

UNIVERSAL ABELIAN EMBEDDING AND AFFINE RELAXATION FOR CSP

SIU ON CHAN* AND HIU TSUN NG†

ABSTRACT. Bhangale, Khot, and Minzer (TheoretCS 2026) proposed a hybrid approximation algorithm for certain satisfiable CSPs to exploit their abelian embeddings. We propose BasicSDP $\wedge$ CSinglAIP to simplify and generalize their hybrid algorithm. We do so by introducing two concepts: (1) universal embedding, which simplifies and generalizes their master embedding, and (2) affine relaxation in an abelian group.

We prove that our algorithm is not simulated by SDP+AIP, providing the first linear-level SDP+AIP lower bound for a tractable CSP. We also prove the first lower bounds for their hybrid algorithm, by turning our algorithm into a hierarchy and proving linear-level lower bounds, while achieving the optimal gap for large-arity CSPs.

A deeper understanding of our algorithms' power may lead to progress on uniform CSP algorithms and the optimal approximability of satisfiable CSPs.

1. INTRODUCTION

The Dichotomy Theorem for Constraint Satisfaction Problems (CSPs), proved independently by Bulatov [Bul17] and Zhuk [Zhu20], has motivated new open problems and lines of research. We focus on two of these problems.

One central open problem is the existence of uniform CSP algorithms (e.g. [Zhu20, Problem 1], [Zhu26, BW26, BHZ26]). It asks for an algorithm that, given both a template $\mathcal{R}$ and a CSP($\mathcal{R}$) instance I as input, and with the promise that CSP($\mathcal{R}$) is in P, decides in polynomial time whether I has a satisfying assignment. Note that the template $\mathcal{R}$ is also part of the input; each relation in $\mathcal{R}$ is given as a list of tuples. Neither Bulatov's nor Zhuk's algorithm is uniform, as they both take super-polynomial time in the size of $\mathcal{R}$.

Another challenging open problem is the optimal approximability of satisfiable CSPs (e.g. [Bha25, Questions 1 to 3], [BKM26, Conjecture 1.1]), assuming the Rich 2-to-1 Games Conjecture [BKM21]. The goal is to extend Raghavendra's algorithm and its inapproximability results for (unsatisfiable) CSPs [Rag08] to satisfiable CSPs. [Rag08] cannot handle satisfiable CSPs due to the use of "uniform noise" in both the hardness reduction and the algorithm.

So far, these two lines of work have often (though not always) used disjoint techniques: the former using universal algebra, and the latter Fourier analysis. Their high-level intersection lies in the fact that, since both must solve tractable CSPs as a special case, they both combine two families of algorithmic techniques: (1) local consistency and its stronger variants, such as semidefinite programming (SDP), which is indispensable in approximation algorithms; and (2) Gaussian elimination and its generalizations, to solve linear equations over abelian groups.

1.1. Hybrid Algorithm. To design approximation algorithms for satisfiable CSPs, the latter line looks for specific solutions to "abelian embedding" equations, a technique pioneered in the "hybrid algorithm" of Bhangale, Khot, and Minzer [BKM26].

Abelian embedding is a concept central to the latter line of work:

*sochan@gmail.com.

†htng0@link.cuhk.edu.hk.

Definition 1.1 (Abelian Embedding). A k -tuple of maps $\sigma_i : D \rightarrow G$ from a domain D to an abelian group G is an embedding of a relation $R \subseteq D^k$ if there exists $g \in G$ such that for every $a \in R$,

$$(1) \quad \sum_{i \in [k]} \sigma_i(a_i) = g.$$

Denote such an embedding by (σ, g) or (G, σ, g) , where $\sigma \stackrel{\text{def}}{=} (\sigma_i)_{i \in [k]}$. An embedding is trivial if σ_i are all constant functions.

To see why this concept is useful, consider the following CSPs.

Example 1.2. The 3-XOR odd-parity constraint $R \stackrel{\text{def}}{=} \{w \in \{0, 1\}^3 \mid w(1) + w(2) + w(3) \equiv 1 \pmod{2}\}$ has an embedding to $\mathbb{Z}_2$ with $\sigma_i(0) = 0_{\mathbb{Z}_2}$ and $\sigma_i(1) = 1_{\mathbb{Z}_2}$, and $g = 1_{\mathbb{Z}_2}$. This embedding is the natural (or identity) embedding of a 3-XOR constraint to $\mathbb{Z}_2$. As another example, the monotone 1-in-3-SAT constraint $R' \stackrel{\text{def}}{=} \{w \in \{0, 1\}^3 \mid w(1) + w(2) + w(3) = 1\}$ also has the same embedding to $\mathbb{Z}_2$, since $R' \subseteq R$.

Using abelian embeddings, one can construct a system of abelian linear equations that provides a one-sided test for whether a CSP instance is satisfiable. For 3-XOR (with both odd and even parity constraints), the system is the traditional system of equations over $\mathbb{Z}_2$. For (non-monotone) 1-in-3 SAT, one can again consider the corresponding equations over $\mathbb{Z}_2$ to reject some (but not all) unsatisfiable instances. Even though the actual equations in [BKM26] are much more complicated, these simple examples illustrate why abelian embeddings may be useful. Outside of approximation of satisfiable CSPs, abelian embeddings have also inspired related results in additive combinatorics [BKLM25] and hardness of approximation [BBK⁺26].

The hybrid algorithm [BKM26, Algorithm 4] works only for certain 3-CSPs and aborts unless certain conditions are met. In particular, it aborts if any CSP constraint non-trivially embeds to $\mathbb{Z}$. It is interesting to go beyond this barrier and generalize the hybrid algorithm further. Bhanagle [Bha25, Section 5.1.2] mentioned their hybrid algorithm “can be viewed as a general algorithm that captures the abelian embedding property of the predicates when designing approximation algorithms.” [Bha25, Question 5] also asked:

Question 1.3. How powerful is the hybrid algorithm [...] towards approximating Max- P -CSPs on satisfiable instances in general?

1.2. Uniform CSP Algorithms. Meanwhile, the pursuit of uniform CSP algorithms has a vastly different focus in solution approaches. While a major innovation in [BKM26]’s hybrid algorithm is the ingredient — the Gaussian elimination subroutine defined using abelian embeddings, this is usually not the case for the former line.

To design uniform CSP algorithms, the ingredients in the solutions proposed are often simply the four standard relaxations: (1) arc consistency; (2) linear programming (LP); (3) semidefinite programming (SDP); and (4) affine integer programming (AIP). When these “basic” relaxations are not strong enough, there is also a standard, systematic technique to strengthen these relaxations into a progressively stronger “hierarchy” of relaxations, by enforcing constraints on small subsets of variables. The hierarchy version of arc consistency is known as local consistency or bounded width (BW). BW, LP, and SDP may be seen as increasingly stronger versions of local consistency, while AIP mimics Gaussian elimination: it solves linear equations over the integers $\mathbb{Z}$ instead of a field.

The main focus, instead, is to cleverly combine these two families of relaxation algorithms in hopes of solving all tractable CSPs. Many different combination approaches have been proposed in the literature. For example, in terms of LP and AIP,

- Simply run both algorithms independently. This is known as the $\text{LP} \wedge \text{AIP}$ algorithm [Zhu26].

- Run the LP algorithm first. Then, augment the AIP algorithm by banning all partial assignments that are not used in any solution of the former algorithm. This is known as the LP + AIP algorithm. [BGWŽ20] introduced LP + AIP. The SDP + AIP version was proposed concurrently by [CŽ25] (known as “SDA”) and [CNP24] (as a hierarchy).
- Iteratively run the above algorithms. Each time, we enforce some additional restriction onto the relaxations, which is a one-sided test to check whether a partial assignment is a part of some satisfying assignment. We ban the partial assignments that fail the test.

The most common test is to simply force the partial assignment onto a constraint or a subset of variables. This is known in the literature under many different names, such as CLAP [CŽ22], C(BLP+AIP) [CŽ22], and cohomological consistency (hierarchy) [ÓC22]. We follow the notations in [Zhu26] and refer to this approach using CSingl or Singl (respectively). The algorithms mentioned above are equivalent to CSinglLP + AIP, CSingl(LP $\wedge$ AIP), and Singl(BW $\wedge$ AIP) (hierarchy) respectively.

More generally, without being specific on the test itself, we shall simply refer to this approach as an iterative algorithm. This iterative approach is used in the hybrid algorithm in [BKM26].

1.3. Our Results. We study the power of the generalized hybrid algorithm for satisfiable CSPs *in general*, beyond the technical restrictions of [BKM26]. We also substantially simplify the algorithm, using techniques from uniform CSP algorithms. We propose a new algorithm, (iterative) BasicSDP $\wedge$ CSinglAIP¹, which combines basic SDP with the constraint-singleton variant of AIP.

Theorem 1.4 (Section 3.6). BasicSDP $\wedge$ CSinglAIP *subsumes the* [BKM26] *hybrid algorithm.*

In this paper, we only consider decision algorithms $\mathcal{A}$ that accept every satisfiable CSP instance, but may erroneously accept unsatisfiable instances. Algorithm $\mathcal{A}$ subsumes algorithm $\mathcal{B}$ if $\mathcal{A}$ rejects every instance that $\mathcal{B}$ rejects. Algorithm $\mathcal{A}$ solves some CSP if $\mathcal{A}$ rejects every unsatisfiable instance.

BasicSDP $\wedge$ CSinglAIP has a number of advantages over the hybrid algorithm. First, it applies immediately to all CSPs, whereas the hybrid algorithm is restricted to certain CSPs mentioned above. In particular, substantial changes to the hybrid algorithm are required to deal with CSPs that non-trivially embed to $\mathbb{Z}$. Second, BasicSDP $\wedge$ CSinglAIP is a candidate uniform CSP algorithm, subject to the caveat about SDP in Theorem 1.5 (the same caveat also applies to the hybrid algorithm). The hybrid algorithm is not a polynomial-time uniform algorithm, as it requires super-polynomial time in the size of the template $\mathcal{R}$ to search for certain abelian embeddings, known as “master embeddings”, and it also aborts under certain conditions [BKM26, Algorithm 4].

Remark 1.5. It is currently unknown how to decide in polynomial time whether the basic SDP relaxation of a CSP instance has an exact solution. Following [BKM26], we will ignore this issue in this paper. When we consider the SDP hierarchy, we will further assume there is a $(n|D|)^{O(d)}$ -time algorithm that decides whether a CSP $(D, \mathcal{R})$ instance on n variables has an exact SDP hierarchy solution of level d . The existence of such algorithms is left as open questions for future work.

BasicSDP $\wedge$ CSinglAIP strengthens the basic SDP+AIP relaxation proposed in [CŽ25] (“SDA”) and its hierarchy proposed concurrently in [CNP24]. We further show that our generalized hybrid algorithm is strictly stronger than these algorithms, because they fail to solve mod-2 and mod-3 equations on disjoint subdomains, known as CSP($\mathbb{Z}_2 \cup \mathbb{Z}_3$) [Zhu26]:

Theorem 1.6 (Theorems 3.8, 3.11, 4.5 and 5.20). BasicSDP $\wedge$ CSinglAIP *solves* CSP($\mathbb{Z}_2 \cup \mathbb{Z}_3$), *but neither SDA nor level- $\Omega(n)$ SDP + AIP hierarchy solves* CSP($\mathbb{Z}_2 \cup \mathbb{Z}_3$), *where n is the number of variables of an instance.*

¹Note that this algorithm is iterative. We will always consider the iterative version, and henceforth omit the adjective “iterative” when referring to this algorithm.

During Bhangale’s talk in a Dagstuhl CSP seminar in May 2025, it was asked if the [BKM26] hybrid algorithm can be implemented using SDP+AIP. Theorems 1.4 and 1.6 rule out this possibility, at least for our generalization of the hybrid algorithm. As we will see, SDP+AIP fails because AIP can only check whether the abelian equations have *any solution*. However, for the hybrid algorithm to solve $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$, it must check whether the equations have *specific solutions corresponding to partial assignments of the CSP instance*. The latter task needs CSinglAIP.

Theorem 1.6 is also interesting in the context of uniform CSP algorithms, because $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$ is tractable. Numerous candidate uniform CSP algorithms have been considered [BGWŻ20, ČŽ22, DO24, ÓC22, ČŽ25, CNP24], and many have been ruled out as candidates [LP25, Zhu26]. Our Theorem 1.6 further rules out SDA and its SDP+AIP hierarchy, resolving a remark in [CGP26] about SDA’s candidacy. Theorem 1.6 provides the first SDA and SDP+AIP lower bounds for a tractable CSP; no CSPs in previous such lower bounds [CNP24, ČŽ25, CdX26] are known to be tractable.

Remark 1.7. AIP is not the only generalization of Gaussian elimination. Polynomial calculus is another, and popular in proof complexity. Conneryd, Ghannane, and Pang [CGP26] showed that the standard lower bound technique for polynomial calculus implies lower bounds to cohomological consistency. However, to efficiently certify mod- p equations to have no solutions, one needs to use polynomial calculus proof over a field of characteristic p [AR01]. The need for a matching characteristic makes polynomial calculus less convenient to use than AIP as a uniform algorithm.

The connection between the hybrid algorithm and $\text{BasicSDP} \wedge \text{CSinglAIP}$ in Theorem 1.4 has another application: translate lower bounds of the latter to those of the former. We will prove $\text{BasicSDP} \wedge \text{CSinglAIP}$ (and hence hybrid algorithm) lower bounds for CSPs that are “pairwise independent” [CdX26] and “lax” [CN25], as defined in Section 6. In fact we prove lower bounds not just for $\text{BasicSDP} \wedge \text{CSinglAIP}$, but also for its hierarchy version introduced in this paper: $\text{Singl}(\text{SDP} \wedge \text{AIP})$.

Below $\text{opt}(I)$ denotes the maximum fraction of constraints satisfied by an assignment to I . Also, given a distribution ρ over k -ary relations on D , let

$$\text{opt}_k(\rho) \stackrel{\text{def}}{=} \max_{\nu \in \Delta(D)} \mathbb{E}_{R \sim \rho} \mathbb{P}_{a \sim \nu^k} [a \in R]$$

be the maximum expected fraction of constraints satisfied by i.i.d. distributions, when the constraints are drawn from the distribution ρ .

Theorem 1.8 (Theorem 6.4). *Suppose a k -CSP $(D, \mathcal{R})$ is pairwise ν -independent and lax for some distribution ν with $\text{supp}(\nu) = D$. For $\varepsilon > 0$ and sufficiently large n , some instance I of the CSP on n variables has a $\text{Singl}(\text{SDP} \wedge \text{AIP})$ hierarchy solution of level $\Omega(n)$ and $\text{opt}(I) \leq \min_{\rho \in \Delta(\mathcal{R})} \text{opt}_k(\rho) + \varepsilon$.*

The level lower bound $\Omega(n)$ is optimal up to a constant factor. Theorem 1.8 strengthens previous $\text{C}(\text{BLP} + \text{AIP})$ lower bounds [CdX26, Theorems 7.15]. [Bha25] remarked “[i]t would be interesting to show the limitation of the hybrid approximation algorithm”, and “[i]t would also be interesting to define the hierarchies of this hybrid algorithm and understand their limitations”. Theorem 1.8 is our answer to these remarks, and forms the first lower bounds to the [BKM26] hybrid algorithm as well as its hierarchy version. As explained in Section 1.4, Theorem 1.8 provides evidence, in the form of an unconditional $\text{Singl}(\text{SDP} \wedge \text{AIP})$ gap, that large-arity k -CSP with domain D may be hard to approximate beyond $O(|D|k/|D|^k)$, matching the approximation ratio of [MM14] when $k \geq \Omega(\log|D|)$.

The $\text{BasicSDP} \wedge \text{CSinglAIP}$ algorithm and $\text{Singl}(\text{SDP} \wedge \text{AIP})$ hierarchy are currently candidate algorithms for both tractable CSPs and the Approximation Dichotomy Conjecture [BKM26, Conjecture 1.1]. Understanding their power may lead to simultaneous progress on both lines of work. An obvious open problem is to investigate whether $\text{BasicSDP} \wedge \text{CSinglAIP}$ solves the coset-CSP on the dihedral group of order 8 that fools $\text{C}(\text{BLP} + \text{AIP})$ introduced in [Zhu26].

[CŽ25] showed SDA lower bounds for approximate graph homomorphism (AGH), which may be seen as evidence for Brakensiek–Guruswami conjecture [BG21] on the NP-hardness of this problem. Now that SDA is known to not solve some tractable CSP (Theorem 5.20), better evidence is needed. To resurrect these SDA lower bounds for AGH, we upgrade them to $\text{BasicSDP} \wedge \text{CSinglAIP}$ lower bounds in Theorem 7.8.

1.4. Gap Instance, Dictator Test, and Inapproximability. A theme of [Rag08] is the equivalence between gap instances, dictator tests, and inapproximability for CSPs without perfect completeness. Let us discuss how our Theorem 1.8 fits in this picture for CSPs with perfect completeness, especially in the context of k -CSP [Bha25, Question 3].

Using the k -CSP from Theorem 6.5, Theorem 1.8 yields an unconditional linear-level $\text{Singl}(\text{SDP} \wedge \text{AIP})$ gap. This is evidence that satisfiable large-arity k -CSP may be hard to approximate beyond $O(|D|k/|D|^k)$. This matches the inapproximability ratio (assuming $P \neq NP$) of almost satisfiable k -CSP, up to a constant factor [Cha16]. It also coincides with the approximation ratio of any k -CSP when $k \geq \Omega(\log|D|)$ [MM14]. See also alternative evidence in the form of dictator tests with perfect completeness [TY15, BKT18]. The current best NP-hardness for satisfiable boolean k -CSP has an inapproximability ratio of $2^{\tilde{O}(k^{1/3})}/2^k$ [Hua14].

A goal of [BKM26] is to convert gap instances (with certain restrictions) of their hybrid algorithms to dictator tests with perfect completeness. Our Theorem 1.8 yields a rich source of such gap instances to their hybrid algorithm, and might become similar dictator tests in the future, when a less restrictive conversion is found.

A potential approach towards inapproximability of satisfiable CSPs is to compose dictator tests with Rich 2-to-1 Games [BKM21]. Inapproximability in turn yields gap instances, because hierarchy lower bounds for 3-SAT go through hardness reduction (see [CNP24, Section C] for the special case of $\text{SDP} + \text{AIP}$ hierarchy). However, this approach cannot lead to our linear-level lower bound in Theorem 1.8, even assuming Rich 2-to-1 Games Conjecture and Exponential Time Hypothesis (ETH): The sublinear-level SDP upper bounds for Unique Games [BRS11] (which built on [ABS15]) lead to the same upper bounds for Rich 2-to-1 Games, because the reduction from Rich 2-to-1 Games to Unique Games [BKM21] is 2-local [CNP24, Lemma C.1]. Eventually, we will be confronted with the approximability of the CSPs in Theorem 1.8 in the subexponential-time regime: Is there a hardness reduction without using Rich 2-to-1 Games that achieves fully exponential-time hardness (under ETH)? Or is there a subexponential-time algorithm beating linear-level $\text{Singl}(\text{SDP} \wedge \text{AIP})$?

1.5. Proof Overview. Our generalization and simplification of the [BKM26] hybrid algorithm is based on four simple ideas, discussed in the next four subsubsections.

1.5.1. Universal Embedding. Consider the mod-4 constraint. $R'' \stackrel{\text{def}}{=} \{w \in \{0, 1, 2, 3\}^3 \mid w(1) + w(2) + w(3) \equiv 0 \pmod{4}\}$. R'' has an embedding to $\mathbb{Z}_4$, but it also has a coarser embedding into $\mathbb{Z}_2$. Intuitively, embedding into $\mathbb{Z}_4$ is more precise than into $\mathbb{Z}_2$, which in turn is more precise than (the trivial embedding) into $\mathbb{Z}_1$. We should look for the “finest” embedding of R . Therefore the authors of [BKM26] define a single embedding, called *master embedding*, “that encapsulates within it all of the Abelian embeddings” of R [BKM24, Section 1.4.1]. Their definition of master embedding, as well as their hybrid algorithm, requires R to have no non-trivial embedding to $\mathbb{Z}$ (Section A).

In this work, we introduce universal embedding (Theorem 2.2), which generalizes and simplifies the master embedding of [BKM24, BKM26] (Section A). Our definition works regardless of whether R has any non-trivial embedding to $\mathbb{Z}$. While it is easy to come up with similar generalizations, our definition is optimized for ease of use when working with concrete relations R (i.e. specific CSPs). We will show that, for example, the universal embedding for 3-XOR is the natural embedding to $\mathbb{Z}_2$ (Section 2.2). We will also identify a relatively simple universal embedding for $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$, with a short proof (Section 2.3). The brevity of our proofs stems from, and signifies the usefulness of, the choice of our Theorem 2.2.

1.5.2. Affine Relaxation. The [BKM26] hybrid algorithm involves a system of abelian linear equations, called the GE System. Given a constraint C in an instance I , consider the master embedding (G_C, σ^C, g_C) of the set of satisfying assignments A_C of C . Let us call such G_C a master group for C . If the CSP is 3-XOR, then the master group is $\mathbb{Z}_2$, and the master embedding is the natural “identity” embedding (Theorem 2.9). In this GE System, each variable v of I should take a value $z(v)$ that, intuitively, belongs to the product group $G^v \stackrel{\text{def}}{=} \prod_{C \in \mathcal{C}, C \ni v} G_C$ of all “master groups” G_C over all constraints C containing v . $z(v, C)$ is meant to be v ’s opinion about its embedded value in the master group G_C .

Equations are added to the GE System, to ensure the values $\{z(v, C)\}_{v \in V(C)}$ across the k different variables $v \in V(C)$ of any constraint C satisfy the embedding equation in Eq. (1). But these are not the only equations. The GE System must also be consistent with the SDP (Theorem 3.2). To express this consistency requirement, the GE System is rewritten so that instead of variables $z(v)$, it has variables $y(v)$ taking values in the double dual group $\widehat{\widehat{G^v}}$. Since $\widehat{\widehat{G^v}}$ is naturally isomorphic to G^v via Pontryagin duality, the GE System is conceptually equivalent to a relaxation in terms of the hypothetical variables $z(v)$. But now the GE System becomes much more complicated and difficult to reason about [BKM26, Sections 7.2, 7.3].

We simplify the GE System by considering a relaxation involving affine hulls in an abelian group (Theorem 3.1). We give an affine relaxation (Eq. (7)) that is equivalent to the GE System whenever the hybrid algorithm applies. Our affine relaxation (together with our definition of universal embedding) readily generalizes to any CSP, even those non-trivially embeddable to $\mathbb{Z}$, a situation where the GE System and the hybrid algorithm break down.

1.5.3. Affine Integer Programming. It is not clear how to solve the affine relaxation Eq. (7) in polynomial time, especially for CSPs non-trivially embeddable to $\mathbb{Z}$ (and hence having infinite master/universal groups by Theorem 2.4). We bypass this issue altogether with the next idea.

In the GE System, if a variable v gives an honest opinion that its relaxed value $z(v)$ corresponds to some assignment $a \in D$, then $z(v, C)$ equals $\sigma_i^C(v)$ for every constraint C containing v , where (G_C, σ^C, g_C) is the universal embedding for the relation A_C on C , and i is the position of v in the scope of C . As such, the GE System depends on the choice of the embedding (G_C, σ^C, g_C) .

What if we modify Eq. (7), so that the honest encoding of a value $a \in D$ does not depend on any embedding? A natural choice is to encode $a \in D$ as the indicator vector $\mathbb{1}_a \in \mathbb{Z}^D$. This encoding is used in the most well-studied affine relaxation: affine integer program. We will also change the second constraint in Eq. (7) similarly, to remove any dependency on embedding. With these changes, we get yet another affine relaxation (Eq. (11)). Eq. (11) subsumes the original Eq. (7) (Theorem 3.9). It turns out Eq. (11) is equivalent to the basic AIP relaxation (Theorem 3.10), despite not being identical to it. Our reformulation of basic AIP as Eq. (11) may be of independent interest.

By transforming Eq. (7) into the basic AIP relaxation (which may be stronger than Eq. (7)), we get a polynomial-time algorithm to solve (a potentially stronger version of) Eq. (7). As a bonus, AIP has a polynomial-time uniform algorithm, whose running time is polynomial in the size of the instance I and the template $\mathcal{R}$. By contrast, there is no known polynomial-time uniform algorithm that constructs Eq. (7) or GE System, because it takes super-polynomial time in the size of $\mathcal{R}$ to find master/universal embeddings.

1.5.4. Constraint Singleton AIP. The previous subsection ignores an important issue: It is not enough to know whether the affine relaxation in Eq. (7) has *any solution*. The hybrid algorithm needs to know, for each constraint C and each partial assignment b to C , whether Eq. (7) has a *solution corresponding to the partial assignment b* . In other words, it looks for a solution of Eq. (7) whose restriction to C is the honest encoding of b ; see Theorems 3.2 and 3.4.

Since we have transformed Eq. (7) into a basic AIP relaxation, this requirement turns out to be fulfilled by constraint singleton AIP (Theorem 3.11). We have therefore justified how constraint

singleton AIP naturally arises as a replacement for the Gaussian Elimination step **GaussElim** in the hybrid algorithm.

We also prove that this constraint singleton AIP step cannot be implemented by (non-singleton) AIP, even when combined with SDP. We prove that SDA and even linear-level SDP+AIP hierarchy fail to solve $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$ ([Theorem 5.20](#)). That the LP+AIP hierarchy fails to solve $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$ was claimed without proof in [\[Zhu26\]](#) (which built on [\[BG17, LP25\]](#)). Here we make the proof explicit, and also prove that SDP+AIP does not help, either. This result is hardly surprising; our simple observation is that the claimed LP+AIP lower bound can be strengthened to SDP+AIP, because both LP and SDP average-case lower bounds are due to pairwise independence [\[KMOW17, CdX26\]](#), and the mixing LP weight idea also works for SDP. On the other hand, we prove that our generalization of **GaussElim** based on [Eq. \(7\)](#) (and hence **CSinglAIP**) solves $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$ ([Theorem 3.8](#)), by exploiting its universal embedding ([Theorem 2.10](#)). This parallels Zhuk’s [\[Zhu26\]](#) proof that **SinglAIP** solves $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$, even though the details diverge.

1.5.5. Singl(SDP $\wedge$ AIP) Hierarchy. Having strengthened and simplified the hybrid algorithm as **BasicSDP $\wedge$ CSinglAIP**, the next step is to consider its hierarchy version. Since both **Singl(BW $\wedge$ AIP)** (known as cohomological consistency [\[ÓC22\]](#)) and **Singl(LP $\wedge$ AIP)** (known as C(BLP+AIP) [\[CŽ22, CN25\]](#)) hierarchies have been considered before, it is natural to consider the **Singl(SDP $\wedge$ AIP)** hierarchy. This appears in [Section 4](#).

Our **Singl(SDP $\wedge$ AIP)** lower bound ([Theorem 6.3](#)) shares many high-level ideas with the **Singl(LP $\wedge$ AIP)** lower bound in [\[CN25\]](#). There is one subtle difference: When proving the **Singl(LP $\wedge$ AIP)** lower bound, [\[CN25\]](#) worked exclusively with the BW closure, and directly constructed an LP solution with full support with respect to the BW closure. Here for our **Singl(SDP $\wedge$ AIP)** lower bound, in order to reuse the SDP solution with full support from [\[CNP24, CdX26\]](#), we have to relate the BW closure and the τ -wise closure somewhat differently than [\[CN25\]](#). See [Section 6](#) for details.

1.5.6. Approximate Graph Homomorphism. In [Section 7](#), we strengthen the SDA lower bounds for approximate graph homomorphism (AGH) by Ciardo and Živný [\[CŽ25\]](#) to **BasicSDP $\wedge$ CSinglAIP** lower bounds. As is well known, it suffices to find a graph I with arbitrarily large chromatic number, and to construct a **BasicSDP $\wedge$ CSinglAIP** solution for I for $\text{CSP}(C_h)$ where h is odd, i.e. the CSP on an arbitrary odd cycle C_h .

For the SDP solution, we just use the same SDP solution as [\[CŽ25\]](#), and make explicit that the solution has full support. For the **CSinglAIP** solution, we prove a stronger property: For every odd $h \geq 3$, every instance I of $\text{CSP}(C_h)$ has a **CSinglAIP** solution ([Theorem 7.5](#)). This implies **CSinglAIP** accepts every instance of $\text{CSP}(C_h)$ and hence of AGH. Thus **CSinglAIP** is completely useless for these two problems. This property has not appeared in the literature.

1.6. Preliminaries. A set is t -small if its cardinality is at most t . $\mathbb{N} \stackrel{\text{def}}{=} \{0, 1, 2, \dots\}$ denotes the set of natural numbers, and $[k] \stackrel{\text{def}}{=} \{1, 2, \dots, k\}$.

An event E_n happens with uniformly positive probability if there is $\varepsilon > 0$ such that $\mathbb{P}[E_n] \geq \varepsilon$ for all sufficiently large n .

Given a function $a : S \rightarrow D$, $a_T : T \rightarrow D$ denotes its restriction to $T \subseteq S$.

The notation $f : (x \in X) \rightarrow Y_x$ means that f is a dependent function mapping every $x \in X$ to $f(x) \in Y_x$, where the codomain Y_x varies with x . Given $Z \subseteq X$, $f|_Z : (z \in Z) \rightarrow Y_z$ is the restriction of f to Z , given by $f|_Z(z) \stackrel{\text{def}}{=} f(z)$ for $z \in Z$.

Given a subset $S \subseteq V$ and assignment $a \in D^S$, denote by $\mathbb{1}_a : D^V \rightarrow \mathbb{Z}$ the indicator function of a , defined as $\mathbb{1}_a(b) = 1$ if $b_S = a_S$, and $\mathbb{1}_a(b) = 0$ otherwise.

For simplicity, we consider only constraint satisfaction problems with k *distinct* variables per constraint (k -CSP). A constraint satisfaction problem $\text{CSP}(\mathcal{R})$ is a pair $(D, \mathcal{R})$. Its domain D is a nonempty finite set. $\mathcal{R}$ is a nonempty family of k -ary relations over D , i.e. $R \subseteq D^k$ for every $R \in \mathcal{R}$. An instance $I = (V, \mathcal{C})$ of the CSP $(D, \mathcal{R})$ consists of a finite set V of variables and a finite set $\mathcal{C}$ of

constraints over V . Every constraint $C \in \mathcal{C}$ is of the form $C = (S, R)$, where the scope $S \in V^k$ of C is a sequence of k distinct variables, and $R \in \mathcal{R}$ is the set of accepting assignments of C . Write $V(C)$ for the set of variables in the scope of C . In other words, an instance I is a hypergraph, where each hyperedge C is associated with (S, R) .

A partial assignment $a \in D^{V(C)}$ on a constraint $C = (S, R)$ with scope $S = (v_1, \dots, v_k) \in V^k$ naturally corresponds to an assignment $a' \in D^k$ via $a' \stackrel{\text{def}}{=} a \circ S$, that is, $a'(i) = a(S(i)) = a(v_i)$ for $i \in [k]$. The set of satisfying assignments for C is $A_C \stackrel{\text{def}}{=} \{a \in D^{V(C)} \mid a \circ S \in R\}$. An assignment $a \in D^V$ satisfies a constraint $C = (S, R)$ if $a_{V(C)} \in A_C$; otherwise a violates C . An assignment $a \in D^V$ satisfies a constraint set $\mathcal{C}$ or an instance $I = (V, \mathcal{C})$ if a satisfies every constraint $C \in \mathcal{C}$; otherwise a violates $\mathcal{C}$ and I .

2. ABELIAN EMBEDDING

In this section, we define universal embedding, which simplifies and generalizes the master embedding of [BKM24, BKM26]. We also compute the universal embedding for linear equations over $\mathbb{Z}_m$ and $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$.

Remark 2.1. Our Theorem 1.1 of abelian embedding allows g to be nonzero, differing from [BKM24, Definition 1.1] and [BKM26, Definition 1.2]. Allowing nonzero g does not affect the existence of embeddings for a relation R , because one can always shift one of the σ_i to accommodate the change in definitions, as mentioned in [BKM26, after Definition 7.2]. However, when abelian embeddings are used in the [BKM26] hybrid algorithm, it will be crucial to allow nonzero g . This is because the hybrid algorithm involves a system of abelian linear equations based on these embeddings. When the CSP is 3-XOR, the corresponding linear equations are equivalent to the natural equations over $\mathbb{Z}_2$, with the right-hand-side being g . For the same reason, [BKM26] also allows nonzero g in their [BKM26, Definition 7.2] when they consider their hybrid algorithm.

2.1. Universal Embedding.

Definition 2.2 (Universal Embedding). An embedding (G, σ, g) of $R \subseteq D^k$ is universal if for every embedding (H, τ, h) of R , there are affine shifts f_i of a common homomorphism $\varphi : G \rightarrow H$ such that $\tau_i = f_i \circ \sigma_i$ for $i \in [k]$. That is, there are $b_i \in H$ such that $f_i(a) = \varphi(a) + b_i$ for $a \in G, i \in [k]$.

This definition allows affine shifts f_i , and not just homomorphisms, because it is possible that $\sigma_i(a) = 0_G$ while $\tau_i(a) \neq 0_H$, and it is easy to come up with examples of such embeddings of 3-XOR to $\mathbb{Z}_2$.

Remark 2.3 (Tight Embedding). All useful information about an embedding (G, σ, g) of R is contained inside the coset $g + \Delta_\sigma$ in G , where Δ_σ denotes the difference subgroup

$$\Delta_\sigma \stackrel{\text{def}}{=} \langle \{\sigma_i(a) - \sigma_i(b) \mid a, b \in D, i \in [k]\} \rangle,$$

and $\langle T \rangle$ denotes the subgroup generated by the subset T . We can therefore shift every σ_i to obtained an equivalent embedding to Δ_σ as follows. Fix any $w \in D$. (G, σ, g) yields an equivalent embedding (Δ_σ, τ, h) , where $\tau_i \stackrel{\text{def}}{=} \sigma_i - \sigma_i(w)$, and $h \stackrel{\text{def}}{=} g - \sum_{i \in [k]} \sigma_i(w)$. Call such (Δ_σ, τ, h) a tightening of (G, σ, g) . Also call an embedding (G, σ, g) tight if $\Delta_\sigma = G$.

The terminology “universal” comes from category theory. Given $R \subseteq D^k$, one can define a category of tight embeddings of R to abelian groups. Its objects are tight embeddings (G, σ, g) of $R \subseteq D^k$ to abelian groups G . A morphism in this category from (G, σ, g) to (H, σ, h) is a k -tuple of affine shifts $f_i : G \rightarrow H$ of a common homomorphism such that $\tau_i = f_i \circ \sigma_i$ for $i \in [k]$. Theorem 2.2 is an initial (also called universal) object in this category, meaning a tight embedding (G, σ, g) having a unique morphism to any embedding (H, τ, h) (Theorem 2.7).

As an initial object, a universal embedding is essentially unique: If $X = (G, \sigma, g)$ and $Y = (H, \tau, h)$ are both universal embeddings of R , then there are morphisms $f = (f_i)_{i \in [k]}$ from X to Y and $f^{-1} = (f_i^{-1})_{i \in [k]}$ from Y to X , such that $\text{id}_X = f^{-1} \circ f$ and $\text{id}_Y = f \circ f^{-1}$.² In other words, for all practical purposes, a tight universal embedding is unique.

When R does not admit any non-trivial embedding to $\mathbb{Z}$, [BKM24, Definition 3.6] implies that R has a master, and hence universal (Theorem A.5), embedding into a finite abelian group. We give a shorter algebraic proof in Theorems 2.4 and 2.5, without relying on the analytic proof of [BKM24, Lemmas 3.2, 3.3, 3.4]. Theorem 2.5 also generalizes this result without assuming absence of non-trivial embedding to $\mathbb{Z}$, but now the target group is finitely generated.

Lemma 2.4. *A relation $R \subseteq D^k$ has an embedding $(\mathbb{Z}, \tau, h)$ to $\mathbb{Z}$ where τ_i are not all constants if and only if the tight universal embedding (G, σ, g) of R has an infinite target G .*

Proof. Let (G, σ, g) be a universal embedding of R , and $(\mathbb{Z}, \tau, h)$ be any embedding of R where τ_i are not all constants. Universality of (G, σ, g) implies there are homomorphism $\varphi : G \rightarrow \mathbb{Z}$ and shifts $c_i \in \mathbb{Z}$ such that $\tau_i(a) = \varphi(\sigma_i(a)) + c_i$ for $a \in D, i \in [k]$. Therefore $\tau_i(a) - \tau_i(b) = \varphi(\sigma_i(a) - \sigma_i(b))$ for $a, b \in D, i \in [k]$. Since τ_i is non-constant for some $i \in [k]$, $\tau_i(a) \neq \tau_i(b)$ for some $a, b \in D$. Then $0 \neq t(\tau_i(a) - \tau_i(b)) = \varphi(t(\sigma_i(a) - \sigma_i(b)))$ for every nonzero $t \in \mathbb{Z}$, and hence $t(\sigma_i(a) - \sigma_i(b)) \neq 0_G$. Therefore the subgroup of G generated by $\sigma_i(a) - \sigma_i(b)$ is infinite.

Conversely, suppose the tight universal embedding (G, σ, g) of R has an infinite G . Since G is tight, G is finitely generated by $\sigma_i(a) - \sigma_i(b)$ for $a, b \in D, i \in [k]$. The fundamental theorem of finitely generated abelian groups implies $G \cong F \oplus H$ for some finite abelian group H , and $F = \mathbb{Z}^s$ for some natural number $s \geq 1$. We will construct an embedding $(\mathbb{Z}, \tau, h)$ of R by mapping $F = \mathbb{Z}^s$ to $\mathbb{Z}$ via weighted coordinate-wise addition and then dropping H . Since (G, σ, g) is tight, G is generated by $\sigma_{i,F}(a) - \sigma_{i,F}(b)$ for $a, b \in D, i \in [k]$, where $\sigma_{i,F}$ denotes the F -coordinates of σ_i . Therefore $\sigma_{i,F}(a) \neq \sigma_{i,F}(b)$ for some $a, b \in D, i \in [k]$. Let $w \stackrel{\text{def}}{=} \sigma_{i,F}(a) - \sigma_{i,F}(b)$, so that $\sum_{j \in [k]} w_j(\sigma_{i,F}(a)_j - \sigma_{i,F}(b)_j) \neq 0$. Define $\tau_\ell(c) \stackrel{\text{def}}{=} \sum_{j \in [s]} w_j \sigma_{\ell,F}(c)_j$ for $c \in D, \ell \in [k]$, and $h \stackrel{\text{def}}{=} \sum_{j \in [s]} w_j (g_F)_j$. In particular $\tau_i(a) \neq \tau_i(b)$. Then for $r \in R$, $\sum_{i \in [k]} \tau_i(r_i) = \sum_{i \in [k]} \sum_{j \in [s]} w_j \sigma_{i,F}(r_i)_j \stackrel{(*)}{=} \sum_{j \in [s]} w_j (g_F)_j = h$, where $(*)$ uses the fact that (G, σ, g) is an embedding of R and thus $\sum_{i \in [k]} \sigma_i(r_i) = g$. $\square$

Lemma 2.5. *Every k -ary relation $R \subseteq D^k$ over a finite domain D has a tight universal embedding to a finitely generated abelian group that has at most $|D| \times k$ generators.*

Proof. Denote by $\mathcal{E}$ the countable³ collection of tight embeddings (H, τ, h) of R to abelian groups H . Define (G, σ, g) to be the direct product of these embeddings:

$$G \stackrel{\text{def}}{=} \prod_{(H, \tau, h) \in \mathcal{E}} H, \quad \sigma_i \stackrel{\text{def}}{=} \prod_{(H, \tau, h) \in \mathcal{E}} \tau_i \text{ for } i \in [k], \quad g \stackrel{\text{def}}{=} \prod_{(H, \tau, h) \in \mathcal{E}} h.$$

Then (G, σ, g) is universal, because for any embedding (H, τ, h) of R , the projection $\pi : G \rightarrow H$ to the (H, τ, h) -coordinate is a homomorphism, such that $\tau_i = \pi \circ \sigma_i$ for $i \in [k]$. A tightening of (G, σ, g) is also universal (Theorem 2.6), and is generated by $\sigma_i(a) - \sigma_i(w)$ over $a \in D, i \in [k]$. $\square$

Lemma 2.6. *If an embedding (G, σ, g) of $R \subseteq D^k$ is universal, so is its tightening (Δ_σ, τ, h) from Theorem 2.3.*

Proof. Consider any embedding (M, μ, m) of $R \subseteq D^k$. Universality of (G, σ, g) (Theorem 2.2) implies there are homomorphism $\varphi : G \rightarrow M$ and shifts $b_i \in M$ such that $\mu_i(a) = \varphi(\sigma_i(a)) + b_i$ for $a \in D, i \in [k]$. Fix any $w \in D$ and consider the tightening (Δ_σ, τ, h) of (G, σ, g) as in Theorem 2.3, where $\tau_i \stackrel{\text{def}}{=} \sigma_i - \sigma_i(w)$ for $i \in [k]$. The restriction $\varphi|_{\Delta_\sigma}$ of φ to Δ_σ is a homomorphism from Δ_σ

² $f \circ g \stackrel{\text{def}}{=} (f_i \circ g_i)_{i \in [k]}$ whenever $f = (f_i)_{i \in [k]}$ and $g = (g_i)_{i \in [k]}$.

³There are countably many finitely generated abelian groups.

to M . Then $\mu_i(a) = \varphi|_{\Delta_\sigma}(\tau_i(a)) + b_i + \varphi(\sigma_i(w))$ for $a \in D, i \in [k]$. Therefore $\mu_i = f_i \circ \sigma_i$, where $f_i(g) = \varphi|_{\Delta_\sigma}(g) + b_i + \varphi(\sigma_i(w))$ for $g \in \Delta_\sigma$. $\square$

We also show uniqueness of morphism from a tight universal embedding to any other embedding:

Lemma 2.7. *Let (G, σ, g) and (H, τ, h) be tight embeddings of $R \subseteq D^k$. Suppose $(e_i)_{i \in [k]}$ and $(f_i)_{i \in [k]}$ are morphisms from (G, σ, g) to (H, τ, h) . Then $e_i = f_i$ for $i \in [k]$.*

Proof. Since $(e_i)_{i \in [k]}$ is a morphism, there are homomorphism $\varphi : G \rightarrow H$ and shifts $r_i \in H$ such that $\tau_i(a) = e_i(\sigma_i(a)) = \varphi(\sigma_i(a)) + r_i$ for $a \in D, i \in [k]$. Since $(f_i)_{i \in [k]}$ is a morphism, there are also homomorphism $\psi : G \rightarrow H$ and shifts $s_i \in H$ such that $\tau_i(a) = f_i(\sigma_i(a)) = \psi(\sigma_i(a)) + s_i$ for $a \in D, i \in [k]$. Therefore $\varphi(\sigma_i(a)) - \psi(\sigma_i(a)) = r_i - s_i$ for $a \in D, i \in [k]$, and $(\varphi - \psi)(\sigma_i(a) - \sigma_i(b)) = 0$ for $a, b \in D, i \in [k]$. Since G is tight, G is generated by such $\sigma_i(a) - \sigma_i(b)$, implying $\varphi(g) = \psi(g)$ for $g \in G$. This in turn implies $r_i = s_i$ for $i \in [k]$. $\square$

Existence of non-trivial embeddings is equivalent to the tight universal embedding having a non-trivial target:

Lemma 2.8. *A relation $R \subseteq D^k$ has an embedding (H, τ, h) where the $\tau_i : D \rightarrow H$ are not all constants if and only if the tight universal embedding (G, σ, g) of R has a non-trivial target G .*

Proof. Suppose G in the tight universal embedding (G, σ, g) of R is non-trivial (i.e. $|G| > 1$). Tightness says G is generated by $\sigma_i(a) - \sigma_i(b)$ over $a, b \in D, i \in [k]$. Therefore $\sigma_i(a) \neq \sigma_i(b)$ for some $a, b \in D, i \in [k]$. Then (G, σ, g) is an embedding of R where $\sigma_i : D \rightarrow G$ are not all constants.

Conversely, suppose G in the tight universal embedding (G, σ, g) of R is trivial. Consider any embedding (H, τ, h) of R . Universality of (G, σ, g) implies there are homomorphism $\varphi : G \rightarrow H$ and shifts $c_i \in H$ such that $\tau_i(a) = \varphi(\sigma_i(a)) + c_i$ for $a \in D, i \in [k]$. Therefore $\tau_i(a) - \tau_i(b) = \varphi(\sigma_i(a) - \sigma_i(b)) = 0_H$ for $a, b \in D, i \in [k]$, where the last equality uses $|G| = 1$ and $\varphi(0_G) = 0_H$. $\square$

2.2. Identity Embedding is Universal. Given $m, k \in \mathbb{N}$, consider a k -ary mod- m constraint $R_{k,m,b} \stackrel{\text{def}}{=} \{a \in \mathbb{Z}_m^k \mid \sum_{i \in [k]} a_i = b\}$ for some $b \in \mathbb{Z}_m$. It has the identity embedding (σ, b) to $\mathbb{Z}_m$, given by $\sigma_i \stackrel{\text{def}}{=} \text{id}_{\mathbb{Z}_m}$ for $i \in [k]$.

Theorem 2.9. *For any $k \geq 3, m \geq 1, b \in \mathbb{Z}_m$, the identity embedding for $R_{k,m,b}$ is universal.*

Proof. Consider any embedding (H, τ, h) of $R_{k,m,b}$. Our goal is to show that τ_i are affine shifts of a common homomorphism $h : \mathbb{Z}_m \rightarrow H$. Then the maps $f_i \stackrel{\text{def}}{=} \tau_i$ satisfy $\tau_i = f_i \circ \text{id}_{\mathbb{Z}_m}$ for $i \in [k]$.

Given any tuple $t \in \mathbb{Z}_m^k$, denote by $\sum \tau(t) \stackrel{\text{def}}{=} \sum_{i \in [k]} \tau_i(t_i)$.

Given distinct $i, j \in [k]$, fix an arbitrary tuple $s \in R_{k,m,b}$ such that $s_i = s_j = 0$ (for example, s may be all-zeros, except an entry b at some position different from i and j). Further given $c \in \mathbb{Z}_m$, define the tuple t that is the same as s , except $t_i \stackrel{\text{def}}{=} s_i + c = c$ and $t_j \stackrel{\text{def}}{=} s_j - c = -c$. Since R contains both s and t , $\sum \tau(t) - \sum \tau(s) = 0_H$, so

$$(2) \quad \tau_i(c) + \tau_j(-c) = \tau_i(0) + \tau_j(0).$$

For $a, b \in \mathbb{Z}_m$ and $r \in [k] \setminus \{i, j\}$, define the tuples u, v, w that are the same as s , except

$$\begin{array}{lll} u_i = s_i + a & & u_r = s_r - a \\ v_i = s_i + b & v_j = s_j - b & \\ w_i = s_i + a + b & w_j = s_j - b & w_r = s_r - a \end{array}$$

Since R contains s, u, v , and w , we have $\sum \tau(u) + \sum \tau(v) - \sum \tau(w) - \sum \tau(s) = 0_H$. Thus

$$(3) \quad \tau_i(a) + \tau_i(b) = \tau_i(a + b) + \tau_i(0).$$

Therefore $h_i \stackrel{\text{def}}{=} a \mapsto \tau_i(a) - \tau_i(0)$ is a homomorphism from $\mathbb{Z}_m$ to H . Eq. (2) further implies $\tau_i(c) - \tau_i(0) = \tau_j(c) - \tau_j(0)$ for $c \in \mathbb{Z}_m$ and distinct $i, j \in [k]$. That is, the maps $f_i \stackrel{\text{def}}{=} \tau_i : \mathbb{Z}_m \rightarrow H$ are affine shifts of a common homomorphism. And for any $c \in \mathbb{Z}_m, i \in [k]$, $f_i \circ \text{id}_{\mathbb{Z}_m}(c) = f_i(c) = \tau_i(c)$. $\square$

2.3. Universal Embedding for $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$. We will define $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$ and compute its universal embedding.

The domain of this k -CSP is the disjoint union of $\mathbb{Z}_2$ and $\mathbb{Z}_3$, denoted $D \stackrel{\text{def}}{=} \{0, 1, 0', 1', 2'\}$. Each $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$ combines a $\mathbb{Z}_2$ constraint and a $\mathbb{Z}_3$ constraint. Therefore each $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$ constraint is of the form

$$\left\{ a \in \{0, 1\}^k \mid \sum_{i \in [k]} a_i \equiv y \pmod{2} \right\} \cup \left\{ a \in \{0', 1', 2'\}^k \mid \sum_{i \in [k]} a_i \equiv z \pmod{3} \right\}$$

where $y \in \{0, 1\}, z \in \{0, 1, 2\}$. Note that this definition is slightly different from that in [Zhu26], whose constraint is either a $\mathbb{Z}_2$ constraint or a $\mathbb{Z}_3$ constraint. We choose to define the CSP this way only for simplicity.

Take any instance of $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$. For variables in the same connected component, any satisfying assignment must assign to these variables only the values in $\mathbb{Z}_2$, or only the values in $\mathbb{Z}_3$. Therefore, a connected component is satisfiable if and only if it is satisfiable after restricting to the $\mathbb{Z}_2$ part, or after restricting to the $\mathbb{Z}_3$ part. In either case, the constraints form a system of linear equations over $\mathbb{Z}_2$ or $\mathbb{Z}_3$, which can be efficiently solved by Gaussian elimination. Thus the CSP is tractable.

The k -CSP $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$ has three embeddings to an abelian group:⁴

- (1) $(\mathbb{Z}_2, \sigma^2, y)$. Each σ_i^2 is the identity function on $\mathbb{Z}_2$. Also, pick a satisfying assignment to the $\mathbb{Z}_2$ part of the constraint, i.e. $a \in \mathbb{Z}_2^k$ such that $\sum_i a_i = y$ in $\mathbb{Z}_2$. Then σ_i^2 maps $\mathbb{Z}_3$ to the constant a_i .
- (2) $(\mathbb{Z}_3, \sigma^3, z)$. (Same as before, reversing the roles of $\mathbb{Z}_2$ and $\mathbb{Z}_3$) Each σ_i^3 is the identity function on $\mathbb{Z}_3$. Also, pick a satisfying assignment to the $\mathbb{Z}_3$ part of the constraint, i.e. $a \in \mathbb{Z}_3^k$ such that $\sum_i a_i = z$ in $\mathbb{Z}_3$. Then σ_i^3 maps $\mathbb{Z}_2$ to the constant a_i .
- (3) $(\mathbb{Z}^{k-1}, \sigma', 0)$. Each σ'_i maps $\mathbb{Z}_2$ to $0 \in \mathbb{Z}^{k-1}$. Also, each σ'_i maps $\mathbb{Z}_3$ to the i -th standard basis vector for $i \in [k-1]$, while σ'_k maps $\mathbb{Z}_3$ to the all- (-1) vector.

Theorem 2.10. *For any $k \geq 3$, the direct product of the above three embeddings, as an embedding of a $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$ constraint R to $\mathbb{Z}_2 \times \mathbb{Z}_3 \times \mathbb{Z}^{k-1}$, is universal.*

Proof. Consider any embedding (H, τ, h) of R . Consider the subgroup H_2 of H generated by $\tau_i(a)$ over all $a \in \{0, 1\}, i \in [k]$. Restricted to the subdomain $\{0, 1\}$, (H, τ, h) induces an embedding (H_2, τ^2, h_2) of $R_{k,2,y}$, where $\tau^2 : \{0, 1\} \rightarrow H_2$ agrees with τ . Theorem 2.9 implies (H_2, τ^2, h_2) is the affine shifts of the embedding $(\mathbb{Z}_2, \text{id}_{\mathbb{Z}_2}, y)$: there are affine shifts $f_i^2 : \mathbb{Z}_2 \rightarrow H_2$ of a common homomorphism φ^2 such that $\tau_i \equiv \tau_i^2 \equiv f_i^2 \circ \sigma_i^2$ on $\{0, 1\}$ for $i \in [k]$. In particular,

$$\begin{aligned} \tau_i(1) - \tau_i(0) &= f_i^2 \circ \sigma_i^2(1) - f_i^2 \circ \sigma_i^2(0) \\ &= f_i^2(1_{\mathbb{Z}_2}) - f_i^2(0_{\mathbb{Z}_2}) \\ &= \varphi^2(1_{\mathbb{Z}_2}) - \varphi^2(0_{\mathbb{Z}_2}) = \varphi^2(1_{\mathbb{Z}_2}) \end{aligned}$$

⁴In this subsection, superscripts ^{2} and ^{3} are labels and do not mean the second or third power. However, superscripts ^{k} or $^{k-1}$ are the corresponding powers.

is a constant $p \in H_2 \subseteq H$ independent of $i \in [k]$. Further,

$$\begin{aligned}
2p &= 2f_i^2 \circ \sigma_i^2(1) - 2f_i^2 \circ \sigma_i^2(0) \\
&= f_i^2 \circ (2\sigma_i^2(1)) - f_i^2 \circ (2\sigma_i^2(0)) \\
&= f_i^2(1_{\mathbb{Z}_2} + 1_{\mathbb{Z}_2}) - f_i^2(0_{\mathbb{Z}_2} + 0_{\mathbb{Z}_2}) \\
&= f_i^2(0_{\mathbb{Z}_2}) - f_i^2(0_{\mathbb{Z}_2}) \\
&= 0_H.
\end{aligned}$$

Apply the same argument to the subdomain $\{0', 1', 2'\}$ and $R_{k,3,z}$, we have $\tau_i(1') - \tau_i(0')$ equals some constant $q \in H$ independent of $i \in [k]$. Further, $3q = 0_H$, and $\tau_i(2') - \tau_i(0') = 2q$ for $i \in [k]$.

Denote by $\sigma_i \stackrel{\text{def}}{=} (\sigma_i^2, \sigma_i^3, \sigma_i') : D \rightarrow \mathbb{Z}_2 \times \mathbb{Z}_3 \times \mathbb{Z}^{k-1}$ the product embedding. Define a homomorphism $\varphi : \mathbb{Z}_2 \times \mathbb{Z}_3 \times \mathbb{Z}^{k-1} \rightarrow H$ by $\varphi(1, 0, 0) \stackrel{\text{def}}{=} p$, $\varphi(0, 1, 0) \stackrel{\text{def}}{=} q$, and $\varphi(\sigma_i(0') - \sigma_i(0)) \stackrel{\text{def}}{=} \tau_i(0') - \tau_i(0)$ for $i \in [k-1]$. φ being a homomorphism implies the last identity holds for $i = k$ as well. Let $f_i \stackrel{\text{def}}{=} \varphi - \varphi \circ \sigma_i(0) + \tau_i(0)$ for $i \in [k]$. It is now a mechanical check that $\tau_i = f_i \circ \sigma_i$ as claimed: for each $i \in [k]$,

$$\begin{aligned}
f_i \circ \sigma_i(0) &= \varphi(\sigma_i(0) - \sigma_i(0)) + \tau_i(0) = \tau_i(0) \\
f_i \circ \sigma_i(1) &= \varphi(\sigma_i(1) - \sigma_i(0)) + \tau_i(0) \\
&= \varphi(1, 0, 0) + \tau_i(0) = p + \tau_i(0) = \tau_i(1) \\
f_i \circ \sigma_i(0') &= \varphi(\sigma_i(0') - \sigma_i(0)) + \tau_i(0) \\
&= \tau_i(0') - \tau_i(0) + \tau_i(0) = \tau_i(0') \\
f_i \circ \sigma_i(1') &= \varphi(\sigma_i(1') - \sigma_i(0)) + \tau_i(0) \\
&= [\varphi(\sigma_i(1') - \sigma_i(0'))] + [\varphi(\sigma_i(0') - \sigma_i(0)) + \tau_i(0)] \\
&= \varphi(0, 1, 0) + \tau_i(0') = q + \tau_i(0') = \tau_i(1') \\
f_i \circ \sigma_i(2') &= \varphi(\sigma_i(2') - \sigma_i(0)) + \tau_i(0) \\
&= [\varphi(\sigma_i(2') - \sigma_i(0'))] + [\varphi(\sigma_i(0') - \sigma_i(0)) + \tau_i(0)] \\
&= \varphi(0, 2, 0) + \tau_i(0') = 2q + \tau_i(0') = \tau_i(2') \quad \square
\end{aligned}$$

3. HYBRID ALGORITHM

In this section, we present the hybrid algorithm of [BKM26]. We will represent the system of equations in the hybrid algorithm as an affine relaxation, significantly simplifying the equations and generalizing them to CSPs embeddable into $\mathbb{Z}$. We then present our **BasicSDP** $\wedge$ **CSingAIP** algorithm, which subsumes the hybrid algorithm.

We now state the hybrid algorithm of [BKM26] using our notation (where the input instance is $I = (V, \mathcal{C})$):

Algorithm 1 [BKM26] Hybrid Algorithm

- 1: **repeat**
 - 2: **for all** constraint $C \in \mathcal{C}$, assignment $b \in A_C$ **do**
 - 3: **if not** (**BasicSDP**($\mathcal{C}, b$) and **GaussElim**($\mathcal{C}, b$)) **then**
 - 4: $\mathcal{C} \leftarrow \text{ReplaceConstraint}(\mathcal{C}, C, A_C \setminus \{b\})$
 - 5: **until** no change to $\mathcal{C}$
 - 6: **Accept if** $A_C \neq \emptyset$ for every $C \in \mathcal{C}$
-

Recall that $A_C \subseteq D^{V(C)}$ denotes the set of satisfying assignments to the constraint C .

On line 3, `BasicSDP`($\mathcal{C}, b$) and `GaussElim`($\mathcal{C}, b$) test whether the partial assignment b of constraint C may extend to a global satisfying assignment, using the basic SDP relaxation and Gaussian Elimination, respectively. The subroutine `GaussElim` will be discussed in [Section 3.1](#), and `BasicSDP` in [Section 4.2](#).

On line 4, `ReplaceConstraint` replaces constraint C with a stronger constraint C' that further forbids b :

$$\text{ReplaceConstraint}(\mathcal{C}, C, R) \stackrel{\text{def}}{=} (\mathcal{C} \setminus \{C\}) \cup \{\text{constraint } C' \text{ with relation } R\}.$$

Every time line 4 is executed, one can imagine replacing the current instance with a *reduced instance*. When presented in the above way, many steps in [\[BKM26, Sections 7.2 and 7.3\]](#) that involve $\text{supp}(\mu_C)$ for some SDP solution μ will be replaced in this paper with just A_C (the set of satisfying assignments of the constraint C in the current reduced instance).

3.1. New Affine Relaxation. The `GaussElim` subroutine from the hybrid algorithm ([Algorithm 1](#)) uses Gaussian Elimination to find specific solutions to a system of abelian linear equations, called the “GE System” in [\[BKM26\]](#). The “GE System” of equations has a complex definition [\[BKM26, Sections 7.2, 7.3\]](#). Rather than presenting the original definition, we give a new affine relaxation that is significantly simpler. Whenever [\[BKM26, Algorithm 4\]](#) does not abort, our affine relaxation is equivalent to the GE System (see [Section 3.3](#)). Our new affine relaxation can also handle CSPs non-trivially embeddable into $\mathbb{Z}$, whereas the GE System cannot.

Our simplification hinges on relaxations involving affine hulls in an abelian group:

Definition 3.1 (Affine hull). The affine hull of a subset S of an abelian group G is

$$\text{aff}(S) \stackrel{\text{def}}{=} S + \langle \{a - b \mid a, b \in S\} \rangle,$$

where $\langle T \rangle$ denotes the subgroup generated by T in the abelian group G . Equivalently,

$$\text{aff}(S) = \left\{ \sum_{s \in S} w(s)s \mid w \in \mathbb{Z}^S, \sum_{s \in S} w(s) = 1, \text{finite supp}(w) \right\},$$

which contains all affine integer combinations of S with finite support. When S is nonempty, $\text{aff}(S)$ coincides with the intersection of cosets containing S , i.e., $\text{aff}(S)$ is the coset generated by S .⁵

The above definition is the abelian algebraic analog of the well-studied concept from Euclidean space. Though simple, this concept is absent from [\[BKM26\]](#).

Fix an instance $I = (V, \mathcal{C})$. For every variable $v \in V$, let $\mathcal{C}_v$ be the collection of constraints $C \in \mathcal{C}$ containing v , and define the group

$$(4) \quad G^v \stackrel{\text{def}}{=} \prod_{C \in \mathcal{C}_v} G_C,$$

where G_C is the target group of the universal embedding ([Theorem 2.2](#)) of the set $A_C \subseteq D^{V(C)}$ of satisfying assignments for the constraint C . Further, for every $b \in D$, we define the embedded tuple

$$(5) \quad r_b^v \stackrel{\text{def}}{=} (\sigma_v^C(b))_{C \in \mathcal{C}_v} \in G^v,$$

where $((\sigma_i^C)_{i \in [k]}, g_C)$ is the universal embedding of A_C into G_C , and $\sigma_v^C \stackrel{\text{def}}{=} \sigma_j^C$ with $j \in [k]$ being the position of v in (the scope of) the constraint C . We extend the definition of the embedded tuple to partial assignments $a \in D^{V(C)}$ for a constraint C :

$$(6) \quad r_a^C \stackrel{\text{def}}{=} (r_{a(v)}^v)_{v \in V(C)} \in \prod_{v \in V(C)} G^v.$$

Our new affine relaxation in [Eq. \(7\)](#) consists of solutions z such that:

- (1) For each variable v , $z(v)$ lies in the affine hull of the embedded tuples r_a^v over $a \in D$.

⁵If S is empty, then so is $\text{aff}(S)$, which cannot be a coset, since cosets are nonempty by definition.

- (2) For each constraint C , the tuple $z_{V(C)} = (z(v))_{v \in V(C)}$ lies in the affine hull of the embedded tuples r_a^C over satisfying partial assignments a of C .

$$(7) \quad \begin{aligned} & z : (v \in V) \rightarrow \text{aff}\{r_a^v \mid a \in D\} \\ & \text{subject to } z_{V(C)} \in \text{aff}\{r_a^C \mid a \in A_C\} \quad \text{for } C \in \mathcal{C} \end{aligned}$$

The goal of the GE System is to achieve consistent solutions [BKM26, Definition 7.7]:

Definition 3.2. $(A_C)_{C \in \mathcal{C}}$ is consistent with the coset $\mathcal{T}$ of solutions of the affine relaxation in Eq. (7) if for every $C \in \mathcal{C}$,

$$(8) \quad \text{aff}\{r_a^C \mid a \in A_C\} = \mathcal{T}|_C,$$

where

$$(9) \quad \mathcal{T}|_C \stackrel{\text{def}}{=} \{(z(v))_{v \in V(C)} \mid z \in \mathcal{T}\}.$$

Remark 3.3. [BKM26, Definition 7.6] contains a small mistake: It considers the *subgroup*, rather than the *coset*, generated by a subset. However, $\mathcal{T}|_C$ is a coset, not necessarily a subgroup; thus, equating it with a subgroup in [BKM26, Definition 7.7] is incorrect. Concretely, for odd-parity 3-XOR (Theorem 1.2), the identity embedding is a master/universal embedding (Theorem 2.9). When the whole instance consists of only a single such constraint C , $\mathcal{T}|_C$ is the coset $Y \stackrel{\text{def}}{=} \{a \in \mathbb{Z}_2^3 \mid \sum_{i \in [3]} a_i = 1\}$ of odd-parity assignments. The generated subgroup is $\mathbb{Z}_2^3 \neq Y$, violating [BKM26, Definition 7.7], even though Eq. (8) holds. The simple fix is to change [BKM26, Definition 7.6] to be “the affine hull of a subset”.

Definition 3.4. The subroutine $\text{GaussElim}(\mathcal{C}, b)$ accepts precisely when $r_b^C \in \mathcal{T}|_C$, where $b \in A_C$ and $\mathcal{T}$ is the coset of solutions of Eq. (7).

3.2. Tightness from Universality. Note that the affine constraints in Eq. (7) depend on the embeddings in Eqs. (4) and (5).

As an extreme example, if the trivial embedding were used, i.e., if G_C in Eq. (4) were replaced by the trivial group (consisting only of the identity element), then all the constraints $z_{V(C)} \in \text{aff}\{r_a^C \mid a \in A_C\}$ in Eq. (7) would be satisfied trivially. Therefore “finer” embeddings $(G_C, (\sigma_i^C)_{i \in [k]}, g_C)$ in Eqs. (4) and (5) are preferred. Theorem 3.6 below shows that if the affine relaxation Eq. (7) has solutions under the universal embedding, then it also has solutions under any abelian embedding. Of course, the converse may not hold. Therefore the affine constraints in Eq. (7) are as tight as possible among all abelian embeddings.

Given any function $f : X \rightarrow Y$, any subset $S \subseteq X$, let $f(S) \stackrel{\text{def}}{=} \{f(s) \mid s \in S\}$ be its image under f . The next result says that given any subset S of an abelian group, the image of its affine hull under an affine map f equals the affine hull of its image under f . Put differently, an affine map commutes with taking the affine hull. This result is the abelian algebraic analog of a well-known fact in Euclidean space.

Lemma 3.5. Let S be a subset of an abelian group G . Let $f : G \rightarrow H$ be an affine map from G to another group H , i.e., there is a homomorphism $\varphi : G \rightarrow H$ and $c \in H$ such that $f(g) = \varphi(g) + c$ for $g \in G$. Then

$$(10) \quad f(\text{aff}(S)) = \text{aff}(f(S)).$$

Proof. Let $T \stackrel{\text{def}}{=} \{a - b \mid a, b \in S\}$.

$$f(\text{aff}(S)) = c + \varphi(\text{aff}(S)) = c + \varphi(S + \langle T \rangle) = c + \varphi(S) + \varphi(\langle T \rangle)$$

since φ is a homomorphism. Now

$$\varphi(\langle T \rangle) = \varphi(\langle \{a - b \mid a, b \in S\} \rangle) = \langle \{\varphi(a) - \varphi(b) \mid a, b \in S\} \rangle = \langle \varphi(T) \rangle$$

again because φ is a homomorphism. On the other hand,

$$\begin{aligned} \text{aff}(f(S)) &= f(S) + \langle \{a - b \mid a, b \in f(S)\} \rangle \\ &= c + \varphi(S) + \langle \{\varphi(a) - \varphi(b) \mid a, b \in S\} \rangle \\ &= c + \varphi(S) + \langle \varphi(T) \rangle = f(\text{aff}(S)). \end{aligned}$$

□

We now prove the main result of this subsection.

Theorem 3.6. *If the affine relaxation Eq. (7) has a solution (under the universal embedding), then it also has a solution when the universal embeddings (G_C, σ^C, h_C) in Eqs. (4) and (5) are replaced with any embedding (H_C, τ^C, h_C) of A_C .*

Proof. Let z be a solution to Eq. (7) under the universal embeddings (G_C, σ^C, g_C) of A_C . For any constraint C , any embedding (H_C, τ^C, h_C) of A_C , there are maps $f_i^C : G_C \rightarrow H_C$ such that $\tau_i^C = f_i^C \circ \sigma_i^C$ for $i \in [k]$. Define a map y by $y(v, C) \stackrel{\text{def}}{=} f_i^C(z(v, C))$ for $v \in V, C \in \mathcal{C}_v$, where i is the position of v in C . Then $y = f(z)$ under some affine map $f : \prod_{v \in V, C \in \mathcal{C}_v} G_C \rightarrow \prod_{v \in V, C \in \mathcal{C}_v} H_C$, whose (v, C) -coordinate is f_i^C as in the previous sentence.

Since z is a solution to Eq. (7), $z(v) \in \text{aff}\{r_a^v \mid a \in D\}$ for $v \in V$. Equivalently, $f_v(z(v)) \in f_v(\text{aff}\{r_a^v \mid a \in D\}) \stackrel{(10)}{=} \text{aff}\{f_v(r_a^v) \mid a \in D\}$, where $f_v : \prod_{C \in \mathcal{C}_v} G_C \rightarrow \prod_{C \in \mathcal{C}_v} H_C$ corresponds to the coordinates of f associated with v . Now $f_v(r_a^v) = (f_i^C(\sigma_i^C(a)))_{C \in \mathcal{C}_v} = (\tau_i^C(a))_{C \in \mathcal{C}_v} = s_a^v$, where s_a^v is defined as in Eq. (5) but using the embeddings (H_C, τ^C, h_C) instead. Then $y(v) \in \text{aff}\{s_a^v \mid a \in D\}$.

Similarly, $z_{V(C)} \in \text{aff}\{r_a^C \mid a \in A_C\}$ for $C \in \mathcal{C}$. Equivalently, $f_C(z_{V(C)}) \in f_C(\text{aff}\{r_a^C \mid a \in A_C\}) \stackrel{(10)}{=} \text{aff}\{f_C(r_a^C) \mid a \in A_C\}$. Now $f_C(r_a^C) = (f_v(r_{a(v)}^v))_{v \in V(C)} = (s_{a(v)}^v)_{v \in V(C)} = s_a^C$, where s_a^C is defined as in Eq. (6) but using the embeddings (H_C, τ^C, h_C) instead. Then $y_{V(C)} \in \text{aff}\{s_a^C \mid a \in A_C\}$. Thus y is a solution to Eq. (7) under the embeddings (H_C, τ^C, h_C) . □

The previous proof further shows that if Eq. (7) (under universal embeddings) has a solution z with $z_{V(C)} = r_b^C$, then the same holds under any abelian embedding. Equivalently, if GaussElim($\mathcal{C}, b$) rejects under any embedding, it also does under the universal embedding.

3.3. Equivalence to the GE System. Whenever [BKM26, Algorithm 4] does not abort, our affine relaxation in Eq. (7) is equivalent to the GE System in [BKM26]. Our G^v in Eq. (4) is essentially the same as their G_{master}^v , because master embeddings are universal (Theorem A.5). The GE System works with assignments y mapping $v \in V$ to the double dual group $\widehat{\widehat{G^v}}$ (in our terminology). But $\widehat{\widehat{G^v}}$ is naturally isomorphic to G^v by Pontryagin duality, which says that any homomorphism φ from $\widehat{\widehat{G^v}}$ to the multiplicative circle group $\mathbb{T}$ of complex numbers of unit magnitude is an evaluation map $\chi \in \widehat{\widehat{G^v}} \mapsto \chi(h)$ for some $h \in G^v$. Therefore, the GE System indirectly specifies an affine relaxation with solutions of the form $z : (v \in V) \rightarrow G^v$. That affine relaxation, implicit in [BKM26], is made explicit in Eq. (7).

The GE System consists of three types of constraints:

- (1) “Valid character constraints” [BKM26, Section 7.2.2]
- (2) “Valid satisfying assignments constraints” [BKM26, Section 7.2.2]
- (3) Character constraints enforcing $\text{aff}\{r_a^C \mid a \in A_C\} \supseteq \mathcal{T}|_C$, to partially achieve Eq. (8) [BKM26, Claim 7.9 and Algorithm 2].

Type 1 constraints merely force the y variables in the GE System to correspond to character evaluations, i.e., $y(v) \in \widehat{\widehat{G^v}}$ for $v \in V$.

Type 2 constraints are subsumed by Type 3. Indeed, for any $C \in \mathcal{C}$, $z_{V(C)} \in \text{aff}\{r_b^C \mid b \in A_C\}$ implies there is $w \in \mathbb{Z}^{A_C}$, $\sum_{b \in A_C} w(b) = 1$ such that $z_{V(C)} = \sum_{b \in A_C} w(b)r_b^C$. Then

$$\sum_{v \in V(C)} z(v, C) \stackrel{(5),(6)}{=} \sum_{b \in A_C} w(b) \sum_{v \in V(C)} \sigma_v^C(b(v)) \stackrel{(*)}{=} \sum_{b \in A_C} w(b)g_C = g_C,$$

where $(*)$ uses the fact that (σ^C, g_C) is an embedding of A_C . The previous equation is precisely a Type 2 character constraint via Pontryagin duality ([Theorem B.1](#)).

For Type 3, applying Pontryagin duality ([Theorem B.2](#)) yields the $z_{V(C)} \in \text{aff}\{r_a^C \mid a \in A_C\}$ constraints in [Eq. \(7\)](#). These constraints further imply the other constraint $z(v) \in \text{aff}\{r_a^v \mid a \in D\}$ in [Eq. \(7\)](#) for every non-isolated variable v . Therefore [Eq. \(7\)](#) is equivalent to the GE System (isolated variables that do not participate in any constraint can be ignored and removed).

The next claim summarizes this subsection.

Claim 3.7. *If a CSP instance I has no isolated variable and [\[BKM26, Algorithm 4\]](#) does not abort, then solutions y to the GE System [\[BKM26, Sections 7.2, 7.3\]](#) are bijective with solutions z to [Eq. \(7\)](#) via Pontryagin duality.*

3.4. Hybrid Algorithm Solves $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$. In this subsection, we prove that our generalized hybrid algorithm ([Section 3.1](#)) solves $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$. In fact, it does so using only its Gaussian Elimination step [GaussElim](#), while SDP plays no role. Recall that [GaussElim](#) $(\mathcal{C}, b)$ accepts precisely when $r_b^C \in \mathcal{T}|_C$, where $b \in A_C$ and $\mathcal{T}$ is the coset of solutions of [Eq. \(7\)](#). The key observation is that $r_b^C \in \mathcal{T}|_C$ for every $C \in \mathcal{C}$ implies the existence of a satisfying assignment for $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$. Our next result exploits the universal embedding of $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$ from [Theorem 2.10](#).

Theorem 3.8. *Consider a $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$ instance $I = (V, \mathcal{C})$. Suppose for every $C \in \mathcal{C}$, there is $b \in A_C$ such that $r_b^C \in \mathcal{T}|_C$. Then I is satisfiable.*

Proof. Consider the universal embedding (G, σ, g) of $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$ into $\mathbb{Z}_2 \times \mathbb{Z}_3 \times \mathbb{Z}^{k-1}$ from [Theorem 2.10](#). For $i \in [k]$, $b \in D$, the $\mathbb{Z}^{k-1}$ -coordinate of $\sigma_i(b)$ is of the form $p\sigma_i(0')$, where $p = 1$ if $b \in \{0', 1', 2'\}$ and $p = 0$ otherwise. Therefore, for any $s_i \in \text{aff}\{\sigma_i(b) \mid b \in D\}$ so that $s_i = \sum_{b \in D} w_i(b)\sigma_i(b)$ with $w_i \in \mathcal{D} \stackrel{\text{def}}{=} \{w \in \mathbb{Z}^D \mid \sum_{a \in D} w(a) = 1\}$, the $\mathbb{Z}^{k-1}$ -coordinate of s_i is of the form $q_i\sigma_i(0')$, where $q_i = \sum_{b \in \{0', 1', 2'\}} w_i(b)$. Further, for any $(s_i)_{i \in [k]}$ such that $s_i \in \text{aff}\{\sigma_i(b) \mid b \in D\}$ for $i \in [k]$ and $\sum_{i \in [k]} s_i = g$, we have $q_i = q_j$ for $i, j \in [k]$ by considering the $\mathbb{Z}^{k-1}$ -coordinate of the universal embedding.

Consider a $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$ instance $I = (V, \mathcal{C})$ and any solution z of [Eq. \(7\)](#). For $v \in V$, there is $w_v \in \mathcal{D}$ such that $z(v) = \sum_{b \in D} w_v(b)r_b^v$. Define $q : V \rightarrow \mathbb{Z}$ by $q(v) = \sum_{b \in \{0', 1', 2'\}} w_v(b)$. Then the $\mathbb{Z}^{k-1}$ -coordinate of $z(v, C)$ is of the form $q(v)\sigma_v(0')$ for $v \in V, C \in \mathcal{C}_v$. Further, $q(v) = q(u)$ if variables u and v are both contained in some constraint $C \in \mathcal{C}$, implying $q(u) = q(v)$ if u and v belong to the same connected component.

Suppose there are $C \in \mathcal{C}, a \in A_C$ such that $r_a^C \in \mathcal{T}|_C$. Then some solution z to [Eq. \(7\)](#) satisfies $z(v) = r_{a(v)}^v$ for $v \in V(C)$. Fix any $v_0 \in V(C)$. Then $q(v_0) \in \{0, 1\}$. The last sentence of the preceding paragraph implies $q(u) = q(v_0)$ for every variable u in the same connected component K as v_0 . If $q(u) = 0$, the $\mathbb{Z}_2$ -coordinate of z is a satisfying assignment for the $\mathbb{Z}_2$ part of $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$ on K . More precisely, let $t(u) = \sum_{b \in \mathbb{Z}_2} w(b)b \in \mathbb{Z}_2$. Then $\sum_{v \in V(C)} t(v) = y$ for every $\mathbb{Z}_2$ -constraint $\{a \in \mathbb{Z}_2^3 \mid \sum_{i \in [k]} a_i = y\}$ in K , because $t(v)$ equals the $\mathbb{Z}_2$ -coordinate of $z(v, C)$ for every $v \in V(C), C \in \mathcal{C}_v$.

Similarly, if $q(u) = 1$, the $\mathbb{Z}_3$ -coordinate of z is a satisfying assignment for the $\mathbb{Z}_3$ part on this connected component, using the fact that $\sum_{b \in \{0, 1\}} w_v(b) = 1 - q(v) = 0$ for variable v in this connected component. Therefore, if for every $C \in \mathcal{C}$, there is $a \in A_C$ such that $r_a^C \in \mathcal{T}|_C$, then the $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$ instance is satisfiable. $\square$

3.5. Yet Another Affine Relaxation and AIP. The constraints in Eq. (7) depend on the embeddings. To avoid such a dependence, we propose the following affine relaxation for any CSP $(D, \mathcal{R})$ (not necessarily embeddable into abelian groups):

$$(11) \quad \begin{aligned} & y : V \rightarrow \text{aff}\{\mathbb{1}_a \in \mathbb{Z}^D \mid a \in D\} \\ & \text{subject to } y_{V(C)} \in \text{aff}\{(\mathbb{1}_{a(v)})_{v \in V(C)} \in \mathbb{Z}^{D \times V(C)} \mid a \in A_C\} \quad \text{for } C \in \mathcal{C} \end{aligned}$$

A merit of this relaxation is that Eq. (11) is a stronger relaxation than Eq. (7):

Lemma 3.9. *If Eq. (11) has a solution, then so does Eq. (7).*

Proof. Let $y : V \rightarrow \mathcal{D}$ be a solution to Eq. (11). For $v \in V$, define a homomorphism $h_v : \mathbb{Z}^D \rightarrow G^v$ by $\mathbb{1}_a \mapsto r_a^v$ for $a \in D$. Define a solution $z : (v \in V) \rightarrow G^v$ for Eq. (7) by $z(v) \stackrel{\text{def}}{=} h_v \circ y(v)$ for $v \in V$. Then $z(v) = \sum_{a \in D} y(v, a) r_a^v \in \text{aff}\{r_a^v \mid a \in D\}$, because $\sum_{a \in D} y(v, a) = 1$.

For each constraint $C \in \mathcal{C}$, $y_{V(C)} \in \text{aff}\{(\mathbb{1}_{a(v)})_{v \in V(C)} \mid a \in A_C\}$, so there is $w(C) \in \mathbb{Z}^{A_C}$ such that

$$(12) \quad \sum_{a \in A_C} w(C, a) = 1 \quad \text{and} \quad (y(v))_{v \in V(C)} = \sum_{a \in A_C} w(C, a) (\mathbb{1}_{a(v)})_{v \in V(C)}.$$

Apply the homomorphism $(h_v)_{v \in V(C)} : \mathbb{Z}^{D \times V(C)} \rightarrow \prod_{v \in V(C)} G^v$ to both sides of the last equality,

$$(z(v))_{v \in V(C)} = \sum_{a \in A_C} w(C, a) (r_{a(v)}^v)_{v \in V(C)} \in \text{aff}\{(r_{a(v)}^v)_{v \in V(C)} \mid a \in A_C\}. \quad \square$$

Another merit is that Eq. (11) is in fact equivalent to basic AIP, which implies Eq. (11) can be solved in polynomial time.

Lemma 3.10. *Eq. (11) has a solution if and only if the basic AIP relaxation does.*

Proof. That y is a solution to Eq. (11) is equivalent to the existence of $w : (C \in \mathcal{C}) \rightarrow \mathbb{Z}^{A_C}$ satisfying Eq. (12). The last equality of Eq. (12) is equivalent to

$$(13) \quad y(v) = \sum_{a \in A_C} w(C, a) \mathbb{1}_{a(v)} = \pi_v(w(C)) \quad \text{for } v \in V(C).$$

This is precisely the consistency condition for (y, w) to be a basic AIP solution. $\square$

3.6. Constraint Singleton AIP simulates GaussElim. The GaussElim($\mathcal{C}, b$) step in the hybrid algorithm of [BKM26] uses Gaussian Elimination to test whether r_a^C belongs to $\mathcal{T}|_C$. Note that this is a different task than testing whether $\mathcal{T}$ is empty, i.e., whether Eq. (7) is satisfiable. We now show that GaussElim can in fact be simulated by constraint singleton AIP (CSinglAIP) — while without access to the affine constraints in Eq. (7).

Recall that given an instance $I = (V, \mathcal{C})$, a basic AIP solution for I is a map $s : (z \in V \cup \mathcal{C}) \rightarrow \mathcal{D}_z$ such that $\pi_v \circ s(C) = s(v)$ for $v \in C \in \mathcal{C}$, where the relaxed domain $\mathcal{D}_z \stackrel{\text{def}}{=} \{w \in \mathbb{Z}^{A_z} \mid \sum_{a \in A_z} w(a) = 1\}$.

To describe constraint singleton algorithms, we need to consider reduced instances. Given an instance $I = (V, \mathcal{C})$, some constraint $C \in \mathcal{C}$, and assignment $b \in A_C$, define the reduced instance $I + b \stackrel{\text{def}}{=} (V, \mathcal{C}_b)$, where $\mathcal{C}_b \stackrel{\text{def}}{=} \text{ReplaceConstraint}(\mathcal{C}, C, \{b\})$ is the same as $\mathcal{C}$ but the assignment b is forced on C .

Theorem 3.11. *Let $\mathcal{T}$ be the coset of solutions of Eq. (7) for an instance $I = (V, \mathcal{C})$. For any $C' \in \mathcal{C}, b \in A_{C'}$, if there exists a basic AIP solution s^b for the reduced instance $I + b$, then $r_b^{C'} \in \mathcal{T}|_{C'}$.*

Proof. Let s^b be a basic AIP solution for $I + b$. The proof of Theorem 3.10 implies $y \stackrel{\text{def}}{=} s^b|_V$ is a solution to Eq. (11) for $I + b$. Then $z \stackrel{\text{def}}{=} v \mapsto \sum_{a \in D} y(v, a) r_a^v$ is a solution to Eq. (7) for $I + b$ (and hence I) by the proof of Theorem 3.9. Since s^b is a basic AIP solution for $I + b$, we have $s^b(C') = \mathbb{1}_b$, so $s^b(u) = \mathbb{1}_{b(u)}$ for $u \in V(C')$. Thus $z(u) = r_{b(u)}^u$ for $u \in V(C')$, and $r_b^{C'} = z_{V(C')} \in \mathcal{T}|_{C'}$. $\square$

Consider the combined basic SDP and CSinglAIP algorithm $\mathcal{A}'$ obtained by replacing GaussElim in the hybrid algorithm (Algorithm 1) with CSinglAIP, where the subroutine CSinglAIP($\mathcal{C}, b$) accepts precisely when $I + b$ has a basic AIP solution. Theorem 3.11 implies that if $\mathcal{A}'$ accepts, then the reduced instance at the end of the algorithm has consistent solutions, i.e., Theorem 3.2 holds. Indeed, “ $\subseteq$ ” in Eq. (8) follows from Theorem 3.11 and

$$\text{aff}\{r_a^C | a \in A_C\} \subseteq \text{aff}(\mathcal{T}|_C) \stackrel{(10)}{=} \mathcal{T}|_C,$$

while “ $\supseteq$ ” follows from Eq. (7). Further, Theorem 3.11 implies any b rejected by GaussElim($\mathcal{C}, b$) is also rejected by CSinglAIP($\mathcal{C}, b$).

Proof of Theorem 1.4. Suppose the hybrid algorithm (Algorithm 1) rejects an instance $I = (V, \mathcal{C})$. Consider the sequence $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_t$ of constraint sets during an execution of the hybrid algorithm, where $\mathcal{C}_1 = \mathcal{C}$. For each $1 \leq j < t$, there is $C_j \in \mathcal{C}_j$ and $b_j \in A_{C_j}$, such that (1) BasicSDP($\mathcal{C}_j, b_j$) $\wedge$ GaussElim($\mathcal{C}_j, b_j$) rejects; and (2) $\mathcal{C}_{j+1} = \text{ReplaceConstraint}(\mathcal{C}_j, C_j, A_{C_j} \setminus \{b_j\})$. Further, some $\mathcal{C}_t \in \mathcal{C}_t$ satisfies $A_{\mathcal{C}_t} = \emptyset$.

Now consider algorithm $\mathcal{A}'$. The paragraph after the proof of Theorem 3.11 implies that BasicSDP($\mathcal{C}_j, b_j$) $\wedge$ CSinglAIP($\mathcal{C}_j, b_j$) also rejects for $1 \leq j < t$. Therefore $\mathcal{C}_1, \dots, \mathcal{C}_t$ is also a valid sequence of constraint sets during an execution of $\mathcal{A}'$. This sequence causes $\mathcal{A}'$ to reject I because some $\mathcal{C}_t \in \mathcal{C}_t$ satisfies $A_{\mathcal{C}_t} = \emptyset$. $\square$

4. SINGLETON (SDP $\wedge$ AIP) HIERARCHY

This section introduces a new hierarchy: the singleton (SDP $\wedge$ AIP) hierarchy. To motivate this hierarchy and to compare it with existing algorithms, we first look at a simpler family of decision algorithms for CSP that iteratively rule out partial assignments to *individual constraints* (Sections 4.1 to 4.3). Next, we introduce more powerful hierarchy algorithms that iteratively rule out partial assignments on *small variable subsets*, and finally, we introduce our singleton (SDP $\wedge$ AIP) hierarchy (Section 4.4).

Our family of simpler iterative algorithms captures or subsumes a few existing algorithms: It captures the hybrid algorithm in [BKM26], and subsumes LP+AIP [BGWŽ20], CLAP [CŽ22], SDA [CŽ25], and all constraint singleton algorithms in [Zhu26]. The more powerful hierarchy algorithms also capture or subsume a few existing algorithms: They capture cohomological consistency [ÓC22] and C(BLP+AIP) hierarchy [CŽ22, CN25], and subsume $\mathbb{Z}$ -affine consistency [DO24] and SDP+AIP hierarchy [CNP24]. The most powerful of all, the new singleton (SDP $\wedge$ AIP) hierarchy, subsumes all the algorithms in this paragraph.

Bhangale [Bha25] remarked it would be “interesting to define the hierarchies of [[BKM26]’s] hybrid algorithm and understand their limitations.” This remark partly motivated our singleton (SDP $\wedge$ AIP) hierarchy, which is a hierarchy analog in the hybrid algorithm in [BKM26] (and, in our opinion, *the* hierarchy analog). Another motivation for this hierarchy comes from the quest towards uniform CSP algorithms. Many algorithms have been proposed as mentioned in the previous paragraph, and our singleton (SDP $\wedge$ AIP) hierarchy subsumes them all.

4.1. Iterative Individual-Constraint Algorithm. The family of iterative local algorithms in Algorithm 2 generalizes the hybrid algorithm in Algorithm 1. It is based on repeatedly

- (1) Running a relaxation subroutine $\mathcal{A}(\mathcal{C}, b)$ to test whether a CSP instance $(V, \mathcal{C})$ has a satisfying assignment extending some partial assignment $b \in D^U$ (where $U \subseteq V$), and
- (2) Reducing the instance to exclude b if the test in (1) rejects.

The decision subroutine $\mathcal{A}(\mathcal{C}, b)$ satisfies the following “relaxation property”:

Property 4.1. If an instance $(V, \mathcal{C})$ has a satisfying assignment $a \in D^V$ such that $a_U = b$, then $\mathcal{A}(\mathcal{C}, b)$ accepts.

Algorithm 2 Iterative $\mathcal{A}$ Algorithm

```

1: repeat
2:   for all constraint  $C \in \mathcal{C}$ , assignment  $b \in A_C$  do
3:     if  $\mathcal{A}(\mathcal{C}, b)$  rejects then
4:        $\mathcal{C} \leftarrow \text{ReplaceConstraint}(\mathcal{C}, C, A_C \setminus \{b\})$ 
5: until no change to  $\mathcal{C}$ 
6: Accept if  $A_C \neq \emptyset$  for every  $C \in \mathcal{C}$ 

```

It is possible that $\mathcal{A}(\mathcal{C}, b)$ accepts even if $(V, \mathcal{C})$ has no satisfying assignment extending b .

Recall that every time line 4 is executed, we imagine replacing the current instance with a *reduced instance*.

An example of the relaxation subroutine $\mathcal{A}(\mathcal{C}, b)$ is the basic LP (when b is an assignment to some constraint $C \in \mathcal{C}$). $\text{BasicLP}(\mathcal{C}, b)$ computes the basic LP relaxation for the instance $(V, \mathcal{C})$, and accepts if b is in the support of the relative interior of the polytope of feasible solutions to the basic LP relaxation. This step appears in the LP+AIP algorithm of [BGWŻ20, Figure 1].

Similar relaxation subroutines include arc consistency, basic SDP, and basic AIP. The adjective “basic” (or “canonical” in some literature) refers to relaxations of partial assignments to *constraints and individual variables*, as opposed to “hierarchy” that are relaxations of partial assignments to *small variable subsets*.⁶

Another relaxation subroutine is the Gaussian Elimination step $\text{GaussElim}(\mathcal{C}, b)$ from the hybrid algorithm in [BKM26]. This step differs from subroutines in the previous paragraph in many ways: e.g. the set of values taken by variable v in GaussElim depends on v ’s neighborhood.

Combining different relaxation subroutines yields even more interesting ones, such as $\mathcal{A} = \text{BasicSDP} \wedge \text{GaussElim}$ as in Algorithm 1, i.e. $\mathcal{A}$ accepts if both BasicSDP and GaussElim do.

4.2. Basic Relaxation Subroutine. We now formally define the basic relaxation subroutines for four elementary relaxations: arc consistency (whose hierarchy is called local consistency or bounded width), LP, SDP, and AIP. For each elementary relaxation $\mathfrak{R}$, we will define the corresponding basic relaxation subroutine $\text{Basic}\mathfrak{R}$ in three steps:

- (1) The relaxed domains $\mathcal{D}_v^{\mathfrak{R}}$ for each variable v and $\mathcal{D}_C^{\mathfrak{R}}$ for each constraint C
- (2) The basic $\mathfrak{R}$ relaxation
- (3) The basic relaxation subroutine $\text{Basic}\mathfrak{R}(\mathcal{C}, b)$

Below $A_S \subseteq D^S$ denotes the set of partial assignments on S that satisfy all constraints in I contained in S , where a constraint $C \in \mathcal{C}$ is contained in S if every variable of C belongs to S .

Definition 4.2 (Relaxed Domain). We now define the relaxed domain $\mathcal{D}_S^{\mathfrak{R}}$ of each elementary relaxation $\mathfrak{R}$ on a subset $S \subseteq V$ of variables.

BW: $\mathcal{D}_S^{\text{BW}} \stackrel{\text{def}}{=} \mathcal{P}(A_S) \setminus \{\emptyset\}$, the family of nonempty subsets over A_S .

LP: $\mathcal{D}_S^{\text{LP}} \stackrel{\text{def}}{=} \Delta(A_S)$, the set of distributions over A_S .

SDP: $\mathcal{D}_S^{\text{SDP}}$ is the set of orthogonal vectors $\{u(a)\}_{a \in A_S} \subseteq \mathcal{X}$ indexed by A_S in an arbitrary inner product space $\mathcal{X}$, such that $\sum_{a \in A_S} \|u(a)\|^2 = 1$. This $\mathcal{X}$ is independent of S .

AIP: $\mathcal{D}_S^{\text{AIP}} \stackrel{\text{def}}{=} \left\{ w : A_S \rightarrow \mathbb{Z} \mid \sum_{a \in A_S} w(a) = 1 \right\}$, containing affine integer weights supported on A_S .

For a variable $v \in V$ and relaxation $\mathfrak{R}$, we also write $\mathcal{D}_v^{\mathfrak{R}}$ for $\mathcal{D}_{\{v\}}^{\mathfrak{R}}$.

For each elementary relaxation $\mathfrak{R}$, we also define the relaxed domain $\mathcal{D}_C^{\mathfrak{R}}$ for a constraint C as in Theorem 4.2, except that A_S is replaced with A_C , the set of satisfying assignments on C .

⁶For this reason, the adjective “basic” should be avoided when discussing hierarchies: LP hierarchy instead of BLP hierarchy, where the B stands for “basic”.

For each elementary relaxation, we consider a relaxed assignment $\alpha_S \in \mathcal{D}_S$ to be a function from the set of satisfying assignments A_S on S to a commutative monoid $\mathcal{M}$.

BW: $\mathcal{M}_{\text{BW}} \stackrel{\text{def}}{=} \mathbb{B}$. The Boolean algebra $\mathbb{B}$ has join $\vee$ as the addition and meet $\wedge$ as the multiplication.

A set $\alpha_S \in \mathcal{P}(A_S)$ is represented by its indicator function $\mathbb{1}_{\alpha_S} : A_S \rightarrow \mathbb{B}$, so that $\mathbb{1}_{\alpha_S}(a) = 1$ if $a \in \alpha_S$, and $\mathbb{1}_{\alpha_S}(a) = 0$ otherwise.

LP: $\mathcal{M}_{\text{LP}} \stackrel{\text{def}}{=} \mathbb{R}_+$, the nonnegative reals.

SDP: $\mathcal{M}_{\text{SDP}} \stackrel{\text{def}}{=} \mathcal{X}$.

AIP: $\mathcal{M}_{\text{AIP}} \stackrel{\text{def}}{=} \mathbb{Z}$.

For any subsets $T \subseteq S \subseteq V$, there is a projection $\pi_{S \rightarrow T} : \mathcal{M}^{D^S} \rightarrow \mathcal{M}^{D^T}$ of functions taking values in $\mathcal{M}$:

$$(14) \quad \pi_{S \rightarrow T}(\alpha_S)(b) \stackrel{\text{def}}{=} \sum_{\substack{a \in D^S \\ a_T = b}} \alpha_S(a) \quad \text{for } \alpha_S \in \mathcal{M}^{D^S}, b \in D^T.$$

Here the sum is over $\mathcal{M}$.

Definition 4.3 (Basic $\mathfrak{R}$ relaxation). For each elementary relaxation $\mathfrak{R}$, a solution $s^{\mathfrak{R}}$ to the basic relaxation is a function mapping $T \in V \cup \mathcal{C}$ to $\mathcal{D}_T^{\mathfrak{R}}$, that is consistent with projection:

$$(15) \quad \pi_v(s^{\mathfrak{R}}(C)) = s^{\mathfrak{R}}(v) \quad \text{for } C \in \mathcal{C}, v \in C.$$

Applying [Theorem 4.3](#) to the four elementary relaxations BW, LP, SDP, and AIP, we get arc consistency, basic LP, basic SDP, and basic AIP relaxations, respectively. The basic SDP relaxation from [Theorems 4.2](#) and [4.3](#) is not the traditional definition [[Rag08](#)], but is equivalent ([[BGS25](#), Section 6.3.2], [[CŽ26](#), Section 4]). Similar basic $\mathfrak{R}$ relaxations have been proposed by [[BGS25](#), Remark 6.11] using the language of minions.

Turning the basic $\mathfrak{R}$ relaxation into a subroutine $\text{Basic}\mathfrak{R}(\mathcal{C}, b)$ requires an additional step. For LP, the additional step is to consider the relative interior mentioned earlier. This means looking for a basic LP solution s^{LP} whose support $\text{supp}(s^{\text{LP}}(C))$ is maximal for each constraint C . Then $\text{BasicLP}(\mathcal{C}, b)$ simply accepts if $b \in \text{supp}(s^{\text{LP}}(C))$, where $b \in A_C$. Similarly for SDP, there also exists a basic SDP solution s^{SDP} with maximal support ([[CNP24](#), Lemma 12.5], [[BKM26](#), Claim 7.11]). Again, $\text{BasicSDP}(\mathcal{C}, b)$ accepts if $b \in \text{supp}(s^{\text{SDP}}(C))$. The same idea of looking for a basic $\mathfrak{R}$ solution with maximal support also works for BW and AIP, but is inferior to singleton subroutine discussed next.

4.3. Constraint Singleton Subroutine. Stronger relaxation subroutines arise by considering reduced instances that force a particular partial assignment. Given an instance $I = (V, \mathcal{C})$, some constraint $C \in \mathcal{C}$, and assignment $b \in A_C$, define the reduced instance $I + b \stackrel{\text{def}}{=} (V, \mathcal{C}_b)$, where $\mathcal{C}_b \stackrel{\text{def}}{=} \text{ReplaceConstraint}(\mathcal{C}, C, \{b\})$ is the same as $\mathcal{C}$ but the assignment b is forced on C . Note that I has a satisfying assignment extending b if and only if $I + b$ is satisfiable. Define a relaxation subroutine $\text{CSingl}\mathfrak{R}(\mathcal{C}, b)$ that accepts precisely when the reduced instance $I + b = (V, \mathcal{C}_b)$ has a basic $\mathfrak{R}$ relaxation solution. The constraint singleton $\mathfrak{R}$ algorithm is the iterative $\text{CSingl}\mathfrak{R}$ algorithm, i.e. [Algorithm 2](#) with $\mathcal{A} = \text{CSingl}\mathfrak{R}$.

The singleton algorithm was proposed by [[DB97](#)] for arc consistency, i.e. basic BW relaxation in our terminology. The constraint counterparts were proposed by [[CŽ22](#)] and their minions were studied systematically in [[Zhu26](#)].

Note that all the algorithms in [Sections 4.1](#) to [4.3](#) so far are iterative: they keep rejecting partial assignments to constraints by calling subroutines repeatedly. There are also non-iterative versions of such algorithms. For example, for each relaxation $\mathfrak{R}$, consider a variant of the $\text{Basic}\mathfrak{R}(\mathcal{C})$ algorithm that accepts precisely when there is a solution to the basic $\mathfrak{R}$ relaxation of the instance $I = (V, \mathcal{C})$, without enforcing any partial assignment b . Such algorithms are the original, non-singleton version of arc consistency, basic LP, basic SDP, and basic AIP.

Our (iterative) $\text{BasicSDP} \wedge \text{CSinglAIP}$ algorithm differs from all the (non-iterative) $\mathfrak{R} \wedge \mathfrak{S}$ algorithms and (iterative) $\text{CSingl}(\mathfrak{R} \wedge \mathfrak{S})$ algorithms in [Zhu26] — all the iterative algorithms in [Zhu26] are of the form $\text{Singl}\mathfrak{R}$ or $\text{CSingl}\mathfrak{R}$ for some (possibly combined) relaxation $\mathfrak{R}$. By contrast, our iterative $\text{BasicSDP} \wedge \text{CSinglAIP}$ algorithm consists of both the non-constraint-singleton part BasicSDP and the constraint-singleton part AIP . Of course, our $\text{BasicSDP} \wedge \text{CSinglAIP}$ can be strengthened to $\text{CSingl}(\text{SDP} \wedge \text{AIP})$, but we prefer to focus on the weaker and simpler version in this paper. $\text{BasicSDP} \wedge \text{CSinglAIP}$ is much closer to the [BKM26] hybrid algorithm, and is still a candidate uniform algorithm for tractable CSPs and the Approximation Dichotomy Conjecture.

We will omit the adjective “iterative” when there is no danger of ambiguity. In this paper, the $\text{BasicSDP} \wedge \text{CSinglAIP}$ algorithm always refers to the iterative version.

Theorem 4.5 below says that our $\text{BasicSDP} \wedge \text{CSinglAIP}$ algorithm ([Algorithm 2](#)) is equivalently looking for certain non-iterative basic solutions ([Theorem 4.4](#)), by considering a family of basic AIP solutions. Related results have appeared before for CLAP [CŽ22] and many singleton algorithms [Zhu26] in terms of minions.

Definition 4.4. A $\text{BasicSDP} \wedge \text{CSinglAIP}$ solution to an instance $I = (V, \mathcal{C})$ consists of:

- A basic SDP solution s^{SDP} to I ; and
- A collection $\{s_b^{\text{AIP}}\}$ of basic AIP solutions to I , one for each $b \in \text{supp}(s^{\text{SDP}}(C)), C \in \mathcal{C}$, such that
 - (1) $\text{supp}(s_b^{\text{AIP}}(C)) \subseteq \text{supp}(s^{\text{SDP}}(C))$ for $C, C' \in \mathcal{C}, b \in \text{supp}(s^{\text{SDP}}(C'))$; and
 - (2) $s_b^{\text{AIP}}(C) = 1_b$ for each $C \in \mathcal{C}, b \in \text{supp}(s^{\text{SDP}}(C))$.

Theorem 4.5. $\text{BasicSDP} \wedge \text{CSinglAIP}$ Algorithm accepts an instance I if and only if I has a $\text{BasicSDP} \wedge \text{CSinglAIP}$ solution.

Proof. If $\text{BasicSDP} \wedge \text{CSinglAIP}$ Algorithm ([Algorithm 2](#)) accepts, let $\{B_C\}$ denotes the set of satisfying assignments to the constraints of the final reduced instance I_{final} at the end of the algorithm. From the final execution of the lines 2–4, there exists a basic SDP solution s^{SDP} for I_{final} , and basic AIP solutions s_b^{AIP} for $I_{\text{final}} + b$, one for each $b \in B_C, C \in \mathcal{C}$. Then $\text{supp}(s^{\text{SDP}}(C)) = B_C$ for $C \in \mathcal{C}$, so $\text{supp}(s^{\text{SDP}}(C)) = B_C \supseteq \text{supp}(s_b^{\text{AIP}}(C))$ for $C \in \mathcal{C}, b \in B_C = \text{supp}(s^{\text{SDP}}(C))$. Also, $s_b^{\text{AIP}}(C) = 1_b$ because s_b^{AIP} is a basic AIP solution to $I_{\text{final}} + b$. Therefore $(s^{\text{SDP}}, \{s_b^{\text{AIP}}\})$ is a solution as in [Theorem 4.4](#).

Conversely, if a $\text{BasicSDP} \wedge \text{CSinglAIP}$ solution $(s^{\text{SDP}}, \{s_B^{\text{AIP}}\})$ to I exists as in [Theorem 4.4](#). Define subsets $B_C \stackrel{\text{def}}{=} \text{supp}(s^{\text{SDP}}(C))$ for $C \in \mathcal{C}$. Each B_C is nonempty since $\sum_{b \in B_C} \|s^{\text{SDP}}(C, b)\|^2 = 1$. We prove by induction that $B_C \subseteq A_C$ for $C \in \mathcal{C}$ throughout the algorithm (where A_C denotes the current set of satisfying assignments to constraint C), which will then imply $\text{BasicSDP} \wedge \text{CSinglAIP}$ Algorithm ([Algorithm 2](#)) accepts I .

Base Case: Initially, $B_C \subseteq A_C$.

Induction Step: Suppose $B_C \subseteq A_C$ before the if statement on lines 3–4 of [Algorithm 2](#). For every $C \in \mathcal{C}, b \in B_C$, we have $b \in \text{supp}(s^{\text{SDP}}(C))$, so $\text{BasicSDP}(\{B_C\}, b)$ (and hence $\text{BasicSDP}(\{A_C\}, b)$) accepts. Also, s_b^{AIP} is a basic AIP solution to the reduced instance $I' = (V, \{B_C\}) + b$ (and hence $(V, \{A_C\}) + b$), so $\text{CSinglAIP}(\{A_C\}, b)$ also accepts. Therefore b is never removed from A_C . Thus $B_C \subseteq A_C$ still holds at the end of the if statement on lines 3–4. $\square$

4.4. Hierarchy and Its Singleton Algorithm. For each elementary relaxation $\mathfrak{R}$, a level- d $\mathfrak{R}$ hierarchy solution $s^{\mathfrak{R}}$ is a function mapping $S \in \binom{V}{\leq d}$ to $\mathcal{D}_S^{\mathfrak{R}}$, that is consistent with projection:

$$(16) \quad \pi_T(s^{\mathfrak{R}}(S)) = s^{\mathfrak{R}}(T) \quad \text{for } T \subseteq S \subseteq V, |S| \leq d.$$

Given a relaxation $\mathfrak{R}$, we define the singleton $\mathfrak{R}$ hierarchy algorithm in [Algorithm 3](#) ($\text{Singl}\mathfrak{R}$ for short). It keeps track of collections B_U of partial assignments $b \in D^U$ for variable subsets U of size at most d that has not been rejected by the $\mathfrak{R}$ hierarchy.

Algorithm 3 Level- d Singleton $\mathfrak{R}$ Hierarchy Algorithm

```

1: Initialize  $B_U \leftarrow D^U$  for  $U \in \binom{V}{\leq d}$ 
2: repeat
3:   for all variable subset  $U \in \binom{V}{\leq d}$ , assignment  $b \in B_U$  do
4:     if instance  $(V, \mathcal{C} \cup \{\{b\}\})$  has no level- $d$   $\mathfrak{R}$  hierarchy solution then
5:        $\mathcal{C} \leftarrow \mathcal{C} \cup \{D^U \setminus \{b\}\}$ 
6:        $B_U \leftarrow B_U \setminus \{b\}$ 
7: until no change to  $\mathcal{C}$ 
8: Accept if  $B_U \neq \emptyset$  for every  $U \in \binom{V}{\leq d}$ 

```

On Line 4, $\mathcal{C} \cup \{\{b\}\}$ represents adding to $\mathcal{C}$ an extra constraint (with relation $\{b\}$) that forces the partial assignment b to U . Note that $\mathcal{C}$ has a satisfying assignment extending b if and only if $\mathcal{C} \cup \{\{b\}\}$ is satisfiable. Line 5 further forbids b in the future.

Algorithm 3 modifies Algorithm 2 in two ways:

- Algorithm 2 iteratively rejects partial assignments to individual constraints, while Algorithm 3 iteratively rejects partial assignments to small variable subsets
- The relaxation subroutine $\mathcal{A}$ in Algorithm 2 is now the $\mathfrak{R}$ hierarchy subroutine applied to $\mathcal{C} \cup \{\{b\}\}$, justifying the name “singleton hierarchy $\mathfrak{R}$ algorithm” for Algorithm 3

Cohomological consistency [ÓC22] is equivalent to the $\text{Singl}(\text{BW} \wedge \text{AIP})$ hierarchy algorithm. Given two relaxations $\mathfrak{R}$ and $\mathfrak{S}$, a $\mathfrak{R} \wedge \mathfrak{S}$ hierarchy solution $s^{\mathfrak{R} \wedge \mathfrak{S}}$ is a pair of solutions $s^{\mathfrak{R}}$ and $s^{\mathfrak{S}}$ to the $\mathfrak{R}$ and $\mathfrak{S}$ hierarchies, respectively. $\text{C}(\text{BLP} + \text{AIP})$ hierarchy [CŽ22, CN25] is equivalent to the $\text{Singl}(\text{LP} \wedge \text{AIP})$ hierarchy algorithm.

Our $\text{Singl}(\text{SDP} \wedge \text{AIP})$ hierarchy algorithm subsumes cohomological consistency, $\text{C}(\text{BLP} + \text{AIP})$ hierarchy, $\text{SDP} + \text{AIP}$ hierarchy [CNP24], as well as the hybrid algorithm of [BKM26].

Remark 4.6. Many previous works, e.g. [BGWŽ20], considered a different combination $\mathfrak{R} + \mathfrak{S}$, whose solution $s^{\mathfrak{R} + \mathfrak{S}}$ is a pair of solutions $s^{\mathfrak{R}}$ and $s^{\mathfrak{S}}$ to the $\mathfrak{R}$ and $\mathfrak{S}$ relaxations that additionally satisfies $\text{supp}(s^{\mathfrak{R}}(T)) \supseteq \text{supp}(s^{\mathfrak{S}}(T))$ for every variable subset/constraint T , whenever $s^{\mathfrak{R}}(T)$ and $s^{\mathfrak{S}}(T)$ are defined. Such $\mathfrak{R} + \mathfrak{S}$ relaxations are often considered without the iterative Algorithm 2 or Algorithm 3. Iterative $\text{CSingl}(\mathfrak{R} \wedge \mathfrak{S})$ algorithm (Algorithm 2) actually subsumes the non-iterative $\text{Basic}(\mathfrak{R} + \mathfrak{S})$ algorithm. Similarly, (iterative) $\text{Singl}(\mathfrak{R} \wedge \mathfrak{S})$ hierarchy algorithm (Algorithm 3) subsumes the non-iterative $\mathfrak{R} + \mathfrak{S}$ hierarchy algorithm (of the same level).

Theorem 4.8 below says that the (iterative) $\text{Singl}(\text{SDP} \wedge \text{AIP})$ hierarchy algorithm (Algorithm 3) accepts if and only if a (non-iterative) $\text{Singl}(\text{SDP} \wedge \text{AIP})$ hierarchy solution (Theorem 4.7) exists, up to at most a factor 2 loss in level. A $\text{Singl}(\text{SDP} \wedge \text{AIP})$ hierarchy solution consists of both an SDP hierarchy solution and a *family* of AIP hierarchy solutions.

Definition 4.7. A level- d $\text{Singl}(\text{SDP} \wedge \text{AIP})$ solution to an instance $I = (V, \mathcal{C})$ consists of:

- A level- d SDP solution s^{SDP} to I ; and
- A collection $\{s_b^{\text{AIP}}\}$ of level- d AIP solutions to I , one for each $b \in \text{supp}(s^{\text{SDP}}(U)), U \in \binom{V}{\leq d}$, such that
 - (1) $\text{supp}(s_b^{\text{AIP}}(T)) \subseteq \text{supp}(s^{\text{SDP}}(T))$ for $U, T \in \binom{V}{\leq d}, b \in \text{supp}(s^{\text{SDP}}(U))$; and
 - (2) $s_b^{\text{AIP}}(U) = 1_b$ for each $U \in \binom{V}{\leq d}, b \in \text{supp}(s^{\text{SDP}}(U))$.

Theorem 4.8. *If the level- d $\text{Singl}(\text{SDP} \wedge \text{AIP})$ algorithm accepts an instance I , then I has a level- d $\text{Singl}(\text{SDP} \wedge \text{AIP})$ solution. Conversely, if I has a level- $2d$ $\text{Singl}(\text{SDP} \wedge \text{AIP})$ solution, then the level- d $\text{Singl}(\text{SDP} \wedge \text{AIP})$ algorithm accepts I .*

Proof. If level- d Algorithm 3 accepts, let $\{B_U\}$ be the non-empty subsets at the end of the algorithm, and $\mathcal{C}_{\text{final}}$ be the constraint set $\mathcal{C}$ at the end of the algorithm. From the final execution of Lines

3–8, there are level- d SDP solutions s_b^{SDP} and level- d AIP solutions s_b^{AIP} for the reduced instances $(V, \mathcal{C}_{\text{final}} \cup \{\{b\}\})$, for every $b \in B_U, U \in \binom{V}{\leq d}$. Define a new level- d SDP solution s^{SDP} for I by

$$s^{\text{SDP}}(T, a) \stackrel{\text{def}}{=} \frac{1}{\sqrt{Z}} \bigoplus_{U \in \binom{V}{\leq d}, b \in B_U} s_b^{\text{SDP}}(T, a), \quad \text{where } Z \stackrel{\text{def}}{=} \left| \biguplus \{B_U \mid U \in \binom{V}{\leq d}\} \right|$$

for $a \in B_T, T \in \binom{V}{\leq d}$. s^{SDP} is an SDP solution for I , because for $T \in \binom{V}{\leq d}$, we have $\text{supp}(s^{\text{SDP}}(T)) \subseteq \bigcup_{U \in \binom{V}{\leq d}, b \in B_U} \text{supp}(s_b^{\text{SDP}}(T)) \subseteq A_T$. Also, for distinct $a, a' \in D^T$,

$$\langle s^{\text{SDP}}(T, a), s^{\text{SDP}}(T, a') \rangle = \frac{1}{\sqrt{Z}} \sum_{U \in \binom{V}{\leq d}, b \in B_U} \langle s_b^{\text{SDP}}(T, a), s_b^{\text{SDP}}(T, a') \rangle = 0.$$

Further,

$$\sum_{a \in D^T} \|s^{\text{SDP}}(T, a)\|^2 = \frac{1}{Z} \sum_{U \in \binom{V}{\leq d}, b \in B_U} \sum_{a \in D^T} \|s_b^{\text{SDP}}(T, a)\|^2 = 1.$$

For $\mathfrak{R} \in \{\text{SDP}, \text{AIP}\}$, $U, T \in \binom{V}{\leq d}, b \in D^U$, we have $\text{supp}(s_b^{\mathfrak{R}}(T)) \subseteq B_T$, because for every $a \in D^T \setminus B_T$, there is some constraint in $\mathcal{C}_{\text{final}}$ with relation $D^T \setminus \{a\}$. This implies $\text{supp}(s^{\text{SDP}}(T)) \subseteq \bigcup_{U \in \binom{V}{\leq d}, b \in B_U} \text{supp}(s_b^{\text{SDP}}(T)) \subseteq B_T$. And $\text{supp}(s^{\text{SDP}}(U)) \supseteq \text{supp}(s_b^{\text{SDP}}(U)) \ni b$ for $U \in \binom{V}{\leq d}, b \in B_U$, because s^{SDP} is a solution for the instance $(V, \mathcal{C}_{\text{final}} \cup \{\{b\}\})$. Therefore $\text{supp}(s^{\text{SDP}}(T)) = B_T \supseteq \text{supp}(s_b^{\text{AIP}}(T))$ for $U, T \in \binom{V}{\leq d}, b \in B_U$. Finally, for $U, T \in \binom{V}{\leq d}, b \in \text{supp}(s^{\text{SDP}}(U)) = B_U$, $s_b^{\text{AIP}}(T) = \mathbb{1}_b$ because s_b^{AIP} is a solution for the instance $(V, \mathcal{C}_{\text{final}} \cup \{\{b\}\})$. Therefore I has a level- d solution in the sense of [Theorem 4.7](#).

Conversely, suppose some level- $2d$ solution $(s^{\text{SDP}}, \{s_b^{\text{AIP}}\})$ to $I = (V, \mathcal{C})$ exists as in [Theorem 4.7](#). Define subsets $B'_U \stackrel{\text{def}}{=} \text{supp}(s^{\text{SDP}}(U))$ for $U \in \binom{V}{\leq d}$. Each B'_U is nonempty since $\sum_{a \in B'_U} \|s^{\text{SDP}}(U, a)\|^2 = 1$. We prove by induction that $B'_U \subseteq B_U$ for $U \in \binom{V}{\leq d}$ throughout the algorithm (where B_U denotes the current set of partial assignments to variable subset U), which will then imply level- d [Algorithm 3](#) accepts I .

Base Case: Initially, $B'_U \subseteq B_U = D^U$.

Induction Step: Suppose $B'_U \subseteq B_U$ for $U \in \binom{V}{\leq d}$ before the if statement on lines 4–6 of [Algorithm 3](#). Let $\mathcal{C}' \stackrel{\text{def}}{=} \mathcal{C} \cup \bigcup_{T \in \binom{V}{\leq d}, a \in D^T \setminus B'_T} \{D^T \setminus \{a\}\}$. Induction Hypothesis implies $\mathcal{C}'$ is a superset of the constraint set at the current iteration. We argue that for every $U \in \binom{V}{\leq d}, b \in B'_U$, there is a level- d SDP $\wedge$ AIP hierarchy solution to $(V, \mathcal{C}' \cup \{\{b\}\})$. For SDP, since $b \in B'_U = \text{supp}(s^{\text{SDP}}(U))$, we may condition the solution s^{SDP} on b to get a level- d SDP solution s_b^{SDP} to I ([Theorem C.1](#)), so that $\text{supp}(s_b^{\text{SDP}}(U)) = \{b\}$ and $\text{supp}(s_b^{\text{SDP}}(T)) \subseteq \text{supp}(s^{\text{SDP}}(T))$ for $T \in \binom{V}{\leq d}$. Then s_b^{SDP} is an SDP solution to $(V, \mathcal{C}' \cup \{\{b\}\})$. For AIP, since $s_b^{\text{AIP}}(U) = \mathbb{1}_b$ and $\text{supp}(s_b^{\text{AIP}}(T)) \subseteq \text{supp}(s^{\text{SDP}}(T)) = B'_T$ for $T \in \binom{V}{\leq d}$, s_b^{AIP} is a level- d AIP solution to $(V, \mathcal{C}' \cup \{\{b\}\})$.

Since there is a level- d SDP $\wedge$ AIP solution to $(V, \mathcal{C}' \cup \{\{b\}\})$, b is never removed from B_U for $U \in \binom{V}{\leq d}, b \in B'_U$. Thus $B'_U \subseteq B_U$ still holds at the end of the if statement on lines 4–6. $\square$

5. SDP+AIP LOWER BOUNDS FOR $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$

In this section, we prove lower bounds for the SDP+AIP hierarchy on a tractable CSP, namely $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$ from [Section 2.3](#).

It is well-known in the literature that LP and SDP hierarchies are powerless against linear equations (of arity $k \geq 3$) because they are “pairwise uniform” (or “ τ -wise uniform” in general), which means each constraint has a probability distribution on its satisfying assignments such that

the marginal distribution on each subset of size 2 (or τ) is uniform. Therefore, the main difficulty here is to fool the AIP hierarchy.

As noted in [Zhu26], we can use the disconnectedness of this CSP to fool the basic AIP relaxation. We consider “ d -solutions” for AIP, which are basic AIP relaxation solutions except the sum of integer weights is d instead of 1. [Zhu26] showed that by assigning a uniform distribution to each constraint on the satisfying assignments on the $\mathbb{Z}_2$ part, we can construct a basic LP relaxation solution whose probability masses are multiples of $1/4$ (assuming $k = 3$ as in [Zhu26]). Multiplying all probability masses by 4 then yields a 4-solution for AIP. Repeating this on the $\mathbb{Z}_3$ part, we obtain a 9-solution for AIP. By subtracting the 8-solution (obtained by doubling the 4-solution) from the 9-solution, we obtain a 1-solution, which is just a basic AIP relaxation solution.

This approach is readily generalizable to the AIP hierarchy. For CSPs that are linear equations, the standard construction of LP hierarchy solutions is also to assign a uniform distribution over satisfying assignments in some carefully chosen vertex subsets. This means that we can recover a “ 2^n -hierarchy-solution” and a “ 3^n -hierarchy-solution” (where $n = |V|$ is the number of variables in the CSP instance), and subsequently a “1-hierarchy-solution” using an appropriate linear combination, which is the AIP hierarchy solution we needed.

We will formalize this observation in the rest of this section.

5.1. Local closures and sparsity. The first step to construct a hierarchy solution is to find some carefully chosen vertex subsets in the CSP instance. Intuitively, each of these vertex subsets contains sufficient information, and the constraints outside cannot impose any additional restrictions. This allows us to construct solutions more easily by ignoring all constraints outside.

This idea has appeared in many previous works under different names, such as advice sets [BGMT12, Section 3.1], closures [KMOW17, Definition 5.3], and local closures [CNP24, Definition 8.7]. Here, we follow the notations of [CNP24, CNP25].

The following definitions are formulated over k -uniform hypergraphs. S and $\mathcal{E}$ denote vertex subsets and hyperedge subsets in the same hypergraph respectively.

Definition 5.1 ([CNP25, Example 5.12]). Define the τ -wise boundary operator

$$\mathcal{B}_S(\mathcal{E}) \stackrel{\text{def}}{=} \{e \in \mathcal{E} \mid \text{at most } \tau \text{ vertices } u \in e \text{ satisfy } u \in S \text{ or } \deg_{\mathcal{E}}(u) > 1\}.$$

Definition 5.2 (S -closed, [CNP25, Remark 5.16]). $\mathcal{E}$ is S -closed if $\mathcal{B}_S(\mathcal{E}) = \emptyset$.

Definition 5.3 (Local closure, [CNP25, Definition 5.6]). Given $t \in \mathbb{N}$, the t -local S -closure $\text{cl}_S^t(\mathcal{E})$ of $\mathcal{E}$ is the union of t -small S -closed $\mathcal{F} \subseteq \mathcal{E}$.

Note that local closures are in fact hyperedge subsets, not vertex subsets as discussed. However, the difference is purely notational. See [CNP25, Section 6.1] for an abstraction of this idea (“insular” subsets) in terms of vertex subsets.

5.2. Random instances. Following [CNP25, CdX26], we consider the model of random instances in [CM13]. In this model, there is a distribution ρ over $\mathcal{R}$. When ρ is uniform over $\mathcal{R}$, this model coincides with the natural model of random CSPs in many previous works [BGMT12, KMOW17, CNP24].

Definition 5.4 (ρ -random instance, [CNP25, Definition 2.1]). Fix a k -CSP $(D, \mathcal{R})$, a distribution ρ over $\mathcal{R}$ with $\text{supp}(\rho) = \mathcal{R}$, a finite set V of variables, and $m \in \mathbb{N}$.

- (1) A random constraint $C = (S, R)$ is chosen by picking a sequence $S \in V^k$ of k distinct variables uniformly, and R according to ρ .
- (2) A random instance consists of choosing with replacement m uniformly random constraints.

Equivalently, a random instance consists of first picking a k -uniform hypergraph whose hyperedges are chosen uniformly and independently, and then choosing the relation R of each hyperedge (i.e. constraint) independently from $\mathcal{R}$ according to ρ .

In a random hypergraph, the set of hyperedges contained inside a small vertex subset S is very sparse. Assuming the CSP is pairwise uniform, then this sparsity property intuitively means that the set of constraints that could impose restrictions onto a small vertex subset S is also small. This property is used universally in the literature to construct hierarchy lower bounds.

In the following, we follow [CNP24] and track this sparsity property using the notion of “ (t, γ) -expanding”. Its precise definition is not useful in this work, and we refer readers to [CNP24, Section 8.1] for the full technical context. In particular, the measure of sparsity $\text{sp}(\cdot, \cdot)$ is defined in [CNP24, Definition 8.1].

Definition 5.5 (Sparsity expansion, [CNP24, Definition 8.4]). Let $t \geq 0$ and $\gamma > 0$. A constraint set $\mathcal{C}$ over V is (t, γ) -expanding if $\text{sp}(\mathcal{C}', S) \geq \gamma |\mathcal{C}'|$ for any S -closed t -small constraint subset $\mathcal{C}' \subseteq \mathcal{C}$ and any $S \subseteq V$. An instance I is (t, γ) -expanding if its constraint set is.

In the following theorem, $o_{\zeta; k}(1)$ represents a function $\varepsilon_k(\zeta)$ that, for every fixed choice of the parameter k , $\varepsilon_k(\zeta) \rightarrow 0$ as $\zeta \rightarrow 0$ from above.

Theorem 5.6 ([CNP24, Theorem B.5]). Let $\lambda \stackrel{\text{def}}{=} \tau - 1 \geq 1$, $0 < \gamma \leq \lambda/2$, $\zeta \stackrel{\text{def}}{=} \Delta^{2/(\lambda-\gamma)}/n$. Except with probability $o_{\zeta; k}(1)$, a random k -CSP instance with n vertices and $m \stackrel{\text{def}}{=} \Delta n$ constraints is (t, γ) -expanding, where

$$t = \frac{n}{\Delta^{2/(\lambda-\gamma)}} \cdot \frac{1}{2O(k)}.$$

5.3. Standard LP and SDP hierarchy solution. As is standard in the literature (e.g. [BGMT12, KMO17, CNP24, CN25]), we construct LP solutions on carefully chosen vertex subsets by sampling an assignment independently on each constraint and each isolated variable, conditioned on agreement on common variables.

Definition 5.7. Suppose $J = (V, \mathcal{C})$ is an instance. Define the μ_J to be the following distribution over satisfying assignments $b \in A_J$:

- (1) Draw b_v uniformly from $D^{\{v\}}$ independently for isolated variable $v \in V \setminus V(\mathcal{C})$; and
- (2) Draw $b_{V(C)}$ from μ_C independently for $C \in \mathcal{C}$, conditioned on agreeing at their common variables.

Note that this definition implicitly depends on the distribution assigned to each isolated variable and each constraint, which we shall specify later.

We shall consider μ to be a function mapping instances J to μ_J .

The following result is a simple case of [CNP24, Theorem 11.26, Lemma 11.27]. Specifically, we choose $K \stackrel{\text{def}}{=} \emptyset$, $\mathcal{E} \stackrel{\text{def}}{=} \emptyset$, and $\eta(C) \stackrel{\text{def}}{=} A_C$ for all constraints $C \in \mathcal{C}$ in the theorem to obtain the following simplification. See [CNP24, Section 11] for the definition of the inner product space $\mathcal{X}$ used in the SDP solution.

Theorem 5.8 ([CNP24, Theorem 11.26, Lemma 11.27]). Suppose an instance $I = (V, \mathcal{C})$ of a τ -wise uniform CSP is $(2t, 3\gamma)$ -expanding, and each constraint $C \in \mathcal{C}$ has a τ -wise uniform distribution with full support on its satisfying assignments A_C . Let $d \stackrel{\text{def}}{=} t\gamma/2$. Define

$$(17) \quad s_R \stackrel{\text{def}}{=} \pi_R \circ \mu \circ \text{cl}_R^t(I) \quad \text{for } R \in \binom{V}{\leq 2d}; \text{ and}$$

$$(18) \quad \alpha_S(a_S) \stackrel{\text{def}}{=} \mathbb{1}_{a_S} \quad \text{for } S \in \binom{V}{\leq d}, a_S \in A_S.$$

Then $\alpha \stackrel{\text{def}}{=} (\alpha_S)_S$ is a level- d SDP hierarchy solution inducing the level- $2d$ LP hierarchy solution $s \stackrel{\text{def}}{=} (s_R)_R$.

5.4. Linear equations. Given $m, k \in \mathbb{N}$, consider k -ary mod- m constraints $R_{k,m,b} \stackrel{\text{def}}{=} \{a \in \mathbb{Z}_m^k \mid \sum_{i \in [k]} a_i = b\}$ defined with respect to some $b \in \mathbb{Z}_m$. Consider the k -ary mod- m CSP $k\text{-Lin}_m \stackrel{\text{def}}{=} (\mathbb{Z}_m, \{R_{k,m,b}\}_{b \in \mathbb{Z}_m})$ formed by such constraints.

We check that [Theorem 5.8](#) is applicable to instances of $k\text{-Lin}_m$ (with $k \geq 3$ so that $\tau \stackrel{\text{def}}{=} k-1 \geq 2$).

Lemma 5.9. *For any $m, k \in \mathbb{N}$, $k\text{-Lin}_m$ is $(k-1)$ -wise uniform, witnessed by the uniform distribution over all satisfying assignments for each $R_{k,m,b}$ (over $b \in \mathbb{Z}_m$).*

Proof. Take any $b \in \mathbb{Z}_m$. Let η be the uniform distribution over all satisfying assignments of $R_{k,m,b}$. For any subset of $(k-1)$ coordinates $S \stackrel{\text{def}}{=} [k] \setminus \{i\}$ (where $i \in [k]$), we need to show that the marginal distribution $\pi_S(\eta)$ is the uniform distribution over $\mathbb{Z}_m^S$.

Indeed, for any fixed choice of assignment $a \in \mathbb{Z}_m^S$, there is a unique extension $c \in \mathbb{Z}_m^{\{i\}}$, given by $c \stackrel{\text{def}}{=} b - \sum_{j \in [k] \setminus \{i\}} a_j$, such that $a \cup c \in \mathbb{Z}_m^{[k]}$ is a satisfying assignment of $R_{k,m,b}$. This means that $\pi_S(\eta)(a) = \eta(a \cup c) = 1/m^{k-1}$. The same holds for all $a \in \mathbb{Z}_m^S$, so $\pi_S(\eta)$ is indeed the uniform distribution over S . $\square$

The following lemma allows us to generalize the idea in [\[Zhu26\]](#) from the basic LP/AIP relaxation to the LP/AIP hierarchy.

Lemma 5.10. *In the context of [Theorem 5.8](#), let $k \geq 3$ be an integer, m be a prime number, and $I = (V, C)$ be an instance of $k\text{-Lin}_m$. Construct the level-2d LP solution s using the uniform distribution on isolated variables and (the satisfying assignments of) constraints in [Theorem 5.7](#). Then, every probability mass $s(S)(a)$ (where $S \in \binom{V}{\leq 2d}$, $a \in \mathbb{Z}_m^S$) in s satisfies $m^{|V|} \cdot s(S)(a) \in \mathbb{N}_0$.*

Proof. Take any $S \in \binom{V}{\leq 2d}$. Let $J \stackrel{\text{def}}{=} \text{cl}_S^t(I)$ and $T \stackrel{\text{def}}{=} V(J)$. Let us firstly consider μ_J . Note that $s(S) \stackrel{(17)}{=} \pi_S \circ \mu_J$.

Claim 5.11. μ_J is the uniform distribution over all satisfying assignments of J .

Proof. The probability that a fixed satisfying assignment $b \in A_J$ is sampled by μ_J is proportional to the product of probabilities of the following events:

- Each isolated variable v in J sampling b_v . Each such event happens with probability $1/m$ regardless of b .
- Each constraint C in J sampling $b_{V(C)}$ in the scope of C . Each such event happens with probability $1/m^{k-1}$ provided b is a satisfying assignment of J .

Therefore, this product of probabilities is the same for all satisfying assignments $b \in A_J$, meaning μ_J is indeed the uniform distribution over all satisfying assignments of J . $\square$

Claim 5.12. *The number of satisfying assignments in J is m^t for some non-negative integer $0 \leq t \leq |T|$.*

Proof. Each constraint in J is a linear equation in $\mathbb{F}_m$. Consider all these constraints as a system of linear equations on $|T|$ variables over $\mathbb{F}_m$. Such a system either has no solutions or has exactly m^t solutions, where t is the dimension of the null space of the coefficient matrix of the system.

This system must have at least one solution because μ_J is a well-defined probability distribution on A_J , which means A_J is nonempty. Also, $0 \leq t \leq |T|$ because there are only $|T|$ variables in the system. $\square$

Claim 5.13. *Let m^t be given by the above claim. Every probability mass $\mu_J(b)$ (where $b \in \mathbb{Z}_m^T$) in μ_J is either 0 or $1/m^t$.*

Proof. If b is not a satisfying assignment of J , then $\mu_J(b) = 0$. Otherwise b is a satisfying assignment of J , and $\mu_J(b) = 1/m^t$ because μ_J is a uniform distribution over its satisfying assignments. $\square$

We are now ready to prove [Theorem 5.10](#). The probability mass $s(S)(a)$ can be written as

$$s(S)(a) \stackrel{(17)}{=} \pi_S(\mu_J) = \sum_{\substack{b \in \mathbb{Z}_m^T \\ b_S = a}} \mu_J(b),$$

which is a sum of probabilities, each being either 0 or $1/m^t$. Therefore, $s(S)(a) = p/m^t$ for some $p \in \mathbb{N}_0$, and therefore

$$m^{|V|} \cdot s(S)(a) = m^{|V|-t} \cdot p \in \mathbb{N}_0$$

where the membership is due to $T \subseteq V$, so $t \leq |T| \leq |V|$, so $|V| - t \in \mathbb{N}_0$, and so $m^{|V|-t} \in \mathbb{N}$. $\square$

5.5. $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$. $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$ is not pairwise uniform by itself, but the CSPs we obtain by restricting the variable domains to the $\mathbb{Z}_2$ part (resp. $\mathbb{Z}_3$) are. Therefore, we will construct two separate SDP (and their induced LP) solutions.

As discussed in the beginning of this section, the AIP solution will be a linear combination of the two LP solutions we constructed. To ensure the support of the AIP solution is contained within the induced LP solution, the latter must be a mixture of the two induced LP solutions we constructed, which requires mixing the corresponding SDP solutions.

The following lemma says that SDP solutions can be mixed in a straightforward manner.

Given an instance I , denote by Σ_d^{SDP} the set of its SDP hierarchy solutions of level d . More precisely, Σ_d^{SDP} is the set of $((\alpha_S)_S, (\mu_R)_R)$, where $(\alpha_S)_S$ is an SDP hierarchy solution for I of level d inducing the LP hierarchy solution $(\mu_R)_R$ of level $2d$.

Lemma 5.14 ([\[CNP24, Lemma 12.5\]](#)). *Let I be an instance and $d \in \mathbb{N}$. Let ρ be a distribution over a finite set Ω . Suppose for every $\mathcal{E} \in \Omega$, there is $((\alpha_S^\mathcal{E})_S, (\mu_R^\mathcal{E})_R) \in \Sigma_d^{\text{SDP}}$. Then $((\alpha_S)_S, (\mu_R)_R) \in \Sigma_d^{\text{SDP}}$, where*

$$(19) \quad \mu_R \stackrel{\text{def}}{=} \mathbb{E}_{\mathcal{E} \sim \rho} \mu_R^\mathcal{E} \quad \text{for } 2d\text{-small } R \subseteq V; \text{ and}$$

$$(20) \quad \alpha_S \stackrel{\text{def}}{=} \bigoplus_{\mathcal{E} \in \Omega} \sqrt{\rho(\mathcal{E})} \cdot \alpha_S^\mathcal{E} \quad \text{for } d\text{-small } S \subseteq V.$$

We now formally generalize the idea in [\[Zhu26\]](#).

Theorem 5.15. *Let $k \geq 3$ be an integer, $\tau \stackrel{\text{def}}{=} 2$, $I = (V, \mathcal{C})$ be a $(2t, 3\gamma)$ -expanding instance of the k -CSP $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$. Then, I has a level- d SDP+AIP hierarchy solution for $d \stackrel{\text{def}}{=} t\gamma/2$.*

Proof. Restricting the variable domains to the $\mathbb{Z}_2$ part, we recover an instance of k -Lin₂. [Theorem 5.10](#) constructs a level- d SDP solution (α_2, s_2) . Similarly, restricting to the $\mathbb{Z}_3$ part recovers an instance of k -Lin₃, and [Theorem 5.10](#) constructs a level- d SDP solution (α_3, s_3) .

Let $n \stackrel{\text{def}}{=} |V|$. [Theorem 5.10](#) guarantees that for any $S \in \binom{V}{\leq 2d}$, $a \in \mathbb{Z}_m^S$, $2^n \cdot s_2(S)(a) \in \mathbb{N}_0$ and $3^n \cdot s_3(S)(a) \in \mathbb{N}_0$.

By Bézout's identity, there are $\alpha, \beta \in \mathbb{Z}$ such that $\alpha \cdot 2^n + \beta \cdot 3^n = 1$. Define $r : (S \in \binom{V}{\leq 2d}) \rightarrow (\mathbb{Q}^{A_S})$ by

$$(21) \quad r \stackrel{\text{def}}{=} \alpha \cdot 2^n \cdot s_2 + \beta \cdot 3^n \cdot s_3.$$

The following claims show that r is indeed a level- $2d$ AIP solution.

Claim 5.16. *For $S \in \binom{V}{\leq 2d}$, $a \in D^S$, $r(S)(a) \in \mathbb{Z}$.*

Proof. $r(S)(a) \stackrel{(21)}{=} \alpha(2^n \cdot s_2(S)(a)) + \beta(3^n \cdot s_3(S)(a)) \in \mathbb{Z}$. $\square$

Claim 5.17. $r(\emptyset)(\emptyset) = 1$.

Proof. $r(\emptyset)(\emptyset) \stackrel{(21)}{=} \alpha \cdot 2^n \cdot s_2(\emptyset)(\emptyset) + \beta \cdot 3^n \cdot s_3(\emptyset)(\emptyset) = \alpha \cdot 2^n \cdot 1 + \beta \cdot 3^n \cdot 1 = 1$. $\square$

Claim 5.18. For any $T \subseteq S \in \binom{V}{\leq 2d}$, $\pi_T(r(S)) = r(T)$.

Proof.

$$\begin{aligned} \pi_T(r(S)) &\stackrel{(21)}{=} \pi_T(\alpha \cdot 2^n \cdot s_2(S) + \beta \cdot 3^n \cdot s_3(S)) \\ &= \alpha \cdot 2^n \cdot \pi_T(s_2(S)) + \beta \cdot 3^n \cdot \pi_T(s_3(S)) \\ &\stackrel{(16)}{=} \alpha \cdot 2^n \cdot s_2(T) + \beta \cdot 3^n \cdot s_3(T) \\ &\stackrel{(21)}{=} r(T) \end{aligned} \quad \square$$

We now mix the two SDP solutions (α_2, s_2) and (α_3, s_3) to obtain (α, s) using [Theorem 5.14](#) (with any ρ with non-zero probability mass on both solutions, such as assigning probability $1/2$ to each of them). The following claim shows that (s, r) is indeed a level- $2d$ LP+AIP solution.

Claim 5.19. For $S \in \binom{V}{\leq 2d}$, $\text{supp}(r(S)) \subseteq \text{supp}(s(S))$.

$$\text{Proof. } \text{supp}(r(S)) \stackrel{(21)}{\subseteq} \text{supp}(s_2(S)) \cup \text{supp}(s_3(S)) \stackrel{(19)}{=} \text{supp}(s(S)). \quad \square$$

Therefore, (α, r) is indeed a level- d SDP+AIP solution. $\square$

In the following theorem, the notation $o_{n;k}(1)$ represents a function $\varepsilon_k(n)$ that, for every fixed k , $\varepsilon_k(n) \rightarrow 0$ as $n \rightarrow \infty$.

Theorem 5.20. Let $k \geq 3$ be an integer. Except with probability $o_{n;k}(1)$, a random instance of the k -CSP $\text{CSP}(\mathbb{Z}_2 \cup \mathbb{Z}_3)$ with n variables and Δn constraints has a SDP+AIP hierarchy solution of level $\Omega_k(n/(\Delta^{2/(k-2)} \log \Delta))$.

Proof. Fix $\gamma \stackrel{\text{def}}{=} 1/\log \Delta$, $\tau \stackrel{\text{def}}{=} k-1$, $\lambda \stackrel{\text{def}}{=} \tau-1 = k-2$. We may assume $\Delta^{2/\lambda} \log \Delta \leq n$, for otherwise the level lower bound is trivial, as every hierarchy has a solution of level 0. Since $\Delta^{2/(\lambda-3/\log \Delta)} = \Theta(\Delta^{2/\lambda})$ for $\lambda \geq 1$, we have $\zeta \stackrel{\text{def}}{=} \Delta^{2/(\lambda-3\gamma)}/n = o_n(1)$. [Theorem 5.6](#) implies that except with probability $o_{n;k}(1)$, a random instance I is $(2t, 3\gamma)$ -expanding, where $t = n/(\Delta^{2/(\lambda-3\gamma)} 2^{O(k)}) = \Omega_k(n/\Delta^{2/\lambda})$. [Theorem 5.15](#) then constructs a level- d SDP+AIP solution with $d \stackrel{\text{def}}{=} t\gamma/2 = \Omega_k(n/(\Delta^{2/\lambda} \log \Delta))$. $\square$

6. SINGLETON (SDP $\wedge$ AIP) LOWER BOUNDS

Definition 6.1 (t -wise independent [[CdX26](#)]). Let ν be a distribution over D . A distribution μ over D^k is t -wise ν -independent if $\pi_T(\mu) = \nu^T$ for $T \in \binom{[k]}{\leq t}$. A k -CSP $(D, \mathcal{R})$ is t -wise ν -independent if every $R \in \mathcal{R}$ has a t -wise ν -independent distribution μ over D^k satisfying $\text{supp}(\mu) \subseteq R$.

Definition 6.2 (Lax [[CN25](#)]). A k -CSP $(D, \mathcal{R})$ is lax if every constraint C and every variable v of C has an assignment b to the remaining variables $V(C) \setminus \{v\}$, so that C is satisfied by b regardless of the value for v . That is, for every $R \in \mathcal{R}$, $v \in [k]$, there exists $b \in D^{[k] \setminus \{v\}}$ such that $D^v \times \{b\} \subseteq R$.

Theorem 6.3. Let $k \geq 3$. Suppose a k -CSP $(D, \mathcal{R})$ is pairwise ν -independent and lax for some distribution ν with $\text{supp}(\nu) = D$. With uniformly positive probability, a random instance of the CSP with n variables and Δn constraints has a $\text{Singl}(\text{SDP} \wedge \text{AIP})$ hierarchy solution of level $\Omega_{k,\Delta}(n)$.

[Theorem 6.3](#) will be proved in [Sections 6.1](#) and [6.2](#). Let us outline the proof that combines ideas from [[CNP24](#), [CN25](#), [CdX26](#)]. Constructing solutions $s^{\mathfrak{R}}(S)$ for small variable subsets S requires considering the closure of S that contains all the constraints needed to define $s^{\mathfrak{R}}(S)$. The choice of closure depends on the relaxation $\mathfrak{R}$ and the assumption on the CSP. For SDP assuming pairwise independence [[CdX26](#)], we use ancestor closure [[CNP24](#)]. For AIP assuming lax, we use BW closure [[CN25](#)]. These two closures have not been considered together in previous works. The support

of the SDP solution is dictated by another closure: τ -wise closure [KMOW17]. To ensure the Singleton AIP solution [CN25] is consistent with the full-support SDP solution [CNP24, CdX26], we relate (partial satisfying assignments to) the τ -wise and BW closures using pairwise BW neutrality [CNP24], which follows from the pairwise independence assumptions in Theorem 6.3.

Theorem 6.3 leads to gap instances for the hybrid algorithm, the BasicSDP $\wedge$ CSingAIP algorithm, as well as linear-level Singl(SDP $\wedge$ AIP) hierarchy, because the optimal value $\text{opt}(I)$ of an instance is concentrated. Here $\text{opt}(I)$ denotes the maximum fraction of constraints satisfied by an assignment to I . Also, given a distribution ρ over k -ary relations on D , let

$$(22) \quad \text{opt}_k(\rho) \stackrel{\text{def}}{=} \max_{\nu \in \Delta(D)} \mathbb{E}_{R \sim \rho} \mathbb{P}_{a \sim \nu^k} [a \in R]$$

be the maximum expected fraction of constraints satisfied by i.i.d. distributions, when the constraints are drawn from the distribution ρ . Theorem D.1 shows that a random instance I with n variables and Δn constraints have $\text{opt}(I) = \text{opt}_k(\rho) + o_n(1) + o_\Delta(1)$ with high probability. Theorems D.1 and 6.3 together imply the following.

Corollary 6.4. *Suppose a k -CSP $(D, \mathcal{R})$ is pairwise ν -independent and lax for some distribution ν with $\text{supp}(\nu) = D$. For $\varepsilon > 0$ and sufficiently large n , some instance I of the CSP on n variables has a Singl(SDP $\wedge$ AIP) hierarchy solution of level $\Omega(n)$ and $\text{opt}(I) \leq \min_{\rho \in \Delta(\mathcal{R})} \text{opt}_k(\rho) + \varepsilon$.*

Theorem 6.4 gives a recipe to generate gap instances for various CSPs. CSPs satisfying Theorems 6.3 and 6.4 include k -SAT (for $k \geq 3$), and the following Max- k -CSP:

Example 6.5. Fix any $k \geq 3$ and $|D| \geq 2$. Let $R \subseteq D^k$ be a relation that is the union of (1) Håstad predicate [Cha16, Section C.2], which is pairwise uniform; and (2) a Hamming ball of radius 1, which is lax [CN25, Lemma 4.13]. Then $|R| = O(|D|k)$. Identifying the domain D with the cyclic group $\mathbb{Z}_{|D|}$, let $\mathcal{R} \stackrel{\text{def}}{=} \{R + b \mid b \in \mathbb{Z}_{|D|}^k\}$ consist of literal shifts of R . Theorem 6.4 yields instances I of the k -CSP $(D, \mathcal{R})$ having a linear-level Singl(SDP $\wedge$ AIP) hierarchy solution, and $\text{opt}(I) = |R|/|D|^k + o_n(1) = O(|D|k/|D|^k) + o_n(1)$ [Tul09, Lemma A.1].

6.1. Multiple Closures. We now start the preparatory work towards Theorem 6.3. Fix $\tau \geq 2$. Recall cl_S^t denotes the local τ -wise closure from Theorem 5.3.

The next theorem is essentially [CNP24, Theorem 12.6] that constructs an SDP hierarchy solution with full support on small closures. [CNP24, Theorem 12.6] concerns a special case of τ -wise independent CSP known as τ -wise uniform CSP. The generalization to τ -wise independent CSP follows by replacing [CNP24, Proposition 10.5] with [CdX26, Lemma 7.14], and replacing [CNP24, Lemma 11.15] with [CdX26, Lemma 7.9]. [CNP24, Theorem 12.6] works with a slightly different version of SDP hierarchy (whose solutions involve both vectors $\{\alpha_S\}$ and distributions $\{\mu_R\}$) that is equivalent to ours (Theorem C.3). [CNP24, Theorem 12.6] only states $\text{supp}(s^{\text{SDP}}(S)) \supseteq \pi_S(A_{S \cup V(\text{cl}_S^t(\mathcal{C})))$, but in fact $\text{supp}(s^{\text{SDP}}(S)) = \text{supp}(\mu_S) = \pi_S(A_{S \cup V(\text{cl}_S^t(\mathcal{C})))$ by [CNP24, Lemma A.8].

Theorem 6.6. *Suppose an instance $I = (V, \mathcal{C})$ of a τ -wise independent CSP is $(4t, 3\gamma)$ -expanding. Let $d \stackrel{\text{def}}{=} t\gamma/4$. Then I has a level- d SDP solution s^{SDP} such that for every $S \in \binom{V}{\leq d}$, $\text{supp}(s^{\text{SDP}}(S)) = \pi_S(A_{V(\text{cl}_S^{2t}(I))})$, where $V(\text{cl}_S^{2t}(I)) \stackrel{\text{def}}{=} S \cup V(\text{cl}_S^{2t}(\mathcal{C}))$.*

In order to relate the τ -wise closure to another closure (BW closure) discussed later, [CN25] considers generic closure, based on generic boundary, of which Theorem 5.1 is an example. [CN25] also introduced the concept of “insular family”. Here $\text{cl}_S = \text{cl}_S^\infty$ denotes the non-local version of the closure operator.

Definition 6.7 (Insular family, [CN25, Definition 6.3]). Given a hypergraph $(V, \mathcal{E})$, a family $\mathcal{Z} \subseteq 2^V$ of vertex subsets is an insular family if:

- $\text{cl}_T(\mathcal{E}[S]) \subseteq \mathcal{E}[T]$ whenever $S, T \in \mathcal{Z}$ and $T \subseteq S$; and

- $S \cap T \in \mathcal{Z}$ for $S, T \in \mathcal{Z}$.

An example of such a family is the following:

$$\mathcal{Z}_t \stackrel{\text{def}}{=} \{S \subseteq V \mid |\text{cl}_S^t(\mathcal{C})| \leq t \text{ and } \text{cl}_S^t(\mathcal{C}) = \mathcal{C}[S]\}$$

that satisfies some nice “insular” property under the τ -wise closure [CN25, Lemma 6.4].

Lemma 6.8. $\text{cl}_S^{2t}(\mathcal{E}) = \text{cl}_S^t(\mathcal{E})$ if $\text{cl}_S^{2t}(\mathcal{E})$ is t -small.

Proof. $\text{cl}_S^{2t}(\mathcal{E}) \supseteq \text{cl}_S^t(\mathcal{E})$ follows easy from [CN25, Definition 5.6]. We will show the reverse inclusion. Let $T \stackrel{\text{def}}{=} S \cup V(\text{cl}_S^{2t}(\mathcal{E}))$. We now show that $\text{cl}_T^t(\mathcal{E}) \subseteq \text{cl}_S^{2t}(\mathcal{E})$. Since $\text{cl}_S^{2t}(\mathcal{E})$ is S -closed by [CN25, Lemma 5.3] and t -small, given any t -small T -closed $\mathcal{F} \subseteq \mathcal{E}$, $\mathcal{F} \cup \text{cl}_S^{2t}(\mathcal{E})$ is S -closed by chaining [CN25, Definition 5.1] and $2t$ -small, so $\mathcal{F} \subseteq \mathcal{F} \cup \text{cl}_S^{2t}(\mathcal{E}) \subseteq \text{cl}_S^{2t}(\mathcal{E})$ by definition of $\text{cl}_S^{2t}(\mathcal{E})$. Taking the union over all such $\mathcal{F}$, $\text{cl}_T^t(\mathcal{E}) \subseteq \text{cl}_S^{2t}(\mathcal{E})$, and $\text{cl}_T^t(\mathcal{E})$ is also t -small.

$\mathcal{E}[T] \subseteq \text{cl}_T^t(\mathcal{E})$ because $\{e\}$ is T -closed for $e \in \mathcal{E}[T]$ and $t \geq 1$. Definition of T implies $\text{cl}_S^{2t}(\mathcal{E}) \subseteq \mathcal{E}[T] \subseteq \text{cl}_T^t(\mathcal{E})$. Therefore $\text{cl}_T^t(\mathcal{E}) = \text{cl}_S^{2t}(\mathcal{E})$. Since $\text{cl}_T^t(\mathcal{E}) = \text{cl}_S^{2t}(\mathcal{E})$ is S -closed by [CN25, Lemma 5.3], [CN25, Lemmas 5.4, 5.9] imply $\text{cl}_T^t(\mathcal{E}) = \text{cl}_S(\text{cl}_T^t(\mathcal{E})) = \text{cl}_S^t(\mathcal{E})$. Thus $\text{cl}_S^{2t}(\mathcal{E}) \subseteq \text{cl}_T^t(\mathcal{E}) = \text{cl}_S^t(\mathcal{E})$. $\square$

Lemma 6.9. Consider any $U \subseteq V$. Suppose some $W \in \mathcal{Z}_t$ contains U and is the minimal such W . If $T \stackrel{\text{def}}{=} \text{cl}_U^{2t}(I) \in \mathcal{Z}_t$, then $W = T$.

Proof. Note that $U \subseteq T$ and hence $W \subseteq T$. For any S containing U , $\text{cl}_S^t(\mathcal{C}) = \mathcal{C}[S]$, $|\mathcal{C}[S]| \leq t$, we have $\text{cl}_S^t(\mathcal{C}) \supseteq \text{cl}_U^t(\mathcal{C}) = \text{cl}_U^{2t}(\mathcal{C})$, where the equality is the previous Lemma. Thus $S \supseteq U \cup \text{cl}_S^t(\mathcal{C}) \supseteq U \cup V(\text{cl}_U^{2t}(\mathcal{C})) = T$, implying $W \supseteq T$. $\square$

The next result is the analog of [CN25, Lemma 8.2] for local closure, allowing us to compare the local τ -wise closure and the local BW closure. Following [CN25, Section 8.1], given boundary operators $\mathcal{B}, \mathcal{B}'$ for $(V, \mathcal{H})$ (where $\mathcal{H}$ is a hyperedge set over V) and their corresponding closure operators cl, cl' , write $\mathcal{B}' \subseteq \mathcal{B}$ if $\text{cl}_S(\mathcal{F}) = \emptyset$ for $\mathcal{E} \subseteq \mathcal{H}, S \subseteq V, \mathcal{F} \in \mathcal{B}'_S(\mathcal{E})$.

Lemma 6.10. Let $\mathcal{B}, \mathcal{B}'$ be boundary operators for $(V, \mathcal{H})$. If $\mathcal{B}' \subseteq \mathcal{B}$, then $\text{cl}_S^t(\mathcal{E}) \subseteq \text{cl}'^t_S(\mathcal{E})$ for any $t \geq 0, S \subseteq V, \mathcal{E} \subseteq \mathcal{H}$.

Proof. [CN25, Lemma 8.2] implies $\text{cl} \subseteq \text{cl}'$ in the sense that $\text{cl}_S(\mathcal{E}) \subseteq \text{cl}'_S(\mathcal{E})$ for $S \subseteq V, \mathcal{E} \subseteq \mathcal{H}$. Now $\mathcal{F} \subseteq \mathcal{H}$ is S -closed under cl if and only if $\text{cl}_S(\mathcal{F}) = \mathcal{F}$ [CN25, Definition 5.1], which implies $\mathcal{F} = \text{cl}_S(\mathcal{F}) \subseteq \text{cl}'_S(\mathcal{F}) \subseteq \mathcal{F}$, which is in turn equivalent to $\mathcal{F}$ being S -closed under cl' .

Since $\text{cl}_S^t(\mathcal{E})$ is the union of t -small $\mathcal{F} \subseteq \mathcal{E}$ such that $\mathcal{F}$ is S -closed under cl [CN25, Definition 5.6], and any such $\mathcal{F}$ is also S -closed under cl' , we have $\text{cl}_S^t(\mathcal{E}) \subseteq \text{cl}'^t_S(\mathcal{E})$. $\square$

6.2. Proof of Combined Lower Bounds. We prove Theorem 6.3 in this subsection. Recall the notions of BW boundary and BW closure from [CN25].

Definition 6.11 (BW Boundary, [CN25, Definition 5.20]). Let $k, \ell \geq 2$. Let $S \subseteq V$ be a vertex subset, and $\mathcal{E}$ a k -uniform hyperedge set over V .

- The first S -boundary of $\mathcal{E}$, $\mathcal{B}_S^1(\mathcal{E})$, is the collection of $\{e\}$ over hyperedges $e \in \mathcal{E} \setminus \mathcal{E}[S]$ such that e contains at most one vertex u satisfying $u \in S$ or $\deg_{\mathcal{E}}(u) > 1$.
- The second S -boundary of $\mathcal{E}$, $\mathcal{B}_{S,\ell}^2(\mathcal{E})$, is the collection of hyperedge sets of pendant paths P of length ℓ in $\mathcal{E} \setminus \mathcal{E}[S]$, such that non-endpoints of P are outside S .
- The S -boundary of $\mathcal{E}$ is $\mathcal{B}_{S,\ell}(\mathcal{E}) \stackrel{\text{def}}{=} \mathcal{B}_S^1(\mathcal{E}) \cup \mathcal{B}_{S,\ell}^2(\mathcal{E})$.

When $S = \emptyset$, we omit S from the notation, i.e. $\mathcal{B}_{\ell}(\mathcal{E}) \stackrel{\text{def}}{=} \mathcal{B}_{\emptyset,\ell}(\mathcal{E})$ and $\mathcal{B}_{\ell}^i(\mathcal{E}) \stackrel{\text{def}}{=} \mathcal{B}_{\emptyset,\ell}^i(\mathcal{E})$ for $i = 1, 2$.

Definition 6.12 ($((S, \ell)$ -closed, [CN25, Remark 5.16]). Let $k, \ell \geq 2$. Given a vertex subset S , a k -uniform hyperedge set $\mathcal{E}$ is $((S, \ell)$ -closed if $\mathcal{B}_{S,\ell}(\mathcal{E}) = \emptyset$.

Definition 6.13 (Local closure, [CN25, Definition 5.6]). Given $t \in \mathbb{N}$, $k, \ell \geq 2$, the t -local (S, ℓ) -closure $\text{cl}_{S, \ell}^t(\mathcal{E})$ of a k -uniform hyperedge set $\mathcal{E}$ is the union of t -small (S, ℓ) -closed $\mathcal{F} \subseteq \mathcal{E}$.

Definition 6.14 (Sparse, [CN25, Definition 5.23]). A k -uniform hyperedge set $\mathcal{E}$ is δ -sparse if $|\mathcal{E}| \leq \frac{1+\delta}{k-1}|V(\mathcal{E})|$. $\mathcal{E}$ is (t, δ) -sparse if every t -small subset of $\mathcal{E}$ is δ -sparse. A k -uniform hypergraph is δ -sparse (resp. (t, δ) -sparse) if its hyperedge set is.

Definition 6.15 (Expanding, [CN25, Definition 5.27]). Let $t \geq 0$ and $\gamma > 0$. A hyperedge set $\mathcal{E}$ over V is (ℓ, t, γ) -expanding if $|\mathcal{B}_\ell(\mathcal{F})| \geq \gamma|\mathcal{F}|$ for any t -small $\mathcal{F} \subseteq \mathcal{E}$. A hypergraph is (ℓ, t, γ) -expanding if its hyperedge set is.

Proof of Theorem 6.3. Fix $\tau = 2, \gamma_1 = 1/\log \Delta$. Theorem 5.6 implies that except with probability $o_{n,k,\Delta}(1)$, a random instance I is $(4t_1, 3\gamma_1)$ -expanding, where $t_1 = n/(\Delta^{2/(1-3\gamma_1)}2^{O(k)}) = \Omega_{k,\Delta}(n)$.

Following the proof of [CN25, Theorem 8.8] (with $9 = \ell' = 3\ell + 3$, where $\ell = 2$), [CN25, Lemmas 5.24, 5.28, 5.31] imply there is $\delta \stackrel{\text{def}}{=} \delta(k, 9)$ given by [CN25, Lemma 5.26], $\gamma_2 > 0$ and $t_2 = \Omega_{k,\Delta}(n)$ such that with uniformly positive probability, a random instance I is $(4t_2, \delta)$ -sparse, $(9, 4t_2, \gamma_2)$ -expanding, and has girth more than 6.

Let $t = \min(t_1, t_2)$. We will construct a Singl(SDP $\wedge$ AIP) hierarchy solution of level $d \stackrel{\text{def}}{=} \min(t_1\gamma_1/4, t_2/2(1 + 9k/\gamma_2)(1 + k/\gamma_1)) = \Omega_{k,\Delta}(n)$. Theorem 6.6 yields the level- d SDP solution s^{SDP} for $I = (V, \mathcal{C})$ with full support, i.e. $\text{supp}(s^{\text{SDP}}(S)) = \pi_S(A_{V(\text{cl}_S^{2t}(I))})$ for $S \in \binom{V}{\leq d}$, where $V(\text{cl}_S^{2t}(I)) \stackrel{\text{def}}{=} S \cup V(\text{cl}_S^{2t}(\mathcal{C}))$.

For AIP, consider the family of subsets

$$\mathcal{Z}_t \stackrel{\text{def}}{=} \{S \subseteq V \mid |\text{cl}_{S,9}^t(\mathcal{C})| \leq t \text{ and } \text{cl}_{S,9}^t(\mathcal{C}) = \mathcal{C}[S]\}$$

that are “insular” under the BW closure. I satisfies a technical condition called $(2t_2, \gamma_2/9)$ -confined by [CN25, Remark 6.9], and $\mathcal{Z}_t$ is 2-nice by [CN25, Theorem 7.6]. Since the CSP is pairwise ν -independent with $\text{supp}(\nu) = D$, the CSP is 1-null-constraining [CN25, Definition 4.2], which is equivalent to pairwise BW neutral [CNP24, Section 4] and is 2-null-constraining.

Given any $U \in \binom{V}{\leq d}$, let $T \stackrel{\text{def}}{=} V(\text{cl}_U^{2t}(I))$ be the pairwise closure of U . [CNP24, Lemma 8.11] implies $|T| \leq (1 + k/\gamma_1)|U| \leq t_2/2(1 + 9k/\gamma_2)$. Let $K \stackrel{\text{def}}{=} \bigcap \{S \in \mathcal{Z}_t \mid S \supseteq T\}$. [CN25, Proof of Lemma 6.12] implies $|\mathcal{C}[K]| \leq |K| \leq (1 + 9k/\gamma_2)|T| \leq t_2/2$.

Consider any $b \in \text{supp}(s^{\text{SDP}}(U)) = \pi_U(A_T) = \pi_U(A_K)$, where the last equality is Theorem 6.16. There is $q \in A_K$ such that $q_U = b$. Consider the subfamily $\mathcal{Z}_{\supseteq K} \stackrel{\text{def}}{=} \{S \in \mathcal{Z}_t \mid S \supseteq K\}$. Then $\mathcal{Z}_{\supseteq K}$ is a 2-nice insular family [CN25, Definition 6.3] with $\min \mathcal{Z}_{\supseteq K} = K$. [CN25, Lemma 6.15, Theorem 7.17] applied to $\mathcal{Z}_{\supseteq K}$ yield an AIP solution s_b^{AIP} by [CN25, Definition 6.14]. Then $s_b^{\text{AIP}}(U) = \pi_U(1_q) = 1_b$. This solution has level $d \leq t_2/(1 + 9k/\gamma_2)$ because for every $S \in \binom{V}{\leq d}$, some $R \in \mathcal{Z}_{\supseteq K}$ minimally contains $S \cup T$ by [CN25, Lemma 6.12]. Let $A_{R,q} \stackrel{\text{def}}{=} \{a \in A_R \mid a_U = q\}$. Let W be the vertex subset that minimally contains S , $\text{cl}_{W,9}^t(\mathcal{C}) = \mathcal{C}[W]$ and $|\mathcal{C}[W]| \leq t$. Theorem 6.9 implies $W = V(\text{cl}_S^{2t}(I))$. Then [CN25, Lemma 6.17] implies

$$(23) \quad \text{supp}(s_b^{\text{AIP}}(S)) \subseteq \pi_S(A_{R,q}) \subseteq \pi_S(A_W) = \pi_S(A_{V(\text{cl}_S^{2t}(I))}).$$

[CN25, Lemma 8.1] and Theorem 6.10 imply $V(\text{cl}_S^{2t}(I)) \subseteq V(\text{cl}_S^{2t}(I))$. Thus

$$\text{RHS of (23)} \subseteq \pi_S(A_{V(\text{cl}_S^{2t}(I))}) = \text{supp}(s^{\text{SDP}}(S)). \quad \square$$

Claim 6.16. For any vertex subset U such that $|\text{cl}_U^{2t}(\mathcal{C})| \leq t$, if $T \stackrel{\text{def}}{=} V(\text{cl}_U^{2t}(I))$, for $K \supseteq T$ such that $|\mathcal{C}[K]| \leq t$, then $\pi_U(A_K) = \pi_U(A_T)$.

Proof. $\mathcal{C}[T] = \text{cl}_T(\mathcal{C}[K])$ because

$$\begin{aligned} \mathcal{C}[T] &\subseteq \text{cl}_T(\mathcal{C}[K]) && \text{(Using } T \subseteq K, \text{ and “containing } \mathcal{E}[S]\text{” in [CN25, Definition 5.1])} \\ &\subseteq \text{cl}_T^t(\mathcal{C}) && \text{([CN25, Lemma 5.8])} \\ &\subseteq \text{cl}_U^{2t}(\mathcal{C}) && \text{(Proof of Theorem 6.8)} \\ &\subseteq \mathcal{C}[T] && (V(\text{cl}_U^{2t}(\mathcal{C})) \subseteq T). \end{aligned}$$

[CNP24, Proposition 5.4] implies $\text{cl}_T(\mathcal{C}[K])$ is obtained from $\mathcal{C}[K]$ by iteratively removing pairwise boundaries according to [CNP24, Eq. (10)]. [CNP24, Lemma 5.11, Proposition 6.2] imply [CNP24, Eq. (11)], i.e. $\pi_U(A_K) = \pi_U(A_T)$. $\square$

6.3. Non-trivially Embeddable CSP. Theorem 6.3 as written only applies to CSPs without non-trivial embeddings. This subsection extends the lower bounds to certain CSPs with non-trivial embeddings using a trick.

Remark 6.17. If a CSP is lax, then the CSP has no non-trivial abelian embedding. Indeed, if (σ, g) is an embedding of $R \in \mathcal{R}$, then for every $v \in [k]$, there is $b \in D^{[k] \setminus \{v\}}$ such that $\sigma_v(a) + \sum_{i \in [k] \setminus \{v\}} \sigma_i(b(v)) = g$ for every $a \in D$, so σ_v is a constant function.

Are there $\text{Singl}(\text{SDP} \wedge \text{AIP})$ hierarchy lower bounds for CSPs with non-trivial embedding? We are able to prove lower bounds for such CSPs as well, by constructing “useless” embeddings using a product construction and applying Theorem 6.3:

Definition 6.18. Given any k -CSP $(D, \mathcal{R})$ and abelian group H , define the product k -CSP $(D \times H, \mathcal{R}_H)$. For every relation R in $\mathcal{R}$, $\mathcal{R}_H$ contains a relation

$$(24) \quad R' \stackrel{\text{def}}{=} \left\{ b \in (D \times H)^k \mid (\pi_D(b_i))_{i \in [k]} \in R, \sum_{i \in [k]} \pi_H(b_i) = 0_H \right\}.$$

Whenever H is not the trivial group, $\mathcal{R}_H$ has a non-trivial embedding $(\sigma, 0_H)$ to H , via the maps $\sigma_i \stackrel{\text{def}}{=} \pi_H$ for $i \in [k]$.

Given an instance $J = (V, \mathcal{C})$ of $\text{CSP}(\mathcal{R})$, replace the relation R of every constraint $C \in \mathcal{C}$ with R' as in Eq. (24) to obtain a new instance $J' = (V, \mathcal{C}')$ of $\text{CSP}(\mathcal{R}_H)$. It is easy to see that $\text{opt}(J) = \text{opt}(J')$. Indeed, for every assignment $b : V \rightarrow D$ of J , there is an assignment $b' : V \rightarrow D \times H$ of J' satisfying the same fraction of constraints, where $b'(v) \stackrel{\text{def}}{=} (b(v), 0_H)$ for $v \in V$. Conversely, for every assignment $b' : V \rightarrow D \times H$ of J' , there is an assignment $b : V \rightarrow D$ of J satisfying at least the same fraction of constraints, where $b(v) \stackrel{\text{def}}{=} \pi_D(b'(v))$ for $v \in V$.

When $k \geq 3$, then every $R \in \mathcal{R}$ is pairwise connected if and only if every $R' \in \mathcal{R}_H$ is. Recall that $R \subseteq D^k$ is pairwise connected [BKM24, Definition 1.3] if for every distinct $i, j \in [k]$, the bipartite graph $(D \uplus D, R_{i,j})$ is connected, where $R_{i,j} = \{(b_i, b_j) \mid b \in R\}$. This is because for every $R \in \mathcal{R}$, we have $R'_{i,j} = \{((b_i, h_i), (b_j, h_j)) \mid (b_i, b_j) \in R_{i,j}, h_i, h_j \in H\}$. Conversely, for every $R' \in \mathcal{R}_H$, we have $R_{i,j} = \{(b_i, b_j) \mid ((b_i, h_i), (b_j, h_j)) \in R'_{i,j}\}$.

Further, $\text{CSP}(\mathcal{R})$ is trivially satisfiable by a single value [CNP24, Section B.1] if and only if $\text{CSP}(\mathcal{R}_H)$ is.

Therefore, if we consider any pairwise independent and lax $\text{CSP}(\mathcal{R})$ (such as 3-SAT, or Theorem 6.5, both of which consist of pairwise connected relations), consider the derived $\text{CSP}(\mathcal{R}_{\mathbb{Z}_2})$. Every relation in $\mathcal{R}_{\mathbb{Z}_2}$ embeds non-trivially to $\mathbb{Z}_2$. Then Theorem 6.4 implies $\text{CSP}(\mathcal{R})$ has an instance J with an arbitrarily large number n of variables, such that J has a level- $\Omega(n)$ $\text{Singl}(\text{SDP} \wedge \text{AIP})$ solution s , while $\text{opt}(J) = \min_{\rho \in \Delta(D)} \text{opt}_k(\rho) + o_n(1)$. This implies the derived instance J' of $\text{CSP}(\mathcal{R}_{\mathbb{Z}_2})$ also has a level- $\Omega(n)$ $\text{Singl}(\text{SDP} \wedge \text{AIP})$ solution s' , where s' is supported on the subdomain $D \times \{0_{\mathbb{Z}_2}\}$,

by copying from the solution s . That is, for $S \in \binom{V}{\leq d}$, $b : S \rightarrow D \times \mathbb{Z}_2$,

$$s'(S, b) \stackrel{\text{def}}{=} \begin{cases} s(S, \pi_D(b)) & \text{if } b(v) \in D \times \{0_{\mathbb{Z}_2}\} \forall v \in S \\ 0 & \text{otherwise} \end{cases}.$$

7. (ITERATIVE) BasicSDP $\wedge$ CSinglAIP LOWER BOUNDS FOR AGH

[CŽ25] proved SDA (i.e., basic SDP+AIP) lower bounds for the approximate graph homomorphism (AGH) problem. AGH includes as a special case the approximate graph coloring problem, which is already wide open. Such unconditional lower bounds may be seen as evidence for the Brakensiek–Guruswami conjecture [BG21] on the NP-hardness of this problem. Now that SDA is known not to solve some tractable CSPs (Theorem 5.20), better evidence is needed. In this section, we strengthen the SDA lower bound of [CŽ25] to BasicSDP $\wedge$ CSinglAIP.

7.1. Approximate Graph Homomorphism. Given two graphs G, H , we write $G \rightarrow H$ if there exists an edge-preserving map $f : V(G) \rightarrow V(H)$ such that if $(u, v) \in E(G)$, then $(f(u), f(v)) \in E(H)$. f is commonly known as a *homomorphism* in the universal algebra literature. We also say G is *H -colorable* in this case.

Given any graph G , the *graph homomorphism* problem $\text{CSP}(G)$ is the problem to decide if the input graph I is G -colorable or not. In other words, for any input graph $I = (V, E)$, we consider the instance $I_G \stackrel{\text{def}}{=} (V, \mathcal{C})$, where each edge $e \in E$ corresponds to a constraint $C \in \mathcal{C}$ with relation $R \stackrel{\text{def}}{=} E(G)$. As a slight abuse of notation, we do not make a distinction between an input graph I and an input instance of $\text{CSP}(G)$.

Given non-bipartite, loopless graphs G, H such that $G \rightarrow H$, the *approximate graph homomorphism* problem $\text{PCSP}(G, H)$ is the problem to decide if the input graph I is G -colorable or not H -colorable, provided one of these two holds. Note that these two possibilities are disjoint: if $g : V(I) \rightarrow V(G)$ is an edge-preserving map from I to G , then $f \circ g : V(I) \rightarrow V(H)$ is an edge-preserving map from I to H .

Conjecture 7.1 ([BG21, Conjecture 1.2]). *Let G and H be undirected nonbipartite graphs with a homomorphism from G to H . Then the promise graph homomorphism problem associated with G and H is NP-hard.*

See e.g. [KOWŽ23, AFO⁺25] for recent efforts towards this conjecture.

To apply relaxation algorithms for CSPs on the approximate graph homomorphism problem $\text{PCSP}(G, H)$ (or any promise CSPs in general), it is standard in the literature to simply discard H and run the algorithm on $\text{CSP}(G)$. We say that the algorithm solves $\text{PCSP}(G, H)$ if, for any input instance I of $\text{CSP}(G)$, the algorithm accepts I implies $I \rightarrow H$. That is, the algorithm always correctly determine if the input graph I is G -colorable or not H -colorable.

7.2. CSinglAIP. For any odd $h \geq 3$, denote by C_h the cycle graph with h vertices. The vertex set of C_h is $\mathbb{Z}_h$. There is an undirected edge $(i, i+1)$ for $i \in \mathbb{Z}_h$.

Lemma 7.2. *Let $h \geq 3$ be an odd integer, $A_C \stackrel{\text{def}}{=} E(C_h) \subseteq \mathbb{Z}_h^2$ be the set of satisfying assignments of a constraint in $\text{CSP}(C_h)$. For any fixed $p, q \in \mathbb{Z}_h$, there exists an AIP weight $w : A_C \rightarrow \mathbb{Z}$ such that $\pi_1(w) = 1_p$ and $\pi_2(w) = 1_q$.*

Proof. We say that a pair (p, q) is *projectable* if such an AIP weight w exists. $(1, 2)$ is projectable using $w \stackrel{\text{def}}{=} 1_{(1,2)}$. We shall incrementally construct the AIP weight w for other pairs of (p, q) .

Claim 7.3. *If (p, q) is projectable, then so is $(p, q+2)$.*

Proof. Let w be the corresponding AIP weight of (p, q) . We construct the corresponding AIP weight w' for $(p, q + 2)$ as $w' \stackrel{\text{def}}{=} w - \mathbb{1}_{(q+1, q)} + \mathbb{1}_{(q+1, q+2)}$.

It is a mechanical check that w' is indeed the desired AIP weight:

- $\pi_1(w') = \mathbb{1}_p$. For any $c \in \mathbb{Z}_h$,

$$\begin{aligned} \pi_1(w')(c) &= \pi_1(w)(c) - \pi_1(\mathbb{1}_{(q+1, q)})(c) + \pi_1(\mathbb{1}_{(q+1, q+2)})(c) \\ &= \mathbb{1}[c = p] - \mathbb{1}[c = q + 1] + \mathbb{1}[c = q + 1] \\ &= \mathbb{1}[c = p]. \end{aligned}$$

- $\pi_2(w') = \mathbb{1}_q$. For any $d \in \mathbb{Z}_h$,

$$\begin{aligned} \pi_2(w')(d) &= \pi_2(w)(d) - \pi_2(\mathbb{1}_{(q+1, q)})(d) + \pi_2(\mathbb{1}_{(q+1, q+2)})(d) \\ &= \mathbb{1}[d = q] - \mathbb{1}[d = q] + \mathbb{1}[d = q + 2] \\ &= \mathbb{1}[d = q + 2]. \end{aligned}$$

□

By a symmetrical argument, we can also increase p by 2.

Claim 7.4. *If (p, q) is projectable, then so is $(p + 2, q)$.*

We are done because any $(p, q) \in \mathbb{Z}_h^2$ can be obtained from $(1, 2)$ by applying the above two claims a finite number of times (because h is odd). As a result, every (p, q) is projectable. □

Lemma 7.5. *For any odd $h \geq 3$, any input graph $I = (V, E)$ for $\text{CSP}(C_h)$, any edge $e \in E$, and any satisfying assignment $b \in A_e$ to e , there is a BasicAIP solution for the reduced instance $I + b$.*

Proof. Let $e = \{x, y\}$. Fix any global assignment $a : V \rightarrow \mathbb{Z}_h$ such that $a(x) = b(x)$ and $a(y) = b(y)$. For any $v \in V$, define $s(v) \stackrel{\text{def}}{=} \mathbb{1}_{a_v}$. We now define s on constraints so that s is consistent with projection. We define $s(e) \stackrel{\text{def}}{=} \mathbb{1}_b$ as required. For any $f \in E \setminus \{e\}$, we construct $s(f)$ using Theorem 7.2 (with $p \stackrel{\text{def}}{=} a(x)$, $q \stackrel{\text{def}}{=} a(y)$). □

7.3. Wrapping Up. Following [CŽ25], the hard instance I will be the Kneser graph $K_{s,t}$. Given $s, t \in \mathbb{N}$ such that $s > 2t$, the Kneser graph $K_{s,t}$ is the undirected graph whose vertices are subsets of $[s]$ of size t , and whose edges are disjoint pairs of such subsets. The next lemma makes explicit the fact that the SDP solution in [CŽ25] has full support.

Lemma 7.6. *Fix any odd $n \geq 3$ and consider $\text{CSP}(C_n)$. There is a rational $0 < \delta < 2$ given by [CŽ25, Proof of Theorem 1], such that for any $s, t \in \mathbb{N}$ satisfying $\frac{t}{\delta} \in \mathbb{N}$ and $s \stackrel{\text{def}}{=} \frac{2t}{\delta} + t$, the Kneser graph $K_{s,t}$ has a basic SDP solution r of full support, i.e. $\text{supp}(r(e)) = A_e$ for $e \in E(K_{s,t})$.*

Proof. Given an odd $n \geq 3$, choose $0 < \delta < 2$ and construct the same SDP C_n -solution r for $K_{s,t}$ as in [CŽ25, Proof of Theorem 1]. It is easy to see that r has full support. Indeed, the SDP solution r is invariant under the automorphisms of C_n (and also $K_{s,t}$, but not useful for us), i.e., $\langle r(u, a), r(v, b) \rangle = \langle r(u, \psi(a)), r(v, \psi(b)) \rangle$ for $\psi \in \text{Aut}(C_n)$, $u, v \in V(K_{s,t})$, $a, b \in V(C_n)$ [CŽ25, Definition 8]. As is well known, C_n is arc-transitive for any $n \geq 3$, since the automorphism $\psi = a \in \mathbb{Z}_n \mapsto a + j - i$ maps any arc $(i, i \pm 1)$ to any arc $(j, j \pm 1)$ of the same direction, and the automorphism $\psi = a \in \mathbb{Z}_n \mapsto 1 - a$ maps the arc $(0, 1)$ to its reverse $(1, 0)$. Now fix any $e \stackrel{\text{def}}{=} (u, v) \in E(K_{s,t})$. As (a, b) ranges over the arcs of C_n (i.e. satisfying assignments of the constraint e), the inner product $\langle r(u, a), r(v, b) \rangle$ remains unchanged. The constraints of the basic SDP imply $\sum_{f \in A_e} \|r(e, f)\|^2 = 1$, so Theorem 7.7 implies $\|r(e, f_{ab})\|^2 = \langle r(u, a), r(v, b) \rangle = 1/|A_e| > 0$ for every $(a, b) \in A_e$, i.e. $(a, b) \in \text{supp}(r(e))$. Therefore $A_e \subseteq \text{supp}(r(e))$. The reverse inclusion $A_e \supseteq \text{supp}(r(e))$ follows from the definition of the basic SDP. □

Claim 7.7. *Let s be a basic SDP solution to a 2-CSP instance $I = (V, \mathcal{C})$, under the non-traditional Theorems 4.2 and 4.3 of basic SDP. Then for any constraint $C \in \mathcal{C}$ with scope (u, v) , any $a, b \in D$, $\langle s(u, a), s(v, b) \rangle = \|s(C, f_{ab})\|^2$, where $f_{ab}(u) = a$ and $f_{ab}(v) = b$.*

Proof. Expand $s(u, a) = \sum_{\substack{f \in A_C \\ f(u)=a}} s(C, f)$ and $s(v, b) = \sum_{\substack{g \in A_C \\ f(v)=b}} s(C, g)$, we have

$$\langle s(u, a), s(v, b) \rangle = \sum_{\substack{f, g \in A_C \\ f(u)=a, g(v)=b}} \langle s(C, f), s(C, g) \rangle \stackrel{(*)}{=} \langle s(C, f_{ab}), s(C, f_{ab}) \rangle = \|s(C, f_{ab})\|^2,$$

where $(*)$ uses the orthogonality of $s(C, f)$ and $s(C, g)$ for distinct $f, g : \{u, v\} \rightarrow D$. $\square$

When combined with [Theorem 4.5](#), the next theorem shows that $\text{BasicSDP} \wedge \text{CSinglAIP}$ does not solve the approximate graph homomorphism problem, because the algorithm erroneously accepts some Kneser graph $K_{s,t}$ even when $K_{s,t} \not\rightarrow H$. By a basic $\mathfrak{R}$ G -solution, we mean a basic $\mathfrak{R}$ solution to $\text{CSP}(G)$.

Theorem 7.8. *For any nonbipartite loopless undirected graphs G and H such that $G \rightarrow H$, some Kneser graph $K_{s,t}$ has a $\text{BasicSDP} \wedge \text{CSinglAIP}$ G -solution but $K_{s,t} \not\rightarrow H$.*

Proof. Following [[CŽ25](#), Proof of Theorem 1], let n be the odd girth of G and m be the chromatic number of H . Then C_n is a subgraph of G , and $H \rightarrow K_m$. Choose rational $0 < \delta < 2$, $\frac{t}{\delta} \in \mathbb{N}$, and $s \stackrel{\text{def}}{=} \frac{2t}{\delta} + t$ as in [Theorem 7.6](#), and further $t \geq \frac{2m}{2-\delta}$. As explained in [[CŽ25](#), Proof of Theorem 1], the Lovász formula for chromatic number implies $K_{s,t} \not\rightarrow K_m$, and hence $K_{s,t} \not\rightarrow H$.

[Theorem 7.6](#) constructs a basic SDP C_n -solution s^{SDP} for $K_{s,t}$ such that $\text{supp}(s^{\text{SDP}}(e)) = A_e$ for $e \in E(K_{s,t})$. For $e \in E(K_{s,t}), f \in A_e$, [Theorem 7.5](#) further constructs basic AIP C_n -solutions s_f^{AIP} for the reduced instances $K_{s,t} + f$. Then $\text{supp}(s_f^{\text{AIP}}(e')) \subseteq A_{e'} = \text{supp}(s^{\text{SDP}}(e'))$ for $e' \in E(K_{s,t})$. Since C_n is a subgraph of G , these basic SDP and AIP solutions are also G -solutions. $\square$

Acknowledgements. We thank Amey Bhangale, Siu Man Chan, Yassine Ghannane, Sasha Razborov, Dmitriy Zhuk, and Standa Živný for helpful discussions. We also thank the anonymous SODA reviewers for helpful feedback.

APPENDIX A. MASTER EMBEDDING

In this section, we recall the definition of master embedding of [[BKM24](#), [BKM26](#)] and show that any master embedding is universal.

Definition A.1 (Linear reduction [[BKM24](#), Definition 3.1]). An embedding (G, σ, g) of R is a linear reduction of another (H, τ, h) if G is a subgroup of H , and there are injective homomorphisms $f_i : G \rightarrow H$ such that $\tau_i = f_i \circ \sigma_i$ for $i \in [k]$.

Lemma A.2 ([[BKM24](#), Lemma 3.4]). *Let $k = 3$. Suppose $R \subseteq D^k$ does not have any non-trivial abelian embedding to $\mathbb{Z}$. There is a finite collection $\mathcal{E}$ of abelian embeddings of R to finite abelian groups, such that for every embedding (M, μ, m) of R , some embedding $(H, \tau, h) \in \mathcal{E}$ is a linear reduction of (M, μ, m) via injective homomorphisms f_i that are all the same.⁷*

Definition A.3 (Master embedding [[BKM24](#), Definition 3.6]). Let $k = 3$. Suppose $R \subseteq D^k$ does not have any non-trivial abelian embedding to $\mathbb{Z}$. Let $\mathcal{E}$ be a finite collection of abelian embeddings of R from [Theorem A.2](#). The master embedding (G, σ, g) of R is the direct product of embeddings from $\mathcal{E}$, i.e.

$$G \stackrel{\text{def}}{=} \prod_{(H, \tau, h) \in \mathcal{E}} H, \quad \sigma_i \stackrel{\text{def}}{=} \prod_{(H, \tau, h) \in \mathcal{E}} \tau_i \text{ for } i \in [k], \quad g \stackrel{\text{def}}{=} \prod_{(H, \tau, h) \in \mathcal{E}} h.$$

Remark A.4. The definition of linear reduction is similar to our definition of universal embedding ([Theorem 2.2](#)), with the following differences:

⁷That the injective homomorphisms f_i are all the same follows from inspecting the proof of [[BKM24](#), Lemma 3.3].

- [BKM24, Definition 3.1] requires G to be a subgroup of H and the homomorphisms f_i to be injective. We do not have such requirements in our [Theorem 2.2](#).
- Our [Theorem 2.2](#) allows f_i to be affine shifts of a common homomorphism, while the f_i are all homomorphisms in [BKM24, Definition 3.1].
- Linear reduction is only part of the definition of the master embedding. On the other hand, our universal embedding plays the same role as their master embedding, while also generalizing the latter in the presence of non-trivial embeddings to $\mathbb{Z}$.

In our opinion, the above differences are the reasons why universal embedding is simpler and easier to work with than master embedding.

Lemma A.5. *A master embedding is universal.*

Proof. Suppose (M, μ, m) is an embedding of R into a finite abelian group M . [Theorem A.2](#) implies there is an embedding $(H, \tau, h) \in \mathcal{E}$ and an injective homomorphism $f : H \rightarrow M$ such that $\mu_i = f \circ \tau_i$. The desired group homomorphism from G to M is given by $f \circ \pi$, where π is the projection to the (H, τ, h) -coordinate of G . Then $\mu_i = f \circ \tau_i = f \circ \pi \circ \sigma_i$, as required. $\square$

APPENDIX B. AFFINE HULL AS HYPERPLANE INTERSECTION

This section shows that the affine hull of a subset is the intersection of all hyperplanes containing it. This algebraic fact mirrors its counterpart in convex analysis and justifies the equivalence between our affine relaxation in [Eq. \(7\)](#) and the linear equations in [BKM26].

Given a subgroup H of a finitely generated abelian group G , recall the subgroup $H^\perp \stackrel{\text{def}}{=} \{\chi \in \hat{G} \mid \chi(h) = 1 \text{ for } h \in H\}$ of characters of G . It is well known that in a finitely generated⁸ abelian group G , every subgroup H satisfies $(H^\perp)^\perp = H$ via Pontryagin duality. This result can be restated as follows. Given a character χ of G , define the subgroup $\chi^\perp \stackrel{\text{def}}{=} \{h \in G \mid \chi(h) = 1\}$. Thus, $(H^\perp)^\perp = \bigcap \{\chi^\perp \mid \chi \in H^\perp\}$.

This conclusion generalizes to cosets:

Claim B.1. *For any finitely generated abelian group G , any coset $a + H$,*

$$a + H = \bigcap_{\chi \in Y} \{g \in G \mid \chi(g) = \chi(a)\},$$

where Y consists of characters $\chi \in \hat{G}$ such that $\chi(g) = \chi(a)$ for every $g \in a + H$.

Proof. The preceding paragraph shows that $H = \bigcap \{\chi^\perp \mid \chi \in H^\perp\}$.

Now $a + H = a + \bigcap \{\chi^\perp \mid \chi \in H^\perp\} = \bigcap \{a + \chi^\perp \mid \chi \in H^\perp\}$.

And $\chi^\perp = \{g \in G \mid \chi(g) = 1\}$, implying $a + \chi^\perp = \{g \in G \mid \chi(g) = \chi(a)\}$.

Finally, $\chi \in H^\perp$ if and only if $\chi \equiv 1$ on H , if and only if $\chi \equiv \chi(a)$ on $a + H$. $\square$

Given a k -tuple of maps $\sigma_i : D \rightarrow G$ and $a \in D^k$, let $\sigma(a) \stackrel{\text{def}}{=} (\sigma_i(a_i))_{i \in [k]} \in G^k$. Given a subset $R \subseteq D^k$, let $\sigma(R) \stackrel{\text{def}}{=} \{\sigma(a) \mid a \in R\} \subseteq G^k$.

Lemma B.2. *If (G, σ, g) is an embedding of $R \subseteq D^k$ into a finitely generated abelian group G ,*

$$\text{aff}(\sigma(R)) = \bigcap_{(\chi, z) \in Y} \{b \in G^k \mid \chi(b) = z\},$$

where Y consists of pairs $(\chi, z) \in \widehat{G^k} \times \mathbb{T}$ such that $\chi \equiv z$ on $\sigma(R)$, and $\mathbb{T}$ is the multiplicative circle group of complex numbers with unit magnitude.

⁸Or even locally compact. See <https://mathoverflow.net/q/85944>. Also [Con, Exercise 3.8 on page 12].

Proof. If R is nonempty, the result follows from [Theorem B.1](#). This follows from the fact that $\text{aff}(\sigma(R))$ is a coset of G^k . Also, $\chi \equiv z$ on $\sigma(R)$ if and only if $\chi \equiv z$ on $\text{aff}(\sigma(R))$.

If R is empty, then $\text{aff}(\sigma(R))$ is empty, as is the right-hand side, by taking any $\chi \in \widehat{G^k}$ and considering $(\chi, 1), (\chi, -1) \in Y$. $\square$

APPENDIX C. SDP SOLUTION

Lemma C.1 (Conditioning). *Given any level- d SDP hierarchy solution s for an instance $I = (V, \mathcal{C})$ and any $U \subseteq V$, if $d \geq 2|U|$, then for any $b \in \text{supp}(s(U))$, there is a level- $(d - |U|)$ SDP hierarchy solution r for I such that $\text{supp}(r(U)) = \{b\}$ and $\text{supp}(r(T)) \subseteq \text{supp}(s(T))$ for $T \in (\leq_{d-|U|}^V)$.*

Proof. Fix $b \in \text{supp}(s(U))$. For $T \in (\leq_{d-|U|}^V)$, $a \in D^T$, define $r(T, a) \stackrel{\text{def}}{=} s(T \cup U, a \cup b) / \|s(U, b)\|$ if $a|_{T \cap U} = b|_{T \cap U}$ (where a is consistent with b), and $r(T, a) \stackrel{\text{def}}{=} 0$ otherwise.

For $T \in (\leq_{d-|U|}^V)$, distinct $a, a' \in D^T$, we have $\langle r(T, a), r(T, a') \rangle = 0$. Indeed, if a is inconsistent with b , then $r(T, a) = 0$. Similarly, if a' is inconsistent with b , then $r(T, a') = 0$. If both a and a' are consistent with b , then $\langle r(T, a), r(T, a') \rangle = \langle s(T \cup U, a \cup b), s(T \cup U, a' \cup b) \rangle / \|s(U, b)\|^2 = 0$.

Also,

$$\sum_{a \in D^T} \|r(T, a)\|^2 = \sum_{\substack{a \in D^T \\ a|_{T \cap U} = b|_{T \cap U}}} \frac{\|s(T \cup U, a \cup b)\|^2}{\|s(U, b)\|^2} \stackrel{(25)}{=} \frac{\|s(U, b)\|^2}{\|s(U, b)\|^2} = 1.$$

For $T \in (\leq_{d-|U|}^V)$, $a \in D^T$, suppose $s(T, a) = 0$. If a is inconsistent with b , then $r(T, a) = 0$. Otherwise

$$\begin{aligned} \|r(T, a)\|^2 \|s(U, b)\|^2 &= \|s(T \cup U, a \cup b)\|^2 \leq \sum_{\substack{b \in D^S \\ b|_{T \cap S} = a|_{T \cap S}}} \|s(T \cup U, a \cup b)\|^2 \\ &\stackrel{(25)}{=} \|s(T, a)\|^2 = 0, \end{aligned}$$

so $r(T, a) = 0$ as well. Therefore $\text{supp}(r(T)) \subseteq \text{supp}(s(T)) \subseteq A_T$. Thus r is a level- d SDP hierarchy solution for I . And $\text{supp}(r(U)) = \{b\}$ because the only $a \in D^U$ consistent with b is b itself. $\square$

Lemma C.2. *Given any level- d SDP hierarchy solution s for an instance $I = (V, \mathcal{C})$, any $S \subseteq T \subseteq V$ such that $|T| \leq d$, any $b \in S$,*

$$(25) \quad \sum_{\substack{a \in D^T \\ a_S = b}} \|s(T, a)\|^2 = \|s(S, b)\|^2.$$

Proof. Since $s(T, a)$ and $s(T, a')$ are orthogonal for distinct $a, a' \in D^T$, the Pythagorean theorem implies that summing over $a \in D^T$ such that $a_S = b$,

$$\sum_{a: a_S = b} \|s(T, a)\|^2 = \left\| \sum_{a: a_S = b} s(T, a) \right\|^2 \stackrel{(14)}{=} \|\pi_S(s(T))(b)\|^2 \stackrel{(16)}{=} \|s(S, b)\|^2. \quad \square$$

In some literature, [\[CNP24\]](#), the SDP hierarchy has a slightly different definition, involving both vectors and distributions. In [\[CNP24\]](#), a level- d SDP solution consists of

- Vector assignments $\alpha_S : A_S \rightarrow \mathcal{X}$ to an arbitrary inner product space $\mathcal{X}$ for variable subsets S of size at most d , and
- Distributions $\mu_R \in \Delta(A_R)$ for variable subsets R of size at most $2d$,

such that

- (a) $\{\alpha_S\}$ is consistent with projection ([Eq. \(16\)](#));
- (b) $\{\mu_R\}$ is consistent with projection; and

(c) for any d -small $S, T \subseteq V$, any $a_S \in A_S, a_T \in A_T$,

$$(26) \quad \langle \alpha_S(a_S), \alpha_T(a_T) \rangle_{\mathcal{X}} = \mathbb{P}_{b \sim \mu_{S \cup T}} [b_S = a_S, b_T = a_T].$$

The above definition is equivalent to our vector-only definition ([Theorem 4.2](#)), losing at most a factor of 2 in level.

Lemma C.3. *If the level- d SDP hierarchy of [\[CNP24\]](#) has a solution, then so does the level- d SDP hierarchy defined in this paper. If the level- $2d$ SDP hierarchy in this paper has a solution, then so does the level- d SDP hierarchy of [\[CNP24\]](#).*

Proof. Suppose the [\[CNP24\]](#) SDP hierarchy has a level- d solution $(\{\alpha_S\}, \{\mu_R\})$. [Eq. \(26\)](#) implies $\alpha_S(a)$ is orthogonal to $\alpha_S(a')$ for $S \in \binom{V}{\leq d}$ and distinct $a, a' \in A_S$. Further, $\{\alpha_S\}$ is consistent with projection by assumption. Therefore $\{\alpha_S\}$ is a level- d solution to the SDP hierarchy in our paper.

Suppose the SDP hierarchy in our paper has a level- $2d$ solution $\{\alpha_S\}$. Define the distributions μ_R by $\mu_R(b) \stackrel{\text{def}}{=} \|\alpha_R(b)\|^2$ for $R \in \binom{V}{\leq 2d}, b \in D^R$. Then $\{\alpha_S\}_{S \in \binom{V}{\leq d}}$ is consistent with projection by assumption. [Theorem C.2](#) implies $\{\mu_R\}_{R \in \binom{V}{\leq 2d}}$ is consistent with projection. Finally, for any $S, T \in \binom{V}{\leq d}$, let $R \stackrel{\text{def}}{=} S \cup T$. Then for $a_S \in D^S, a_T \in D^T$,

$$\begin{aligned} \langle \alpha_S(a_S), \alpha_T(a_T) \rangle_{\mathcal{X}} &= \sum_{\substack{b \in D^R \\ b_S = a_S}} \sum_{\substack{b' \in D^R \\ b'_T = a_T}} \langle \alpha_R(b), \alpha_R(b') \rangle_{\mathcal{X}} = \sum_{\substack{b \in D^R \\ b_S = a_S \\ b_T = a_T}} \|\alpha_R(b)\|^2 \\ &= \mathbb{P}_{b \sim \mu_{S \cup T}} [b_S = a_S, b_T = a_T], \end{aligned}$$

where the first equality follows from the consistency of $\{\alpha_S\}_{S \in \binom{V}{\leq 2d}}$ with projection, and the second equality uses orthogonality of $\alpha_R(b)$ and $\alpha_R(b')$ for distinct $b, b' \in D^R$. $\square$

APPENDIX D. VALUE OF RANDOM INSTANCE

A variant of the following lemma has appeared in [\[CdX26, Lemma F.5\]](#). The only difference is that [\[CdX26, Lemma F.5\]](#) concerns the binomial (also known as Erdős–Rényi) model of random hypergraph, while [Theorem D.1](#) has m independent hyperedges with replacement ([Theorem 5.4](#)).

Lemma D.1. *Fix any distribution ρ over k -ary relations on D . As long as $\Delta \geq \Omega(\log|D|)$, except with probability $o_n(1)$ over a ρ -random CSP instance $I = (V, \mathcal{C})$ with n variables and Δn constraints,*

$$\text{opt}(I) = \text{opt}_k(\rho) + o_n(1) + o_{\Delta}(1).$$

Proof. Fix any assignment $b \in D^V$. Let ν be the distribution over D of $b(v)$ by sampling a uniformly random $v \in V$. Consider picking a single random constraint with scope $S \sim V^k$ and relation R . Since the uniform distributions over V^k and over V^k have statistical distance $o_{n;k}(1)$,

$$\mathbb{P}_{S \sim V^k} [b_S \in R] = \mathbb{P}_{S \sim V^k} [b_S \in R] + o_n(1) = \mathbb{P}_{a \sim \nu^k} [a \in R] + o_n(1).$$

Further taking expectation over $R \sim \rho$, an assignment b satisfies a random constraint with probability $s \leq \text{opt}_k(\rho) + o_n(1)$ ([Eq. \(22\)](#)). With m independent constraints, Chernoff bound implies b satisfies more than $sm + \alpha$ constraints with probability at most $\exp(-\alpha^2/m)$. A union bound over all $b \in D^n$ implies $\text{opt}(I) \leq \text{opt}_k(\rho) + o_n(1) + o_{\Delta}(1)$.

For the reverse inequality, consider a distribution η that maximizes [Eq. \(22\)](#). Enumerate the domain D as $[q] = \{1, \dots, q\}$ and the vertex set V as $\{v_1, \dots, v_n\}$. Define an assignment $c \in D^V$ from η by “rounding”: Let $f : [q] \rightarrow \mathbb{R}$ be the cdf of η , so that $f(i) := \sum_{j \leq i} \eta(j)$. Then $c(v_i) \stackrel{\text{def}}{=} \arg \min_{j \in [q]} f(j) \geq i/n$. Let ν be the distribution over D of $c(v)$ by sampling a uniformly random $v \in V$. Then $|\nu(a) - \eta(a)| = O(1/n)$ for every $a \in D$, and hence $\|\nu - \eta\|_1 = o_n(1)$. Now

c satisfies a random constraint with probability $r \stackrel{\text{def}}{=} \mathbb{E}_{R \sim \rho} \mathbb{P}_{S \sim V^k}[c_S \in R] = \text{opt}_k(\rho) + o_n(1)$. Out of m independent constraints, c satisfies fewer than $rm - \alpha$ constraints with probability at most $\exp(-\alpha^2/m)$. Therefore $\text{opt}(I) \geq \text{opt}_k(\rho) + o_n(1) + o_\Delta(1)$ except with probability $o_n(1)$. $\square$

REFERENCES

- [ABS15] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. *J. ACM*, 62(5), November 2015.
- [AFO⁺25] Sergey Aravakumov, Marek Filakovský, Jakub Opršal, Gianluca Tasinato, and Uli Wagner. Hardness of 4-Colouring G -Colourable Graphs. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, page 72–83. Association for Computing Machinery, 2025.
- [AR01] M. Alekhnovich and A. Razborov. Lower Bounds for Polynomial Calculus: Non-Binomial Case. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, FOCS '01, page 190, USA, 2001. IEEE Computer Society.
- [BBK⁺26] Amey Bhangale, Mark Braverman, Subhash Khot, Yang Liu, Dor Minzer, and Kunal Mittal. An Analytical Approach to Parallel Repetition via CSP Inverse Theorems. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing*, STOC '26, page 298–304. Association for Computing Machinery, 2026.
- [BG17] Christoph Berkholz and Martin Grohe. Linear Diophantine Equations, Group CSPs, and Graph Isomorphism. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '17, page 327–339, USA, 2017. Society for Industrial and Applied Mathematics.
- [BG21] Joshua Brakensiek and Venkatesan Guruswami. Promise Constraint Satisfaction: Algebraic Structure and a Symmetric Boolean Dichotomy. *SIAM Journal on Computing*, 50(6):1663–1700, 2021.
- [BGMT12] Siavosh Benabbas, Konstantinos Georgiou, Avner Magen, and Madhur Tulsiani. SDP Gaps from Pairwise Independence. *Theory of Computing*, 8(12):269–289, 2012.
- [BGS25] Joshua Brakensiek, Venkatesan Guruswami, and Sai Sandeep. SDPs and Robust Satisfiability of Promise CSP. *Discrete Analysis*, sep 9 2025.
- [BGWŽ20] Joshua Brakensiek, Venkatesan Guruswami, Marcin Wrochna, and Stanislav Živný. The Power of the Combined Basic Linear Programming and Affine Relaxation for Promise Constraint Satisfaction Problems. *SIAM Journal on Computing*, 49(6):1232–1248, 2020.
- [Bha25] Amey Bhangale. The Quest Towards Optimal Approximability of CSPs. *ACM SIGLOG News*, 12(1):4–30, March 2025.
- [BHZ26] Libor Barto, Maximilian Hadek, and Dmitriy Zhuk. Toward a Uniform Algorithm and Uniform Reduction for Constraint Problems. *CoRR*, abs/2604.06335, 2026.
- [BKLM25] Amey Bhangale, Subhash Khot, Yang P. Liu, and Dor Minzer. On Inverse Theorems and Combinatorial Lines. In *66th IEEE Annual Symposium on Foundations of Computer Science*, FOCS '25, pages 1672–1684. IEEE, 2025.
- [BKM21] Mark Braverman, Subhash Khot, and Dor Minzer. On Rich 2-to-1 Games. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.
- [BKM24] Amey Bhangale, Subhash Khot, and Dor Minzer. On Approximability of Satisfiable k -CSPs: IV. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, page 1423–1434. Association for Computing Machinery, 2024.
- [BKM26] Amey Bhangale, Subhash Khot, and Dor Minzer. On Approximability of Satisfiable k -CSPs: V. *TheoretCS*, Volume 5, May 2026.
- [BKT18] Amey Bhangale, Subhash Khot, and Devanathan Thiruvengatathari. An Improved Dictatorship Test with Perfect Completeness. In *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2017)*, volume 93 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 15:1–15:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018.
- [BRS11] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding Semidefinite Programming Hierarchies via Global Correlation. In *FOCS 2011*, pages 472–481, 2011.
- [Bul17] Andrei A. Bulatov. A Dichotomy Theorem for Nonuniform CSPs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science*, pages 319–330, 2017.
- [BW26] Manuel Bodirsky and Armin Weiß. The Complexity of Finding Coset-Generating Polymorphisms and the Promise Metaproblem. In *53rd International Colloquium on Automata, Languages, and Programming (ICALP 2026)*, volume 374 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 169:1–169:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026.
- [CdX26] Siu On Chan, Tommaso d’Orsi, and Jeff Xu. Strongly Refuting Random CSP without Literals. In *FOCS '26*, 2026. To appear.

- [CGP26] Jonas Conneryd, Yassine Ghannane, and Shuo Pang. Lower Bounds for CSP Hierarchies Through Ideal Reduction. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 6436–6463, 2026.
- [Cha16] Siu On Chan. Approximation Resistance from Pairwise-Independent Subgroups. *Journal of the ACM*, 63(3), 2016.
- [CM13] Siu On Chan and Michael Molloy. A Dichotomy Theorem for the Resolution Complexity of Random Constraint Satisfaction Problems. *SIAM Journal on Computing*, 42(1):27–60, 2013.
- [CN25] Siu On Chan and Hiu Tsun Ng. How Random CSPs Fool Hierarchies: II. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC ’25, page 1710–1721. Association for Computing Machinery, 2025.
- [CNP24] Siu On Chan, Hiu Tsun Ng, and Sijin Peng. How Random CSPs Fool Hierarchies. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, page 1944–1955. Association for Computing Machinery, 2024.
- [Con] Keith Conrad. Characters of finite abelian groups. <https://kconrad.math.uconn.edu/blurbs/grouptheory/charthy.pdf>.
- [CŽ22] Lorenzo Ciardo and Stanislav Živný. CLAP: A New Algorithm for Promise CSPs. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1057–1068, 2022.
- [CŽ25] Lorenzo Ciardo and Stanislav Živný. Semidefinite Programming and Linear Equations vs. Homomorphism Problems. *SIAM Journal on Computing*, 54(3):545–584, 2025.
- [CŽ26] Lorenzo Ciardo and Stanislav Živný. Hierarchies of Minion Tests for PCSPs through Tensors. *ACM Trans. Algorithms*, 22(2), April 2026.
- [DB97] Romuald Debruyne and Christian Bessière. Some Practicable Filtering Techniques for the Constraint Satisfaction Problem. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, IJCAI 97*, pages 412–417. Morgan Kaufmann, August 1997.
- [DO24] Víctor Dalmau and Jakub Opršal. Local Consistency as a Reduction Between Constraint Satisfaction Problems. In *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS ’24*. Association for Computing Machinery, 2024.
- [Hua14] Sangxia Huang. Approximation Resistance on Satisfiable Instances for Sparse Predicates. *Theory of Computing*, 10(14):359–388, 2014.
- [KMOW17] Pravesh K. Kothari, Ryuhei Mori, Ryan O’Donnell, and David Witmer. Sum of Squares Lower Bounds for Refuting Any CSP. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, page 132–145. ACM, 2017.
- [KOWŽ23] Andrei Krokhin, Jakub Opršal, Marcin Wrochna, and Stanislav Živný. Topology and Adjunction in Promise Constraint Satisfaction. *SIAM Journal on Computing*, 52(1):38–79, 2023.
- [LP25] Moritz Lichter and Benedikt Pago. Limitations of Affine Integer Relaxations for Solving Constraint Satisfaction Problems. In *52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025)*, volume 334 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 166:1–166:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025.
- [MM14] Konstantin Makarychev and Yuri Makarychev. Approximation Algorithm for Non-Boolean Max- k -CSP. *Theory of Computing*, 10(13):341–358, 2014.
- [ÓC22] Adam Ó Conghaile. Cohomology in Constraint Satisfaction and Structure Isomorphism. In *47th International Symposium on Mathematical Foundations of Computer Science, 2022*, volume 241, pages 75:1–75:16, 2022.
- [Rag08] Prasad Raghavendra. Optimal Algorithms and Inapproximability Results for Every CSP? In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC ’08, page 245–254. Association for Computing Machinery, 2008.
- [Tul09] Madhur Tulsiani. CSP Gaps and Reductions in the Lasserre Hierarchy. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, page 303–312. ACM, 2009.
- [TY15] Suguru Tamaki and Yuichi Yoshida. A Query Efficient Non-Adaptive Long Code Test with Perfect Completeness. *Random Structures & Algorithms*, 47(2):386–406, 2015.
- [Zhu20] Dmitriy Zhuk. A Proof of the CSP Dichotomy Conjecture. *Journal of the ACM*, 67(5), 2020.
- [Zhu26] Dmitriy Zhuk. Singleton Algorithms for the Constraint Satisfaction Problem. In *FOCS ’26*, 2026. To appear.