Strong Sparsification for 1-in-3-SAT via Polynomial Freiman-Ruzsa

Benjamin Bedert
Mathematical Institute
University of Oxford
Oxford, UK
benjamin.bedert@maths.ox.ac.uk

Tamio-Vesa Nakajima
Department of Computer Science
University of Oxford
Oxford, UK
tamio-vesa.nakajima@cs.ox.ac.uk

Karolina Okrasa
Department of Computer Science
University of Oxford
Oxford, UK
karolina.okrasa@cs.ox.ac.uk

Stanislav Živný
Department of Computer Science
University of Oxford
Oxford, UK
standa.zivny@cs.ox.ac.uk

***Abstract*—We introduce a new notion of sparsification, called *strong sparsification*, in which constraints are not removed but variables can be merged. As our main result, we present a strong sparsification algorithm for 1-in-3-SAT. The correctness of the algorithm relies on establishing a sub-quadratic bound on the size of certain sets of vectors in $\mathbb{F}_2^d$. This result, obtained using the recent *Polynomial Freiman-Ruzsa Theorem* (Gowers, Green, Manners and Tao, Ann. Math. 2025), could be of independent interest. As an application, we improve the state-of-the-art algorithm for approximating linearly-ordered colourings of 3-uniform hypergraphs (Håstad, Martinsson, Nakajima and Živný, APPROX 2024).**

***Index Terms*—sparsification, 1-in-3-SAT, additive combinatorics, approximation, linearly-ordered colourings**

I. Introduction

Sparsification, the idea of reducing the size of an object of interest (such as a graph or a formula) while preserving its inherent properties, has been tremendously successful in many corners of computer science.

One notion of sparsification comes from the influential paper of Benczúr and Karger [11], who showed that, for any n-vertex graph G, one can efficiently find a weighted subgraph G' of G with $O(n \log n)$ many edges, so that the size of *all* cuts in G is approximately preserved in G', up to a small multiplicative error. The bound on the size of G' was later improved to linear by Batson, Spielman and Srivastava [9]. Andoni, Chen, Krauthgamer, Qin, Woodruff. and Zhang showed that the dependency on ε is optimal [1]. From the many follow-up works, we mention the paper of Kogan and Krauthgamer [35], who initiated the study of sparsification for constraint satisfaction problems (CSPs), Filtser and Krauthgamer [23], who classified sparsifiable Boolean binary CSPs, and Butti and Živný [15], who classified sparsifiable binary CSPs on all finite domains. Some impressive progress on this line of work has been made in recent years by Khanna, Putterman, and Sudan [32]–[34], establishing optimal sparsifiers for classes of CSPs and codes, and Brakensiek and Guruswami [14], who pinned down the sparsifiability of all CSPs (up to polylogarithmic factors, and non-efficiently). A different but related notion of sparsification is the concept of (unweighted) additive cut sparsification, introduced by Bansal, Svensson and Trevisan [6], later studied for other CSPs by Pelleg and Živný [38].

Another study on sparsification comes from computational complexity. The *Exponential Time Hypothesis* (ETH) of Impagliazzo, Paturi and Zane [30] postulates that 3-SAT requires exponential time: there exists $\delta > 0$ such that an n-variable 3-SAT instance requires time $O(2^{\delta n})$.[1] The *sparsification lemma* from the same paper [30] is then used to establish that ETH implies that exponential time is needed for a host of other problems. The lemma roughly says that any n-variable 3-SAT instance is equisatisfiable to an OR of exponentially many 3-SAT formulae, each of which has only linearly many clauses in n.[2] This should be contrasted with what can (or rather cannot) be done in polynomial time: under the assumption that $\mathrm{NP} \not\subseteq \mathrm{coNP}/\mathrm{poly}$, Dell and Melkebeek showed that 3-SAT cannot be sparsified in polynomial time into an equivalent formula with $O(n^{3-\varepsilon})$ clauses [19].

Drawing on techniques from fixed-parameter tractability [18] and kernelisation [18], [24], Jansen and Pieterse [31] and Chen, Jansen and Pieterse [17] studied which NP-complete Boolean CSPs admit a non-trivial sparsification. In particular,

This work was supported by UKRI EP/X024431/1 and a Clarendon Fund Scholarship. For the purpose of Open Access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. All data is provided in full in the results section of this paper. The full version of this paper is arXiv:2507.17878 [10].

[1] The weaker hypothesis $\mathrm{P} \neq \mathrm{NP}$ only postulates that 3-SAT requires super-polynomial time.

[2] The precise statement includes a universal quantification over an arbitrarily small $\varepsilon > 0$ that controls the growth of the exponentials involved.

they observed that 1-in-3-SAT admits a linear-size sparsifier, meaning an equivalent instance with $O(n)$ many clauses, where n is the number of variables [31]. Building on techniques from the algebraic approach to CSPs, Lagerkvist and Wahlström then considered CSPs over domains of larger size [36].

In the present article we will be interested in sparsifying *approximate* problems. When dealing with NP-hard problems, there are two natural ways to relax the goal of exact solvability and turn to approximation: a quantitative one and a qualitative one. The first one seeks to maximise the number of satisfied constraints. A canonical example is the max-cut problem: finding a cut of maximum size is NP-hard, but a cut of size at least roughly 0.878 times the optimum can be efficiently found by the celebrated result of Goemans and Williamson [26]. The second goal seeks to satisfy all constraints but in a weaker form. Here are a few examples of such problems. Firstly, the approximate graph colouring problem, studied by Garey and Johnson in the 1970s [25]: given a k-colourable graph G, find an ℓ-colouring of G for some $\ell \geq k$. Secondly, finding a satisfying assignment to a k-SAT instance that is promised to admit an assignment satisfying at least $\lceil k/2 \rceil$ literals in each clause — a problem coined $(2+\varepsilon)$-SAT by Austrin, Guruswami and Håstad [4]. Finally, given a satisfiable instance of (monotone) 1-in-3-SAT, find a satisfying not-all-equal assignment [13]. The former, quantitative notion of approximation has led to many breakthroughs in the last three decades, including the Probabilistically Checkable Proof (PCP) theorem [2], [3], [20]. The latter, qualitative notion has been investigated systematically only very recently under the name of *Promise Constraint Satisfaction Problems* (PCSPs) [8], [13]. Unfortunately, traditional notions of sparsification, involving removing constraints, fail when applied to qualitative approximation.

To illustrate that, consider the following naive procedure for graph 2-colouring: given an n-vertex instance graph G, if G *is* bipartite then return a spanning forest of G; if G is *not* bipartite then return one of the odd cycles in G. This simple algorithm is essentially the best possible: it outputs an instance G' with at most n edges, whose set of 2-colourings is exactly the same as that of G. However, the above approach breaks down for approximate solutions: there are 3-colourings of G' that are not 3-colourings of G (see Figure 1a).

Therefore, we need a notion that allows us to turn approximate solutions of our simplified instance into the solutions of the original one. In the above example of 2-colouring, a desired outcome would be a sparse graph G' that is 2-colourable if and only if G is and, furthermore, any k-colouring of G' translates into a k-colouring of G. Luckily, it is easy to see how to do this here. Suppose there exist two vertices with a common neighbour in G, say $x - y - z$. Then, x and z must be coloured identically in *all* 2-colourings of G. Hence, we can identify x with z; that is, we replace x and z with a new vertex x', both in the vertex set of G and the edges of G.[3] Let G' be the result of applying the identification procedure iteratively for all such triples. Now, if G was originally 2-colourable, so is G'. (Indeed, there is a 1-to-1 correspondence between the 2-colourings of G and those of G'.) Furthermore, any k-colouring of G' can be easily extended to a k-colouring of G: we colour each vertex of G according to the colour of the vertex it was merged into in G' (see Figure 1b).

Observe that the key property of the procedure outlined above is that, since all we do is merge variables that are equal in all solutions, no constraints are deleted. There is nothing special about 2-colourings in this argument — an analogous method, which we call a *strong sparsification*, can be applied to other computational problems, including variants of graph and hypergraph colouring problems, cf. the full version of this paper [10]. Here, we focus on a particular generalisation of 2-colouring, namely (monotone) 1-in-3-SAT: given a set of variables $X = \{x_1, \ldots, x_n\}$, together with a set $C \subseteq X^3$ of clauses, assign values 0 and 1 to the variables so that for every clause $(x_i, x_j, x_k) \in C$ exactly one variable among x_i, x_j, x_k is set to 1, with the remaining two set to 0.[4]

Definition 1. A *strong sparsification algorithm for monotone 1-in-3-SAT* is an algorithm which, given an instance $\mathcal{X} = (X, C)$ of monotone 1-in-3-SAT, outputs an equivalence relation $\sim$ on X such that if $x_i \sim x_j$ then x_i and x_j have the same value in all solutions to $\mathcal{X}$. We define the instance $\mathcal{X}/{\sim} = (X/{\sim}, C/{\sim})$ of monotone 1-in-3-SAT as follows: $X/{\sim}$ is the set of the equivalence classes $[x_1]_\sim, \ldots, [x_n]_\sim$, and each clause $(x_i, x_j, x_k) \in C$ induces a clause $([x_i]_\sim, [x_j]_\sim, [x_k]_\sim)$ in $C/{\sim}$. The *performance* of the algorithm is given by the number of clauses in $\mathcal{X}/{\sim}$, as a function of $n = |X|$.

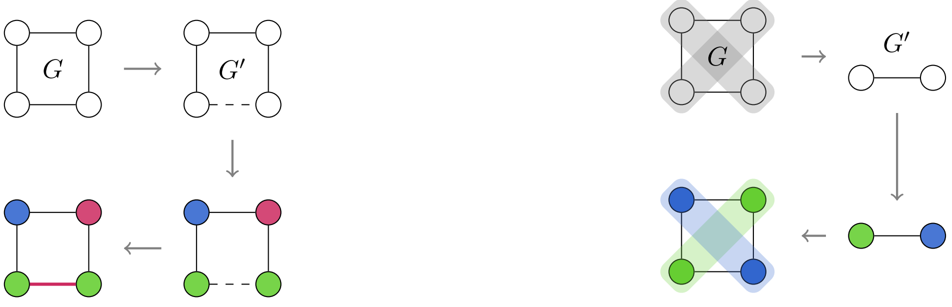
We emphasise that the notion of strong sparsification can be defined in an analogous way for other satisfiability problems. However, since a strong sparsification is, in particular, a sparsification in the sense of the aforementioned work of Dell and van Melkebeek [19], for some classic problems of this type (3-SAT in particular), it is unlikely to obtain any non-trivial results. We focus on 1-in-3-SAT, one of the first problems for which positive sparsification results were obtained [31].

The trivial strong sparsification for monotone 1-in-3-SAT (the one that does not merge any variables) has worst-case performance $O(n^3)$. There is a slightly cleverer approach that has performance $O(n^2)$ (noted e.g. in [22]). Suppose there exist clauses (x, y, z) and (x, y, t). There are only 3 possible assignments to (x, y, z, t): $(1, 0, 0, 0)$, $(0, 1, 0, 0)$ and $(0, 0, 1, 1)$. We see immediately that z and t are the same in all solutions and thus can be merged. After the exhaustive application of this rule, we get an instance in which, for every pair of variables x, y, there is at most one z such that $(x, y, z) \in C$, so the number of clauses is $O(n^2)$.

With this approach, the $O(n^2)$ bound is essentially tight: let $X = \{0, 1\}^d$, and $C = \{(i, j, k) \mid i, j, k \in X, i \oplus j \oplus k = 0\}$, where $\oplus$ denote the bitwise Boolean XOR operation. It is not

[3] Blum used the idea of merging two vertices that must be assigned the same colour in his paper on approximate graph colouring [12]. He calls this "Type 3 progress."

[4] A strong sparsification algorithm for monotone 1-in-3-SAT can be generically transformed into one for *non-monotone* 1-in-3-SAT (i.e. allowing negated literals), cf. Section IV. Thus, we shall focus on monotone 1-in-3-SAT.



(a) (b)

Fig. 1

difficult to see that, for any $i, j \in X$, there is exactly one variable k (namely $k = i \oplus j$) such that (i, j, k) is a clause. Hence, the above strong sparsification does nothing and outputs the original instance with $\Theta(n^2)$ clauses.

It is much harder to find a strong sparsification with better than quadratic performance, and in fact the existence of such an algorithm is not a priori clear. As our main contribution, we show that such an algorithm exists and thus improve the trivial quadratic upper bound.

Theorem 2 (Main). *There exists a polynomial-time strong sparsification algorithm for monotone 1-in-3-SAT with performance $O(n^{2-\varepsilon})$, for $\varepsilon \approx 0.0028$.*

A proof of Theorem 2 can be found in Section III. The main technical ingredient is the following theorem, which could be of independent interest. It is proved in Section II using tools from additive combinatorics.

Theorem 3. *Fix n, d. Consider $V = \{v_1, \dots, v_n\} \subseteq \mathbb{F}_2^d$ and let $N_1, \dots, N_n \subseteq V$ satisfy*

(i) for all $i \in [n]$, $v_i + N_i = N_i$, and

(ii) for all $i \in [n]$, $v_1, v_2 \dots, v_{i-1} \notin \langle N_i + N_i \rangle$.

Then $\sum_{i=1}^{n} |N_i| = O(n^{2-\varepsilon})$ for $\varepsilon \approx 0.0028$.

Note that the answer is polynomial in n since, for example, a linear lower bound is achieved by taking $N_i = \{0, v_i\}$. In fact, in the remark following Proposition 10, we provide an example where $\sum_i |N_i| = \Omega(n^{\log_2 3})$. Note also that, since $|N_i| \leq n$, there is a trivial bound $\sum_i |N_i| \leq n^2$. Hence, our contribution consists in improving the trivial bound by a polynomial saving of n^ε.

In the full version of this paper [10], we also show that no algorithm (even one with exponential runtime) can output a strong sparsifier for monotone 1-in-3-SAT with n variables with performance $o(n^{1.725\cdots})$. This is because there are instances with $\Omega(n^{1.725\cdots})$ constraints in which no merges of any two variables are possible.

Application: As an application of our result, we improve the state-of the-art approximation of hypergraph colourings. There are several different notions of colourings for hypergraphs, the classic one being nonmonochromatic colourings [21]. We shall focus on *linearly-ordered* (LO) colourings [7], also known as *unique-maximum* colourings [16]: the colours are taken from a linearly-ordered set, such as the integers, with the requirement that the maximum colour in each hyperedge is unique.

Notice that finding an LO 2-colouring of a 3-uniform hypergraph H is precisely the same problem as monotone 1-in-3-SAT (interpret the clauses of an instance as edges of H, and take the order $0 < 1$). Hence, our strong sparsification algorithm from Theorem 2 also applies to LO 2-colouring 3-uniform hypergraphs. Thus, we can improve the state-of-the-art algorithms for *approximate LO colouring*, where we are given an LO 2-colourable 3-uniform hypergraph H, and are asked to find an LO-colouring with as few colours as possible. The best known algorithm so far is the following, due to Håstad, Martinsson, Nakajima and Živný.

Theorem 4 ([29, Theorem 1]). *There is a polynomial-time algorithm that, given an n-vertex 3-uniform LO 2-colourable hypergraph H with $n \geq 4$,[5] returns an LO $(\log_2 n)$-colouring of H.*

Using our sparsification algorithm, we improve this to the following.

Corollary 5. *There is a polynomial-time algorithm that, given an n-vertex 3-uniform LO 2-colourable hypergraph H with $n \geq 5$, returns an LO $(0.999 \log_2 n)$-colouring of H.*

We will use the following result from [29]. Since its performance depends on the number of edges in the input, it benefits from our sparsification scheme.

Theorem 6 ([29, Theorem 3]). *There is a polynomial-time algorithm that, given a 3-uniform LO 2-colourable hypergraph H with m edges, returns an LO $(2 + \frac{1}{2}\log_2 m)$-colouring of H.*

Proof of Corollary 5. Recall that a hypergraph $H = (V, E)$ can be interpreted as an instance $\mathcal{X} = (V, E)$ of monotone 1-in-3-SAT, where V is the set of variables, the clauses of $\mathcal{X}$ are the edges of H, and any solution to $\mathcal{X}$ correspond to an LO 2-colouring of H and vice-versa. Thus, using Theorem 2, we compute an equivalence relation $\sim$ on V so that, if $u \sim v$, then u and v get the same colour in any LO 2-colouring; and

[5] Assumptions like this are to make sure that $\log_2 n \geq 2$. We will have similar assumptions for other algorithms.

$H' = H/\sim$ has $O(n^{2-\varepsilon})$ edges for $\varepsilon \approx 0.0028$. Since H is LO 2-colourable, H' is as well (and, in fact, the LO 2-colourings of H and H' are in a 1-to-1 correspondence).

Next, using Theorem 6, find an LO $(2+\frac{1}{2}\log_2(O(n^{2-\varepsilon})))$-colouring of H', i.e. an LO colouring with $O(1)+\frac{2-\varepsilon}{2}\log_2 n$ colours. Note that $(2-\varepsilon)/2 \approx 0.9986$, and so, for n larger than some constant, we have $O(1)+\frac{2-\varepsilon}{2}\log_2 n \le 0.999\log_2 n$; whereas, for n smaller than a constant, we can find an LO 2-colouring by brute force. Since for every $n \ge 5$ any LO 2-colouring is in particular an LO $(0.999\log_2 n)$-colouring, in all cases we have found an LO $(0.999\log_2 n)$-colouring of H'. Note that any LO colouring of $H' = H/\sim$ immediately gives rise to an LO colouring of H, by assigning each vertex of H the colour given to its equivalence class in $H/\sim$. □

Finally, we remark that we find the introduced notion of strong sparsification interesting and worth exploring for other satisfiability problems and CSPs. This could unveil a new and exciting line of work, going beyond the results of the present article. We start this exploration in the full version of this paper [10], where we obtain bounds for some CSPs, including monotone 1-in-k-SAT and, more generally, ℓ-in-k-SAT, Not-All-Equal-k-SAT, graph k-colouring, and systems of linear equations.

Paper structure: Section II gives a prof of the main technical result, Theorem 3. Section III gives a proof of our sparsification algorithm, Theorem 2. Section IV shows that monotone strong sparsification implies non-monotone strong sparsification.

Acknowledgements: We thank the anonymous reviewers of FOCS 2025 for their very useful and detailed feedback on an extended abstract of this work. We also thank Alexandru Pascadi for introducing the authors to each other.

II. Additive combinatorics

In this section we prove Theorem 3. We reformulate it into an equivalent, notationally more convenient statement as follows.

Theorem 7. *Let*

$$\mathscr{C}(n) := \max_{V,N_i}\left\{\sum_{i=1}^{n}|N_i| \;\middle|\; \begin{array}{l} V\subseteq\mathbb{F}_2^d,\ |V|=n, \textit{ and} \\ N_1,\dots,N_n\subseteq V \textit{ satisfy} \\ \textit{(i), (ii) of Theorem 3}\end{array}\right\}.$$

Then, $\mathscr{C}(n) = O(n^{2-\varepsilon})$ *for* $\varepsilon = 0.0028$.

Our proof of this bound employs two prominent results from the area of additive combinatorics, namely the Balog-Szemerédi-Gowers and Polynomial Freiman-Ruzsa Theorems. In particular, the value of ε that we obtain with this approach depends (essentially linearly) on the strongest known constants in these theorems. To state them, we need to introduce some standard concepts from additive combinatorics.

For a set $A \subseteq \mathbb{F}_2^d$ and an integer k, we define its *k-energy*

$$E_k(A) := \#\{(a_1,a_2,\dots,a_k)\in A^k \mid a_1+a_2+\dots+a_k = 0\}.$$

Note that if $r_A(x) := \#\{(a_1,a_2)\in A^2 \mid a_1+a_2 = x\}$, then we can equivalently write $E_4(A) = \sum_{x\in\mathbb{F}_2^d} r_A(x)^2$.[6] It may be helpful to keep in mind the trivial upper bound $|E_k(A)| \le |A|^{k-1}$ which holds since after choosing $a_1,\dots,a_{k-1}$, the element a_k, if exists, is fixed by the equation. We also define the *sumset* $A+A := \{a_1+a_2 \mid a_j \in A\}$, and the ratio $|A+A|/|A|$, known as the *doubling constant* of A. Again, there are the trivial bounds $|A| \le |A+A| \le |A|^2$. One should think of sets with "large energy" (say $E_4(A) \ge |A|^3/K$) and sets with "small doubling" (say $|A+A| \le K|A|$) as being highly additively structured. The latter notion is strictly stronger, and it is not hard to show that a set B with doubling constant L automatically satisfies $E_4(B) \ge |B|^3/L$.

The following Balog-Szemerédi-Gowers Theorem is a standard result in additive combinatorics [5], [27], and provides a partial converse. Roughly speaking, it states that a set with large additive energy contains a rather large subset with small doubling. We will use a recent version due to Reiher and Schoen with the current best dependence on K.

Theorem 8 (Balog-Szemerédi-Gowers Theorem [39])**.** *Let $K \ge 1$ and let B have additive energy $E_4(B) \ge |B|^3/K$. Then there exists a subset $B' \subseteq B$ of size $|B'| \ge |B|/(2K^{1/2})$ with doubling $|B'+B'| = O(K^4|B'|)$.*

Another very recently proved celebrated theorem, known as the Polynomial Freiman-Ruzsa Conjecture or Marton's Conjecture, describes the structure of sets B with small doubling in $\mathbb{F}_2^d$, stating that they must essentially be contained in a small number of translates of a subgroup.

Theorem 9 (Polynomial Freiman-Ruzsa Theorem [28])**.** *Let $K \ge 1$ and let $B \subseteq \mathbb{F}_2^d$ be a set with doubling $|B+B| \le K|B|$. Then there exists a subspace $G \le \mathbb{F}_2^d$ such that $|G| \le |B|$ and B is contained in at most $2K^9$ translates of G.*

We begin with a proposition proving Theorem 7 in the setting where $V = \mathbb{F}_2^d$ is itself a vector space. The argument in this special case relies on the so-called polynomial method, and does not require the two theorems above. The power of the Balog-Szemerédi-Gowers and Polynomial Freiman-Ruzsa Theorems will be required to deal with general sets V, essentially by reducing the general problem in Theorem 7 to this special setting in the following proposition (or more accurately Corollary 11).

Proposition 10. *Let $V = \{v_1,v_2,\dots,v_n\} = \mathbb{F}_2^d$ and $N_1,\dots,N_n \subseteq V$ satisfy conditions (i) and (ii) from Theorem 3. Then $\sum_{i=1}^n |N_i| = O(n^{\log_2 3})$, where we note that* $\log_2 3 \approx 1.585$.

Interestingly, this bound is optimal as the following example shows. Let $V = \mathbb{F}_2^d$; by abuse of notation we let $S \subseteq [d]$ denote the indicator vector for the set $S \subseteq [d]$. Thus $V = \{S \mid S \subseteq [d]\}$; also we have implicitly defined addition on sets to be the symmetric difference. We order V in decreasing

[6] We remark that $E_4(A) = \#\{(a_1,a_2,a_3,a_4)\in A^4 \mid a_1+a_2 = a_3+a_4\}$ is commonly known as the *additive energy* of V in the additive combinatorics community.

order of size i.e. S comes before T if $|S| > |T|$. Next, define $N_S :- \{T \in V \mid T \subseteq S\}$. Note that (i) is satisfied: $S + N_S = N_S$, since if $T \subseteq S$ then $S + T = (S \setminus T) \subseteq S$. One can also check that (ii) is satisfied because if $T \in N_S$ for $S \neq T$, then $S \supset T$, hence $|S| > |T|$ which implies that T comes after S in our ordering. Finally, $|N_S| = 2^{|S|}$ (it contains one vector for each subset of S) and we calculate $\sum_{S \subseteq [d]} |N_S| = \sum_{i=0}^{n} \binom{d}{i} 2^i = 3^d = (2^d)^{\log_2 3}$.

Before giving the proof, we note that Proposition 10 implies a power saving bound whenever V is contained in subspace H whose size is not much larger than that of V itself.

Corollary 11. *Let*

$$V = \{v_1, v_2, \dots, v_n\} \subseteq \mathbb{F}_2^d$$

and $N_1, \dots, N_n \subseteq V$ satisfy conditions (i) and (ii) from Theorem 3. If $V \subseteq H$ is contained in a subspace H, then $\sum_{i=1}^{n} |N_i| = O(|H|^{\log_2 3})$.

Proof of Corollary 11. Define

$$V' = \{v_1, \dots, v_n, v_{n+1}, \dots, v_{|H|}\}$$

where $v_{n+1}, \dots, v_{|H|}$ is an arbitrary ordering of the elements of $H \setminus V$. Let $N_i' = N_i$ if $i \in [n]$ and $N_i' = \varnothing$ if $i > n$. By Proposition 10, the claimed bound clearly follows. □

Proof of Proposition 10. Let the sets $V = \mathbb{F}_2^d$ and N_i be given. Since conditions (i) and (ii) (as well as the sizes $|N_i|$) are preserved under translating N_i (i.e. replacing N_i by $N_i + x$ for some $x \in \mathbb{F}_2^d$), we may assume without loss of generality that $0 \in N_i$ for each i. Hence, $\langle N_i + N_i \rangle = \langle N_i \rangle -: H_i$ for subspaces $H_i \leq \mathbb{F}_2^d$. We may further assume without loss of generality that for each i we have $N_i = H_i$. Indeed, condition (ii) remains unaffected, while $N_i + v_i = N_i$ implies that $\langle N_i \rangle + v_i = \langle N_i \rangle$ (and replacing N_i by $\langle N_i \rangle$ can also only increase the sizes $|N_i|$).

Thus, it is enough to show that, if $v_1, v_2, \dots, v_n$ is an ordering of $\mathbb{F}_2^d$ and H_i are subspaces such that

(i) $v_i \in H_i$, and
(ii) $v_1, v_2, \dots, v_{i-1} \notin H_i$,

then $\sum_i |H_i| = O(n^{\log_2 3})$. Let us write $h_i = \operatorname{codim} H_i = d - \dim H_i$, so that for each i we may find h_i many vectors $u_1^{(i)}, u_2^{(i)}, \dots, u_{h_i}^{(i)} \in \mathbb{F}_2^d$ for which

$$H_i = \{x \in \mathbb{F}_2^d \mid x \cdot u_r^{(i)} = 0 \text{ for all } r = 1, 2, \dots, h_i\}. \quad (1)$$

We emphasise that for $x, y \in \mathbb{F}_2^d$ we write $x \cdot y = \sum_{j=1}^{d} x_j y_j \in \mathbb{F}_2$.

Consider the following polynomial in the variable $X = (X_1, X_2, \dots, X_d)$ for each $i \in [n]$:

$$F_i(X) = F_i(X_1, X_2, \dots, X_d) :- \prod_{r=1}^{h_i} \left(X \cdot u_r^{(i)} - 1 \right), \quad (2)$$

which is simply a product of h_i many linear polynomials. Since we will only ever evaluate this polynomial for $X \in \mathbb{F}_2^d$, we may employ a *multilinearisation* trick which replaces each occurrence of a power X_s^t by X_s, for $t = 1, 2, 3, \dots$ and $s \in [n]$. One should note that this does not affect evaluations of $F_i(X)$ for $X \in \mathbb{F}_2^d$, since $Y^t = Y$ for $Y \in \mathbb{F}_2$. Multilinearizing each $F_i(X)$, we obtain the polynomials $\tilde{F}_i$ which are linear combinations of monomials in d variables, having degree 0 or 1 in each variable:

$$\tilde{F}_i(X) \in \langle 1, X_1, \dots, X_d, X_1X_2, \dots, X_1X_2 \dots X_d \rangle.$$

By construction, $F_i(X) = \tilde{F}_i(X)$ for each $X \in \mathbb{F}_2^d$. The crucial property of these polynomials is the following.

Lemma 12. *We have that* $\tilde{F}_j(v_i) = F_j(v_i) = \begin{cases} 0 & \textit{if } i < j, \\ 1 & \textit{if } i = j. \end{cases}$

Proof of Lemma 12. Note that if $i < j$, then by condition (ii) we have $v_i \notin H_j$ and hence from (1) there exists $r \in \{1, 2, \dots, h_j\}$ such that $v_i \cdot u_r^{(j)} = 1$. Clearly (2) then shows that $F_j(v_i) = 0$. Now, if $i = j$, then by condition (i) we have $v_i = v_j \in H_i$ which by (1) means precisely that $v_i \cdot u_r^{(i)} = 0$ for all $r \in \{1, 2, \dots, h_i\}$. Hence, $F_j(v_i) = 1$. *(End of proof of Lemma 12)* ■

Lemma 12 easily implies the following.

Lemma 13. *The polynomials $\tilde{F}_1, \tilde{F}_2, \dots, \tilde{F}_n$ are linearly independent in the polynomial vector space given by*

$$\langle 1, X_1, \dots, X_k, X_1X_2, \dots, X_1X_2 \dots X_d \rangle.$$

Proof of Lemma 13. Suppose not. Then there is a dependence relation

$$\tilde{F}_{j_1} + \tilde{F}_{j_2} + \dots + \tilde{F}_{j_m} = 0$$

for some $1 \leq j_1 < \dots < j_m \leq n$. But then, evaluating this polynomial at $X = v_{j_1} \in \mathbb{F}_2^d$ would give a contradiction by Lemma 12: $0 = \tilde{F}_{j_1}(v_{j_1}) + \tilde{F}_{j_2}(v_{j_1}) + \dots + \tilde{F}_{j_m}(v_{j_1}) = 1 + 0 + \dots + 0 = 1$. *(End of proof of Lemma 13)* ■

To use this information to bound $\sum_{i=1}^{n} |N_i| = \sum_i |H_i|$, we find a bound on the sizes of the *level sets* $V_\alpha :- \{i \in [n] \mid |H_i| \geq \alpha n = \alpha 2^d\}$ for each $\alpha \in [0, 1]$. As each H_i is a subspace, it suffices to bound $|V_\alpha|$ when $\alpha = 2^{-b}$ for some $b \in [d]$. Now note that if $i \in V_{2^{-b}}$, then $|H_i| \geq 2^{d-b}$ so that $h_i = \operatorname{codim} H_i \leq b$. Hence, from the definition (2) we see that the collection of polynomials

$$\{\tilde{F}_i(X) \mid i \in V_{2^{-b}}\} \subseteq \left\langle \prod_{i=1}^{d} X_i^{m_i} \;\middle|\; m_i \in \{0, 1\} \text{ and } \sum_{i=1}^{d} m_i \leq b \right\rangle -: P_b$$

is a set of $|V_{2^{-b}}|$ many linearly independent polynomials which are all contained in the subspace

$$P_b \leq \langle 1, X_1, \dots, X_1X_2, \dots, X_1 \dots X_d \rangle$$

of polynomials of total degree at most b. This implies

$$|V_{2^{-b}}| \leq \dim P_b = \sum_{r=0}^{b} \binom{d}{r}.$$

Hence, in total we can bound

$$\sum_{i=1}^{n} |N_i| = \sum_{i=1}^{n} |H_i| \le \sum_{b=0}^{d} 2^{d-b} |V_{2^{-b}}| \le \sum_{b=0}^{d} 2^{d-b} \sum_{r=0}^{b} \binom{d}{r}$$

$$= \sum_{r=0}^{d} \binom{d}{r} (1 + 2 + \cdots + 2^{d-r}) \le 2 \sum_{r=0}^{d} \binom{d}{r} n/2^r.$$

By the binomial formula and as $n = 2^d$, the final bound gives $\sum_i |N_i| = O(n(3/2)^{\log_2 n})$. Observing that $n(3/2)^{\log_2 n} = n^{\log_2 3}$ concludes the proof of Proposition 10. □

We proceed to the proof of the general case.

Proof of Theorem 7. We will prove the bound $\mathscr{C}(n) \le c_0 n^{2-\varepsilon}$ for all $n \in \mathbb{N}$, for some constant $c_0 > 0$. At the end we will find some necessary lower bounds on c_0. In particular, we will proceed by induction on n, assuming that the bound $\mathscr{C}(m) \le c_0 m^{2-\varepsilon}$ holds for all $m < n$.

Suppose that the set $V = \{v_1, v_2, \ldots, v_n\} \subseteq \mathbb{F}_2^d$ and sets $N_1, \ldots, N_n \subseteq V$ satisfy conditions (i), (ii) and are such that $\sum_{i=1}^n |N_i| = \mathscr{C}(n)$. We may assume that

$$\sum_{i=1}^{n} |N_i| \ge n^{2-\varepsilon}, \tag{3}$$

as otherwise we are done. The first step in this proof consists in showing, using tools from additive combinatorics, that under the assumption (3), a large subset of V must be rather densely contained in a subspace of $\mathbb{F}_2^d$. We show that (3) implies that $E_3(V)$, and hence $E_4(V)$, are large.

Lemma 14. *Let V and the sets $N_i \subseteq V$ satisfy* (3)*. Then $E_3(V) \ge \sum_{i=1}^n |N_i| \ge n^{2-\varepsilon}$. Moreover, $E_4(V) \ge n^{3-2\varepsilon}$*

Proof of Lemma 14. The bound for $E_3(V)$ is trivial from condition (i), since whenever $j, k \in [n]$ are such that $v_k \in N_j$, then $v_j + v_k \in N_j \subseteq V$ so that $(v_j, v_k, v_j + v_k)$ is a tuple that contributes to $E_3(V)$.

Note that $\sum_{v \in V} r_V(v) = E_3(V) \ge n^{2-\varepsilon}$. Also, we observed above that $\sum_x r_V(x)^2 = E_4(V)$. By Cauchy-Schwarz, we then get

$$E_4(V) = \sum_{x \in \mathbb{F}_2^d} r_V(x)^2 \ge \sum_{v \in V} r_V(v)^2 \ge$$

$$\frac{1}{|V|} \left(\sum_{v \in V} r_V(v) \right)^2 = \frac{E_3(V)^2}{n} \ge n^{3-2\varepsilon}.$$

(End of proof of Lemma 14) ■

We may now combine Lemma 14 with Theorem 8 and Theorem 9. By Theorem 8 (Balog-Szemerédi-Gowers) and as $E_4(V) \ge n^{3-2\varepsilon} = n^3/K$ for $K = n^{2\varepsilon}$, there exists a subset $A \subseteq V$ of size $|A| \ge n^{1-\varepsilon}/2$ with $|A - A| = O(n^{8\varepsilon}|A|)$. Now, by Theorem 9 (Polynomial-Freiman-Ruzsa), we may find a subspace $H \le \mathbb{F}_2^d$ such that A is covered by $O(n^{72\varepsilon})$ translates of H, and where $|H| \le |A|$. In particular, there is one such translate $x_0 + H$ such that

$$|V \cap (x_0 + H)| \ge |A|/O(n^{72\varepsilon}) \ge \Omega(n^{1-73\varepsilon}).$$

Also, without loss of generality we may take $x_0 = 0$, as otherwise we may replace H by $\langle H, x_0 \rangle$, which still satisfies the two properties above up to an additional factor of 2, namely:

- $|H| \le 2|A|$,
- $|V \cap H| = \Omega(|A|/n^{72\varepsilon}) = \Omega(n^{1-73\varepsilon})$.

Thus we have completed the first step of the proof. To use this information for estimating $\sum_i |N_i|$, we split $N_i = N_i^H \cup N_i^C$ for each $i \in [n]$, where

$$N_i^H = N_i \cap (H \cup (H + v_i)) = N_i \cap \langle H, v_i \rangle$$

and $N_i^C = N_i \setminus N_i^H$. It is notationally convenient to also define $N_v = N_i$ if $v = v_i \in V$, and similarly for N_v^H, N_v^C. We can calculate

$$\sum_{i=1}^{n} |N_i| = \sum_{v \in V \cap H} |N_v^H| + \sum_{v \in V \cap H} |N_v^C| + \sum_{v \in V \setminus H} |N_v^H| + \sum_{v \in V \setminus H} |N_v^C|. \tag{4}$$

The reason for the definitions of the sets N_i^H, N_i^C will become clear shortly: essentially, the idea is that, because $V \cap H$ is rather dense in the subspace H by the first step, one may expect to obtain good bounds for the first three terms by applying Corollary 11. The final term may be bounded using the induction hypothesis, since it will be clear from our choice that the sets N_v^C still satisfy conditions (i), (ii). The second step therefore consists in making this approach precise and bounding each of the four terms above.

1) First, we immediately deduce from Corollary 11 that

$$\sum_{v \in V \cap H} |N_v^H| = O(|H|^{\log_2 3}),$$

since the sets $V' :- V \cap H$ (ordered in the same way as in V) and $N_v' :- N_v^H$ for $v \in V \cap H$ still satisfy conditions (i) and (ii). Only that $N_v^H + v = N_v^H$ for $v \in V \cap H$ is perhaps non-trivial, but this is satisfied since $N_v + v = N_v$ holds for the original sets N_v and, as $v \in H$, we may take the intersection of both sides with H.

2) The final term may be bounded by $\sum_{v \in V \setminus H} |N_v^C| \le \mathscr{C}(|V \setminus H|)$, since the set $\tilde{V} :- V \setminus H$ with $\tilde{N}_v :- N_v^C \subseteq V \setminus H$ for $v \in \tilde{V}$ is again a system satisfying conditions (i), (ii). Indeed, (ii) is straightforward as $\tilde{N}_v \subseteq N_v$. Moreover, (i) holds: if $v \in V \setminus H$, then, as $v + N_v = N_v$, the set N_v consists of pairs $x, x + v$. Recall that, by definition, $N_v^C = N_v \setminus (N_v \cap \langle H, v \rangle)$, thus we have indeed also only removed elements in pairs (i.e. $x \in N_v \cap \langle H, v \rangle$ if and only if $x + v \in N_v \cap \langle H, v \rangle$). Therefore, as we showed above that $|V \cap H| \ge \Omega(n^{1-73\varepsilon})$, and as $\mathscr{C}(n)$ is clearly increasing in n, we can bound

$$\sum_{v \in V \setminus H} |N_v^C| \le \mathscr{C}(n - \Omega(n^{1-73\varepsilon})).$$

3) To bound the middle sums in (4), we will use the following lemma, whose proof we postpone to the end of the section.

Lemma 15. *We have that*

(a) $\displaystyle\sum_{v\in V\setminus H} |N_v^H| = O(n|H|^{\frac{1}{2}\log_2 3})$,

(b) $\displaystyle\sum_{v\in V\cap H} |N_v^C| = O(n|H|^{\frac{1}{2}\log_2 3})$.

It remains to show how the three bounds above may be combined to complete the proof of Theorem 7 (and hence Theorem 3). Using these bounds in (4), we get

$$\sum_{i=1}^{n} |N_i| = O(|H|^{\log_2 3} + n|H|^{\frac{1}{2}\log_2 3}) + \mathscr{C}(n - \Omega(n^{1-73\varepsilon})).$$

Let $c_1 > 0$ be a constant that can be used so that $\Omega(n^{1-73\varepsilon}) \geq c_1 n^{1-73\varepsilon}$ in the equation above. Recall that $|H| \leq 2|A| \leq 2n$ and that we assumed that $\mathscr{C}(n) = \sum_{i=1}^{n} |N_i|$. Thus, we conclude that

$$\begin{aligned}\mathscr{C}(n) &= \mathscr{C}(n(1 - c_1 n^{-73\varepsilon})) + O(n^{1+\frac{1}{2}\log_2 3}) \\ &\leq \mathscr{C}(n(1 - c_1 n^{-73\varepsilon})) + c_2 n^{1+\frac{1}{2}\log_2 3},\end{aligned}$$

for some constant $c_2 > 0$. By the induction hypothesis, we may bound

$$\begin{aligned}\mathscr{C}(n(1 - c_1 n^{-73\varepsilon})) &\leq c_0 n^{2-\varepsilon}(1 - c_1 n^{-73\varepsilon})^{2-\varepsilon} \\ &\leq c_0(n^{2-\varepsilon} - c_1 n^{2-74\varepsilon}),\end{aligned}$$

since certainly $(1 - c_1 n^{-73\varepsilon})^{2-\varepsilon} \leq 1 - c_1 n^{-73\varepsilon}$. We deduce that

$$\mathscr{C}(n) \leq c_0 n^{2-\varepsilon} - c_0 c_1 n^{2-74\varepsilon} + c_2 n^{1+\frac{1}{2}\log_2 3}.$$

This implies the desired bound $\mathscr{C}(n) \leq c_0 n^{2-\varepsilon}$ so long as $c_0 c_1 n^{2-74\varepsilon} \geq c_2 n^{1+\frac{1}{2}\log_2 \varepsilon}$. If we choose $\varepsilon > 0$ such that $2 - 74\varepsilon > 1 + \frac{1}{2}\log_2 3$, then we can choose c_0 large enough (in terms of the absolute constants c_1, c_2) so that the required bound holds for all $n \in \mathbb{N}$. Hence, taking ε with this property suffices — and any value of ε less than $\frac{1-\frac{1}{2}\log_2 3}{74} \approx 0.002804$ works. This concludes the proof of Theorem 7. □

Our final task is then to prove Lemma 15.

Proof of Lemma 15. (a) Recall that we want to bound $\sum_{v\in V\setminus H} |N_v^H|$ where $N_v^H = N_v \cap \langle H, v\rangle$. We begin with an estimate for the contributions of the sets $(V \setminus H)_a :- \{v \in V \setminus H \mid |N_v^H| \geq a\}$ for each $a \in [n]$. If $v \in V \setminus H$, then $N_v^H \subseteq N_v \subseteq V$, and, as $N_v^H = N_v \cap (H \cup (v + H))$ satisfies $N_v^H = v + N_v^H$, by condition (i), it follows that exactly half of the elements of $N_v^H = v + N_v^H$ lie in H and the other half lie in $v + H$. This means that if $v \in (V \setminus H)_a$, then the non-trivial coset $v + H$ intersects V in at least $|N_v^H|/2 \geq a/2$ elements.

Let $y_1 + H, y_2 + H, \ldots, y_\ell + H$ be all the distinct non-trivial cosets of H which each contain at least $a/2$ elements of V. We note two things. First, observe that $(V \setminus H)_a \subseteq \bigcup_{j=1}^{\ell}(y_j + H)$. Indeed, if $v \in (V \setminus H)_a$, then $v \in v + H$, and we proved above that $v + H$ contains at least $a/2$ elements of V. Second, since distinct cosets are disjoint, it is clear that $\ell \leq 2n/a$ as $|V| \leq n$. We claim that if we fix one such large coset, call it $y + H$, then $\sum_{v\in y+H} |N_v^H| = O(|H|^{\log_2 3})$. By the two observations above this gives the bound $\sum_{v\in(V\setminus H)_a} |N_v^H| \leq \sum_{j=1}^{\ell}\sum_{v\in y_j+H} |N_v^H| = O(|H|^{\log_2 3}\ell) = O(|H|^{\log_2 3}n/a)$.

To see why the claim holds, simply note that we may apply Corollary 11 to the set $V' :- V \cap (H \cup (y + H)) = V \cap \langle H, y\rangle$, with the sets $N_v' :- N_v^H \subseteq V'$ for all $v \in V'$. It is easy as always to see that these satisfy conditions (i), (ii), and note also that $|\langle H, y\rangle| = 2|H|$, so that Corollary 11 gives $\sum_{v\in y+H} |N_v^H| = O(|\langle H, y\rangle|^{\log_2 3}) = O(|H|^{\log_2 3})$.

From the definition of $(V \setminus H)_a$ we also know that $\sum_{v\in V\setminus(V\setminus H)_a} |N_v^H| \leq a|V \setminus (V \setminus H)_a| \leq na$. Finally, we can bound in total

$$\begin{aligned}\sum_{v\in V\setminus H} |N_v^H| &\leq \sum_{v\in V\setminus(V\setminus H)_a} |N_v^H| + \sum_{v\in(V\setminus H)_a} |N_v^H| \\ &= O(na + |H|^{\log_2 3}n/a) \\ &= O(|H|^{\frac{1}{2}\log_2 3}n),\end{aligned}$$

if we choose $a = (|H|^{\log_2 3})^{1/2}$.

(b) We now find a good way to bound $\sum_{v\in V\cap H} |N_v^C|$, where we recall that $N_v^C = N_v \setminus (H \cup (v + H))$. Note in particular that $N_v^C \subseteq V \setminus H$. We again proceed by considering estimates for the contributions of the level sets

$$\begin{aligned}&(V \setminus H)^{(a)} :- \\ &\qquad \left\{ w \in V \setminus H \;\middle|\; \begin{array}{l} w \text{ appears in at least } a \text{ many} \\ \text{sets } N_v^C \text{ with } v \in V \cap H \end{array} \right\}.\end{aligned}$$

These are perhaps slightly more complicated than the level sets above; note that these level sets are not subsets of the set $V \cap H$ over which we are summing, but of $V \setminus H$. However, it is clear that for any a:

$$\begin{aligned}\sum_{v\in V\cap H} |N_v^C| &= \sum_{v\in V\cap H} |N_v^C \setminus (V \setminus H)^{(a)}| \\ &\quad + \sum_{v\in V\cap H} |N_v^C \cap (V \setminus H)^{(a)}| \\ &\leq na + \sum_{v\in V\cap H} |N_v^C \cap (V \setminus H)^{(a)}|,\end{aligned}$$

so it is sufficient to find, for each a, good bounds on $\sum_{v\in V\cap H} |N_v^C \cap (V \setminus H)^{(a)}|$.

Similarly as above, if we fix $a \in [n]$ then we claim that we may find cosets $y_1 + H, \ldots, y_\ell + H$ such that $(V \setminus H)^{(a)} \subseteq \bigcup_{j=1}^{\ell}(y_j + H)$ and $\ell \leq 2n/a$. Indeed, it is enough to take all the distinct cosets $y_j + H$ which each contain at least $a/2$ elements of V, and note first that there clearly are at most $2n/a$ such cosets as in the proof of the first bound. To see why $(V \setminus H)^{(a)} \subseteq \bigcup_{j=1}^{\ell}(y_j + H)$, pick a $w \in (V \setminus H)^{(a)}$ and recall that by definition, $w \in N_v^C \subseteq N_v$ for at least a many $v \in V \cap H$. By condition (i), this means that $w + v \in N_v \subseteq V$ for at least a many vectors $v \in H$, so that the coset $w + H$ contains at least a elements of V. This completes the proof of the claim.

Again, we apply Corollary 11 for each subspace $\langle H, y_i\rangle$ with the set $V' :- V \cap \langle H, y_i\rangle$ and $N_v' :- N_v \cap \langle H, y_i\rangle \subseteq V'$

for $v \in V \cap H$ (and to be fully rigorous we may take $N'_v = \varnothing$ for $v \in V \cap (y_i + H)$). We deduce that

$$\sum_{v \in V \cap H} |N_v^C \cap (y_i + H)| \leq \sum_{v \in V'} |N'_v| \\ = O(|\langle H, y_i \rangle|^{\log_2 3}) = O(|H|^{\log_2 3}),$$

since $N_v^C \cap (y_i + H) \subseteq N'_v$. Hence, summing over all $\ell \leq 2n/a$ cosets, we get

$$\sum_{v \in V \cap H} |N_v^C \cap (V \setminus H)^{(a)}| \\ \leq \sum_{j=1}^{\ell} \sum_{v \in V \cap H} |N_v^C \cap (y_j + H)| = O(|H|^{\log_2 3} n/a),$$

and, in total,

$$\sum_{v \in V \cap H} |N_v^C| \leq an + \sum_{v \in V \cap H} |N_v^C \cap (V \setminus H)^{(a)}| \\ \leq an + O(|H|^{\log_2 3} n/a) = O(|H|^{\frac{1}{2}\log_2 3} n),$$

if we choose $a = (|H|^{\log_2 3})^{1/2}$. That concludes the proof of Lemma 15. □

III. Sparsification algorithm

In this section, we prove Theorem 2, i.e. present an efficient algorithm for strong sparsification of monotone 1-in-3-SAT. For convenience, we consider the problem of monotone *2-in-3*-SAT which is obtained by swapping the roles of 0 and 1 in the definition of 1-in-3-SAT — it is clear that for our purposes these two problems are equivalent. Exploiting the ideas of [29], [37], we will make use of the *linear structure* of 2-in-3-SAT: a clause (x, y, z) of a monotone 2-in-3-SAT instance is satisfied if and only if $x + y + z = 2$, $x, y, z \in \{0, 1\}$.

Definition 16. Consider an instance $\mathcal{X} = (X, C)$ of monotone 2-in-3-SAT. We define a system of modulo 2 linear equations $A_{\mathcal{X}}$ as follows. The set of variables of $A_{\mathcal{X}}$ is X, and for every clause $(x, y, z) \in C$, $A_{\mathcal{X}}$ contains the linear equation $x + y + z \equiv 0 \bmod 2$.

Clearly $A_{\mathcal{X}}$ is a relaxation of $\mathcal{X}$ — every solution to $\mathcal{X}$ is also a solution to $A_{\mathcal{X}}$; in particular, if two variables are always equal in every solution to $A_{\mathcal{X}}$, then they are always equal in every solution to $\mathcal{X}$. We say that two distinct variables x and y are *twins* if $\hat{x} = \hat{y}$ for every solution $(\hat{v})_{v \in X}$ to $A_{\mathcal{X}}$ — and we say that $\mathcal{X}$ is *twin-free* if no such pair of variables exists. Note that it is easy to check in polynomial time whether x and y are twins — simply solve $A_{\mathcal{X}}$ with $(\hat{x}, \hat{y})$ set to $(0, 1)$ and $(1, 0)$.

Fix a twin-free instance $\mathcal{X} = (X, C)$ of monotone 2-in-3-SAT and consider the vector space $\mathbb{F}_2[X]$, i.e. the space of formal linear combinations of elements in X with coefficients in $\mathbb{F}_2$. Each equation $x + y + z \equiv 0 \bmod 2$ in $A_{\mathcal{X}}$ can be associated with an element $x + y + z$ of $\mathbb{F}_2[X]$. We let $\langle C \rangle$ denote the subspace generated by all of these equations.

Lemma 17. *For every solution $(\hat{v})_{v \in X}$ of $A_{\mathcal{X}}$, and for any $x_1 + \cdots + x_k \in \langle C \rangle$, we have $\hat{x}_1 + \cdots + \hat{x}_k \equiv 0 \bmod 2$.*

Proof. The term $x_1 + \cdots + x_k$ can be formed by summing together multiple equations from $A_{\mathcal{X}}$. Hence it must equal 0 in any solution to $A_{\mathcal{X}}$. □

Thus, whenever $\mathcal{X} = (X, C)$ is twin-free, for any distinct $x, y \in X$ we have $x + y \notin \langle C \rangle$, and hence x and y are different *as elements of* $\mathbb{F}_2[X]/\langle C \rangle$. Observing that $\mathbb{F}_2[X]/\langle C \rangle$ is just some finite-dimensional vector space of the form $\mathbb{F}_2^d$, we have the following.

Lemma 18. *Whenever $\mathcal{X} = (X, C)$ is twin-free, we can compute in polynomial time an integer d and an* injective *map $\alpha : X \to \mathbb{F}_2^d$ so that the following holds. For any $x_1, \ldots, x_k \in X$ with $\alpha(x_1) + \cdots + \alpha(x_k) = 0$, we have $\hat{x}_1 + \cdots + \hat{x}_k \equiv 0 \bmod 2$ in any solution $(\hat{v})_{v \in X}$ to $A_{\mathcal{X}}$. Furthermore, for every equation $x + y + z \equiv 0$ in $A_{\mathcal{X}}$, we have $\alpha(x) + \alpha(y) + \alpha(z) = 0$.*

Proof. Note that $X \to \mathbb{F}_2[X] \to \mathbb{F}_2[X]/\langle C \rangle \cong \mathbb{F}_2^d$. We output this composite as α. For every input in X it is straightforward to see what it should be mapped to in $\mathbb{F}_2[X]$ and further in $\mathbb{F}_2[X]/\langle C \rangle$. Moreover, from the previous discussion it follows that α satisfies the required conditions. Finally, the image in $\mathbb{F}_2^d$ can be computed simply by finding a basis for the quotient space, which can be done in polynomial time. □

Fix a twin-free instance $\mathcal{X} = (X, C)$, and let α be given by Lemma 18. Consider two variables $x, y \in X$. If there exists an even number of neighbours[7] of x, say $z_1, \ldots, z_{2k}$, so that $\alpha(z_1) + \cdots + \alpha(z_{2k}) = \alpha(y)$, then write $x \succeq y$. This suggestive notation has the following justification.

Lemma 19. *Suppose $\mathcal{X} = (X, C)$ is twin-free and $x \succeq y$. Then, in any solution $(\hat{v})_{v \in X}$ to $\mathcal{X}$, we have $\hat{x} \geq \hat{y}$.*

Proof. If $\hat{x} = 1$ then the claim holds, so suppose $\hat{x} = 0$. Since $(\hat{v})_{v \in X}$ is a solution to $\mathcal{X}$ (which, recall, is a 2-in-3-SAT instance), it follows that for all neighbours z of x we have $\hat{z} = 1$. Now, by assumption we have that $\alpha(z_1) + \cdots + \alpha(z_{2k}) = \alpha(y)$ i.e. $\alpha(z_1) + \cdots + \alpha(z_{2k}) + \alpha(y) = 0$. Hence

$$\hat{y} \equiv \hat{y} + 2k \equiv \hat{y} + \hat{z}_1 + \cdots + \hat{z}_{2k} \equiv 0 \mod 2.$$

Thus $\hat{y} = 0$ and $\hat{x} \geq \hat{y}$ as required. □

We can check whether $x \succeq y$ in polynomial time, as we will now describe. Let $z_1, \ldots, z_t$ be the neighbours of x. For $j \in [d]$ let α^j be defined so that $\alpha(x) = (\alpha^1(x), \ldots, \alpha^d(x))$. We check whether there exist $b_1, \ldots, b_t \in \mathbb{F}_2$ so that $\sum_{i=1}^t b_i = 0$ and, for all $j \in [d]$, we have $\sum_{i=1}^t b_i \alpha^j(z_i) = \alpha^j(y)$. If they exist, let Z be the set of these z_i's for which $b_i = 1$. Clearly, the first sum guarantees that Z has an even number of elements, while the others give that $\sum_{z \in Z} \alpha(z) = \alpha(y)$. Therefore, $x \succeq y$ if and only if $b_1, \ldots, b_t$ exist, and we can decide that by solving a system of linear equations.

Call a sequence of variables $x_1, \ldots, x_k$ a *cycle* if $x_1 \succeq \cdots \succeq x_k \succeq x_1$. If $\mathcal{X}$ does not admit any cycles, call it *cycle-free*. With these definitions in place, we can finally apply Theorem 3 in the following theorem.

[7] We say two variables are neighbours if they belong to the same clause.

Theorem 20. *Suppose $\mathcal{X} = (X, C)$ is an n-variable, m-clause instance of monotone 2-in-3-SAT that is twin-free and cycle-free. Then $m = O(n^{2-\varepsilon})$, for the same ε as in Theorem 3.*

Proof. Let α, d be given by Lemma 18. Consider the relation $\succeq$. As $\succeq$ is assumed to be acyclic, there exists a topological sort of V with respect to $\succeq$. In other words, we order $X = \{x_1, \ldots, x_n\}$ in such a way that for $j \leq i$ we have $x_i \not\succeq x_j$.

With this in hand, we apply Theorem 3 to

$$V = \{\alpha(x_1), \ldots, \alpha(x_n)\}$$
$$N_i = \{\alpha(y) \mid y \text{ is a neighbour of } x_i\}.$$

Let us first check that the properties of Theorem 3 are satisfied.

(i) Consider any element $\alpha(y) \in N_i$. There exists z so that (x_i, y, z) is a clause of $\mathcal{X}$. The properties of α guarantee that $\alpha(x_i) + \alpha(y) + \alpha(z) = 0$, hence $\alpha(y) + \alpha(x_i) = \alpha(z) \in N_i$, as z is also a neighbour of x_i.

(ii) Consider any $j \leq i$. We have $x_i \not\succeq x_j$ by our choice of ordering of $x_1, \ldots, x_n$. Hence, there is no collection of an even number of neighbours $z_1, \ldots, z_{2k}$ of x_i so that $\alpha(x_j) = \sum_{\ell=1}^{2k} \alpha(z_\ell)$. The set of possible sums on the right ranges over $\langle N_i + N_i \rangle$, thus we obtain that $\alpha(x_j) \notin \langle N_i + N_i \rangle$.

Hence, we can apply Theorem 3, and thus we conclude that $\sum_{i=1}^{n} \#\{\alpha(y) \mid y \text{ is a neighbour of } x_i\} = O(n^{2-\varepsilon})$ for $\varepsilon \approx 0.0028$. Since α is injective, this is the same as saying that the total number of pairs of variables (x, y) that are neighbours is at most $O(n^{2-\varepsilon})$.

Observe that, for any four distinct variables $x, y, z, t \in X$, it is impossible that there is a clause on variables x, y, and z and another clause on variables x, y, and t, as then z and t would be twins. In other words, each pair of variables that are in a clause together are in *exactly* one clause together. Thus, the number of clauses is at most the number of such pairs — whence the conclusion. □

We are now ready to prove Theorem 2.

Proof of Theorem 2. Suppose we are given an instance $\mathcal{X}$ of monotone 2-in-3-SAT. We will construct $\sim$ by repeatedly merging pairs of variables that are the same in all solutions to $\mathcal{X}$. The relation $\sim$ will then be the transitive, reflexive closure of all the merges.

Let us now describe what variables we merge. While there are any twins x and y, merge them. If there are no twins, but there exists a cycle $x_1 \succeq \ldots \succeq x_k \succeq x_1$, then merge all the variables $x_1, \ldots, x_k$. Since variables in a twin-pair have the same value in all solutions to $A_{\mathcal{X}}$, they must have the same value in all solutions to $\mathcal{X}$ — and the latter is true for all variables in a cycle as well, due to Lemma 19. Detecting twins can be done in polynomial time; furthermore, computing $\preceq$ and then finding cycles in it can also be done in polynomial time. Suppose we started with an n-variable instance. We get, at the end, a twin-free, cycle-free instance on at most n variables. By Theorem 20 this instance has $O(n^{2-\varepsilon})$ clauses, as desired. □

IV. Monotone strong sparsification implies non-monotone

Theorem 21. *Suppose that there is a polynomial-time strong sparsification algorithm $\mathscr{A}$ for monotone 1-in-3-SAT with performance $f(n)$. Then there is a polynomial-time strong sparsification algorithm for* non-monotone *1-in-3-SAT with performance at most $8f(2n)$.*

Proof. Suppose we are given an instance $\mathcal{X}$ of non-monotone 1-in-3-SAT with variables $X = \{x_1, \ldots, x_n\}$ and clauses C. Add variables $y_1, \ldots, y_n$, and create an instance $\mathcal{Y}$ with variable set $Y = \{x_1, y_1, \ldots, x_n, y_n\}$ and whose set C' of clauses is the same as in $\mathcal{X}$, but with the literal $\neg x_i$ replaced by the variable y_i in every clause. By construction $\mathcal{Y}$ is an instance of monotone 1-in-3-SAT with $2n$ variables; also every solution to $\mathcal{X}$ extends to a solution to $\mathcal{Y}$, by setting $y_i = \neg x_i$.

Thus, we can apply $\mathscr{A}$ to $\mathcal{Y}$ to create an equivalence relation $\sim$ on Y such that, in any solution to $\mathcal{Y}$, if $v \sim w$ then v and w are assigned the same value. Furthermore, by our assumption on $\mathscr{A}$, we have that the size of $C'/{\sim}$ is $f(2n)$. By construction, in every solution to $\mathcal{X}$ extended to $\mathcal{Y}$, different values are assigned to x_i and y_i. Define a graph $G = (Y, E)$, where $(v, w) \in E$ if there exists $i \in [n]$ such that $v \sim x_i$ and $w \sim y_i$. Note that if $(v, w) \in E$ then, in any solution to $\mathcal{X}$ extended to $\mathcal{Y}$, different values are assigned to v and w.

If the graph G is non-bipartite then $\mathcal{X}$ is unsatisfiable, and our strong sparsification algorithm can vacuously return any equivalence relation; hence, suppose G is bipartite, and denote by A and B its bipartition classes (if G is not connected, fix one choice for A and B). We write $x_i \sim_G x_j$ if x_i and x_j belong to the same connected component of G, and are both in A or both in B. If $x_i \sim_G x_j$, then there exists a path $x_i - z_1 - \ldots - z_{2k-1} - x_j$ in G, which implies that x_i and x_j are assigned the same value in each solution to $\mathcal{X}$. Since the above procedure can return $\sim_G$ in polynomial time, it is a strong sparsification algorithm for 1-in-3-SAT. It only remains to prove that this algorithm has the advertised performance; i.e. that $\mathcal{X}/{\sim_G}$ does not have too many clauses.

We claim that each element of $C'/{\sim}$ corresponds to at most 8 clauses from $C/{\sim_G}$. We will prove this by constructing a mapping that associates, with each clause $c \in C/{\sim_G}$, a non-empty subset $A(c) \subseteq C'/{\sim}$, in such a way that every clause from $C'/{\sim}$ belongs to at most 8 different sets $A(c)$. We illustrate the construction by example. Consider a clause $(P, \neg Q, R) \in C/{\sim_G}$ (and recall that P, Q, R are equivalence classes of $\sim_G$). Define

$$A(P, \neg Q, R) = \{([x_i]_\sim, [y_j]_\sim, [x_k]_\sim) \mid x_i \in P, y_j \in Q, x_k \in R\}.$$

Consider some clause (S, T, U) of $\mathcal{Y}/{\sim}$ (recall again that S, T, U are equivalence classes of $\sim$). For each of the 8 sign patterns in $\{+, -\}^3$ there is at most one clause $c = \mathcal{X}/{\sim_G}$ for which $(S, T, U) \in A(c)$ which follows that sign pattern. For example, for the sign pattern $(+, -, +)$, consider any $x_i \in S, y_j \in T, x_k \in U$, and note that there is at most one clause of form $c = ([x_i]_{\sim_G}, \neg[x_j]_{\sim_G}, [x_k]_{\sim_G})$ in $C/{\sim_G}$.

This is because, for any other $x_{i'} \in P, y_{j'} \in Q, x_{k'} \in R$, we have $x_i \sim x_{i'}, y_j \sim y_{j'}, x_k \sim x_{k'}$ and hence $x_i \sim_G x_{i'}, x_j \sim_G x_{j'}, x_k \sim_G x_{k'}$. Noting that only such a clause can have $(S, T, U) \in A(c)$ and follow the sign pattern $(+, -, +)$ completes the proof. □

References

[1] A. Andoni, J. Chen, R. Krauthgamer, B. Qin, D. P. Woodruff, and Q. Zhang, "On sketching quadratic forms," in *Proc. 7th ACM Conference on Innovations in Theoretical Computer Science (ITCS'16)*. ACM, 2016, pp. 311—319.

[2] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, "Proof verification and the hardness of approximation problems," *J. ACM*, vol. 45, no. 3, pp. 501–555, 1998.

[3] S. Arora and S. Safra, "Probabilistic checking of proofs: A new characterization of NP," *J. ACM*, vol. 45, no. 1, pp. 70–122, 1998.

[4] P. Austrin, V. Guruswami, and J. Håstad, "(2+ε)-Sat is NP-hard," *SIAM J. Comput.*, vol. 46, no. 5, pp. 1554–1573, 2017.

[5] A. Balog and E. Szemerédi, "A statistical theorem of set addition," *Comb.*, vol. 14, no. 3, pp. 263–268, 1994.

[6] N. Bansal, O. Svensson, and L. Trevisan, "New notions and constructions of sparsification for graphs and hypergraphs," *Proc. 60th IEEE Annual Symposium on Foundations of Computer Science (FOCS'19)*, pp. 910–928, 2019.

[7] L. Barto, D. Battistelli, and K. M. Berg, "Symmetric Promise Constraint Satisfaction Problems: Beyond the Boolean Case," in *Proc. 38th International Symposium on Theoretical Aspects of Computer Science (STACS'21)*, ser. LIPIcs, vol. 187. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, pp. 10:1–10:16.

[8] L. Barto, J. Bulín, A. A. Krokhin, and J. Opršal, "Algebraic approach to promise constraint satisfaction," *J. ACM*, vol. 68, no. 4, pp. 28:1–28:66, 2021.

[9] J. Batson, D. A. Spielman, and N. Srivastava, "Twice-Ramanujan sparsifiers," *SIAM Review*, vol. 56, no. 2, pp. 315–334, 2014.

[10] B. Bedert, T. Nakajima, K. Okrasa, and S. Živný, "Strong Sparsification for 1-in-3-SAT via Polynomial Freiman-Ruzsa," Tech. Rep., 2025, arXiv:2507.17878.

[11] A. A. Benczúr and D. R. Karger, "Approximating s-t Minimum Cuts in $\tilde{O}(n^2)$ Time," in *Proc. 28th Annual ACM Symposium on Theory of Computing (STOC'96)*. ACM, 1996, pp. 47–55.

[12] A. Blum, "New approximation algorithms for graph coloring," *J. ACM*, vol. 41, no. 3, pp. 470–516, 1994.

[13] J. Brakensiek and V. Guruswami, "Promise constraint satisfaction: Algebraic structure and a symmetric Boolean dichotomy," *SIAM J. Comput.*, vol. 50, no. 6, pp. 1663–1700, 2021.

[14] ——, "Redundancy is all you need," in *Proc. 57th Annual ACM Symposium on Theory of Computing (STOC'25)*. ACM, 2025.

[15] S. Butti and S. Živný, "Sparsification of binary CSPs," *SIAM J. Discret. Math.*, vol. 34, no. 1, pp. 825–842, 2020.

[16] P. Cheilaris, B. Keszegh, and D. Pálvölgyi, "Unique-maximum and conflict-free coloring for hypergraphs and tree graphs," *SIAM J. Discret. Math.*, vol. 27, no. 4, pp. 1775–1787, 2013.

[17] H. Chen, B. M. P. Jansen, and A. Pieterse, "Best-case and worst-case sparsifiability of Boolean CSPs," *Algorithmica*, vol. 82, no. 8, pp. 2200–2242, 2020.

[18] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, *Parameterized algorithms*. Springer, 2015.

[19] H. Dell and D. van Melkebeek, "Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses," *J. ACM*, vol. 61, no. 4, pp. 23:1–23:27, 2014.

[20] I. Dinur, "The PCP theorem by gap amplification," *J. ACM*, vol. 54, no. 3, p. 12, 2007.

[21] I. Dinur, O. Regev, and C. Smyth, "The Hardness of 3-Uniform Hypergraph Coloring," *Comb.*, vol. 25, no. 5, pp. 519–535, 2005.

[22] L. Drori and D. Peleg, "Faster exact solutions for some NP-hard problems," *Theor. Comput. Sci.*, vol. 287, no. 2, pp. 473–499, 2002.

[23] A. Filtser and R. Krauthgamer, "Sparsification of two-variable valued constraint satisfaction problems," *SIAM J. Discret. Math.*, vol. 31, no. 2, pp. 1263–1276, 2017.

[24] F. V. Fomin, D. Lokshtanov, S. Saurabh, and M. Zehavi, *Kernelization: theory of parameterized preprocessing*. Cambridge University Press, 2019.

[25] M. R. Garey and D. S. Johnson, "The complexity of near-optimal graph coloring," *J. ACM*, vol. 23, no. 1, pp. 43–49, 1976.

[26] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, no. 6, pp. 1115–1145, 1995.

[27] W. T. Gowers, "A new proof of Szemerédi's theorem for arithmetic progressions of length four," *Geom. Funct. Anal.*, vol. 8, no. 3, pp. 529–551, 1998.

[28] W. T. Gowers, B. Green, F. Manners, and T. Tao, "On a conjecture of Marton," *Ann. Math.*, vol. 201, no. 2, pp. 515–549, 2025.

[29] J. Håstad, B. Martinsson, T.-V. Nakajima, and S. Živný, "A Logarithmic Approximation of Linearly-Ordered Colourings," in *Proc. Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2024)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, pp. 7:1–7:6.

[30] R. Impagliazzo, R. Paturi, and F. Zane, "Which problems have strongly exponential complexity?" *J. Comput. Syst. Sci.*, vol. 63, no. 4, pp. 512–530, 2001.

[31] B. M. P. Jansen and A. Pieterse, "Optimal Sparsification for Some Binary CSPs Using Low-Degree Polynomials," *ACM Trans. Comput. Theory*, vol. 11, no. 4, pp. 28:1–28:26, 2019.

[32] S. Khanna, A. Putterman, and M. Sudan, "Near-optimal size linear sketches for hypergraph cut sparsifiers," in *Proc. 65th IEEE Annual Symposium on Foundations of Computer Science (FOCS'24)*. IEEE, 2024, pp. 1669–1706.

[33] S. Khanna, A. L. Putterman, and M. Sudan, "Code sparsification and its applications," in *Proc. 2024 ACM-SIAM Symposium on Discrete Algorithms (SODA'24)*. SIAM, 2024, pp. 5145–5168.

[34] ——, "Efficient algorithms and new characterizations for CSP sparsification," in *Proc. 57th Annual ACM Symposium on Theory of Computing (STOC'25)*. ACM, 2025, pp. 407–416.

[35] D. Kogan and R. Krauthgamer, "Sketching cuts in graphs and hypergraphs," in *Proc. 6th ACM Conference on Innovations in Theoretical Computer Science (ITCS'15)*. ACM, 2015, pp. 367–376.

[36] V. Lagerkvist and M. Wahlström, "Sparsification of SAT and CSP problems via tractable extensions," *ACM Trans. Comput. Theory*, vol. 12, no. 2, pp. 13:1–13:29, 2020.

[37] T.-V. Nakajima and S. Živný, "Linearly ordered colourings of hypergraphs," *ACM Trans. Comput. Theory*, vol. 13, no. 3–4, 2022.

[38] E. Pelleg and S. Živný, "Additive sparsification of CSPs," *ACM Trans. Algorithms*, vol. 20, no. 1, pp. 1:1–1:18, 2024.

[39] C. Reiher and T. Schoen, "Note on the Theorem of Balog, Szemerédi, and Gowers," *Comb.*, vol. 44, no. 3, pp. 691–698, 2024.