ETH-Hardness of Approximating 2-CSPs and Directed Steiner Network*

Irit Dinur[1] and Pasin Manurangsi[2]

1 **Weizmann Institute of Science, Rehovot, Israel**
irit.dinur@weizmann.ac.il

2 **University of California, Berkeley, USA**
pasin@berkeley.edu

Abstract

We study 2-ary constraint satisfaction problems (2-CSPs), which can be stated as follows: given a constraint graph $G = (V, E)$, an alphabet set Σ and, for each edge $\{u, v\} \in E$, a constraint $C_{uv} \subseteq \Sigma \times \Sigma$, the goal is to find an assignment $\sigma : V \to \Sigma$ that satisfies as many constraints as possible, where a constraint C_{uv} is said to be satisfied by σ if $(\sigma(u), \sigma(v)) \in C_{uv}$.

While the approximability of 2-CSPs is quite well understood when the alphabet size $|\Sigma|$ is constant (see e.g. [37]), many problems are still open when $|\Sigma|$ becomes super constant. One open problem that has received significant attention in the literature is whether it is hard to approximate 2-CSPs to within a polynomial factor of both $|\Sigma|$ and $|V|$ (i.e. $(|\Sigma||V|)^{\Omega(1)}$ factor). As a special case of the so-called Sliding Scale Conjecture, Bellare et al. [5] suggested that the answer to this question might be positive. Alas, despite many efforts by researchers to resolve this conjecture (e.g. [39, 4, 20, 21, 35]), it still remains open to this day.

In this work, we separate $|V|$ and $|\Sigma|$ and ask a closely related but weaker question: is it hard to approximate 2-CSPs to within a polynomial factor of $|V|$ (while $|\Sigma|$ may be super-polynomial in $|V|$)? Assuming the exponential time hypothesis (ETH), we answer this question positively: unless ETH fails, no polynomial time algorithm can approximate 2-CSPs to within a factor of $|V|^{1-1/\log^{\beta}|V|}$ for some $\beta > 0$. Note that our ratio is not only polynomial but also almost linear. This is almost optimal since a trivial algorithm yields an $O(|V|)$-approximation for 2-CSPs.

Thanks to a known reduction [25, 16] from 2-CSPs to the Directed Steiner Network (DSN) problem, our result implies an inapproximability result for the latter with polynomial ratio in terms of the number of demand pairs. Specifically, assuming ETH, no polynomial time algorithm can approximate DSN to within a factor of $k^{1/4-o(1)}$ where k is the number of demand pairs. The ratio is roughly the square root of the approximation ratios achieved by best known polynomial time algorithms [15, 26], which yield $O(k^{1/2+\varepsilon})$-approximation for every constant $\varepsilon > 0$.

Additionally, under Gap-ETH, our reduction for 2-CSPs not only rules out polynomial time algorithms, but also fixed parameter tractable (FPT) algorithms parameterized by the number of variables $|V|$. These are algorithms with running time $g(|V|) \cdot |\Sigma|^{O(1)}$ for some function g. Similar improvements apply for DSN parameterized by the number of demand pairs k.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Hardness of Approximation, Constraint Satisfaction Problems, Directed Steiner Network, Parameterized Complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.36

* This work was supported by BSF grant 2014371 and NSF grants CCF 1540685, CCF 1655215.

9th Innovations in Theoretical Computer Science Conference (ITCS 2018).
Editor: Anna R. Karlin; Article No. 36; pp. 36:1–36:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

We study 2-ary constraint satisfaction problems (2-CSPs): given a constraint graph $G = (V, E)$, an alphabet Σ and, for each edge $\{u, v\} \in E$, a constraint $C_{uv} \subseteq \Sigma \times \Sigma$, the goal is to find an assignment $\sigma : V \to \Sigma$ that satisfies as many constraints as possible, where a constraint C_{uv} is satisfied by σ if $(\sigma(u), \sigma(v)) \in C_{uv}$. Throughout the paper, we use k to denote the number of variables $|V|$, n to denote the alphabet size $|\Sigma|$, and N to denote nk.

Constraint satisfaction problems (CSPs) and their inapproximability have been studied extensively since the proof of the PCP theorem in the early 90's [3, 2]. Most of the effort has been directed towards understanding the approximability of CSPs with constant arity and constant alphabet size, leading to a reasonable if yet incomplete understanding of the landscape [27, 31, 37, 12]. When the alphabet size grows, the Sliding Scale Conjecture (SSC) [5] predicts that the hardness of approximation ratio will grow as well, and be at least a constant power of the alphabet size n. This has been confirmed for values of n up to $2^{(\log N)^{1-\delta}}$ (see [39, 4, 20]). Proving the same for n that is itself a constant power of N is the so-called polynomial SSC and is still open. Before we proceed, let us note that the results of [39, 4, 20] work only for arity larger than two and, hence, do not imply inapproximability for 2-CSPs. We will discuss the special case of 2-CSPs in more details below.

The polynomial SSC has been approached from different angles. In [21] the authors try to find the smallest arity and alphabet size such that the hardness factor is a constant power of n, and in [19] the conjecture is shown to follow (in some weaker sense) from the Gap-ETH hypothesis, which we discuss in more details later. In this work we focus on yet another angle, which is to separate n and k and ask whether it is hard to approximate constant arity CSPs to within a factor that is a constant power of k (but possibly not a constant power of n). Observe here that obtaining NP-hardness of $k^{\Omega(1)}$ factor is likely to be as hard as obtaining one with $N^{\Omega(1)}$; this is because CSPs can be solved exactly in time $n^{O(k)}$, which means that, unless $\mathsf{NP} \not\subseteq \bigcap_{\varepsilon>0} \mathsf{DTIME}(2^{n^{\varepsilon}})$, NP-hard instances of CSPs must have $k = \text{poly}(N)$.

This motivates us to look for hardness from assumptions stronger than $\mathsf{P} \neq \mathsf{NP}$. Specifically, our result will be based on the Exponential Time Hypothesis (ETH), which states that no subexponential time algorithm can solve 3-SAT (Conjecture 5). We show that, assuming ETH, no polynomial time algorithm can approximate 2-CSPs to within an almost linear ratio in k, as stated below. This is almost optimal since there is a straightforward $(k/2)$-approximation for 2-CSPs, by simply satisfying all constraints that touch a variable with highest degree.

▶ **Theorem 1** (Main Theorem). *Assuming ETH, there exists a constant $\beta > 0$ such that, no algorithm can, given a 2-CSP instance Γ with alphabet size n and k variables such that the constraint graph is complete, distinguish between the following two cases in polynomial time:*
- *(Completeness)* $\text{val}(\Gamma) = 1$*, and,*
- *(Soundness)* $\text{val}(\Gamma) < 2^{(\log k)^{1-\beta}}/k$*.*

Here $\text{val}(\Gamma)$ *denotes the maximum fraction of edges satisfied by any assignment.*

To paint a full picture of how our result stands in comparison to previous results, let us state what is known about the approximability of 2-CSPs; due to the vast literature regarding 2-CSPs, we will focus only the regime of large alphabets which is most relevant to our setting. In terms of NP-hardness, the best known inapproximability ratio is $(\log N)^c$ for every constant $c > 0$; this follows from Moshkovitz-Raz PCP [36] and the Parallel Repetition Theorem for the low value regime [24]. Assuming a slightly stronger assumption that NP is not contained in quasipolynomial time (i.e. $\mathsf{NP} \not\subseteq \bigcup_{c>0} \mathsf{DTIME}(n^{(\log n)^c})$), 2-CSP is hard to approximate to within a factor of $2^{(\log N)^{1-\delta}}$ for every constant $\delta > 0$; this can be proved by applying Raz's original Parallel Repetition Theorem [38] to the PCP Theorem. In [19], the author observed that running time for parallel repetition can be reduced by looking at unordered sets instead of ordered tuples. This observation implies that[1], assuming ETH, no polynomial time $N^{1/(\log\log\log N)^c}$-approximation algorithm exists for 2-CSPs for some constant $c > 0$. Moreover, under Gap-ETH (which will be stated shortly), it was shown that, for every sufficiently small $\varepsilon > 0$, any N^{ε}-approximation algorithm must run in time $N^{\Omega(\exp(1/\varepsilon))}$. Note that, while this latest result comes close to the polynomial sliding scale conjecture, it does not quite resolve the conjecture yet. In particular, even the weak form of the conjecture which postulates that there exists $\delta > 0$ for which no polynomial time algorithm can approximate 2-CSPs to within N^{δ} factor of the optimum does not follow from [19]. Nevertheless, the Gap-ETH-hardness of [19] does imply that, for any function $f = o(1)$, no polynomial time algorithm can approximate 2-CSPs to within a factor of $N^{f(N)}$.

In all results mentioned above, the constructions give 2-CSP instances in which the alphabet size n is smaller than the number of variables k. In other words, even if we aim for an inapproximability ratio in terms of k instead of N, we still get the same ratios as stated above. Thus, our result is the first hardness of approximation for 2-CSPs with $k^{\Omega(1)}$ factor. Note again that our result rules out any polynomial time algorithm and not just $N^{O(\exp(1/\varepsilon))}$-time algorithm ruled out by [19]. Moreover, our ratio is almost linear in k whereas the result of [19] only holds for ε that is sufficiently small depending on the parameters of Gap-ETH.

An interesting feature of our reduction is that it produces 2-CSP instances with the alphabet size n that is much larger than k. This is reminiscent of the setting of 2-CSPs parameterized by the number of variables k. In this setting, the algorithm's running time is allowed to depend not only polynomially on N but also on any function of k (i.e. $g(k) \cdot \text{poly}(N)$ running time for some function g); such algorithm is called a *fixed parameter tractable (FPT)* algorithm parameterized by k. The question here is whether this added running time can help us approximate the problem beyond the $O(k)$ factor achieved by the straightforward algorithm. We show that, even in this parameterized setting, the trivial algorithm is still essentially optimal (up to lower order terms). This result holds under the Gap Exponential Time Hypothesis (Gap-ETH), a strengthening of ETH which states that, for some $\varepsilon > 0$, even distinguishing between a satisfiable 3-CNF formula and one which is not even $(1-\varepsilon)$-satisfiable cannot be done in subexponential time (see Conjecture 7). Moreover, under this stronger assumption, we improve the lower order term in our inapproximability ratio from $2^{(\log k)^{1-\beta}}$ for some $\beta > 0$ to $2^{(\log k)^{1/2+\rho}}$ for any $\rho > 0$. This result is stated formally below.

▶ **Theorem 2.** *Assuming Gap-ETH, for any constant $\rho > 0$ and any function g, no algorithm can, given a 2-CSP instance Γ with alphabet size n and k variables such that the constraint graph is complete, distinguish between the following two cases in $g(k) \cdot (nk)^{O(1)}$ time:*

- *(Completeness)* $\text{val}(\Gamma) = 1$*, and,*
- *(Soundness)* $\text{val}(\Gamma) < 2^{(\log k)^{1/2+\rho}}/k$*.*

To the best of our knowledge, the only previous inapproximability result for parameterized 2-CSPs is from [16]. There the authors showed that, assuming Gap-ETH, no $k^{o(1)}$-approximation $g(k) \cdot (nk)^{O(1)}$-time algorithm exists; this is shown via a simple reduction from parameterized inapproximbability of Densest-k Subgraph from [11] (which is in turn based on a construction from [33]). Our result is a direct improvement over this result.

[1] In [19], only the Gap-ETH-hardness result is stated. However, the ETH-hardness result follows easily by invoking a PCP theorem (Theorem 6 below) to get a gap instance.

We end our discussion on 2-CSPs by noting that, while our results suggest that the trivial algorithm achieves an essentially optimal ratio in terms of k, non-trivial approximation is possible when we measure the ratio in terms of N instead of k: in particular, a polynomial time $O(N^{1/3})$-approximation algorithm is known for the problem [14].

Direct Steiner Network

As a corollary of our hardness of approximation results for 2-CSPs, we obtain an inapproximability result for Directed Steiner Network with polynomial ratio in terms of the number of demand pairs. In the Directed Steiner Network (DSN) problem (sometimes referred to as the Directed Steiner Forest problem [26, 17]), we are given an edge-weighed directed graph G and a set $\mathcal{D}$ of k demand pairs $(s_1, t_1), \dots, (s_k, t_k) \in V \times V$ and the goal is to find a subgraph H of G with minimum weight such that there is a path in H from s_i to t_i for every $i \in [k]$. DSN was first studied in the approximation algorithms context by Charikar et al. [13] who gave a polynomial time $\tilde{O}(k^{2/3})$-approximation algorithm for the problem. This ratio was later improved to $O(k^{1/2+\varepsilon})$ for every $\varepsilon > 0$ by Chekuri et al. [15]. Later, a different algorithm with similar approximation ratio was proposed by Feldman et al. [26].

Algorithms with approximation ratios in terms of the number of vertices n have also been devised [26, 9, 17, 1]. In this case, the best known algorithm is that of Berman et al. [9], which yields an $O(n^{2/3+\varepsilon})$-approximation for every constant $\varepsilon > 0$ in polynomial time.

On the hardness side, there exists a known reduction from 2-CSP to DSN that preserves approximation ratio to within polynomial factor[2] [25]. Hence, known hardness of approximation of 2-CSPs translate immediately to that of DSN: it is NP-hard to approximate to within any polylogarithmic ratio, it is hard to approximate to within $2^{\log^{1-\varepsilon} n}$ factor unless $\mathsf{NP} \subseteq \mathsf{QP}$, and it is Gap-ETH-hard to approximate to within $n^{o(1)}$ factor. Note that, since k is always bounded above by n^2, these hardness results also hold when n is replaced by k in the ratios. Recently, this reduction was also used by Chitnis et al. [16] to rule out $k^{o(1)}$-FPT-approximation algorithm for DSN parameterized by k assuming Gap-ETH. Alas, none of these results achieve ratios that are polynomial in either n or k and it remains open whether DSN is hard to approximate to within a factor that is polynomial in n or in k.

By plugging our hardness results for 2-CSPs into the reduction, we immediately get (Gap)-ETH-hardness of approximating DSN to within a factor of $k^{1/4-o(1)}$ as stated below.

▶ **Corollary 3.** *Assuming ETH, there exists a constant $\beta' > 0$ such that, there is no polynomial time $\frac{k^{1/4}}{2^{(\log k)^{1-\beta'}}}$-approximation algorithm for DSN.*

▶ **Corollary 4.** *Assuming Gap-ETH, for any constant $\rho' > 0$ and any function g, there is no $g(k) \cdot (nk)^{O(1)}$-time $\frac{k^{1/4}}{2^{(\log k)^{1/2+\rho'}}}$-approximation algorithm for DSN.*

In other words, if one wants a polynomial time approximation algorithm with ratio depending only on k and not n, then the approximation ratios from the algorithms of [15, 26] are roughly within a square of the best possible approximation ratio. To the best of our knowledge, these are the first inapproximability results of DSN whose ratios are polynomial in k.

---

[2] That is, for any non-decreasing function ρ , if DSN admits $\rho(nk)$-approximation in polynomial time, then 2-CSP also admits $\rho(nk)^c$-approximation polynomial time for some absolute constant c.

Agreement tests

Our main result is proved through an agreement testing argument. In agreement testing there is a universe V, a collection of subsets $S_1, \ldots, S_k \subseteq V$, and for each subset S_i we are given a local function $\sigma_{S_i} : S_i \to \{0,1\}$. A pair of subsets are said to *agree* if their local functions agree on every element in the intersection. The goal is, given a non-negligible fraction of agreeing pairs, to deduce the existence of a global function $g : V \to \{0,1\}$ that coincides with many of the local functions. For a more complete description see [22].

Agreement tests capture a natural local to global statement and are present in essentially all PCPs, for example they appear explicitly in the line vs. line and plane vs. plane low degree tests [40, 4, 39]. Our reduction is based on a combinatorial agreement test, where the universe is $[n]$ and the subsets $S_1, \ldots, S_k$ have $\Omega(n)$ elements each and are "in general position", namely they behave like subsets chosen independently at random. A convenient feature about this setting is that every pair of subsets intersect.

Since we are aiming for a large gap, the agreement test must work (i.e., yield a global function) with a very small fraction of agreeing pairs, which in our case is close to $1/k$.

In this small agreement regime the idea, as pioneered in the work of Raz-Safra citeRazS97, is to zero in on a sub-collection of subsets that is (almost) perfectly consistent. From this sub-collection it is easy to recover a global function and show that it coincides almost perfectly with the local functions in the sub-collection. A major difference between our combinatorial setting and the algebraic setting of Raz-Safra is the lack of "distance" in our case: we can not assume that two distinct local functions differ on many points (in contrast, this is a key feature of low degree polynomials). We overcome this by considering different "strengths" of agreement, depending on the fraction of points on which the two subsets agree. This notion too is present in several previous works on combinatorial agreement tests [28, 23].

Hardness of Approximation through Subexponential Time Reductions

Our result is one of the many results in recent years that show hardness of approximation via subexponential time reductions. These results are often based on ETH and its variants. Proposed by Impagliazzo and Paturi [29], ETH can be formally stated as follows:

▶ **Conjecture 5** (Exponential Time Hypothesis (ETH) [29]). *There exist constants $\delta > 0$ such that no algorithm can decide whether any given 3-CNF formula is satisfiable in time $O(2^{\delta m})$ where m denotes the number of clauses*[3].

A crucial ingredient in most reductions in this line of work is a nearly-linear size PCP Theorem. For the purpose of our work, the PCP Theorem can be viewed as a polynomial time transformation of a 3-SAT instance $\widetilde{\Phi}$ to another 3-SAT instance Φ that creates a gap between the YES and NO cases. Specifically, if $\widetilde{\Phi}$ is satisfiable, Φ remains satisfiable. On the other hand, if $\widetilde{\Phi}$ is unsatisfiable, then Φ is not only unsatisfiable but it is also not even $(1-\varepsilon)$-satisfiable for some constant $\varepsilon > 0$ (i.e. no assignment satisfies $(1-\varepsilon)$ fraction of clauses). The "nearly-linear size" part refers to the size of the new instance Φ compared to that of $\widetilde{\Phi}$. Currently, the best known dependency in this form of the PCP Theorem between the two sizes is quasi-linear (i.e. with a polylogarithmic blow-up), as stated below.

[3] The original conjecture states the lower bound as exponential in terms of the number of variables not clauses. However, thanks to the sparsification lemma [30], these two versions are equivalent.

► **Theorem 6** (Quasi-Linear Size PCP [8, 18])**.** *For some constants $\varepsilon, \Delta, c > 0$, there is a polynomial time algorithm that, given any 3-CNF formula $\widetilde{\Phi}$ with m clauses, produces another 3-CNF formula Φ with $O(m \log^c m)$ clauses such that*

- *(Completeness) if* $\text{val}(\widetilde{\Phi}) = 1$*, then* $\text{val}(\Phi) = 1$*, and,*
- *(Soundness) if* $\text{val}(\widetilde{\Phi}) < 1$*, then* $\text{val}(\Phi) < 1 - \varepsilon$*, and,*
- *(Bounded Degree) each variable in Φ appears in at most Δ clauses.*

ETH-hardness of approximation proofs usually proceed in two steps. First, the PCP Theorem is invoked to reduce a 3-SAT instance $\widetilde{\Phi}$ of size m to an instance of the gap version of 3-SAT Φ of size $m' = O(m \log^c m)$. Second, the gap version of 3-SAT is reduced in subexponential time to the problem at hand. As long as the reduction takes time $2^{o(m'/\log^c m')} = 2^{o(m)}$, we can obtain hardness of approximation result for the latter problem. This is in contrast to proving NP-hardness of approximation for which a polynomial time reduction is required.

Another related but stronger version of ETH that we will also employ is Gap-ETH, which states that even the gap version of 3-SAT cannot be solved in subexponential time:

► **Conjecture 7** (Gap Exponential Time Hypothesis (Gap-ETH) [19, 34])**.** *There exist constants $\delta, \varepsilon, \Delta > 0$ such that no algorithm can, given any 3-CNF formula Φ such that each of its variable appears in at most Δ clauses*[4]*, distinguish between the following two cases in time $O(2^{\delta m})$ time where m denotes the number of clauses:*

- *(Completeness)* $\text{val}(\Phi) = 1$.
- *(Soundness)* $\text{val}(\Phi) < 1 - \varepsilon$.

By starting with Gap-ETH instead of ETH, there is no need to apply the PCP Theorem and hence a polylogarithmic loss in the size of the 3-SAT instance does not occur. As demonstrated in previous works, this allows one to improve the ratio in hardness of approximation results [19, 34, 33] and, more importantly, prove inapproximability results for some parameterized problems [10, 11, 16], which are not known to be hard to approximate under ETH. Specifically, for many parameterized problems, the reduction from the gap version of 3-SAT to the problem has size $2^{m'/f(k)}$ for some function f that grows to infinity with k, where m' is the number of clauses in the 3-CNF formula and k is the parameter of the problem. For simplicity, let us focus on the case where $f(k) = k$. If one wishes to derive a meaningful result starting form ETH, $2^{m'/k}$ must be subexponential in terms of m, the number of clauses in the original (no-gap) 3-CNF formula. This means that the term k must dominate the $\log^c m$ factor blow-up from the PCP Theorem. However, since FPT algorithms are allowed to have running time of the form $g(k)$ for any function g, we can pick g to be 2^{2^k}. In this case, the algorithm runs in $2^{\omega(m)}$ time and we cannot deduce anything regarding the algorithm. On the other hand, if we start from Gap-ETH, we can pick k to be a large constant independent of m, which indeed yields hardness of the form claimed in Theorem 2 and Corollary 4.

Finally, we remark that Gap-ETH would follow from ETH if a linear-size (constant-query) PCP exists. While constructing short PCPs has long been an active area of research [6, 8, 18, 36, 7], no such PCP is yet known. For a more in-depth discussion, please refer to [19].

Organization of the Paper. In the next section, we describe our reduction and give an overview of the proof. After defining additional notations in Section 3, we proceed to provide

[4] This bounded degree assumption can be assumed without loss of generality; see [34] for more details.

the soundness analysis of our construction in Section 4. In Section 5, we briefly discuss the setting of parameters that give the desired inapproximability results for 2-CSPs and DSN. Finally, we conclude our work with some discussions and open questions in Section 6.

2 Proof Overview

Like other (Gap-)ETH-hardness of approximation results, our proof is based on a subexponential time reduction from the gap version of 3-SAT to our problem of interest, 2-CSPs. Before we describe our reduction, let us define more notations for 2-CSPs and 3-SAT.

2-CSPs. For notational convenience, we will modify the definition of 2-CSPs slightly so that each variable is allowed to have different alphabets; this definition is clearly equivalent to the more common definition used above. Specifically, an instance Γ of 2-CSP now consists of (1) a constraint graph $G = (V, E)$, (2) for each vertex (or variable) $v \in V$, an alphabet set Σ_v, and, (3) for each edge $\{u, v\} \in E$, a constraint $C_{uv} \subseteq \Sigma_u \times \Sigma_v$. Additionally, to avoid confusion with 3-SAT, we refrain from using the word *assignment* for 2-CSPs and instead use *labeling*, i.e., a labeling of Γ is a tuple $\sigma = (\sigma_v)_{v \in V}$ such that $\sigma_v \in \Sigma_v$ for all $v \in V$. An edge $\{u, v\} \in E$ is said to be *satisfied* by a labeling σ if $(\sigma_u, \sigma_v) \in \Sigma_u \times \Sigma_v$. The value of a labeling σ, denoted by $\text{val}(\sigma)$, is defined as the fraction of edges that it satisfies, i.e., $|\{\{u, v\} \in E \mid (\sigma_u, \sigma_v) \in C_{uv}\}|/|E|$. The goal of 2-CSPs is to find σ with maximum value; we denote the such optimal value by $\text{val}(\Gamma)$, i.e., $\text{val}(\Gamma) = \max_\sigma \text{val}(\sigma)$.

3-SAT. An instance Φ of 3-SAT consists of a variable set X and a clause set $\mathcal{C}$ where each clause is a disjunction of at most three literals. For any assignment $\psi : \mathrm{X} \to \{0, 1\}$, $\text{val}(\psi)$ denotes the fraction of clauses satisfied by ψ. The goal is to find an assignment ψ that satisfies as many clauses as possible; let $\text{val}(\Phi) = \max_\psi \text{val}(\psi)$ denote the fraction of clauses satisfied by such assignment. For each $C \in \mathcal{C}$, we use $\text{var}(C)$ to denote the set of variables whose literals appear in C and, for each $S \subseteq \mathcal{C}$, we use $\text{var}(S)$ to denote $\bigcup_{C \in S} \text{var}(C)$.

Our Construction

Before we state our reduction, let us again reiterate the objective of our reduction. Given a 3-SAT instance $\Phi = (\mathrm{X}, \mathcal{C})$, we would like to produce a 2-CSP instance Γ_Φ such that

- (Completeness) If $\text{val}(\Phi) = 1$, then $\text{val}(\Gamma_\Phi) = 1$,
- (Soundness) If $\text{val}(\Phi) < 1 - \varepsilon$, $\text{val}(\Gamma_\Phi) < k^{o(1)}/k$ where k is number of variables of Γ_Φ,
- (Reduction Time) The time it takes to produce Γ_Φ should be $2^{o(m)}$ where $m = |\mathcal{C}|$,

where $\varepsilon > 0$ is some absolute constant.

Observe that, when plugging a reduction with these properties to Gap-ETH, we directly arrive at the claimed $k^{1-o(1)}$ inapproximability for 2-CSPs. However, for ETH, since we start with a decision version of 3-SAT without any gap, we have to first invoke the PCP theorem to produce an instance of the gap version of 3-SAT before we can apply our reduction. Since the shortest known PCP has a polylogarithmic blow-up in the size, the running time lower bound for gap 3-SAT will not be exponential anymore, rather it will be of the form $2^{\Omega(m/\text{polylog} m)}$ instead. Hence, our reduction will need to produce Γ_Φ in $2^{o(m/\text{polylog} m)}$ time. As we shall see below, this will also be possible with appropriate settings of parameters.

We now move on to state our reduction. In addition to a 3-CNF formula Φ, the reduction also takes in a collection S of subsets of clauses of Φ. For now, the readers should think of the subsets in S as random subsets of $\mathcal{C}$ where each element is included in each subset independently at random with probability α, which will be specified later. As we will see below, we only need two simple properties that the subsets in S are "well-behaved" enough and we will later give a deterministic construction of such well-behaved subsets. With this in mind, our reduction can be formally described as follows.

▶ **Definition 8** (The Reduction). Given a 3-CNF formula $\Phi = (\mathrm{X}, \mathcal{C})$ and a collection S of subsets of $\mathcal{C}$, we define a 2-CSP instance $\Gamma_{\Phi,\mathsf{S}} = (G = (V, E), \Sigma, \{C_{uv}\}_{\{u,v\}\in E})$ as follows:

- The graph G is the complete graph where the vertex set is S, i.e., $V = \mathsf{S}$ and $E = \binom{\mathsf{S}}{2}$.
- For each $S \in \mathsf{S}$, the alphabet set Σ_{S} is the set of all partial assignments to $\mathrm{var}(S)$ that satisfies every clause in S, i.e., $\Sigma_S = \{\psi_S : \mathrm{var}(S) \to \{0,1\} \mid \forall C \in S, \psi_S \text{ satisfies } C\}$.
- For every $S_1 \neq S_2 \in \mathsf{S}$, (ψ_{S_1}, ψ_{S_2}) is included in $C_{S_1S_2}$ if and only if there are consistent, i.e., $C_{S_1S_2} = \{(\psi_{S_1}, \psi_{S_2}) \in \Sigma_{S_1} \times \Sigma_{S_2} \mid \forall x \in \mathrm{var}(S_1) \cap \mathrm{var}(S_2), \psi_{S_1}(x) = \psi_{S_2}(x)\}$.

Let us now examine the properties of the reduction. The number of vertices in $\Gamma_{\Phi,\mathsf{S}}$ is $k = |\mathsf{S}|$. For this proof overview, α should be thought of as $1/\mathrm{polylog}(m)$ whereas k should be thought of as much larger than $\exp(1/\alpha)$ (e.g. $k = \exp(1/\alpha^2)$). For such value of k, all random sets in S have size $O(\alpha m)$ w.h.p., meaning that the reduction time is $2^{m/\mathrm{polylog} m}$.

Moreover, when Φ is satisfiable, it is not hard to see that $\mathrm{val}(\Gamma_{\Phi,\mathsf{S}}) = 1$; specifically, if $\psi : \mathrm{X} \to \{0,1\}$ is the assignment that satisfies every clause of Φ, then we can label each vertex $S \in \mathsf{S}$ of $\Gamma_{\Phi,\mathsf{S}}$ by $\psi|_{\mathrm{var}(S)}$, the restriction of ψ on $\mathrm{var}(S)$. Since ψ satisfies every clause, $\psi|_{\mathrm{var}(S)}$ satisfies all clauses in S and this is a valid labeling. Moreover, since these are restrictions of the same global assignment ψ, they are consistent, i.e., every edge is satisfied.

Hence, we are only left to show that, if $\mathrm{val}(\Phi) < 1 - \varepsilon$, then $\mathrm{val}(\Gamma_{\Phi,\mathsf{S}}) < k^{o(1)}/k$; this is indeed our main contribution. We will show this contrapositively: assuming that $\mathrm{val}(\Gamma_{\Phi,\mathsf{S}}) \geqslant k^{o(1)}/k$, we will "decode" back an assignment to Φ that satisfies at least $1 - \varepsilon$ fraction of clauses.

We remark that our task at hand can be viewed as agreement testing. Informally, in agreement testing, the input is a collection $\{f_T\}_T$ of local functions $f_T : T \to \{0,1\}$ where T is a subset of some universe $\mathcal{U}$ such that, for many pairs T_1 and T_2, f_{T_1} and f_{T_2} agree, i.e., $f_{T_1}(u) = f_{T_2}(u)$ for all $u \in T_1 \cap T_2$. An agreement theorem says that there must be a global function $f : \mathcal{U} \to \{0,1\}$ that coincides (exactly or approximately) with many of the local functions, and thus explains the pairwise "local" agreements. (See e.g. Section 1.1 [22] for a formal definition.) In our case, a labeling $\sigma = \{\sigma_S\}_{S\in\mathsf{S}}$ with high value is exactly a collection of functions $\sigma_S : S \to \{0,1\}$ such that, for many pairs of S_1 and S_2, σ_{S_1} and σ_{S_2} agrees. Our proof of soundness indeed recovers a global function $\psi : \mathrm{X} \to \{0,1\}$ that coincides with many of the local functions σ_S's and thus satisfies $1 - \varepsilon$ fraction of clauses of Φ.

A Simplified Proof: $k^{1/2-o(1)}$ Ratio Inapproximability

Before we describe how we can decode an assignment for Φ when $\mathrm{val}(\Gamma_{\Phi,\mathsf{S}}) \geqslant k^{o(1)}/k$, let us sketch the proof assuming a stronger assumption that $\mathrm{val}(\Gamma_{\Phi,\mathsf{S}}) \geqslant \Theta(1/\alpha)/k^{1/2}$. Since $1/\alpha = k^{o(1)}$, this already implies a $k^{1/2-o(1)}$ factor ETH-hardness of approximating 2-CSPs. In the next subsection, we will refine the arguments and arrive at the desired $k^{1-o(1)}$ factor.

Let D be a large constant to be chosen later. Recall that $\mathrm{val}(\Gamma_{\Phi,\mathsf{S}}) \geqslant (D/\alpha)/k^{1/2}$ implies that there is a labeling $\sigma = \{\sigma_S\}_{S\in\mathsf{S}}$ that satisfies $(D/\alpha)/k^{1/2} \cdot \binom{k}{2} \geqslant \left(\frac{D}{4\alpha}\right) k^{3/2}$ edges.

Let us consider the *consistency graph* of $\Gamma_{\Phi,\mathsf{S}}$ with respect to σ. This is the graph G^σ whose vertex set is S and there is an edge between S_1 and S_2 iff σ_{S_1} and σ_{S_2} are consistent. Note that the number of edges in G^σ is equal to the number of edges satisfied by σ.

Previous works on agreement testers exploit particular structures of the consistency graph to decode a global function. One such property that is relevant to our proof is the notion of *almost transitivity* defined by Raz and Safra in the analysis of their test [39]. More specifically, a graph $G = (V, E)$ is said to be q-transitive for some $q > 0$ if, for every non-edge $\{u, v\}$ (i.e. $\{u, v\} \in \binom{V}{2} \notin E$), u and v can share at most q common neighbors[5]. Raz and Safra showed that their consistency graph is $(k^{1-\Omega(1)})$-transitive where k denote the number of vertices of the graph. They then proved a generic theorem regarding $(k^{1-\Omega(1)})$-transitive graphs: its vertex set can be partitioned so that the subgraph induced by each partition is a clique and the number of edges between different partitions is small. Since a sufficiently large clique corresponds to a global function in their setting, they immediately arrive at their result.

Observe that, in our setting, a large clique also corresponds to an assignment that satisfies almost all clauses of Φ. In particular, suppose that there exists $\mathsf{S}' \subseteq \mathsf{S}$ of size sufficiently large size such that S induces a clique in G^σ. Since σ_S are perfectly consistent among all $S \in \mathsf{S}'$, these local functions induce a function $\psi : \text{var}(\bigcup_{S \in \mathsf{S}'} S) \to \{0, 1\}$ that satisfies all clauses in $\bigcup_{S \in \mathsf{S}'} S$. If S is larger than $\Omega(1/(\varepsilon\alpha))$, then, with high probability, $\bigcup_{S \in \mathsf{S}'} S$ contains all but ε fraction of clauses, which means that ψ satisfies $1 - \varepsilon$ fraction of clauses as desired. Hence, if we could show that our consistency graph G^σ is $(k^{1-\Omega(1)})$-transitive, then we could use the same argument as Raz and Safra's to deduce our desired result. Alas, our graph G^σ does not necessarily satisfy this transitivity property; for instance, consider any two sets $S_1, S_2 \in \mathsf{S}$ and let $\sigma_{S_1}, \sigma_{S_2}$ be such that they disagree on only one variable, i.e., there is a unique $x \in S_1 \cap S_2$ such that $\sigma_{S_1}(x) \neq \sigma_{S_2}(x)$. It is possible that, for every $S \in \mathsf{S}$ that does not contain x, σ_S agrees with both σ_{S_1} and σ_{S_2}; in other words, every such S can be a common neighbor of S_1 and S_2. Since each variable x appears roughly in only $\Theta(\alpha)$ fraction of the sets, there can be as many as $(1 - \Theta(\alpha))k = (1 - o(1))k$ common neighbors of S_1 and S_2 even when there is no edge between S_1 and S_2!

Fortunately for us, a weaker statement holds. If σ_{S_1} and σ_{S_2} disagree on ζn variables (instead of just one variable as above) where n denotes the number of variables in the 3-CNF formula, then we say that they *strongly* disagree. In this case, S_1 and S_2 can have at most $O(\ln(1/\zeta)/\alpha)$ common neighbors in G^σ. Here ζ should be thought of as α^2 times a small constant which will be specified later. To see why this statement holds, observe that, since every $S \in \mathsf{S}$ is a random subset that includes each clause $C \in \mathcal{C}$ with probability α, Chernoff bound implies that, for every subcollection $\widetilde{\mathsf{S}} \subseteq \mathsf{S}$ of size $\Omega(\ln(1/\zeta)/\alpha)$, $\bigcup_{S \in \widetilde{\mathsf{S}}} S$ contains all but $O(\zeta)$ fraction of clauses. Let $\widetilde{\mathsf{S}}_{S_1,S_2} \subseteq \mathsf{S}$ denote the set of common neighbors of S_1 and S_2. It is easy to see that S_1 and S_2 can only disagree on variables that do not appear in $\text{var}(\bigcup_{S \in \widetilde{\mathsf{S}}_{S_1,S_2}} S)$. If $\widetilde{\mathsf{S}}_{S_1,S_2}$ is of size $\Omega(\ln(1/\zeta)/\alpha)$, then $\bigcup_{S \in \widetilde{\mathsf{S}}_{S_1,S_2}} S$ contains all but $O(\zeta)$ fraction of clauses. Hence, assuming that each variable appears in bounded number of clauses, $\text{var}(\bigcup_{S \in \widetilde{\mathsf{S}}_{S_1,S_2}} S)$ also contains all but $O(\zeta)$ fraction of variables. This means that S_1 and S_2 disagrees only on $O(\zeta)$ fraction of variables. By selecting the constant appropriately inside $O(\cdot)$, we arrive at the claim statement.

In other words, while the transitive property does not hold for every edge, it holds for the edges $\{S_1, S_2\}$ where σ_{S_1} and σ_{S_2} strongly disagree. This motives us to define a two-level consistency graph, where the edges with strong disagreement are referred to as *red* edges whereas the original edges in G^σ are now referred to as *blue* edges, as formalized below.

▶ **Definition 9** (Two-Level Consistency Graph)**.** A red/blue graph is an undirected graph $G = (V, E = E_r \cup E_b)$ where its edge set E is partitioned into two sets E_r, the set of red edges, and E_b, the set of blue edges. We use prefixes "blue-" and "red-" to refer to quantities of (V, E_b) and (V, E_r) respectively. (E.g. u is a blue-neighbor of v if $\{u, v\} \in E_b$).

---

[5] In [39], the parameter q denotes the *fraction* of vertices that are neighbors of both u and v rather than the *number* of such vertices. However, we use the latter notion as it is more convenient for us.

Given a labeling σ of $\Gamma_{\Phi,\mathsf{S}}$ and a real number $0 < \zeta < 1$, the two-level consistency graph $G^{\sigma,\zeta} = (V^{\sigma,\zeta}, E_r^{\sigma,\zeta} \cup E_b^{\sigma,\zeta})$ is a red/blue graph defined as follows.

- The vertex set $V^{\sigma,\zeta}$ is simply S.
- The blue edges are the pairs $\{S_1, S_2\}$ satisfied by σ, i.e., $E_b = \{\{S_1, S_2\} \in \binom{\mathsf{S}}{2} \mid \text{disagr}(\sigma_{S_1}, \sigma_{S_2}) = 0\}$.
- The red edges are the pairs $\{S_1, S_2\}$ whose the assignments to the two endpoints disagree on more than ζn variables, i.e., $E_r = \{\{S_1, S_2\} \in \binom{\mathsf{S}}{2} \mid \text{disagr}(\sigma_{S_1}, \sigma_{S_2}) > \zeta n\}$.

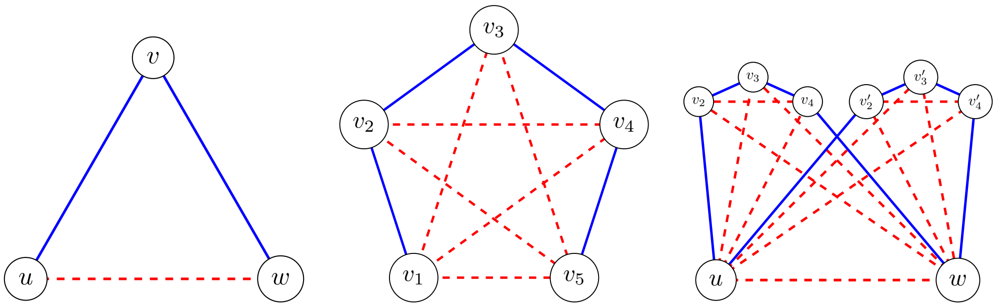
Here $\text{disagr}(\sigma_{S_1}, \sigma_{S_2})$ denotes the number of variables that $\sigma_{S_1}, \sigma_{S_2}$ disagree on, i.e., $\text{disagr}(\sigma_{S_1}, \sigma_{S_2}) = |\{x \in \text{var}(S_1) \cap \text{var}(S_2) \mid \sigma_{S_1}(x) \neq \sigma_{S_2}(x)\}|$.

Note that, when $\text{disagr}(\sigma_{S_1}, \sigma_{S_2}) \in [1, \zeta n]$, S_1, S_2 constitute neither a blue nor a red edge.

Now, the transitivity property above can be stated as follows: for every red edge $\{S_1, S_2\}$ of $G^{\sigma,\zeta}$, there are at most $O(\ln(1/\zeta)/\alpha)$ different S's such that both $\{S, S_1\}$ and $\{S, S_2\}$ are blue edges. For brevity, let us call any red/blue graph $G = (V, E_r \cup E_b)$ *q-red/blue-transitive* if, for every red edge $\{u, v\} \in E_r$, u and v have at most q common blue-neighbors. We will now argue that in any q-red/blue-transitive of average blue-degree d, there exists a subset $U \subseteq V$ of size $\Omega(d)$ such that, only $O(qk/d^2)$ fraction of pairs of vertices in U form red edges.

Before we prove this, let us state why this is useful for decoding a good assignment for the 3-CNF formula Φ. Observe that such a subset U of vertices in the two-level consistency graph translates to a subcollection $\mathsf{S}' \subseteq \mathsf{S}$ such that, for all but $O(qk/d^2)$ fraction of pairs of sets $S_1, S_2 \subseteq \mathsf{S}'$, $\{S_1, S_2\}$ does not form a red edge, meaning that σ_{S_1} and σ_{S_2} disagrees on at most ζn variables. In other words, S' is similar to a clique in the (not two-level) consistency graph, except that (1) $O(qk/d^2)$ fraction of pairs $\{S_1, S_2\}$ are allowed to disagree on as many variables as they like, and (2) even for the rest of pairs, the guarantee now is that they agree on all but at most ζn variables, instead of total agreement as in the previous case of clique. Fortunately, when S' satisfies a certain uniformity condition (which random subsets satisfy w.h.p.), this still suffices to find an assignment to Φ that satisfies $1 - O(qk/d^2) - O(\zeta/\alpha^2)$ fraction of the clauses. One way construct a good assignment is to simply assign each variable $x \in \mathrm{X}$ according to the majority of $\sigma_S(x)$ for all $S \in \mathsf{S}'$ such that $x \in \text{var}(S)$. Our actual proof proceeds slightly differently for technical reasons. Note that in our case $q = O(\ln(1/\zeta)/\alpha)$ and $d = \Omega(Dk^{1/2}/\alpha)$; if we pick $\zeta \ll \varepsilon\alpha^2$ and $D \gg 1/\sqrt{\varepsilon}$, we indeed get an assignment that satisfies $1 - \varepsilon$ fraction of clauses.

We now move on to sketch how one can find such a clique-like subgraph. For simplicity, let us assume that every vertex has the same blue-degree (i.e. (V, E_b) is d-regular). Let us count the number of *red-blue-blue triangle* (or *rbb triangle*), which is a 3-tuple (u, v, w) of vertices in V such that $\{u, v\}, \{v, w\}$ are blue edges whereas $\{u, w\}$ is a red edge. An illustration of a rbb triangle can be found in Figure 1a. The red/blue transitivity can be used to bound the number of rbb triangles as follows. For each $(u^*, w^*) \in V^2$, since the graph is q-red/blue-transitive there are at most q rbb triangle with $u = u^*$ and $w = w^*$. Hence, in total, there can be at most qk^2 rbb triangles. As a result, there exists $v^* \in V$ such that the number of rbb triangles (u, v, w) such that $v = v^*$ is at most qk. Let us now consider the set $U = N_b(v^*)$ that consists of all blue-neighbors of v^*. There can be at most qk red edges with both endpoints in $N_b(v^*)$ because each such edge corresponds to a rbb triangle with $v = v^*$. From our assumption that every vertex has blue degree d, we indeed have that $|U| = d$ and that the fraction of pairs of vertices in U that are linked by red edges is $O(qk/d^2)$ as desired. This completes our overview for $k^{1/2-o(1)}$ factor inapproximability result for 2-CSPs.



(a) a red-blue-blue triangle **(b)** a red-filled 4-walk **(c)** disjoint red-filled 4-walks

Figure 1 Illustrations of red-filled walks. Figures 1a and 1b demonstrate a red-filled 2 walk (rbb triangle) and a red-filled 4-walk respectively. Figure 1c shows two disjoint red-filled 4-walks.

Towards Nearly Linear Ratio Inapproximability

To improve our inapproximability ratio from $k^{1/2-o(1)}$ to $k^{1-o(1)}$, we need to first understand why the approach above fails to work beyond the $k^{1/2}$ ratio regime. To do so, note that the above proof sketch can be summarized into three main steps as follows:

(1) Show that the two-level consistency graph $G^{\sigma,\zeta}$ is q-red/blue-transitive for some $q = k^{o(1)}$.
(2) Use red/blue transitivity to find a large subgraph of $G^{\sigma,\zeta}$ with few induced red edges.
(3) Decode a good assignment to Φ from such "clique-like" subgraph.

The reason that we need $d \geqslant k^{1/2}$ lies in Step 2. Although not stated as such above, our argument in this step can be described as follows. Consider all length-2 blue-walks, i.e., all $(u, v, w) \in V^3$ such that $\{u, v\}$ and $\{v, w\}$ are both blue edges. Using the red/blue transitivity of the graph, we argue that, for almost of all these walks, $\{u, w\}$ is not a red edge (i.e. (u, v, w) is not a rbb triangle), which then allows us to find the "clique-like" subgraph. For this argument to work, we need the number of length-2 blue-walks to exceed the number of rbb triangles. The former is kd^2 whereas the latter is bounded by k^2q in q-red/blue-transitive graphs. This means that we need $kd^2 \geqslant k^2q$, which implies that $d \geqslant k^{1/2}$.

To overcome this limitation, we will instead consider length-ℓ blue-walks for $\ell > 2$ and define a "rbb-triangle-like" structure on these walks. Our goal is again to show that this structure appears rarely in random length-ℓ blue-walks and then use this to find a subgraph that allows us to decode a good assignment for Φ. Observe that the number of length-ℓ blue walks is kd^ℓ. We hope that the number of "rbb-triangle-like" structures is still small; in particular, we will still get a similar bound $k^{2+o(1)}$ for such generalized structure, similar to our previous bound for the red-blue-blue triangles. When this is the case, we need $kd^\ell \geqslant k^{2+o(1)}$, meaning that when $\ell = \omega(1)$ it suffices to select $d = k^{o(1)}$, which yields $k^{1-o(1)}$ factor inapproximability as desired. To facilitate our discussion, let us define notations for ℓ-walks here.

▶ **Definition 10** (ℓ-Walks)**.** For any red/blue graph $G = (V, E_r \cup E_b)$ and any integer $\ell \geqslant 2$, an *ℓ-blue-walk*, abbreviated as an *ℓ-walk*, in G is an $(\ell+1)$-tuple of vertices $(v_1, v_2, \dots, v_{\ell+1}) \in V^{\ell+1}$ such that every pair of consecutive vertices is joined by a blue edge, i.e., $\{v_i, v_{i+1}\} \in E_b$ for every $i \in [\ell]$. We use $\mathcal{W}_\ell^G$ to denote the set of all ℓ-walks in G.

The structure we will consider is a natural generalization of rbb triangles to an ℓ-walk in which every pair of non-consecutive vertices of the walk must be joined by a red edge. We call such a walk a *red-filled ℓ-walk* (see Figure 1b):

▶ **Definition 11** (Red-Filled ℓ-Walks). For any red/blue graph $G = (V, E_r \cup E_b)$, a *red-filled ℓ-walk* is an ℓ-walk $(v_1, v_2, \dots, v_{\ell+1})$ such that every pair of non-consecutive vertices is joined by a red edge, i.e., $\{v_i, v_j\} \in E_r$ for every $i, j \in [\ell+1]$ such that $j > i+1$. Let $\widehat{\mathcal{W}}_\ell^G$ denote the set of all red-filled ℓ-walks in G. Moreover, for every $u, v \in V$, let $\widehat{\mathcal{W}}_\ell^G(u, v)$ denote the set of all red-filled ℓ-walks from u to v, i.e., $\mathcal{W}_\ell^G(u, v) = \{(v_1, \dots, v_{\ell+1}) \in \widehat{\mathcal{W}}_\ell^G \mid v_1 = u \wedge v_{\ell+1} = v\}$.

As mentioned earlier, we will need a generalized transitivity property for our new structure. This can be defined analogously to q-red/blue transitivity as follows.

▶ **Definition 12** ((q, ℓ)-Red/Blue Transitivity). For any $q, \ell \in \mathbb{N}$, a red/blue graph $G = (V, E_r \cup E_b)$ is said to be *(q, ℓ)-red/blue-transitive* if, for every pair of $u, v \in V$ that is joined by a red edge, there are at most q red-filled ℓ-walks from u to v, i.e., $|\widehat{\mathcal{W}}_\ell^G(u, v)| \leqslant q$.

Similar to before, we can argue that, when S consists of random subsets where each element is included in a subset w.p. α, the two-level agreement graph is (q, ℓ)-red/blue transitive for some parameter q that is a function of only α and ℓ. When $1/\alpha$ and ℓ are both small enough in terms of k, q can made to be $k^{o(1)}$. The details of the proof can be found in Section 4.1.

Once this is proved, it is not hard (using a similar argument as before) to show that, when $d \gg (kq)^{1/\ell}$, most ℓ-walks are not red-filled, i.e., $|\mathcal{W}_\ell^G| \gg |\widehat{\mathcal{W}}_\ell^G|$. Even with this, it is still unclear how we can get back a "clique-like" subgraph; in the case of $\ell = 2$ above, this implies that a blue-neighborhood induces few red edges, but the argument does not seem to generalize to larger ℓ. Fortunately, it is still quite easy to find a large subgraph that a non-trivial fraction of pairs of vertices do *not* form red edges; specifically, we will find two subsets $U_1, U_2 \subseteq V$ each of size d such that for at least $1/\ell^2$ fraction of $(u_1, u_2) \in U_1 \times U_2$, $\{u_1, u_2\}$ is not a red edge. To find such sets, observe that, if $|\mathcal{W}_\ell^G| \geqslant 2|\widehat{\mathcal{W}}_\ell^G|$, then for a random $(v_1, \dots, v_{\ell+1}) \in \mathcal{W}_\ell^G$ the probability that there exists non-consecutive vertex v_i, v_j in the walk that are joined by a red edge is at least $1/2$. Since there are less than $\ell^2/2$ such i, j, union bound implies that there must be non-consecutive i^*, j^* such that the probability that v_{i^*}, v_{j^*} are not joined by a red edge is at least $1/\ell^2$. Let us assume without loss of generality that $i^* < j^*$; since they are not consecutive, we have $i^* + 1 < j^*$.

Let us consider v_{i^*+1}, v_{j^*-1}. There must be u^* and w^* such that, conditioning on $v_{i^*+1} = u^*$ and $v_{j^*+1} = w^*$, the probability that $\{v_{i^*}, v_{j^*}\} \notin E_r$ is at least $1/\ell^2$. However, this conditional probability is exactly equal to the fraction of $(u_1, u_2) \in N_b(u^*) \times N_b(w^*)$ such that u_1, u_2 are not joined by a red edge. (Recall that $N_b(v)$ is the set of all blue-neighbors of v.) As a result, $U_1 = N_b(u^*)$ and $U_2 = N_b(w^*)$ are the sets with desired property.

We are still not done yet since we have to use these sets to decode back a good assignment for Φ. This is still not obvious: the guarantee we have for our sets U_1, U_2 are rather weak since we only know that at least $1/\ell^2$ of the pairs of vertices from the two sets do not form red edges. This is in contrast to the $\ell = 2$ case where we have a subgraph such that almost all induced edges are *not* red. Fortunately, there is a well-known fact in combinatorics called the Kővári-Sós-Turán Theorem [32] which roughly states that every bipartite graph that is not too sparse has a reasonably large biclique (a complete bipartite subgraph). We apply this theorem on the bipartite graph between U_1 and U_2 where there is an edge between $u_1 \in U_1$ and $u_2 \in U_2$ iff $\{u_1, u_2\}$ is not a red edge. This gives us $V_1 \subseteq U_1$, $V_2 \subseteq U_2$ of reasonably large sizes such that for all $(u_1, u_2) \in V_1 \times V_2$, u_1 and u_2 are not joined by a red edge.

Once we have such a "non-red biclique", we can decode a good assignment of Φ by taking the majority assignment on one side of the biclique. A simple counting argument again shows that, when V_1 and V_2 are "sufficiently uniform", this majority assignment cannot violate too many clauses of Φ. This wraps up our proof overview.

3 Preliminaries

We next define two properties of collections of subsets, which will be useful in our analysis. First, recall that, in our proof overview for the weaker $k^{1/2-o(1)}$ factor hardness, we need the following to show the red/blue transitivity of the consistency graph: for any r subsets from the collection, their union must contain almost all clauses. Here r is a positive integer that effects the red/blue transitivity parameter. Collections with this property are sometimes called *dispersers*. For walks with larger lengths, we need a stronger property that any union of r intersections of ℓ subsets are large. We call such collections *intersection dispersers*:

▶ **Definition 13** (Intersection Disperser)**.** Given a universe $\mathcal{U}$, a collection S of subsets of $\mathcal{U}$ is an (r, ℓ, η)-*intersection disperser* if, for any r disjoint subcollections $\mathsf{S}^1, \dots, \mathsf{S}^r \subseteq \mathsf{S}$ each of size at most ℓ, we have $\left|\bigcup_{i=1}^{r} \left(\bigcap_{S \in \mathsf{S}^i} S\right)\right| \geqslant (1-\eta)|\mathcal{U}|$.

Another property we need is that any sufficiently large subcollection $\widetilde{\mathsf{S}}$ of S is "sufficiently uniform". This is used when we decode a good assignment from a non-red biclique. More specifically, the uniformity condition requires that almost all clauses appear in not too small number of subsets in $\widetilde{\mathsf{S}}$, as formalized below.

▶ **Definition 14** (Uniformity)**.** For a universe $\mathcal{U}$, a collection $\widetilde{\mathsf{S}}$ of subsets of $\mathcal{U}$ is (γ, μ)-uniform if, for at least $(1-\mu)$ fraction of elements $u \in \mathcal{U}$, u appears in at least γ fraction of the subsets in $\widetilde{\mathsf{S}}$. In other words, $\widetilde{\mathsf{S}}$ is (γ, μ)-uniform iff $|\{u \in \mathcal{U} \mid |\{S \in \widetilde{\mathsf{S}} \mid u \in S\}| \geqslant \gamma|\widetilde{\mathsf{S}}|\}| \geqslant (1-\mu)|\mathcal{U}|$.

Using standard concentration bounds, it is not hard to show that, when m is sufficiently large, a collection of random subsets where each element is included in each subset independently with probability α is an $(O(\alpha^\ell), \ell, O(1))$-disperser and every subcollection of size $\Omega(1/\alpha)$ is $(\alpha, O(1))$-uniform. The exact parameter dependencies are shown in the lemma below.

▶ **Lemma 15** (Deterministic Construction of Well-Behaved Set)**.** *For any $0 < \alpha, \mu, \eta < 1$ and any $k, \ell \in \mathbb{N}$, let m_0 be $1000(\log k \log(1/\mu)/(\alpha\mu^2) + \ell \log(1/\eta) \log k/(\alpha^\ell \eta) + 1/\alpha + 1)$. For any integer $m \geqslant m_0$ and any m-element universe $\mathcal{U}$, there exists a collection S of subsets of $\mathcal{U}$ with the following properties.*

- *(Size) Every subset in S has size at most $2\alpha m$.*
- *(Intersection Disperser) S is a $(\lceil \ln(2/\eta)/\alpha^\ell \rceil, \ell, \eta)$-disperser.*
- *(Uniformity) Any subcollection $\widetilde{\mathsf{S}} \subseteq \mathsf{S}$ of size $\lceil 8\ln(2/\mu)/\alpha \rceil$ is $(\alpha/2, \mu)$-uniform.*

Moreover, such a collection S can be deterministically constructed in time poly$(m)2^{O((m_0)^3)}$.

The deterministic construction is via a standard technique of using random variables with limited independence instead of total independence; we defer the full proof of Lemma 15 to the full version of the paper.

4 Soundness Analysis

Let us now turn our focus back to the soundness analysis, which is our main technical contribution. As stated in the proof overview, our main goal is to show that, if a 3-CNF Φ has small value, then, for any well-behaved collection S of subsets of clauses of Φ, the value of $\Gamma_{\Phi,\mathsf{S}}$ must be small. More precisely, the main theorem of this section is stated below.

▶ **Theorem 16.** *For any $\Delta \in \mathbb{N}$, let Φ be any 3-CNF formula with variable set $\mathcal{X}$ and clause set $\mathcal{C}$ such that each variable appears in at most Δ clauses. Moreover, for any $0 < \eta, \zeta, \gamma, \mu < 1$ and $r, \ell, k, h \in \mathbb{N}$ such that $\ell \geqslant 2$ and $h \leqslant \log k/(\ell \log(4\ell^2))$, let S be any collection of k*

subsets of $\mathcal{C}$ such that S *is* $(r, \ell, \zeta/(3\Delta))$*-intersection disperser and every subcollection* $\widetilde{\mathsf{S}} \subseteq \mathsf{S}$ *of size* h *is* (γ, μ)*-uniform (with respect to the universe* $\mathcal{C}$*). If* $\mathrm{val}(\Phi) < 1 - 2\mu - 6\Delta\zeta/\gamma^2$*, then* $\mathrm{val}(\Gamma_{\Phi,\mathsf{S}}) < \frac{32k^{1/\ell}(r\ell)^2}{k}$.

To prove this theorem, we follow the general outline as stated in the proof overview. In particular, the proof contains three main steps, as elaborated below.

(1) First, we will show that when S is an intersection disperser with appropriate parameters, the two-level consistency graph satisfies red/blue transitivity with certain parameters.

(2) Next, we will argue that, for any red/blue transitive graphs that contains sufficiently many blue edges, we can find a non-red biclique of large size; recall that non-red biclique is two subsets V_1, V_2 of vertices such that there is no red-edge between them.

(3) Finally, we show that, if we can find a large non-red biclique in the two-level consistency graph such that the two subcollections corresponding to each side of the biclique are sufficiently uniform, then we can decode a good assignment to our 3-CNF formula Φ.

Each of the next three subsections is dedicated to each part of the proof. The main lemmas from these subsections (Lemmas 17, 20 and 23) together imply Theorem 16.

Unless stated otherwise, we note that, all results in this section hold for any parameters $\Delta, \ell, k \in \mathbb{N}$ such that $\ell \geqslant 2$, any $0 < \eta, \zeta, \gamma, \mu < 1$ and any 3-CNF formula Φ such that each variables appears in at most Δ clauses. To avoid notational cumbersomeness, we will leave these quantifiers out of the lemma statements. Moreover, throughout the section, we use m and n to denote $|\mathcal{C}|$ and $|\mathrm{X}|$ respectively. To avoid degeneracy cases, we will also assume without loss of generality that each variable appears in at least one clause.

4.1 Red/Blue-Transitivity of Two-Level Consistency Graph

The first step in our proof is to show that the two-level consistency graph $G^{\sigma,\zeta}$ is red/blue-transitive, assuming that S is an intersection disperser, as formalized below.

▶ **Lemma 17.** *If* S *is an* $(r, \ell, \zeta/(3\Delta))$*-intersection disperser, then, for any labeling* σ *of* $\Gamma_{\Phi,\mathsf{S}}$*,* $G^{\sigma,\zeta}$ *is* $((r\ell)^{2(\ell-1)}, \ell)$*-red/blue-transitive.*

In other words, we would like to show that, for every $S_1, S_2 \in \mathsf{S}$ that are joined by a red edge in $G^{\sigma,\zeta}$, there are at most $(r\ell)^{2(\ell-1)}$ red-filled ℓ-walks from S_1 to S_2. The intersection disperser does not immediately imply such a bound, due to the requirement in the definition that the subcollections are disjoint. Rather, it only directly implies a bound on number of *disjoint* ℓ-walks from S_1 to S_2, where two ℓ walks from S_1 to S_2, $(T_1 = S_1, \dots, T_{\ell+1} = S_2), (T'_1 = S_1, \dots, T'_{\ell+1} = S_2) \in \mathcal{W}_\ell^{G^{\sigma,\zeta}}(S_1, S_2)$, are said to be *disjoint* if they do not share any vertex except the starting and ending vertices, i.e., $\{T_2, \dots, T_\ell\} \cap \{T'_2, \dots, T'_\ell\} = \emptyset$. Multiple walks are said to be disjoint if they are mutually disjoint. The following claim is immediate from the definition of intersection dispersers; its proof is omitted here.

▶ **Claim 18.** *If* S *is an* $(r, \ell, \zeta/(3\Delta))$*-intersection disperser, then, for any labeling* σ*,* $2 \leqslant p \leqslant \ell$ *and* $\{S_1, S_2\} \in E_r^{\sigma,\zeta}$*, there are less than* r *disjoint* p*-walks from* S_1 *to* S_2 *in* $G^{\sigma,\zeta}$*.*

Since all 2-walks from S_1 to S_2 are disjoint, the above claim immediately gives a bound on the number of red-filled 2-walks from S_1 to S_2. To bound the number of red-filled walks of larger lengths, we will use induction on the length of the walks. Suppose that we have bounded the number of red-filled i-walks sharing starting and ending vertices for $i \leqslant z - 1$. The key idea in the proof is that we can use this inductive hypothesis to show that, for any $S_1, S_2, S \in \mathsf{S}$, few z-walks from S_1 to S_2 contain S. Here we say that a z-walk $(T_1 = S_1, \dots, T_z = S_2)$ from S_1 to S_2 *contains* S if $S \in \{T_2, \dots, T_z\}$. In other words, each

z-walk from S_1 to S_2 is not disjoint with only few other z-walks from S_1 to S_2. This allows us to show that, if there are too many z-walks, then there must also be many disjoint z-walks as well, which would violate Claim 18. A formal proof of Lemma 17 based on this intuition is given below.

Proof of Lemma 17. For every integer i such that $2 \leqslant i \leqslant \ell$, let $P(i)$ denote the following: for every $S_1, S_2 \in \mathsf{S}$, $|\widehat{\mathcal{W}}_i^{G^{\sigma,\zeta}}(S_1, S_2)| \leqslant (ri)^{2(i-1)}$. For convenient, let $B_i = (ri)^{2(i-1)}$.

$P(2)$ follows from Claim 18. Now, suppose that, for some integer z such that $3 \leqslant z \leqslant \ell$, $P(2), \ldots, P(z-1)$ are true. To prove $P(z)$, let us first show that, for any fixed starting and ending vertices, any vertex cannot appear in too many red-filled z-walks:

▶ **Claim 19.** *For every $S_1, S_2, S \in \mathsf{S}$, the number of red-filled z-walks from S_1 to S_2 containing S in $G^{\sigma,\zeta}$ is at most $B_z/(zr)$.*

Proof. First, observe that the number of red-filled z-walks from S_1 to S_2 containing S is at most the sum over all positions $2 \leqslant j \leqslant z$ of the number of z-walks from S_1 to S_2 such that the j-th vertex in the walk is S, i.e., $\sum_{j=2}^{z} |\{(T_1, \ldots, T_{z+1}) \in \widehat{\mathcal{W}}_z^{G^{\sigma,\zeta}}(S_1, S_2) \mid T_j = S\}|$.

Now, for each $2 \leqslant j \leqslant z$, to bound the number of red-filled z-walks from S_1 to S_2 whose j-th vertex is S, let us consider the following three cases based on the value of j:

1. $3 \leqslant j \leqslant z - 1$. Observe that, for any such walk $(T_1 = S_1, T_2, \ldots, T_j = S, \ldots, T_z, T_{z+1} = S_2)$, the subwalk $(T_1 = S_1, \ldots, T_j = S)$ and $(T_j = S, \ldots, T_{z+1} = S_2)$ must be red-filled walks as well. Since the numbers of red-filled $(j-1)$-walks from S_1 to S and red-filled $(z+1-j)$-walks from S to S_2 are bounded by B_{j-1} and B_{z+1-j} respectively (from the inductive hypothesis), there are at most B_{j-1} choices of $(T_1 = S_1, \ldots, T_j = S)$ and B_{z+1-j} choices of $(T_j = S, \ldots, T_{z-1}, T_z = S_2)$. Hence, there are at most $B_{j-1}B_{z+1-j}$ red-filled z-walks from S_1 to S_2 whose j-th vertex is S.
2. $j = 2$. In this case, the subwalk $(T_2, \ldots, T_{z+1})$ must be a red-filled $(z-1)$-walk from S to S_2. Hence, the number of red-filled z-walks from S_1 to S_2 where $T_j = S$ is at most B_{z-1}.
3. $j = z$. Similar to the previous case, we also have the bound of B_{z-1}.

Summing the above bounds over all j's, the number of red-filled z-walks from S_1 to S_2 containing S is at most $\sum_{j=2}^{z} B_{j-1}B_{z+1-j} \leqslant \sum_{j=2}^{z}(rz)^{2(z-2)} \leqslant B_z/(zr)$ as desired. ◀

Having proved the above claim, it is now easy to show that $P(z)$ is true. Suppose for the sake of contradiction that there exists $S_1, S_2 \in \mathsf{S}$ such that $|\widehat{\mathcal{W}}_z^{G^{\sigma,\zeta}}(S_1, S_2)| > B_z$. Consider the following procedure of selecting disjoint walks from $\widehat{\mathcal{W}}_z^{G^{\sigma,\zeta}}(S_1, S_2)$. First, initialize $U = \widehat{\mathcal{W}}_z^{G^{\sigma,\zeta}}(S_1, S_2)$ and repeat the following process as long as $U \neq \emptyset$: select any $(T_1, \ldots, T_{z+1}) \in U$ and remove every $(T'_1, \ldots, T'_{z+1})$ that is not disjoint with $(T_1, \ldots, T_{z+1})$ from U. Observe that, each time a walk $(T_1, \ldots, T_{z+1})$ is selected, the number of walks removed from U is at most B_z/r; this is because each removed walk must contain at least one of $T_2, \ldots, T_z$, but, from the above claim, each of these vertices are contained in at most $B_z/(zr)$ walks. Since we start with more than B_z walks, at least r disjoint walks are picked, which, due to Claim 18, is a contradiction. Thus, $P(z)$ is true as desired. ◀

4.2 Finding Non-Red Biclique in Red/Blue-Transitive Graph

In the second step of our proof, we will show that any (q, ℓ)-red/blue transitive graph with sufficiently many edges must contain a sufficiently large non-red biclique, as stated below.

▶ **Lemma 20.** *For every $k, q, \ell, d \in \mathbb{N}$ such that $d \geqslant \max\{(2qk)^{1/\ell}, 2\ell^2\}$ and every k-vertex (q, ℓ)-red/blue-transitive graph $G = (V, E_r \cup E_b)$ such that $|E_b| \geqslant 2kd$, there exist $V_1, V_2 \subseteq V$ each of size at least $\log d / \log(4\ell^2) - 1$ such that, for every $u \in V_1$ and $v \in V_2$, $\{u, v\} \notin E_r$.*

As stated in the outline, we prove Lemma 20 by first finding subsets of vertices $U_1, U_2 \subseteq V$ such that for $1/\ell^2$ fraction of $(u_1, u_2) \in U_1 \times U_2$, u_1 and u_2 are not joined by a red edge and then use the Kővári-Sós-Turán Theorem to find the desired non-red biclique. Specifically, to prove Lemma 20, we show the following:

▶ **Lemma 21.** *For every $k, q, \ell, d \in \mathbb{N}$ such that $d \geqslant (2qk)^{1/\ell}$ and every k-vertex (q, ℓ)-red/blue-transitive graph $G = (V, E_r \cup E_b)$ such that $|E_b| \geqslant 2kd$, there exists subsets of vertices $U_1, U_2 \subseteq V$ each of size at least d such that $|\{(u, v) \in U_1 \times U_2 \mid \{u, v\} \notin E_r\}| \geqslant |U_1||U_2|/\ell^2$.*

The Kővári-Sós-Turán Theorem can be stated as follows.

▶ **Theorem 22** (Kővári-Sós-Turán (KST) Theorem [32]). *For every $t, M, N \in \mathbb{N}$ such that $t \leqslant \min\{M, N\}$, any $K_{t,t}$-free bipartite graph with N vertices one side and M vertices on the other contain at most $(t-1)^{1/t}(N - t + 1)M^{1-1/t} + (t-1)M$ edges.*

A simple calculation shows that Lemma 20 follows from Lemma 21 and the KST Theorem. We now move on to the proof of Lemma 21, which is exactly as sketched earlier in Subsection 2.

Proof of Lemma 21. We start by preprocessing the graph so that every vertex has blue-degree at least d. In particular, as long as there exists a vertex v whose blue-degree is at most d, we remove v from G. Let $G' = (V', E'_r \cup E'_b)$ be the graph at the end of this process. Note that we remove less than kd blue edges in total. Since at the beginning $|E_b| \geqslant 2kd$, we have $|E'_b| \geqslant kd$. Observe also that G' remains (q, ℓ)-red/blue-transitive.

Since V' is (q, ℓ)-red/blue-transitive, for every $u, v \in V'$, there can be at most q red-filled ℓ-walks from u to v. Summing this up over all pairs (u, v)'s implies that the number of red-filled ℓ-walk in G' is at most qk^2.

Moreover, notice that $|\mathcal{W}_\ell^{G'}| \geqslant (kd) \cdot d^{\ell-1} \geqslant 2qk^2$; this is because there are at least kd choices for (v_1, v_2) (i.e. all blue edges) and, for $(v_1, \dots, v_{i-1})$, there are at least d choices for v_i.

Hence, we have $|\widehat{\mathcal{W}}_\ell^{G'}|/|\mathcal{W}_\ell^{G'}| \leqslant 1/2$. This implies that $1/2 \leqslant \Pr_{(v_1,\dots,v_{\ell+1}) \in \mathcal{W}_\ell^{G'}}[(v_1, \dots, v_{\ell+1}) \notin \widehat{\mathcal{W}}_\ell^{G'}]$. By union bound, this probability is at most $\sum_{\substack{i,j \in [\ell+1] \\ j > i+1}} \Pr_{(v_1,\dots,v_{\ell+1}) \in \mathcal{W}_\ell^{G'}}[\{v_i, v_j\} \notin E'_r]$.

Now, note that the number of pairs of $i, j \in [\ell + 1]$ such that $j > i + 1$ is $\binom{\ell+1}{2} - \ell \leqslant \ell^2/2$. This implies that there exists one such i, j such that $\Pr_{(v_1,\dots,v_{\ell+1}) \in \mathcal{W}_\ell^{G'}}[\{v_i, v_j\} \notin E'_r] \geqslant 1/\ell^2$. Observe that the probability $\Pr_{(v_1,\dots,v_{\ell+1}) \in \mathcal{W}_\ell^{G'}}[\{v_i, v_j\} \notin E'_r]$ is bounded above by

$$\max_{u,v} \Pr_{(v_1,\dots,v_{\ell+1}) \in \mathcal{W}_\ell^{G'}}[\{v_i, v_j\} \notin E'_r \mid v_{i+1} = u \wedge v_{j-1} = v]$$

where the maximization is taken over all $u, v \in V'$ such that $\Pr_{(v_1,\dots,v_{\ell+1}) \in \mathcal{W}_\ell^{G'}}[v_{i+1} = u \wedge v_{j-1} = v] > 0$. Hence, we can conclude that there exists $u^*, v^* \in V'$ such that

$$\Pr_{(v_1,\dots,v_{\ell+1}) \in \mathcal{W}_\ell^{G'}}[\{v_i, v_j\} \notin E'_r \mid v_{i+1} = u^* \wedge v_{j-1} = v^*] \geqslant 1/\ell^2.$$

The expression on the left is exactly $|\{(u, v) \in N_b(u^*) \times N_b(v^*) \mid \{u, v\} \notin E'_r\}|/(|N_b(u^*)| \cdot |N_b(v^*)|)$. From this and from every vertex in G' has blue-degree at least d, $U_1 = N_b(u^*), U_2 = N_b(v^*)$ are the desired sets. ◀

4.3 Decoding a Good Assignment From Non-Red Biclique

Finally, we will decode a good assignment for Φ from a sufficiently large non-red biclique in the consistency graph $G^{\sigma,\zeta}$. Recall that a non-red biclique in $G^{\sigma,\zeta}$ simply corresponds to two subcollections $\mathsf{S}_1, \mathsf{S}_2$ such that, for every $(S_1, S_2) \in \mathsf{S}_1 \times \mathsf{S}_2$, $\text{disagr}(\sigma_{S_1}, \sigma_{S_2}) \leqslant \zeta n$. The main result of this subsection is that, given such $\mathsf{S}_1, \mathsf{S}_2$, if both S_1 and S_2 are sufficiently uniform, then we can find a good assignment for Φ. This is stated more precisely below.

▶ **Lemma 23.** *Let $\mathsf{S}_1, \mathsf{S}_2$ be any (γ, μ)-uniform collections of subsets of $\mathcal{C}$. If there is a labeling σ of $\mathsf{S}_1 \cup \mathsf{S}_2$ such that $\text{disagr}(\sigma_{S_1}, \sigma_{S_2}) \leqslant \zeta n$ for every $S_1 \in \mathsf{S}_1$ and $S_2 \in \mathsf{S}_2$, then $\text{val}(\Phi) \geqslant 1 - 2\mu - 6\Delta\zeta/\gamma^2$.*

As outlined earlier, the assignment we take is the majority assignment ψ_{maj} of $\{\sigma_{S_1}\}_{S_1 \in \mathsf{S}_1}$. The key to proving that ψ_{maj} violates few clauses is that, if a clause C is violated, then, for each $S_2 \in \mathsf{S}_2$ that contains C, σ_{S_2} and ψ_{maj} must disagree on at least one variable in $\text{var}(C)$ because σ_{S_2} satisfies C but ψ_{maj} violates it. Hence, if C appears often in both S_1 and S_2, then it contributes to many disagreements between S_1 and S_2; the uniformity condition help us ensure that most C indeed appears often in S_1 and S_2. On the other hand, $\text{disagr}(\sigma_{S_1}, \sigma_{S_2})$ is small for every $S_1 \in \mathsf{S}_1$ and $S_2 \in \mathsf{S}_2$, meaning that there cannot be too many disagreements in total. Comparing this upper and lower bound gives us the desired result. Due to space constraint, we omit the full analysis from this version of the paper.

5 Inapproximability Results of 2-CSPs and DSN

The inapproximability results for 2-CSPs can be shown simply by plugging in the appropriate parameters to Theorem 16. More specifically, for ETH-hardness, since there is a $\text{polylog}m$ loss in the PCP Theorem (Theorem 6), we need to select our $\alpha = 1/\text{polylog}m$ so that the size (and running time) of the reduction is $2^{o(m)}$. Recall in Lemma 15 that we need $m \geqslant \Omega(\alpha^\ell)$, meaning that ℓ can be at most $O(\log m / \log\log m)$. We will pick ℓ to be just $\sqrt{\log m}$. We will finally pick k to be $\exp(\ell \log \ell/\alpha) = \exp(\text{polylog}m)$; this is so that the non-edge biclique size $\log k/(\ell \log(4\ell^2))$ (from Theorem 16) is large enough that we can use Lemma 15 to guarantee its uniformity. Other parameters are chosen accordingly. We omit the full proof, which consists almost solely of calculations, from this version of the paper.

For the inapproximability based on Gap-ETH, we do not incur a loss of $\text{polylog}m$ from the PCP Theorem anymore. Thus, we can choose α to be any function that converges to zero as k goes to infinity, e.g., $\alpha = 1/\log\log k$. Now note that the parameter r in Theorem 16 for the intersection disperser property grows with $(1/\alpha)^\ell$ (see Lemma 15). Since the soundness guarantee in Theorem 16 is of the form $k^{O(1/\ell)}(r\ell)^{O(1)}/k = k^{O(1/\ell)}(1/\alpha)^{O(\ell)}/k$, it is minimized when ℓ is roughly $\sqrt{\log k}$, which yields the bound $2^{(\log k)^{1/2+o(1)}}/k$.

The inapproximability results for DSN can be proved by simply plugging the hardness of approximation of 2-CSPs to the following known reduction from 2-CSPs to DSN.

▶ **Lemma 24** ([16, Lemma 27][6]). *There is a polynomial time reduction that, given a 2-CSP instance Γ where the constraint graph is a complete graph on k variables, produces an edge-weighted directed graph G and a set of demands $\mathcal{D} = \{(s_1, t_1), \dots, (s_{k^2-k}, t_{k^2-k})\}$ s.t.*

- *If $\text{val}(\Gamma) = 1$, then there exists a subgraph H of cost 1 that satisfies all demands.*
- *If $\text{val}(\Gamma) < \gamma$, then every subgraph satisfying all demand pairs has cost more than $\sqrt{2/\gamma}$.*

[6] While the reduction is attributed to Dodis and Khanna [25], the lemma below is extracted from [16] since, in [25], the full description of the reduction and its properties are left out due to space constraint.

6 Conclusion and Discussions

We prove ETH-hardness of approximating 2-CSPs within $k^{1-o(1)}$ factor where k denotes the number of variables. This ratio is nearly tight since a trivial algorithm yields an $O(k)$-approximation. Under Gap-ETH, we strengthen our result to rule out not only polynomial time but also FPT time algorithms parameterized by k. Due to a known reduction, our result implies $k^{1/4-o(1)}$ factor inapproximability of DSN where k is the number of demand pairs.

Of course the polynomial SSC still remains open and resolving it will advance our understanding of approximability of many problems. Even without fully resolving the conjecture, it may still be good to further study the interaction between the number of variables k and the alphabet size n. For instance, while we show the inapproximability result with ratio $k^{1-o(1)}$, the dependency between n and k is quite bad; in our ETH-hardness reduction, n is $2^{2^{(\log k)^{\Theta(1)}}}$. Would it be possible to improve this dependency (say, to $n = k^{\text{polylog} k}$)?

Another interesting direction is to try to prove similar hardness results as ours for other problems. For example, Densest k-Subgraph (DkS) is one such candidate problem; similar to 2-CSPs with k variables, the problem can be approximated trivially to within $O(k)$-factor and no polynomial (or even FPT) time $k^{1-\varepsilon}$-approximation algorithm is known for the problem. Hence, it may also be possible to prove ETH-hardness of factor $k^{1-o(1)}$ for DkS.

Acknowledgments. We would like to thank Prahladh Harsha for useful discussions. Pasin would also like to thank Rajesh Chitnis and Andreas Emil Feldmann for insightful discussions regarding DSN.

References

1 Amir Abboud and Greg Bodwin. Reachability preservers: New extremal bounds and approximation algorithms. *CoRR*, abs/1710.11250, 2017. `arXiv:1710.11250`.

2 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, may 1998.

3 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

4 Sanjeev Arora and Madhu Sudan. Improved low degree testing and its applications. In *STOC*, pages 485–495, 1997.

5 Mihir Bellare, Shafi Goldwasser, Carsten Lund, and Alexander Russell. Efficient probabilistically checkable proofs and applications to approximations. In *STOC*, pages 294–304, 1993.

6 Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust PCPs of proximity, shorter PCPs, and applications to coding. *SIAM J. Comput.*, 36(4):889–974, 2006.

7 Eli Ben-Sasson, Yohay Kaplan, Swastik Kopparty, Or Meir, and Henning Stichtenoth. Constant rate PCPs for Circuit-SAT with sublinear query complexity. *J. ACM*, 63(4):32:1–32:57, 2016.

8 Eli Ben-Sasson and Madhu Sudan. Short PCPs with polylog query complexity. *SIAM J. Comput.*, 38(2):551–607, 2008.

9 Piotr Berman, Arnab Bhattacharyya, Konstantin Makarychev, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Approximation algorithms for spanner problems and directed steiner forest. *Inf. Comput.*, 222:93–107, 2013.

10 Edouard Bonnet, Bruno Escoffier, Eun Jung Kim, and Vangelis Th. Paschos. On subexponential and FPT-time inapproximability. *Algorithmica*, 71(3):541–565, 2015.

11 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From Gap-ETH to FPT-inapproximability: Clique, Dominating Set, and more. In *FOCS*, pages 743–754, 2017.

12 Siu On Chan. Approximation resistance from pairwise-independent subgroups. *J. ACM*, 63(3):27:1–27:32, 2016.

13 Moses Charikar, Chandra Chekuri, To-Yat Cheung, Zuo Dai, Ashish Goel, Sudipto Guha, and Ming Li. Approximation algorithms for directed steiner problems. *J. Algorithms*, 33(1):73–91, 1999.

14 Moses Charikar, MohammadTaghi Hajiaghayi, and Howard J. Karloff. Improved approximation algorithms for label cover problems. *Algorithmica*, 61(1):190–206, 2011.

15 Chandra Chekuri, Guy Even, Anupam Gupta, and Danny Segev. Set connectivity problems in undirected graphs and the directed steiner network problem. *ACM Trans. Algorithms*, 7(2):18:1–18:17, 2011.

16 Rajesh Chitnis, Andreas Emil Feldmann, and Pasin Manurangsi. Parameterized approximation algorithms for directed steiner network problems. *CoRR*, abs/1707.06499, 2017.

17 Eden Chlamtác, Michael Dinitz, Guy Kortsarz, and Bundit Laekhanukit. Approximating spanners and directed steiner forest: Upper and lower bounds. In *SODA*, pages 534–553, 2017.

18 Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007.

19 Irit Dinur. Mildly exponential reduction from gap 3SAT to polynomial-gap label-cover. *ECCC*, 23:128, 2016.

20 Irit Dinur, Eldar Fischer, Guy Kindler, Ran Raz, and Shmuel Safra. PCP characterizations of NP: Toward a polynomially-small error-probability. *Computational Complexity*, 20(3):413–504, 2011.

21 Irit Dinur, Prahladh Harsha, and Guy Kindler. Polynomially low error PCPs with polyloglog n queries via modular composition. In *STOC*, pages 267–276, 2015.

22 Irit Dinur and Tali Kaufman. High dimensional expanders imply agreement expanders. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 974–985. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.94.

23 Irit Dinur and Inbal Livni Navon. Exponentially small soundness for the direct product z-test. In *CCC*, pages 29:1–29:50, 2017.

24 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *STOC*, pages 624–633, 2014.

25 Yevgeniy Dodis and Sanjeev Khanna. Design networks with bounded pairwise distance. In *STOC*, pages 750–759, 1999.

26 Moran Feldman, Guy Kortsarz, and Zeev Nutov. Improved approximation algorithms for directed steiner forest. *J. Comput. Syst. Sci.*, 78(1):279–292, 2012.

27 Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

28 Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. New direct-product testers and 2-query pcps. *SIAM J. Comput.*, 41(6):1722–1768, 2012. doi:10.1137/09077299X.

29 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

30 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

31 Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.*, 37(1):319–357, 2007.

32 Tamás Kővári, Vera T. Sós, and Pál Turán. On a problem of K. Zarankiewicz. *Colloquium Mathematicae*, 3(1):50–57, 1954.

33 Pasin Manurangsi. Almost-polynomial ratio ETH-hardness of approximating densest k-subgraph. In *STOC*, pages 954–961, 2017.

34 Pasin Manurangsi and Prasad Raghavendra. A birthday repetition theorem and complexity of approximating dense CSPs. *CoRR*, abs/1607.02986, 2016.

35 Dana Moshkovitz. Low-degree test with polynomially small error. *Computational Complexity*, 26(3):531–582, 2017.

36 Dana Moshkovitz and Ran Raz. Sub-constant error probabilistically checkable proof of almost-linear size. *Computational Complexity*, 19(3):367–422, 2010.

37 Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *STOC*, pages 245–254, 2008.

38 Ran Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998.

39 Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *STOC*, pages 475–484, 1997.

40 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.