associated “polymorphisms” are then analyzed.¹ In the cases when the polymorphisms are severely limited, one can show hardness via a reduction from Label Cover. The approach to study PCSP using polymorphisms originated in [AGH17] and was used to show hardness results for graph and hypergraph coloring in [BG16]. The algebraic theory was further developed significantly in [BKO18] leading among other results to a proof of NP-hardness of 5-coloring 3-colorable graphs. Recently, [KO19] and [WZ19] used topological ideas to make further progress.

In this work, we follow the algebraic approach to prove Theorem 1. In fact, our main motivation is to understand Promise CSPs better. A promise CSP (defined formally in Section 2) is a relaxation of the traditional CSP where one is allowed to find an assignment that satisfies a relaxed version of the predicates underlying the CSP. Approximate graph coloring with more colors than the promised chromatic number is a quintessential example of a promise CSP. Rainbow coloring with fewer colors also naturally falls in this framework. As proved in [BKO18, BG18], as with normal CSP, the complexity of a promise CSP is captured by its associated polymorphisms. Polymorphisms (defined formally in Section 2) of a PCSP are ways to combine multiple solutions of an instance satisfying the stronger predicate to obtain a solution to the instance satisfying the weaker predicate. The high-level principle behind the algebraic approach is that the problem should be easy when it has a rich enough set of polymorphisms that include functions with strong symmetries, and hard when all its polymorphisms are somehow skewed and lack symmetries. This has been fully established for CSPs — when there are polymorphisms which obey weak near-unanimity, the CSP is polytime solvable, and otherwise NP-complete.² The hardness part of this dichotomy is easier and was known for a while [MM08]; the much harder algorithmic part was established only recently in [Bul17, Zhu17].

For promise CSPs, which form a much richer class, our current understanding is rather limited, for both the algorithmic and hardness sides. It is not clear (to even conjecture) what kind of lack in symmetries in the polymorphisms might dictate hardness, and how one might show the corresponding hardness. A simple (but rather limited) sufficient condition for hardness is when all the polymorphisms are dictators that depend on a single coordinate. In [AGH17], it

¹The proof in [DMR09] also implicitly studies polymorphisms and proves that they must have a small number of coordinates with sizeable influence and thus are not too symmetric. This influence-type characterization interfaces better with Unique Games or other highly structured forms of Label Cover.

²For the case of Boolean CSPs, the CSP is hard iff all polymorphisms are essentially at most unary, i.e., either the dictator function, its complement or a constant function.

has been proved that if all the polymorphisms of a PCSP are juntas³, then the PCSP is NP-hard. This is the basis of the hardness results for $(2+\epsilon)$ -SAT [AGH17] and 3-coloring graphs that admit a homomorphism to C_k for any fixed odd integer k [KO19]. The recent hardness of 5-coloring 3-colorable graphs in [BKO18] proceeds by showing that the absence of arity 6 polymorphisms with the so-called Olšák symmetry implies NP-hardness, and then verifying that 3 vs. 5-coloring lacks such polymorphisms.

It turns out that the polymorphisms of rainbow coloring can have Olšák symmetries *and* be non-juntas. We will get around this by proving that these polymorphisms are C -fixing in the sense that there exists a constant number of coordinates and an assignment to them such that if we fix these coordinates to the assignment, the value of the function is fixed. This is also studied as certificate complexity in Boolean function analysis [APV16]. We then prove that if the polymorphisms of a PCSP are C -fixing, then the PCSP is NP-hard.

In order to prove that the polymorphisms have low certificate complexity, we use the connection between sensitivity and certificate complexity of functions. These two ways of characterizing the complexity of functions are well studied in the context of Boolean functions. It is worth emphasizing that for our purposes, all we need is to show that low sensitivity (even sensitivity 2 suffices) implies constant certificate complexity, and thus we are not interested in optimal gaps between sensitivity and certificate complexity. The famous sensitivity vs. block sensitivity conjecture [Nis89] states that these two parameters are in fact polynomially related. In one of the earliest works related to this problem, Simon [Sim83] proved that certificate complexity is at most exponential in sensitivity. We extend this to larger domains and then use it to prove that the polymorphisms that we study have low certificate complexity. We remark that in a striking breakthrough, Huang [Hua19] recently proved the sensitivity vs block sensitivity conjecture for Boolean domains.

The second step is to then use the C -fixing property to show NP-hardness of the PCSP. This is done by the usual paradigm of reducing from Label Cover using polymorphism tests (better known as long code tests) of functions associated with vertices of the Label Cover instance. A more structured form of the C -fixing property where the C variables are fixed to the same value, is used in [BG18] to show NP-hardness of certain Boolean PCSPs. However, in order to prove NP-hardness using our more general notion of C -fixing, we end up needing stronger properties of the Label Cover instance. As a result, our reduction is from the *smooth* Label Cover problem that was introduced and shown to be NP-hard in [Kho02], and has found many applications in inapproximability since.

A natural question is to understand how far we can push these techniques. Our NP hardness reduction from smooth Label Cover works when the polymorphisms of the PCSP in hand are C -fixing for some constant C . As k increases, the polymorphisms of PCSP of 2-coloring a k -uniform hypergraph that is promised to be $(k - 1)$ -rainbow colorable get richer. When k is at most 6, the polymorphisms are C -fixing. At $k = 7$, we show that there is a polymorphism that is not C -fixing for any constant C . In fact, one would need C to be linear in the arity of polymorphisms which also rules out using smooth Label Cover with very strong soundness.

³A C -junta is a function that depends on at most C inputs

1.2 Prior work on rainbow coloring and related problems

Various notions of approximate coloring with rainbow colorability guarantees have been studied in the literature. Bansal and Khot [BK10] prove that when the input hypergraph is promised to be almost k -rainbow colorable, it is Unique Games hard to color it with $O(1)$ colors. Sachdeva and Saket [SS13] establish NP-hardness of $O(1)$ coloring a k -uniform hypergraph when it is promised to be almost $(k/2)$ -rainbow colorable. This was extended by Guruswami and Lee [GL18] to perfectly $(k/2)$ -rainbow colorable hypergraphs. Guruswami and Saket [GS17] prove similar results assuming stronger forms of rainbow colorability in the completeness case. In [ABP18], Austrin, Bhangale, and Potukuchi proved that it is NP-hard to 2-color a k -uniform hypergraph when it is promised to be $(k - 2\sqrt{k})$ -rainbow colorable. On the other hand, when the hypergraph is promised to be $(k - \sqrt{k})$ -rainbow colorable, Bhattiprolu, Guruswami and Lee [BGL15] give algorithms to color the hypergraph with two colors that miscolors at most $k^{-\Omega(k)}$ fraction of edges; this beats the 2^{k+1} fraction achieved by random coloring that is the best possible for general 2-colorable hypergraphs [Hås01]. Brakensiek and Guruswami [BG17] put forth a problem called V label cover (to possibly serve as a perfect completeness variant surrogate for Unique games), and under its conjectured inapproximability proved that it is hard to color a k -uniform $(k - 1)$ -rainbow colorable hypergraph with $O(1)$ colors.

A related notion of hypergraph coloring is strong coloring where we color a k -uniform hypergraph with $s > k$ colors such that in any edge, all the k vertices are colored with distinct colors. Brakensiek and Guruswami [BG16] prove that it is NP-hard to 2-color a k -uniform hypergraph that is promised to be strongly colorable with $\lceil \frac{3k}{2} \rceil$ colors. Assuming the V Label Cover conjecture, it is hard to $O(1)$ -color k -uniform hypergraphs with strong chromatic number at most $k + \sqrt{k}$ [BG17].

1.3 Outline

We start with a few notations and definitions in Section 2. In Section 3, we study polymorphisms of rainbow coloring. We first prove a result on sensitivity and certificate complexity and use it to prove properties of polymorphisms of the PCSP that we are studying. Then, we use these in Section 4 to prove NP hardness. Finally, we conclude in Section 5 by mentioning some open questions.

2 Preliminaries

2.1 Notations

We use $[n]$ to denote the set $\{1, 2, \dots, n\}$. Vectors are represented using bold face letters. We abuse the notation of k -ary relation A to use it both as a set $A \subseteq [q]^k$ and indicator function $A : [q]^k \rightarrow \{0, 1\}$.

2.2 PCSP and Polymorphisms

We will now formally define CSP, PCSP, and Polymorphisms.

Definition 3. (CSP) Given a k ary relation $A : [q]^k \rightarrow \{0, 1\}$ over $[q]$, the Constraint Satisfaction problem associated with A takes input as a set of variables $V = \{a_1, a_2, \dots, a_n\}$ which are to

be assigned values from $[q]$. There are m constraints (e_1, e_2, \dots, e_m) each consisting of $e_i = ((e_i)_1, (e_i)_2, \dots, (e_i)_k) \subseteq V^k$ that indicate that the corresponding assignment should belong to A . The problem is to check if we can satisfy all the constraints or not.

In general, we can have multiple relations A_1, A_2, \dots, A_m , and different constraints can use different relations. We denote such a CSP by $CSP(A_1, A_2, \dots, A_m)$.

Promise CSP (PCSP) is a gap or promise version of CSP. Here, we have a pair of relations such that one is a relaxed form of other and given a CSP instance, and the objective is to decide if there is a satisfying assignment from stronger relation or we cannot even satisfy the CSP using the relaxed relation. One canonical example of PCSP is the promise graph coloring: Given a graph G , distinguish between the case that G can be 3-colored versus G cannot even be colored with five colors. We can formally define PCSP as below:

Definition 4. (*Promise CSP*) In the promise CSP problem, we have a set of pairs of relations $(A_1, B_1), (A_2, B_2), \dots, (A_m, B_m)$ such that for every i , A_i is a subset of $[q_1]^{k_i}$ and B_i is a subset of $[q_2]^{k_i}$. Furthermore, there is a homomorphism $h : [q_1] \rightarrow [q_2]$ such that, for all i , $x \in A_i$ implies $h(x) \in B_i$ for all $x \in [q_1]^{k_i}$. Given a $CSP(A_1, A_2, \dots, A_m)$ instance, the goal is to distinguish between the two cases:

1. There is a solution to the instance assigning values from $[q_1]$ that satisfies every constraint when viewed as $CSP(A_1, A_2, \dots, A_m)$.
2. There is no solution to the instance assigning values from $[q_2]$ that satisfies every constraint when viewed as $CSP(B_1, B_2, \dots, B_m)$.

We now turn our attention towards rainbow coloring which is the PCSP that we study in this paper. In $RAINBOW(k, r, q)$ problem, the input is a k -uniform hypergraph. The goal is to distinguish between the cases when the hypergraph is rainbow colorable with r colors and when it cannot be rainbow colorable with q colors. More formally, we can define the problem as below:

Definition 5. (*RAINBOW(k, r, q)*) In the $RAINBOW(k, r, q)$ promise CSP, $q \leq r \leq k$, we have the relation pair (A, B) defined as follows:

- $A : [r]^k \rightarrow \{0, 1\} : A(x_1, x_2, \dots, x_k) = 1$ if and only if $\{x_1, x_2, \dots, x_k\} = [r]$.
- $B : [q]^k \rightarrow \{0, 1\} : B(y_1, y_2, \dots, y_k) = 1$ if and only if $\{y_1, y_2, \dots, y_k\} = [q]$.

Note that we need q, r to be at most k since we cannot rainbow color a k -uniform hypergraph with more than k colors. We also need the condition that $q \leq r$ for the promise problem to make sense: If the hypergraph is r rainbow colorable, we can infer that it is already $q < r$ rainbow colorable too. Thus, the promise problem is to identify if the hypergraph is r rainbow colorable or it *cannot even be* rainbow colorable with q colors. Furthermore, in this paper we will be only dealing with near perfect completeness case when hypergraph is $(k - 1)$ -partite i.e. $r = k - 1$.

Associated with every promise CSP, there are polymorphisms. Polymorphisms capture the symmetries in the PCSP. They are ways in which we combine solutions to obtain new solutions that are still valid.

Definition 6. (*Polymorphisms*) For a PCSP problem (A, B) , $A : [q_1]^k \rightarrow \{0, 1\}, B : [q_2]^k \rightarrow \{0, 1\}$, a polymorphism is a function f from $[q_1]^n \rightarrow [q_2]$, where n is the arity of the polymorphism that satisfies the property $(f(\mathbf{v}_1), f(\mathbf{v}_2), \dots, f(\mathbf{v}_k)) \in B$ for all $(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k)$ such that for all $i \in [n], ((\mathbf{v}_1)_i, (\mathbf{v}_2)_i, \dots, (\mathbf{v}_k)_i) \in A$.

In the above, we defined polymorphisms for a single relation PCSP. When the PCSP has multiple relations, the polymorphism should satisfy the above property for all the relations. Informally, the arity n polymorphisms are precisely the functions $f : [q_1]^n \rightarrow [q_2]$ such that for every $k \times n$ matrix M with elements from $[q_1]$ whose columns are satisfying tuples of A , the k tuple obtained by applying f to the rows of M should be in B . We refer the reader to [BG18, BKO18] for a detailed introduction to PCSP and various examples of polymorphisms.

We now direct our attention to polymorphisms of $\text{RAINBOW}(k, r, q)$. By definition, the polymorphisms of hypergraph coloring PCSPs turn out to be colorings of certain tensor product hypergraphs. Fix n to be arity of the polymorphisms. We can infer that the polymorphisms of $\text{RAINBOW}(k, r, q)$ are proper q -rainbow colorings of the following k -uniform hypergraph $\text{RH}_n(k, r)$:

- The vertex set of hypergraph is the set $V = [r]^n$.
- A k element set $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k\}$, where each $\mathbf{v}_i \in [r]^n$ is an edge if and only if for every $j \in [n]$, the set $\{(\mathbf{v}_1)_j, (\mathbf{v}_2)_j, \dots, (\mathbf{v}_k)_j\}$ is equal to $[r]$.

That is, a set of k vectors from $[r]^n$ forms an edge if in the matrix obtained by plugging these vectors as rows, all the r elements from $[r]$ occur in every column.

2.3 Complexity measures of functions

Finally, we define the notions of sensitivity and C -fixing of functions.

Definition 7. (*Sensitivity at \mathbf{x}*) For a function $f : [r]^n \rightarrow [q]$, and an input $\mathbf{x} \in [r]^n$, the sensitivity of f at \mathbf{x} , denoted by $S(f, \mathbf{x})$ is defined as the number of coordinates i such that changing \mathbf{x} at i can change the value of f i.e. $S(f, \mathbf{x}) = |\{i \in [n] \mid \exists a : f(\mathbf{x}) \neq f(\mathbf{x} : x_i \leftarrow a)\}|$.

Definition 8. (*Sensitivity*) The sensitivity of a function $f : [r]^n \rightarrow [q]$, denoted by $S(f)$ is defined as the maximum sensitivity of f over all \mathbf{x} in $[r]^n$ i.e. $S(f) = \max_{\mathbf{x}} S(f, \mathbf{x})$.

Definition 9. (*C fixing*) A function f from $[r]^n$ to $[q]$ is said to be C -fixing for some integer C if there exists a set $S = \{s_1, s_2, \dots, s_C\} \subseteq [n]$ and a vector $\alpha \subseteq [r]^n$ such that $f(\mathbf{x}) = c$ whenever $x_{s_i} = \alpha_{s_i}$ for all integers $1 \leq i \leq C$, for some fixed $c \in [q]$.

3 Polymorphisms

In this section, we will analyze the properties of polymorphisms of rainbow coloring. In order to do so, we will prove that low sensitivity implies low certificate complexity. Using this, we will establish that the polymorphisms for $\text{RAINBOW}(k, k-1, \lceil \frac{k-2}{2} \rceil)$ are C -fixing. Along the way, we will study rainbow colorings of various hypergraphs related to $\text{RH}_n(k, r)$. Finally, we will show that our techniques cannot prove hardness of $\text{RAINBOW}(7, 6, 2)$ by presenting a polymorphism that is not C -fixing for any constant C .

3.1 Sensitivity vs certificate complexity

We extend a lemma of [Sim83] that proves that if a function has low sensitivity then the function is C fixing, to larger domains. The proof is along the same lines as the original proof.

Lemma 10. *Let $f : [r]^n \rightarrow [q]$ be a function with sensitivity s , and let $b \in [q]$ such that $f^{-1}(b)$ is non empty. Then, $|f^{-1}(b)| \geq r^{n-s}$.*

Proof. Fix s , and induct on n . The case $n = s$ is trivial. Let $\mathbf{x} \in [r]^n$ be such that $f(\mathbf{x}) = b$. Since $s < n$, there is a coordinate in \mathbf{x} that is not sensitive. Without loss of generality, let it be 1, and let $\mathbf{x} = (x_1, \mathbf{y})$. As the first coordinate is not sensitive for \mathbf{x} , we can conclude that $f(\alpha, \mathbf{y}) = b$ for all $\alpha \in [r]$.

Consider the set of functions $g_i : [r]^{n-1} \rightarrow [q], g_i(\mathbf{u}) = f(i, \mathbf{u}), i \in [r]$. Note that for each such g_i , the set $g_i^{-1}(b)$ is non-empty. In addition, for every $i \in [r]$, sensitivity of g_i is at most the sensitivity of f . Thus, by induction, we know that each such g_i has at least r^{n-1-s} elements \mathbf{u} in $[r]^{n-1}$ such that $g_i(\mathbf{u}) = b$. Note that every such \mathbf{u} gives $f(i, \mathbf{u}) = b$. By combining over all i s, we can conclude that there are at least $r \cdot r^{n-1-s} = r^{n-s}$ elements $\mathbf{x} \in [r]^n$ such that $f(\mathbf{x}) = b$. \square

Lemma 11. *Let $f : [r]^n \rightarrow [q]$ be a function with sensitivity $s < n/2$. Then, it is C -fixing for $C = s(r-1)r^{2s+1}$.*

Proof. We will actually prove a stronger statement that f is a C -junta. Let A denote the set of coordinates with non-zero influence in f i.e. the coordinates that are sensitive for some input. Our goal is to upper bound the cardinality of A .

For a function $f : [r]^n \rightarrow [q]$, let the set of sensitive edges $E(f)$ be defined as the set of pairs of elements $\mathbf{x}, \mathbf{y} \in [r]^n$ such that $f(\mathbf{x}) \neq f(\mathbf{y})$, and \mathbf{x}, \mathbf{y} differ on exactly one coordinate. From the sensitivity bound on f , we can deduce that

$$|E(f)| \leq s(r-1)r^n \tag{1}$$

Fix an arbitrary coordinate $i \in A$. There are elements $\mathbf{x}, \mathbf{y} \in [r]^n$ such that $x_i = \alpha, y_i = \beta, \alpha \neq \beta, f(\mathbf{x}) \neq f(\mathbf{y})$, and \mathbf{x}, \mathbf{y} differ only in i th coordinate. Define a function $g : [r]^{n-1} \rightarrow \{0, 1\}$ as $g(\mathbf{z})$ is 1 if and only if $f(\alpha, \mathbf{z}) = f(\mathbf{x})$, and $f(\beta, \mathbf{z}) = f(\mathbf{y})$ where we use the notation (α, \mathbf{z}) to denote the vector in $[r]^n$ obtained by inserting α in i th position to $\mathbf{z} \in [r]^{n-1}$. Now, since $f(\alpha, \mathbf{z})$ and $f(\beta, \mathbf{z})$ are both sensitive to at most s coordinates, $g(\mathbf{z})$ is sensitive to at most $2s$ coordinates. Also note that $g^{-1}(1)$ is non-empty. Thus, by Lemma 10, we can conclude that $|g^{-1}(1)|$ is at least r^{n-1-2s} . In other words, each sensitive coordinate contributes at least r^{n-2s-1} edges to $E(f)$. Thus, we can conclude that

$$|E(f)| \geq |A|r^{n-2s-1} \tag{2}$$

Combining Equation (1) and Equation (2), we get

$$|A| \leq s(r-1)r^{2s+1} \tag{3}$$

which proves the required claim. \square

3.2 Low sensitivity polymorphisms of rainbow coloring

We now turn our attention towards our main goal in this section: to show that polymorphisms of $\text{RAINBOW}(k, k-1, q)$ are C -fixing for $q = \lceil \frac{k-2}{2} \rceil$. As we have already mentioned earlier, the polymorphisms of rainbow coloring themselves are rainbow colorings of certain tensor product

hypergraphs. To be precise, the n -ary polymorphisms of $\text{RAINBOW}(k, r, q)$ are precisely q -rainbow colorings of $\text{RH}_n(k, r)$. Thus our new goal is to prove that for any integer $q \geq 2$, any q -rainbow coloring of $\text{RH}_n(2q + 2, 2q + 1)$ is a C -fixing function.

In order to achieve this, we will first define certain hypergraphs similar to $\text{RH}_n(k, r)$.

Definition 12. $\text{H}_n(r, s) = (V, E)$ is a r -uniform hypergraph where the vertex set V is equal to $[r]^n$. A set of vectors $(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r)$ is an edge if and only if

1. In every coordinate $i \subseteq [n]$, at least $r - 1$ elements occur i.e. $\left| \bigcup_j (\mathbf{u}_j)_i \right| \geq r - 1 \quad \forall i \in [n]$.
2. All the r elements occur in at least $n - s$ coordinates i.e. $\left| \bigcup_j (\mathbf{u}_j)_i \right| = r$ for at least $n - s$ choices of i in $[n]$.

The reason behind studying these hypergraphs is that the q -rainbow colorings of $\text{RH}_n(2q + 2, 2q + 1)$ are very closely related to q -rainbow colorings of $\text{H}_n(2q + 1, c)$ for any absolute constant c . In fact if we can prove that q -rainbow colorings of $\text{H}_n(2q + 1, c)$ are C -fixing, it implies that q -rainbow colorings of $\text{RH}_n(2q + 2, 2q + 1)$ are $\max(C, c)$ -fixing. This is formally proved in Lemma 16. Thus our modified objective is to argue that q -rainbow colorings of $\text{H}_n(2q + 1, c)$ are C -fixing. In order to do so, we inductively relate q -rainbow colorings of $\text{H}_n(t, c)$ and $\text{H}_n(t - 1, c - 1)$. As a base case, we have the following lemma:

Lemma 13. For all integers $q \geq 2$ and $n \geq 1$, the hypergraph $\text{H}_n(2q - 1, 1)$ cannot be rainbow colored with q colors.

Proof. We will use induction on q . For the case $q = 2$, rainbow coloring with 2 colors is the same as proper coloring the hypergraph with 2 colors. The fact that $\text{H}_n(3, 1)$ cannot be two colored follows from [ABP18] (Lemma 3.2 with $d = 3$).

Suppose for contradiction that f is a valid q -rainbow coloring of $\text{H}_n(2q - 1, 1)$. Let $r = 2q - 1$ denote the uniformity of the hypergraphs. Consider the set of r vectors in $[r]^n$: $\{\bigcup_{i \in [r]} (i, i, \dots, i)\}$. As there are at most $q < r$ colors, some two elements of this set should have same f value. Without loss of generality, let $f(r - 1, r - 1, \dots, r - 1) = f(r, r, \dots, r) = c$ for some $c \in [q]$. Consider the $(r - 2)$ -uniform hypergraph $H = \text{H}_n(r - 2, 1)$. Note that every edge in H together with $\mathbf{u} = (r - 1, r - 1, \dots, r - 1)$ and $\mathbf{v} = (r, r, \dots, r)$ forms an edge in $\text{H}_n(r, 1)$. Thus, all the $q - 1$ colors in $[q] \setminus \{c\}$ occur in every edge of coloring of $\text{H}_n(r - 2, 1)$ using f . This implies that we can get a valid $(q - 1)$ -rainbow coloring of $\text{H}_n(r - 2 = 2(q - 1) - 1, 1)$ by restricting f to $[r - 2]^n$, and replacing the color c using arbitrary color from $[q] \setminus \{c\}$. However, by the induction hypothesis, such a coloring cannot exist, and thus we have arrived at a contradiction. \square

Now, we will use this to argue about q -rainbow colorings of $\text{H}_n(2q + 1, 3)$ via q -rainbow colorings of $\text{H}_n(2q, 2)$. Consider the hypergraph $\text{H}_n(2q, 2)$. A trivial way to q -rainbow color this hypergraph is to pick a coordinate $i \in [n]$, and partition the set $[2q]$ into q disjoint sets of size two, let's say A_1, A_2, \dots, A_q and assign the value $p \in [q]$ to $f(\mathbf{x})$ for $\mathbf{x} = (x_1, x_2, \dots, x_n)$ if and only if $x_i \in A_p$. It turns out that this is the only way to q -rainbow color $\text{H}_n(2q, 2)$. We prove it in the lemma below:

Lemma 14. Let f be a q -rainbow coloring of $\text{H}_n(r = 2q, 2)$. Then, there exists an index $i \in [n]$, sets $A_1, A_2, \dots, A_q \subseteq [r]$ mutually disjoint and each of size 2, such that $f(\mathbf{x}) = j$ iff $x_i \in A_j$.

Proof. First we will prove that the sensitivity of f is at most 1. Let $\mathbf{x} = (x_1, x_2, \dots, x_n)$ be an arbitrary vector in $[r]^n$. Consider a $(r-1)$ -uniform hypergraph $H(\mathbf{x})$ defined on $([r] \setminus \{x_1\}) \times ([r] \setminus \{x_2\}) \times \dots \times ([r] \setminus \{x_n\})$. We add a $r-1$ vector set as edge of $H(\mathbf{x})$ if and only if it has at most one coordinate where there are missing elements i.e. all the $[r] \setminus \{x_i\}$ occur in all but one coordinate i , and in that coordinate, at most one value is missing.

Note that $H(\mathbf{x})$ is isomorphic to $\mathbb{H}_n(2q-1, 1)$. From Lemma 13, we know that $H(\mathbf{x})$ cannot be rainbow colored with q colors. Thus, when we view f as a coloring of $H(\mathbf{x})$, there is an edge that has a color missing. Let it be denoted by $e = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{r-1})$. Let j be the coordinate where there is a missing element in e . If there is no coordinate with a missing element, j can be arbitrary. Without loss of generality, let color $1 \subseteq [q]$ be missing in e . Note that $\{\mathbf{x}\} \cup e$ is an edge of $\mathbb{H}_n(r, 1)$, and thus an edge of $\mathbb{H}_n(r, 2)$ as well. Since f is a proper q -rainbow coloring of $\mathbb{H}_n(2q, 2)$, we can conclude that $f(\mathbf{x}) = 1$. In fact, we can actually deduce something stronger. Let $\mathbf{y} \in [r]^n$ such that \mathbf{x} and \mathbf{y} differ on exactly one coordinate $j' \in [r]^n, j' \neq j$. Note that $\{\mathbf{y}\} \cup e$ is also a valid edge of $\mathbb{H}_n(2q, 2)$ since it has at most two coordinates where there are missing elements i.e. j' and j . Thus, $f(\mathbf{y}) = 1 = f(\mathbf{x})$. Thus, for every \mathbf{x} , in except for one coordinate, changing the value of the coordinate preserves the value of \mathbf{x} . In other words, the sensitivity of f is at most 1.

Using this, we will now prove that f is a dictator. Let i be an influential coordinate of f i.e. there exists $\mathbf{x}, \mathbf{y} \in [r]^n$ differing only in i th coordinate such that $f(\mathbf{x}) \neq f(\mathbf{y})$. We claim that $f(\mathbf{u}) = f(\mathbf{x})$ for all $\mathbf{u} = (u_1, u_2, \dots, u_n) \in [r]^n$ such that $u_i = x_i$, and $f(\mathbf{u}) = f(\mathbf{y})$ if $u_i = y_i$. We will prove this by induction on the number of coordinates in which \mathbf{x} and \mathbf{u} differ excluding i . Since f has sensitivity at most 1, the only sensitive coordinate of \mathbf{x} and \mathbf{y} is i . Thus, for any \mathbf{u} differing only in one coordinate from \mathbf{x} (other than i) such that $u_i = x_i$ or y_i will have corresponding f value. Suppose that the statement holds for all \mathbf{u} differing from \mathbf{x} in t coordinates excluding i .

Now, let \mathbf{u} differ from \mathbf{x} in $t+1$ coordinates excluding i . We can find $\mathbf{v} \in [r]^n, \mathbf{w} \in [r]^n$ such that \mathbf{v} and \mathbf{x} differ in t coordinates excluding $i, v_i = x_i$; \mathbf{w} and \mathbf{y} differ in t coordinates excluding $i, w_i = y_i$, and one of \mathbf{v} and \mathbf{w} differs from \mathbf{u} in at most one coordinate. By the induction hypothesis, $f(\mathbf{v}) = f(\mathbf{x}), f(\mathbf{w}) = f(\mathbf{y})$. Since \mathbf{v} and \mathbf{w} differ in a single coordinate i, i is the only sensitive coordinate of \mathbf{v} and \mathbf{w} . Thus, $f(\mathbf{u})$ is equal to either $f(\mathbf{v})$ or $f(\mathbf{w})$ depending on $u_i = x_i$ or y_i . This completes the inductive proof.

To complete the proof that f is a dictator, we will use this to show that there cannot be two influential coordinates. Suppose that there are two influential coordinates i and j . From the previous argument, we can infer that there are assignments $i_1, i_2, j_1, j_2 \in [r]$ such that assigning these to corresponding coordinates fixes the value of f . Also note that assigning i as i_1 and i_2 fixes f to different values. Similarly, assigning j as j_1 and j_2 fixes f to different values. This gives rise to contradiction since if we set coordinate i to i_1, f should be fixed irrespective of j is equal to j_1 or j_2 . Thus, there can be only one influential coordinate for f , or in other words, f is a dictator.

Let p be the dictator coordinate of f i.e. there exists a function $g : [r] \rightarrow [q]$ such that $f(\mathbf{x}) = g(x_p)$. From the definition of the hypergraph $\mathbb{H}_n(r, 2)$, for every $j \in [r]$, the set $\{\bigcup_i g(x_i)\} \setminus \{g(x_j)\}$ should be equal to $[q]$. This proves that there exists sets $A_1, A_2, \dots, A_q \subseteq [r]$ each of size two, and mutually disjoint such that $g(\alpha) = j$ if and only if $\alpha \in A_j$, which proves the required claim. \square

We finish the chain of inductive arguments by proving a key property of q -rainbow colorings of $\mathbb{H}_n(2q+1, 3)$.

Lemma 15. *Let $f : [2q+1]^n \rightarrow [q]$ be a q -rainbow coloring of $\mathbb{H}_n(r=2q+1, 3)$. Then, there exists an index $i \in [n]$, and $\alpha \in [r]$ such that $S(f, \mathbf{x}) \leq 2$ for all $\mathbf{x} \in [r]^n$ such that $x_i = \alpha$.*

Proof. Let $\mathbf{x} = (x_1, x_2, \dots, x_n) \in [r]^n$ be an arbitrary vector in $[r]^n$. Similar to the previous lemma, we define the complement hypergraph associated with \mathbf{x} . Consider a $(r-1)$ -uniform hypergraph $H(\mathbf{x})$ defined on $([r] \setminus \{x_1\}) \times ([r] \setminus \{x_2\}) \times \dots \times ([r] \setminus \{x_n\})$. We add a $r-1$ vector set as edge of $H(\mathbf{x})$ if and only if it has at most two coordinates where there are missing elements i.e. all the $[r] \setminus \{x_i\}$ occur in all but two coordinates i , and in these two coordinates, at least $r-2$ values occur. Note that $H(\mathbf{x})$ is isomorphic to $\mathbb{H}_n(r-1, 2)$.

We can view $f : [2q+1]^n \rightarrow [q]$ as a q -coloring of $H(\mathbf{x})$. If f is not a valid q -rainbow coloring of $H(\mathbf{x})$, by the same argument as in Lemma 14, we can deduce that $S(f, \mathbf{x}) \leq 2$. If f is a valid q -rainbow coloring of $H(\mathbf{x})$, we will use the properties proved in Lemma 14. Let us define a function $g : [r]^n \rightarrow [n] \cup \{\perp\}$ such that for a vector $\mathbf{x} \in [r]^n$,

1. If f is a valid q -rainbow coloring of $H(\mathbf{x})$, then Lemma 14 implies that there exists a coordinate $i \in [n]$ such that f is a dictator in i th coordinate in $H(\mathbf{x})$. In this case, set $g(\mathbf{x}) = i$.
2. If f is not a valid q -rainbow coloring of $H(\mathbf{x})$, let $g(\mathbf{x}) = \perp$.

First, we will prove that there exists an index $i \in [n]$ such that $g(\mathbf{x}) \in \{i, \perp\}$ for all $\mathbf{x} \in [r]^n$. Suppose $g(\mathbf{x}) = i \in [n]$, and $g(\mathbf{y}) = j \in [n]$ where $\mathbf{x}, \mathbf{y} \in [r]^n$ and $i \neq j$. Since $g(\mathbf{x}) = i$, there exist sets $S_1, S_2, \dots, S_n \subseteq [r]$ such that f is a dictator on i th coordinate in $S = S_1 \times S_2 \times \dots \times S_n \subseteq [r]^n$. In particular, there is a subset $A \subseteq S_i$ such that $|A| = 2$, and $f(\mathbf{x}), \mathbf{x} \in S$, is equal to 1 if and only if $x_i \in A$. Similarly, there exist sets $T_1, T_2, \dots, T_n \subseteq [r]$ such that f is a dictator on j th coordinate in $T = T_1 \times T_2 \times \dots \times T_n \subseteq [r]^n$. There exists a subset $B \subseteq T_j$ such that $|B| = 2$, and $f(\mathbf{x}), \mathbf{x} \in T$ is equal to $c \neq 1$ if and only if $x_j \in B$ for some $c \in [q]$. Combining the both, let $U_i = S_i \cap T_i, |U_i| \geq r-2 \forall i \in [n]$. We can deduce that f is a dictator in both i and j coordinates in $U = U_1 \times U_2 \times \dots \times U_n$. This implies that f is a constant function in U . Recall that there are two assignments in S_i that make f equal to 1 and two assignments in T_j that make f equal to $c \neq 1$. Thus, $f(\mathbf{x}')$ is equal to 1 for some $\mathbf{x}' \in U$ and $f(\mathbf{y}') = c \neq 1$ for some $\mathbf{y}' \in U$. This contradicts the fact that f is a constant function in U . Thus, there exists an index $i \in [n]$ such that $g(\mathbf{x})$ is either equal to i or is equal to \perp for all $\mathbf{x} \in [r]^n$. Without loss of generality let that be the first coordinate i.e. for all $\mathbf{x} \in [r]^n, g(\mathbf{x}) \in \{1, \perp\}$.

Consider the case when $g(\mathbf{x}) = \perp$ for every $\mathbf{x} \in [r]^n$. In this case, we know that $S(f, \mathbf{x}) \leq 2$ for all $\mathbf{x} \in [r]^n$. In particular, we can set α arbitrary and say that $S(f, \mathbf{x}) \leq 2$ whenever $x_1 = \alpha$. So we are only left with the case when there exists a $\mathbf{x} \in [r]^n$ such that $g(\mathbf{x}) = 1$. We will now prove that there exists $\alpha \in [r]$ such that $g(\mathbf{x}) = \perp$ whenever $x_1 = \alpha$, thus proving the required sensitivity bound.

Suppose for contradiction that for every $\alpha \in [r]$, there exists $\mathbf{x} \in [r]^n$ such that $x_1 = \alpha$, and $g(\mathbf{x}) = 1$. Consider a pair $\mathbf{u}, \mathbf{v} \in [r]^n$ such that $u_1 = \alpha, v_1 = \beta \neq \alpha$ and $g(\mathbf{u}) = g(\mathbf{v}) = 1$. Let $\mathbf{u} = (u_1, u_2, \dots, u_n), S_i = [r] \setminus \{u_i\}$ and f is dictator on 1st coordinate in $S = S_1 \times S_2 \times \dots \times S_n$. There is a function $h_1 : S_1 \rightarrow [q]$ such that $f(\mathbf{x}) = h_1(x_1)$ if $\mathbf{x} \in S$ and $|h_1^{-1}(c)| = 2 \forall c \in [q]$.

Similarly, let $\mathbf{v} = (v_1, v_2, \dots, v_n)$, $T_i = [r] \setminus \{v_i\}$ and f is dictator on first coordinate in $T = T_1 \times T_2 \times \dots \times T_n$. There is a function $h_2 : T_1 \rightarrow [q]$ such that $f(\mathbf{x}) = h_2(x_1)$ if $\mathbf{x} \in T$ and $|h_2^{-1}(c)| = 2 \quad \forall c \in [q]$. Let $U_i = S_i \cap T_i$. Note that $U = U_1 \times U_2 \times \dots \times U_n$ is non empty and f is dictator on 1st coordinate in U as well. Note that $|U_i| \geq r - 2$ for all $i \in [n]$. Thus, we can conclude that if $\gamma \in U_1$, then $h_1(\gamma) = h_2(\gamma)$.

Applying this to all pairs \mathbf{u}, \mathbf{v} such that $g(\mathbf{u}) = g(\mathbf{v}) = 1$, we can infer that there exists a function $h : [r] \rightarrow [q]$ that satisfies the property that for all $\mathbf{x} \in [r]^n$ such that $g(\mathbf{x}) = 1$, let $\mathbf{x} = (x_1, x_2, \dots, x_n)$, $S_i = [r] \setminus \{x_i\}$, $S = S_1 \times S_2 \times \dots \times S_n$, then $f(\mathbf{y}) = h(y_1)$ for all $\mathbf{y} \in S$. As $r = 2q + 1 > 2q$, there exists $b \in [q]$ such that $|h^{-1}(b)| \geq 3$. Let $\gamma \in [r]$ be such that $h(\gamma) \neq b$. From our assumption that for every $\alpha \in [r]$ there exists $\mathbf{x} \in [r]^n$ such that $g(\mathbf{x}) = 1$ and $x_1 = \alpha$, there exists $\mathbf{u} \in [r]^n$ such that $u_1 = \gamma$ and $g(\mathbf{u}) = 1$. Now, let $\mathbf{u} = (u_1, u_2, \dots, u_n)$, $S_i = [r] \setminus \{u_i\}$, $S = S_1 \times S_2 \times \dots \times S_n$, and we know that $f(\mathbf{x}) = h(x_1)$ if $\mathbf{x} \in S$, and $|h^{-1}(c) \cap S_1| = 2 \forall c \in [q]$. However, this contradicts the fact that $h(u_1) = h(\gamma) \neq b$, and $|h^{-1}(b)| = 3$. Thus, there exists $\alpha \in [r]$ such that $g(\mathbf{x}) = \perp$ for all $\mathbf{x} \in [r]^n$ such that $x_1 = \alpha$. \square

Finally, we will use the previous hypergraph coloring properties to argue about polymorphisms of rainbow coloring.

Lemma 16. *There exist constant $C = C(q)$ independent of n such that every $f : [2q + 1]^n \rightarrow [q]$ that is an n -ary polymorphism of $\text{RAINBOW}(2q + 2, 2q + 1, q)$ i.e. f is a proper q -rainbow coloring of $\text{RH}_n(2q + 2, 2q + 1)$ is C -fixing.*

Proof. Let $r = 2q + 1$. Let $f : [r]^n \rightarrow [q]$ be a polymorphism of $\text{RAINBOW}(2q + 2, 2q + 1, q)$. We can view f as a q -rainbow coloring of $\text{H}_n(r, 3)$ as the vertex set of $\text{RH}_n(r + 1, r)$ and of $\text{H}_n(r, 3)$ is equal to $[r]^n$. If it is not a valid q -rainbow coloring, there is an edge in which not all q colors appear. Let that edge be $e = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r)$ and $c \in [q]$ be a missing color in $\{f(\mathbf{v}_1), f(\mathbf{v}_2), \dots, f(\mathbf{v}_r)\}$. Since this edge is part of $\text{H}_n(r, 3)$, except for 3 values of i , for all other i , the set $((\mathbf{v}_1)_i, (\mathbf{v}_2)_i, \dots, (\mathbf{v}_r)_i)$ is equal to $[r]$. Let the missing coordinates be the set $S = \{i_1, i_2, i_3\}$. Now consider an element \mathbf{u} of $[r]^n$ such that it has missing values of e in S . From the definition of $\text{RH}_n(r + 1, r)$, we can deduce that the set $e \cup \mathbf{u}$ is an edge of $\text{RH}_n(r + 1, r)$. Since f is a valid q -rainbow coloring of $\text{RH}_n(r + 1, r)$, $f(\mathbf{u})$ is equal to c . Note that this should hold irrespective of what values \mathbf{u} has in coordinates outside S . This proves that f is C -fixing with $C = 3$.

On the other hand if f is a valid q -rainbow coloring of $\text{H}_n(r, 3)$, using Lemma 15, we can deduce that there exists an index $i \in [n]$, and $\alpha \in [r]$ such that $S(f, \mathbf{x}) \leq 2$ whenever $x_i = \alpha$. Now, we can consider a function $g : [r]^{n-1} \rightarrow [q]$ which on an input $\mathbf{y} \in [r]^{n-1}$, is equal to $f(\mathbf{x})$, $\mathbf{x} = \mathbf{y}, x_i \leftarrow \alpha \in [r]^n$ i.e. we first insert α in i th position to \mathbf{y} and then apply f . Note that g has sensitivity at most 2. From Lemma 11, we can conclude that g is C -fixing for $C = 2(r - 1) \cdot r^5$. In other words, g is fixed by assigning values to a set of C indices. This implies that f is also $C' = C + 1$ -fixing since we can first assign i th index to α , then use C -fixing property of g . \square

3.3 High sensitivity polymorphism of $\text{RAINBOW}(7, 6, 2)$

We show that there exists a function $f : [6]^n \rightarrow \{0, 1\}$ that is a polymorphism of $\text{RAINBOW}(7, 6, 2)$ that is not C -fixing for any constant C . We start with a dictator but add just

enough noise such that the function still remains being a polymorphism, but it is no longer C -fixing. Let $wt(\mathbf{x})$ denote the number of $i \in [n], i > 1$ such that $x_i = 1$. Let $S \subseteq [6]^n$ denote the set of $\mathbf{x} \in [6]^n$ such that $wt(\mathbf{x}) > \frac{2n}{3}$. Let $h : [6]^n \rightarrow \{0, 1\}$ be noise function defined below. For a given $\mathbf{x} \in [6]^n$, we define $f(\mathbf{x})$ as follows:

1. If $\mathbf{x} \notin S$
 - (a) If $x_1 \leq 3$, $f(\mathbf{x}) = 0$
 - (b) Else, $f(\mathbf{x}) = 1$
2. Else $f(\mathbf{x}) = h(\mathbf{x})$.

A choice of noise function that works is inverting the original function: $h(\mathbf{x})$ is defined as 1 if and only if $x_1 \leq 3$.

Proposition 17. *The function $f : [6]^n \rightarrow \{0, 1\}$ defined above is a polymorphism of $\text{RAINBOW}(7, 6, 2)$ and it is not C -fixing for any $C < \frac{n}{3}$.*

Proof. Any polymorphism of $\text{RAINBOW}(7, 6, 2)$ is a proper 2-rainbow coloring of $\text{RH}_n(7, 6)$. Recall that rainbow coloring with two colors is the same as standard hypergraph coloring with two colors.

Polymorphism. In any set of 7 vectors E in $[6]^n$ such that all the 6 elements occur in all the coordinates, at most two vectors can be in S . This is because, in any set of three vectors in S , there exists a coordinate in which all three values are equal to 1. Thus, there are vectors $\mathbf{x} \notin S$ with $x_1 \leq 3$ and vector $\mathbf{y} \notin S$ such that $y_1 \geq 3$ in E , which together ensures that E is not monochromatic.

C -fixing. Suppose that there exists a set $T = \{t_1, t_2, \dots, t_m\} \subseteq [n]$ and $(\alpha_1, \alpha_2, \dots, \alpha_m) \subseteq [6]^m$ such that $f(\mathbf{x}) = b$ for all \mathbf{x} such that $x_i = \alpha_i$ for all $1 \leq i \leq m$, for some fixed $b \in \{0, 1\}$. We will prove that $|T| \geq \frac{n}{3}$. Suppose for contradiction that $|T| < \frac{n}{3}$. First, if $1 \notin T$, we can set all coordinates outside T to be equal to $\beta \neq 1$, and in this case $f(\mathbf{x}) = x_1$, which cannot be fixed if $1 \notin T$. Thus $1 \in T$. Next, if all the coordinates outside T are all equal to 1, then $f(\mathbf{x})$ is equal to noise function, which is different from the case when the rest are equal to $\beta \neq 1$. Thus, if f is indeed a C -fixing function, for the C -fixing assignment, the value of f should be independent of the assignment to the coordinates outside T . However, that is not the case as the value of f changes when we set all the coordinates outside T to be 1 or $\beta \neq 1$. \square

4 NP-Hardness

In this section, we will use the properties of polymorphisms proved so far to argue about NP hardness of rainbow coloring PCSP. We will prove the below theorem:

Theorem 18. *Suppose that there exists a constant C such that for all integers $n \geq 1$, every n -ary polymorphism of $\text{RAINBOW}(k, k - 1, q)$ is C -fixing. Then, the corresponding decision problem $\text{RAINBOW}(k, k - 1, q)$ is NP hard.*

Before delving into the proof of Theorem 18, we first mention that this theorem together with Lemma 16 implies Theorem 1. In Lemma 16, we have proved that for every $q \geq 2$, the

polymorphisms of $\text{RAINBOW}(2q+2, 2q+1, q)$ are C -fixing. This fact combined with Theorem 18 implies that $\text{RAINBOW}(2q+2, 2q+1, q)$ is NP hard for every $q \geq 2$. This already proves Theorem 1 when k is even. When k is odd, we can combine Lemma 14 and Lemma 16 to prove that the polymorphisms of $\text{RAINBOW}(k = 2q+1, 2q, q)$ are C -fixing. We can combine this with Theorem 18 to prove Theorem 1 when k is odd.

The rest of this section is dedicated to proving Theorem 18. Like various other hardness of approximation results, we will use the standard label cover with long code framework. We reduce *smooth* label cover introduced in [Kho02] to rainbow coloring PCSP. First we define Label Cover problem below:

Definition 19. (*Label Cover*) In an instance of Label Cover problem, we are given a tuple $(G = (L, R, E), \Sigma, \Pi)$ where

1. G is a bipartite multi graph between vertex sets L and R
2. Each vertex in G has to be assigned a label from Σ
3. For each edge $e = (u, v) \in E$, there is a projection constraint Π_e from u to v that is a function from Σ to itself. This corresponds to a constraint between u and v .

A labelling of graph is a function $\sigma : L \cup R \rightarrow \Sigma$ that assigns a label to each vertex of G . A labelling σ is said to satisfy constraint Π_e if and only if $\Pi_e(\sigma(u)) = \sigma(v)$.

We refer to L and R as left and right vertices respectively. We are now ready to define Gap Label Cover.

Definition 20. ($(1, \epsilon_{LC})$ Gap Label Cover) In $(1, \epsilon_{LC})$ Gap Label Cover, we are given a Label Cover instance $(G = (L, R, E), \Sigma, \Pi)$, and the goal is to distinguish between the following two cases:

1. There is a labelling $\sigma : G \rightarrow \Sigma$ that satisfies all the constraints.
2. No labelling can satisfy ϵ_{LC} fraction of constraints.

As mentioned earlier, we need stronger properties of the Label Cover instance that we are starting with. One such property is smoothness.

Definition 21. (*Smoothness*) A Label Cover instance $(G = (L, R, E), \Sigma, \Pi)$ is said to be (J, ϵ) -smooth if for any vertex $u \in L$ and a set of labels $S \subseteq \Sigma, |S| \leq J$, over a uniformly random neighbor $v \in R$, $\Pr(|\bigcup_{s \in S} \Pi_{u,v}(s)| < |S|) \leq \epsilon$.

The following is a special case of Theorem 1.17 in [Wen13].

Theorem 22. For every $\epsilon, \epsilon_{LC} > 0$ and $J \in \mathbb{Z}_+$, there exists $n = n(\epsilon, \epsilon_{LC}, J)$ such that $(1, \epsilon_{LC})$ Gap Label Cover with $|\Sigma| = n$ that is promised to be (J, ϵ) -smooth is NP hard.

We now prove Theorem 18.

Reduction. We start with $(1, \epsilon_{LC})$ Gap Label Cover instance $(G = (L, R, E), \Sigma, \Pi)$ that is promised to be (C, ϵ) -smooth, for ϵ and ϵ_{LC} to be set later, and output a PCSP instance. The reduction described here is the same as the general one from Label Cover to PCSP in e.g. [BKO18]. Let n denote the label size $n = |\Sigma|$. For each vertex $v \in L \cup R$, we add a set of nodes K_v of size $[k-1]^n$, indexed by n length vectors. We add two types of constraints:

1. Coloring constraints: Inside every vertex of the Label Cover instance, we add the following constraints among the $[k - 1]^n$ nodes. We add the constraint that the promise relation should be satisfied in the set T of k nodes $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\}$ in $[k - 1]^n$, if for every $i \in [n]$, the set $\{\bigcup_j (\mathbf{x}_j)_i\}$ has cardinality $k - 1$.
2. Equality constraints: For every constraint $\Pi_e : u \rightarrow v$ of the Label Cover, we add a set of equality constraints between nodes $\mathbf{x} \in K_u, \mathbf{y} \in K_v$ if for all $i \in [n]$, $\mathbf{x}_i = \mathbf{y}_{\Pi_e(i)}$.

Note that the Coloring constraints give rise to rainbow colorings of k -uniform hypergraphs. It is yet unclear how we can justify adding equality constraints. One way to handle the equality constraints is to have a single node for all the vertices corresponding to equality constraint. However, this fails if we want to add a coloring constraint that involves two copies of the same vertex. A neater way to get around this is to argue that adding equality constraints does not change the set of polymorphisms, and thus the hardness of the predicate remains the same with or without equality constraints. This simple fact is proved in e.g. [BG18, BKO18].

Completeness. If the label cover instance is satisfiable, then PCSP instance that is being output can be satisfied by assignment from $[k - 1]$. Suppose $\sigma : L \cup R \rightarrow \Sigma$ is a labeling that satisfies all the constraints of the Label Cover. For every vertex $\mathbf{x} \in K_u$ corresponding to the vertex $u \in L \cup R$, we can assign the value $\mathbf{x}_{\sigma(u)}$. In other words, in every long code, we are assigning corresponding dictator function. The coloring constraints are defined precisely such that this assignment satisfies the constraints. The equality constraints also follow since the labeling σ satisfies all the constraints of the Label Cover.

Soundness. If the Label Cover is not ϵ_{LC} satisfiable, we need to show that there is no assignment of the PCSP instance in $[q]$ that satisfies all the constraints. Taking contrapositive, if there is an assignment in $[q]$ to PCSP instance that satisfies all the constraints, then we will prove that there is an assignment to the Label Cover instance that can satisfy a c fraction of constraints, for an absolute constant c . Taking $\epsilon_{LC} < c$, we can arrive at a contradiction, thus proving that there is no assignment in $[q]$ to PCSP that satisfies all the constraints.

Let $f_v : [k - 1]^n \rightarrow [q]$ denote the assignment to the PCSP instance that satisfies all the constraints for $v \in L \cup R$. From the coloring constraints, we can infer that f_v is a n -ary polymorphism of $\text{RAINBOW}(k, k - 1, q)$. Thus, it is C -fixing for a constant C independent of n .

For every vertex $v \in L \cup R$ of the Label Cover instance, we will assign a set of labels $A(v) \subseteq [n]$. For vertices v in L , $A(v)$ is the C -fixing set. Since the Label Cover instance is smooth, we will only consider the constraints where all these labels go to distinct labels on the right under projections. We can set the smoothness parameter ϵ to be 0.1 for example, and we will be left with $\frac{9}{10}$ fraction of original constraints. We will prove that there is an assignment that satisfies a c fraction of these constraints, for an absolute constant c , which will prove the original soundness claim. Thus in all the remaining constraints, the set of labels in $A(v)$ go to distinct labels on the right. Thus, for a vertex $v \in R$, each constraint (u, v) gives rise to C coordinates $\Pi_{u,v}(A(u))$. Note that these C coordinates are in fact C -fixing for v for every constraint (u, v) . For a given $v \in R$, there are several such C -fixing sets. Let the set of these C -fixing sets be denoted by $B(v) = \{S_1, S_2, \dots\}$ where each $S_i \subseteq [n]$ is a C -fixing set of f_v . Now we define $A(v)$ for $v \in R$ to be the set of union of an arbitrary fixed maximal disjoint sets in $B(v)$.

In order to prove that there is a good labeling to the Label Cover, we assign a label to every vertex v from $A(v)$ uniformly at random and prove that it satisfies a constant fraction

of constraints with non-zero probability. We will, in fact, show that the random assignment satisfies a constant fraction of constraints in expectation. We prove this in two steps. First, we show that for every constraint (u, v) of the Label Cover, there exists $x \in A(u), y \in A(v)$ such that $\Pi_{u,v}(x) = y$. This follows from the definitions of $A(v)$: suppose the projection of $A(u)$ is disjoint from $A(v)$. In that case, we can add the projection of $A(u)$ to $A(v)$ to get a larger set in v , which contradicts the fact that $A(v)$ is the maximal such union of disjoint projections. This implies that the uniformly random labelling satisfies each constraint (u, v) of Label Cover with probability at least $\frac{1}{|A(u)||A(v)|}$.

To complete the proof, we need to bound the sizes of $A(v)$. As we have already mentioned, for $v \in L, |A(v)| \leq C$. We bound the size of $A(v)$ for vertices v in R using the below lemma.

Lemma 23. *Suppose $f : [k - 1]^n \rightarrow q$ is a polymorphism of $\text{RAINBOW}(k, k - 1, q)$. Let A_1, A_2, \dots, A_t be mutually disjoint subsets of $[n]$ such that each of them is a C -fixing set of f . Then, $t < k$.*

Proof. First note that all the A_i s should fix f to the same value in $[q]$ since otherwise, the vector $\mathbf{u} \in [k - 1]^n$ that has all the fixing sets in A_i s is forced to be equal to multiple colors in $[q]$ at the same time. Let all the A_i s be C -fixing with respect to value $b \in [q]$ i.e. for each $i \in [t]$, there exists an assignment to A_i such that if the value of \mathbf{x} in A_i is equal to the assignment, then the value of $f(\mathbf{x})$ is equal to b irrespective of values of coordinates outside A_i . If $t \geq k$, we can find $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k$ such that all $[k - 1]$ occur in every coordinate, and \mathbf{y}_i has the fixing assignment of A_i . This implies that $f(\mathbf{y}_i) = b$ for all i . However, note that $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k\}$ is an edge of $\text{RH}_n(k, k - 1)$, and thus if f is a polymorphism of $\text{RAINBOW}(k, k - 1, q)$, all the $[q]$ elements should occur in $\{f(\mathbf{y}_1), f(\mathbf{y}_2), \dots, f(\mathbf{y}_k)\}$. This is a contradiction since for all $i, f(\mathbf{y}_i) = b$. \square

From the lemma, we can infer that the cardinality of $A(v)$ for $v \in R$ is at most kC . Combining this with the above, we can deduce that there is an assignment that satisfies $\frac{1}{kC^2}$ fraction of constraints, which is a constant fraction of constraints, independent of n .

5 Conclusion

In this paper, we have proved that given a k -uniform hypergraph that is promised to be $(k - 1)$ -rainbow colorable, it is NP hard to rainbow color it with $\lceil \frac{k-2}{2} \rceil$ colors. As a corollary, we can deduce that for $k \leq 6$, it is NP hard to 2-color a k -uniform hypergraph that is promised to be $(k - 1)$ -rainbow colorable. An immediate question is whether $\text{RAINBOW}(7, 6, 2)$ is NP hard. It would be interesting to get an efficient algorithm though we believe it is unlikely. In Section 3.3, we have provided a polymorphism of $\text{RAINBOW}(7, 6, 2)$ that is not C -fixing. The polymorphisms for this PCSP also have other symmetries (in the form of identities) discussed in [BKO18].

However, it should be noted the polymorphism that we have given in Section 3.3 is very far from symmetric, it seems that we should decode to the unique special coordinate. What we are missing here is a characterization of lack of symmetries that works well with Label Cover to give NP-hardness. We believe that resolving the hardness of this particular PCSP can shed light on identifying criteria for lack of symmetries that imply hardness, beyond C -fixing. Another direction to explore is whether we can further strengthen the completeness in our result. More concretely, given a k -rainbow colorable k -uniform hypergraph, can we efficiently rainbow color it with 3 colors?

Acknowledgments

We thank Amey Bhangale, Joshua Brakensiek, Jakub Opršal, and Xinyu Wu for useful discussions. We would also like to thank anonymous reviewers for helpful comments.

References

- [ABP18] Per Austrin, Amey Bhangale, and Aditya Potukuchi. Improved inapproximability of rainbow coloring. *CoRR*, abs/1810.02784, 2018.
- [ABP19] Per Austrin, Amey Bhangale, and Aditya Potukuchi. Simplified inapproximability of hypergraph coloring via t-agreeing families. *CoRR*, abs/1904.01163, 2019.
- [AGH17] Per Austrin, Venkatesan Guruswami, and Johan Håstad. $(2+\epsilon)$ -sat is NP-hard. *SIAM J. Comput.*, 46(5):1554–1573, 2017.
- [APV16] Andris Ambainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. Sensitivity versus certificate complexity of boolean functions. In *Proceedings of the 11th International Computer Science Symposium on Computer Science — Theory and Applications - Volume 9691*, CSR 2016, pages 16–28, 2016.
- [BG16] Joshua Brakensiek and Venkatesan Guruswami. New hardness results for graph and hypergraph colorings. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 14:1–14:27, 2016.
- [BG17] Joshua Brakensiek and Venkatesan Guruswami. The quest for strong inapproximability results with perfect completeness. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, pages 4:1–4:20, 2017.
- [BG18] Joshua Brakensiek and Venkatesan Guruswami. Promise constraint satisfaction: Structure theory and a symmetric boolean dichotomy. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1782–1801, 2018.
- [BGL15] Vijay V. S. P. Bhattiprolu, Venkatesan Guruswami, and Euiwoong Lee. Approximate hypergraph coloring under low-discrepancy and related promises. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24-26, 2015, Princeton, NJ, USA*, pages 152–174, 2015.
- [Bha18] Amey Bhangale. NP-hardness of coloring 2-colorable hypergraph with polylogarithmically many colors. In *45th International Colloquium on Automata, Languages, and Programming*, pages 15:1–15:11, 2018.
- [BK10] Nikhil Bansal and Subhash Khot. Inapproximability of hypergraph vertex cover and applications to scheduling problems. In *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part I*, pages 250–261, 2010.

- [BKO18] Jakub Bulín, Andrei A. Krokhin, and Jakub Oprsal. Algebraic approach to promise constraint satisfaction. *CoRR*, abs/1811.00970, 2018. STOC 2019, to appear.
- [Bul17] Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 319–330, 2017.
- [DMR09] Irit Dinur, Elchanan Mossel, and Oded Regev. Conditional hardness for approximate coloring. *SIAM J. Comput.*, 39(3):843–873, 2009.
- [DRS05] Irit Dinur, Oded Regev, and Clifford D. Smyth. The hardness of 3-uniform hypergraph coloring. *Combinatorica*, 25(5):519–535, 2005.
- [GHS02] Venkatesan Guruswami, Johan Håstad, and Madhu Sudan. Hardness of approximate hypergraph coloring. *SIAM J. Comput.*, 31(6):1663–1686, 2002.
- [GL18] Venkatesan Guruswami and Euiwoong Lee. Strong inapproximability results on balanced rainbow-colorable hypergraphs. *Combinatorica*, 38(3):547–599, 2018.
- [GS17] Venkatesan Guruswami and Rishi Saket. Hardness of rainbow coloring hypergraphs. In *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2017, December 11-15, 2017, Kanpur, India*, pages 33:33–33:15, 2017.
- [Hås01] Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.
- [Hol02] Jonas Holmerin. Vertex cover on 4-regular hyper-graphs is hard to approximate within 2-epsilon. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing*, pages 544–552, 2002.
- [Hua19] Hao Huang. Induced subgraphs of hypercubes and a proof of the sensitivity conjecture. *arXiv preprint arXiv:1907.00847*, 2019.
- [Kho02] Subhash Khot. Hardness results for coloring 3-colorable 3-uniform hypergraphs. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 23–32. IEEE, 2002.
- [KO19] Andrei A. Krokhin and Jakub Oprsal. The complexity of 3-colouring H-colourable graphs. *arXiv preprint arXiv:1904.03214*, 2019.
- [KS14] Subhash Khot and Rishi Saket. Hardness of finding independent sets in 2-colorable and almost 2-colorable hypergraphs. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1607–1625, 2014.
- [McD93] Colin McDiarmid. A random recolouring method for graphs and hypergraphs. *Combinatorics, Probability and Computing*, 2(3):363–365, 1993.
- [MM08] Miklós Maróti and Ralph McKenzie. Existence theorems for weakly symmetric operations. *Algebra universalis*, 59(3):463–489, Dec 2008.
- [MOO10] Elchanan Mossel, Ryan O’Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low influences: invariance and optimality. *Annals of Mathematics*, 171:295–341, 2010.

- [Mos10] Elchanan Mossel. Gaussian bounds for noise correlation of functions. *Geometric and Functional Analysis*, 19:1713–1756, 2010.
- [Nis89] N. Nisan. CREW PRAMs and decision trees. In *Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing*, STOC '89, pages 327–335. ACM, 1989.
- [Sak14] Rishi Saket. Hardness of finding independent sets in 2-colorable hypergraphs and of satisfiable csps. In *Proceedings of the 29th IEEE Conference on Computational Complexity*, pages 78–89, 2014.
- [Sim83] Hans-Ulrich Simon. A tight $\omega(\log \log n)$ -bound on the time for parallel RAMs to compute nondegenerated boolean functions. In *Foundations of Computation Theory*, pages 439–444. Springer Berlin Heidelberg, 1983.
- [SS13] Sushant Sachdeva and Rishi Saket. Optimal inapproximability for scheduling problems via structural hardness for hypergraph vertex cover. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013*, pages 219–229, 2013.
- [Wen13] Cenny Wenner. Circumventing d -to-1 for approximation resistance of satisfiable predicates strictly containing parity of width at least four. *Theory of Computing*, 9(23):703–757, 2013.
- [WZ19] Marcin Wrochna and Stanislav Zivny. Improved hardness for H-colourings of G-colourable graphs. *arXiv preprint arXiv:1907.00872*, 2019.
- [Zhu17] Dmitriy Zhuk. A proof of CSP dichotomy conjecture. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 331–342, 2017.