Electronic Colloquium on Computational Complexity, Report No. 210 (2018)

On Closest Pair in Euclidean Metric: Monochromatic is as Hard as Bichromatic

Karthik C. S.[*]
Weizmann Institute of Science
karthik.srikanta@weizmann.ac.il

Pasin Manurangsi[†]
University of California, Berkeley
pasin@berkeley.edu

Abstract

Given a set of n points in $\mathbb{R}^d$, the (monochromatic) *Closest Pair* problem asks to find a pair of distinct points in the set that are closest in the ℓ_p-metric. Closest Pair is a fundamental problem in Computational Geometry and understanding its fine-grained complexity in the Euclidean metric when $d = \omega(\log n)$ was raised as an open question in recent works (Abboud-Rubinstein-Williams [FOCS'17], Williams [SODA'18], David-Karthik-Laekhanukit [SoCG'18]).

In this paper, we show that for every $p \in \mathbb{R}_{\geq 1} \cup \{0\}$, under the Strong Exponential Time Hypothesis (SETH), for every $\varepsilon > 0$, the following holds:

- No algorithm running in time $O(n^{2-\varepsilon})$ can solve the Closest Pair problem in $d = (\log n)^{\Omega_\varepsilon(1)}$ dimensions in the ℓ_p-metric.
- There exists $\delta = \delta(\varepsilon) > 0$ and $c = c(\varepsilon) \geq 1$ such that no algorithm running in time $O(n^{1.5-\varepsilon})$ can approximate Closest Pair problem to a factor of $(1 + \delta)$ in $d \geq c \log n$ dimensions in the ℓ_p-metric.

In particular, our first result is shown by establishing the computational equivalence of the *bichromatic* Closest Pair problem and the (monochromatic) Closest Pair problem (up to n^ε factor in the running time) for $d = (\log n)^{\Omega_\varepsilon(1)}$ dimensions.

Additionally, under SETH, we rule out nearly-polynomial factor approximation algorithms running in subquadratic time for the (monochromatic) *Maximum Inner Product* problem where we are given a set of n points in $n^{o(1)}$-dimensional Euclidean space and are required to find a pair of distinct points in the set that maximize the inner product.

At the heart of all our proofs is the construction of a dense bipartite graph with low *contact dimension*, i.e., we construct a balanced bipartite graph on n vertices with $n^{2-\varepsilon}$ edges whose vertices can be realized as points in a $(\log n)^{\Omega_\varepsilon(1)}$-dimensional Euclidean space such that every pair of vertices which have an edge in the graph are at distance exactly 1 and every other pair of vertices are at distance greater than 1. This graph construction is inspired by the construction of locally dense codes introduced by Dumer-Miccancio-Sudan [IEEE Trans. Inf. Theory'03].

[*] Supported by Irit Dinur's ERC-CoG grant 772839 and BSF grant 2014371.
[†] Supported by NSF under Grants No. CCF 1655215 and CCF 1815434.

ISSN 1433-8092

Contents

1 Introduction

The Closest Pair of Points problem or *Closest Pair* problem (CP) is a fundamental problem in computational geometry: given n points in a d-dimensional metric space, find a pair of distinct points with the smallest distance between them. The Closest Pair problem for points in the Euclidean plane [SH75, BS76] stands at the origins of the systematic study of the computational complexity of geometric problems [PS85, Man89, KT05, CLRS09]. Since then, this problem has found abundant applications in geographic information systems [Hen06], clustering [Zah71, Alp10], and numerous matching problems (such as stable marriage [WTFX07]).

The trivial algorithm for CP examines every pair of points in the point-set and runs in time $O(n^2 d)$. Over the decades, there have been a series of developments on CP in low dimensional space for the Euclidean metric [Ben80, HNS88, KM95, SH75, BS76], leading to a deterministic $O(2^{O(d)} n \log n)$-time algorithm [BS76] and a randomized $O(2^{O(d)} n)$-time algorithm [Rab76, KM95]. For low (i.e., constant) dimensions, these algorithms are tight as a matching lower bound of $\Omega(n \log n)$ was shown by Ben-Or [Ben83] and Yao [Yao91] in the *algebraic decision tree* model, thus settling the complexity of CP in low dimensions. On other hand, for very high dimensions (i.e., $d = \Omega(n)$) there are subcubic algorithms [GS16, ILLP04] in the ℓ_1, ℓ_2, and ℓ_∞-metrics using fast matrix multiplication algorithms [Gal14]. However, CP in medium dimensions, i.e., $d = \text{polylog}(n)$, and in various ℓ_p-metrics, have been a focus of study in machine learning and analysis of Big Data [Kle97], and it is surprising that, even with the tools and techniques that have been developed over many decades, when $d = \omega(\log n)$, there is no known subquadratic-time (i.e., $O(2^{o(d)} n^{2-\varepsilon})$-time) algorithm, for CP in any standard distance measure [Ind00, AC09, ILLP04] . The absence of such algorithms was explicitly observed as early as the late nineties by Cohen and Lewis [CL99] but there was not any explanation until recently.

David, Karthik, and Laekhanukit [DKL18] showed that for all $p > 2$, assuming the *Strong Exponential Time Hypothesis* (SETH), for every $\varepsilon > 0$, no algorithm running in $n^{2-\varepsilon}$ time can solve CP in the ℓ_p-metric, even when $d = \omega(\log n)$. Their conditional lower bound was based on the conditional lower bound (again assuming SETH) of Alman and Williams [AW15] for the *Bichromatic Closest Pair* problem[1] (BCP) where we are given two sets of n points in a d-dimensional metric space, and the goal is to find a pair of points, one from each set, with the smallest distance between them. Alman and Williams showed that for all $p \in \mathbb{R}_{\geq 1} \cup \{0\}$, assuming SETH, for every $\varepsilon > 0$, no algorithm running in $n^{2-\varepsilon}$ time can solve BCP in the $\omega(\log n)$-dimensional ℓ_p-metric space. Given that [AW15] show their lower bound on BCP for all ℓ_p-metrics, the lower bound on CP of [DKL18] feels unsatisfactory, since the ℓ_2-metric is arguably the most interesting metric to study CP on. On the other hand, the answer to the complexity of CP in the Euclidean metric might be on the positive side, i.e., there might exist an algorithm that performs well in the ℓ_2-metric because there are more tools available, e.g., Johnson-Lindenstrauss dimension reduction [JL84]. Thus we have the following question:

Open Question 1.1 (Abboud-Rubinstein-Williams[2] [ARW17b], Williams [Wil18a], David -Karthik-Laekhanukit [DKL18]). *Is there an algorithm running in time $n^{2-\varepsilon}$ for some $\varepsilon > 0$*

[1] We remark that BCP is of independent interest as it's equivalent to finding the *Minimum Spanning Tree* in ℓ_p-metric [AESW91, KLN99]. Moreover, understanding the fine-grained complexity of BCP has lead to better understanding of the query time needed for *Approximate Nearest Neighbor* search problem (see Razenshteyn's thesis [Raz17] for a survey about the problem) with polynomial preprocessing time [Rub18].

[2] Please see the erratum in [ARW17a].

which can solve CP *in the Euclidean metric when the points are in* $\omega(\log n)$ *dimensions?*

Even if the answer to the above question is negative, this does not rule out strong approximation algorithms for CP in the Euclidean metric, which might suffice for all applications. Indeed, we do know of subquadratic approximation algorithms for CP. For example, LSH based techniques can solve $(1+\delta)$-CP (i.e., $(1+\delta)$ factor approximate CP) in $n^{2-\Theta(\delta)}$ time [IM98], but cannot do much better [MNP07, OWZ14]. In a recent breakthrough, Valiant [Val15] obtained an approximation algorithm for $(1+\delta)$-CP with runtime of $n^{2-\Theta(\sqrt{\delta})}$. The state of the art is an $n^{2-\widetilde{\Theta}(\delta^{1/3})}$-time algorithm by Alman, Chan, and Williams [ACW16]. Can the dependence on δ be improved indefinitely? For the case of $(1+\delta)$-BCP, assuming SETH, Rubinstein [Rub18] answered the question in the negative. Does $(1+\delta)$-CP also admit the same negative answer?

Open Question 1.2. *Is there an algorithm running in time* $n^{2-\varepsilon}$ *for some* $\varepsilon > 0$ *which can solve* $(1+\delta)$-CP *in the Euclidean metric when the points are in* $\omega(\log n)$ *dimensions for every* $\delta > 0$*?*

Another important geometric problem is the *Maximum Inner Product* problem (MIP): given n points in the d-dimensional Euclidean space, find a pair of distinct points with the largest inner product. This problem along with its bichromatic variant (*Bichromatic Maximum Inner Product* problem, denoted BMIP) is extensively studied in literature (see [ARW17b] and references therein). Abboud, Rubinstein, and Williams [ARW17b] showed that assuming SETH, for every $\varepsilon > 0$, no $2^{(\log n)^{1-o(1)}}$-approximation algorithm running in $n^{2-\varepsilon}$ time can solve BMIP when $d = n^{o(1)}$. It is a natural question to ask if their inapproximability result can be extended to MIP:

Open Question 1.3. *Is there an algorithm running in time* $n^{2-\varepsilon}$ *for some* $\varepsilon > 0$ *which can solve* γ-MIP *in* $n^{o(1)}$ *dimensions for even* $\gamma = 2^{(\log n)^{1-o(1)}}$*?*

1.1 Our Results

In this paper we address all three previously mentioned open questions. First, we almost completely resolve Open Question 1.1. In particular, we show the following.

Theorem 1.4 (Subquadratic Hardness of CP; Informal, See Theorem 4.3)**.** *Let* $p \in \mathbb{R}_{\geq 1} \cup \{0\}$*. Assuming* SETH*, for every* $\varepsilon > 0$*, no algorithm running in* $n^{2-\varepsilon}$ *time can solve* CP *in the* ℓ_p*-metric, even when* $d = (\log n)^{\Omega_\varepsilon(1)}$*.*

In particular we would like to emphasize that the dimension for which we show the lower bound on CP depends on ε. We would also like to remark that our lower bound holds even when the input point-set of CP is a subset of $\{0,1\}^d$. Finally, we note that the centerpiece of the proof of the above theorem (and also the proofs of the other results that will be subsequently mentioned) is the construction of a dense bipartite graph with low *contact dimension*, i.e., we construct a balanced bipartite graph on n vertices with $n^{2-\varepsilon}$ edges whose vertices can be realized as points in a $(\log n)^{\Omega_\varepsilon(1)}$-dimensional ℓ_p-metric space such that every pair of vertices which have an edge in the graph are at distance exactly 1 and every other pair of vertices are at distance greater than 1. This graph construction is inspired by the construction of locally dense codes introduced by Dumer, Miccancio, and Sudan [DMS03] and uses special density properties of Reed Solomon codes. A detailed proof overview is given in Section 2.1.

Next, we improve our result in Theorem 1.4 in some aspects by showing $1 + o(1)$ factor inapproximability of CP even in $O_\varepsilon(\log n)$ dimensions, but can only rule out algorithms running in $n^{1.5-\varepsilon}$ time (as opposed to Theorem 1.4 which rules out exact algorithms for CP running in $n^{2-\varepsilon}$ time). More precisely, we show the following.

Theorem 1.5 (Subquadratic Hardness of gap-CP)**.** *Let $p \in \mathbb{R}_{\geq 1} \cup \{0\}$. Assuming* SETH*, for every $\varepsilon > 0$, there exists $\delta(\varepsilon) > 0$ and $c(\varepsilon) > 1$ such that no algorithm running in $n^{1.5-\varepsilon}$ time that can solve $(1 + \delta)$-*CP *in the ℓ_p-metric, even when $d = c \log n$.*

We remark that the $n^{1.5-\varepsilon}$ lower bound on approximate CP is an artifact of our proof strategy and that a different approach or an improvement in the state-of-the-art bound on the number of minimum weight codewords in algebraic geometric codes (which are used in our proof), will lead to the complete resolution of Open Question 1.2.

It should also be noted that the approximate version of CP and the dimension are closely related. Namely, using standard dimensionality reduction techniques [JL84][3] for $(1 + \delta)$-CP, one can always assume that $d = O_\delta(\log n)$. In other words, hardness of $(1 + \delta)$-CP immediately yields logarithmic dimensionality bound as a byproduct.

Finally, we completely answer Open Question 1.3 by showing the following inapproximability result for MIP, matching the hardness for BMIP from [ARW17b].

Theorem 1.6 (Subquadratic Hardness of gap-MIP)**.** *Assuming* SETH*, for every $\varepsilon > 0$, no algorithm running in $n^{2-\varepsilon}$ time can solve γ-*MIP *for any $\gamma \leq 2^{(\log n)^{1-o(1)}}$, even when $d = n^{o(1)}$.*

Recently, there have been a lot of results connecting BCP or $(1 + o(1))$-BCP to other problems (see [Rub18, Che18a, Che18b, CW19]). Now such connections can be extended to CP as well. For example, the following conditional lower bound follows from [Rub18] for gap-CP in the edit distance metric and for completeness a proof is given in Appendix A.

Theorem 1.7 (Subquadratic Hardness of gap-CP in edit distance metric)**.** *Assuming* SETH*, for every $\varepsilon > 0$, there exists $\delta(\varepsilon) > 0$ and $c(\varepsilon) > 1$ such that no algorithm running in $n^{1.5-\varepsilon}$ time can solve $(1 + \delta)$-*CP *in the edit distance metric, even when $d = c \log n \log \log n$.*

2 Proof Overview

In this section, we provide an overview of our proofs. For ease of presentation, we will sometimes be informal here; all notions and proofs are formalized in subsequent sections. Our overview is organized as follows. First, in Subsection 2.1, we outline our proof of running time lower bounds for exact CP (Theorem 1.4). Then, in Subsection 2.2, we abstract part of our reduction using error-correcting codes, and relate them back to the works on locally dense codes [DMS03, CW12, Mic14] that inspire our constructions. Finally, in Subsection 2.3, we briefly discuss how to modify the base construction (i.e. code properties) to give conditional lower bounds for approximate CP and MIP (Theorems 1.5 and 1.6).

[3] In fact, since our results applies to $\{0, 1\}$-vectors, simply subsampling coordinates would also work.

2.1 Conditional Lower Bound on Exact Closest Pair

In this subsection, we provide a proof overview of a slightly weaker version of Theorem 1.4, i.e., we show that assuming SETH, for every $p \in \mathbb{R}_{\geq 1} \cup \{0\}$, no subquadratic time algorithm can solve CP in the ℓ_p-metric when $d = (\log n)^{\omega(1)}$. We prove such a result by reducing BCP in dimension d to CP in dimension $d + (\log n)^{\omega(1)}$, and the subquadratic hardness for CP follows from the subquadratic hardness of BCP established by [AW15]. Note that the results in this paper remain interesting even if SETH is false, as our reduction shows that BCP and CP are computationally equivalent[4] (up to $n^{o(1)}$ factor in the running time) when $d = (\log n)^{\omega(1)}$. The conditional lower bound on CP is merely a consequence of this computational equivalence. Finally, we note that a similar equivalence also holds between MIP and BMIP.

Understanding an obstacle of [DKL18]. Our proof builds on the ideas of [DKL18] who showed that assuming SETH, for every $p > 2$, no subquadratic time algorithm can solve CP in the ℓ_p-metric when $d = \omega(\log n)$. They did so by connecting the complexity of CP and BCP via the *contact dimension* of the balanced complete bipartite graph (biclique), denoted by $K_{n,n}$. We elaborate on this below.

To motivate the idea behind [DKL18], let us first consider the trivial reduction from BCP to CP: given an instance A, B of BCP, we simply output $A \cup B$ as an instance of CP. This reduction fails because there is no guarantee on the distances of a pair of points both in A (or both in B). That is, there could be two points $\mathbf{a}, \mathbf{a}' \in A$ such that $\|\mathbf{a} - \mathbf{a}'\|_p$ is much smaller than the optimum of BCP on A, B. If we simply solve CP on $A \cup B$, we might find such $\mathbf{a}, \mathbf{a}'$ as the optimal pair but this does not give the answer to the original BCP problem. In order to circumvent this issue, one needs a gadget that "stretch" pairs of points both in A or both in B further apart while keeping the pairs of points across A and B close (and preserving the optimum of BCP on A, B). It turns out that this notion corresponds exactly to the contact dimension of the biclique, which we define below.

Definition 2.1 (Contact Dimension [Pac80])**.** *For any graph $G = (V, E)$, a mapping $\tau : V \to \mathbb{R}^d$ is said to* realize G (in the ℓ_p-metric) *if for some $\beta > 0$, the following holds for every distinct vertices u, v:*

$$\|\tau(u) - \tau(v)\|_p = \beta \text{ if } \{u, v\} \in E, \text{ and,} \tag{1}$$

$$\|\tau(u) - \tau(v)\|_p > \beta \text{ otherwise.} \tag{2}$$

The contact dimension *(in the ℓ_p-metric) of G, denoted by $\mathsf{cd}_p(G)$, is the minimum $d \in \mathbb{N}$ such that there exists $\tau : V \to \mathbb{R}^d$ realizing G in the ℓ_p-metric.*

In this paper, we will be mainly interested in the contact dimension of bipartite graphs. Specifically, [DKL18] only consider the contact dimension of the biclique $K_{n,n}$. Notice that a realization of biclique ensures that vertices on the same side are far from each other while vertices on different sides are close to each other preserving the optimum of BCP; these are exactly the desired properties of a gadget outlined above. Using this, [DKL18] give a reduction from BCP to CP which shows that the two are computationally equivalent whenever $d = \Omega(\mathsf{cd}_p(K_{n,n}))$, as follows.

[4] We can reduce an instance of CP to an instance of BCP by randomly partitioning the input set of CP instance into two, and the optimal closest pair of points will be in different sets with probability 1/2 (and this reduction can be made deterministic).

Let $A, B \subseteq \mathbb{R}^d$ each of cardinality n be an instance of BCP and let $\tau : A \dot\cup B \to \mathbb{R}^{\mathsf{cd}_p(K_{n,n})}$ be a map realizing the biclique $(A \dot\cup B, A \times B)$ in the ℓ_p-metric; we may assume w.l.o.g. that $\beta = 1$. Let δ be the distance between any point in A and any point in B (i.e., δ is an upper bound on the optimum of BCP). Let $\rho > 0$ be such that $\|\tau(\mathbf{a}) - \tau(\mathbf{b})\|_p > 1 + \rho$ for all $\mathbf{a} \in A, \mathbf{b} \in B$ (and this is guaranteed to exist by (2)). Moreover, let $k > \delta/\rho$ be any sufficiently large number. Consider the point-sets $\widetilde{A}, \widetilde{B} \subseteq \mathbb{R}^{d+\mathsf{cd}_p(K_{n,n})}$ of cardinality n each defined as

$$\widetilde{A} = \{\mathbf{a} \circ (k \cdot \tau(\mathbf{a})) \mid \mathbf{a} \in A\}, \ \widetilde{B} = \{\mathbf{b} \circ (k \cdot \tau(\mathbf{b})) \mid \mathbf{b} \in B\},$$

where $\circ$ denotes the concatenation between two vectors and $k \cdot \mathbf{x}$ denotes the usual scalar-vector multiplication (i.e. scaling $\mathbf{x}$ up by a factor of k). For brevity, we write $\widetilde{\mathbf{a}}$ and $\widetilde{\mathbf{b}}$ to denote $\mathbf{a} \circ (k \cdot \tau(\mathbf{a}))$ and $\mathbf{b} \circ (k \cdot \tau(\mathbf{b}))$ respectively.

We now argue that, if we can find the closest pair of points in $\widetilde{A} \cup \widetilde{B}$, then we also immediately solve BCP for (A, B). More precisely, we claim that $(\mathbf{a}^*, \mathbf{b}^*) \in A \times B$ is a bichromatic closest pair of (A, B) if and only if $(\widetilde{\mathbf{a}^*}, \widetilde{\mathbf{b}^*})$ is a closest pair of $\widetilde{A} \cup \widetilde{B}$.

To see that this is the case, observe that, for cross pairs $(\widetilde{\mathbf{a}}, \widetilde{\mathbf{b}}) \in \widetilde{A} \times \widetilde{B}$, (1) implies that the distance $\|\widetilde{\mathbf{a}} - \widetilde{\mathbf{b}}\|_p$ is exactly $(k^p + \|\mathbf{a} - \mathbf{b}\|_p^p)^{1/p}$; hence, among these pairs, $(\widetilde{\mathbf{a}^*}, \widetilde{\mathbf{b}^*})$ is a closest pair iff $(\mathbf{a}^*, \mathbf{b}^*)$ is a bichromatic closest pair in A, B. Notice also that, since the bichromatic closest pair in A, B is of distance at most δ, the closest pair in $\widetilde{A} \cup \widetilde{B}$ is of distance at most $(k^p + \delta^p)^{1/p} \leq k + \delta$.

On the other hand, for pairs both from $\widetilde{A}$ or both from $\widetilde{B}$, the distance must be at least $k(1 + \rho)$, which is more than $k + \delta$ from our choice of k. As a result, these pairs cannot be a closest pair in $\widetilde{A} \cup \widetilde{B}$, and this concludes the sketch of the proof.

There are a couple of details that we have glossed over here: one is that the gap ρ cannot be too small (e.g., ρ cannot be as small as $1/2^n$) and the other is that we should be able to construct τ efficiently. Nevertheless, these are typically not an issue.

[DKL18] show that $\mathsf{cd}_p(K_{n,n}) = \Theta(\log n)$ when $p > 2$ and that the realization can be constructed efficiently and with sufficiently large ρ. This implies the subquadratic hardness of CP (by reduction from BCP) in the ℓ_p-metric for all $p > 2$ and $d = \omega(\log n)$. However, it was known that $\mathsf{cd}_2(K_{n,n}) = \Theta(n)$ [FM88]. Thus, they could *not* extend their conditional lower bound to CP in the Euclidean metric[5] even when $d = o(n)$. In fact, this is a serious obstacle as it rules out many natural approaches to reduce BCP to CP in a black-box manner. Elaborating, the lower bound on $\mathsf{cd}_2(K_{n,n})$ rules out local gadget reductions which would replace each point with a composition of that point and a gadget with a small increase in the number of dimensions, as such gadgets can be used to construct a realization of $K_{n,n}$ in the Euclidean metric in a low dimensional space, contradicting the lower bound on $\mathsf{cd}_2(K_{n,n})$.

Overcoming the Obstacle: Beyond Biclique. We overcome the above obstacle by considering dense bipartite graphs, instead of the biclique. More precisely, we show that there exists a balanced bipartite graph $G^* = (A^* \dot\cup B^*, E^*)$ on $2n$ vertices such that $|E^*| \geq n^{2-o(1)}$ and $\mathsf{cd}_p(G^*)$ is small (i.e. $\mathsf{cd}_p(G^*) \leq (\log n)^{\omega(1)}$). We give a construction of such a graph below but before we do so, let us briefly argue why this suffices to show

[5] Note that plugging in the bound on $\mathsf{cd}_2(K_{n,n})$ in the result of [DKL18] yields that assuming SETH, no subquadratic in n running time algorithm can solve CP when $d = \Omega(n)$. This is not a meaningful lower bound as just the input size of CP when $d = \Omega(n)$ is $\Omega(n^2)$.

that BCP and CP are computationally equivalent (up to $n^{o(1)}$ multiplicative overhead in the running time) for dimension $d = \Omega(\mathsf{cd}_p(G^*))$.

Let us consider the same reduction which produces $\widetilde{A}, \widetilde{B}$ as before, but instead of using a realization of the biclique, we use a realization τ of G^*. This reduction is of course incorrect: if $(\mathbf{a}^*, \mathbf{b}^*)$ is not an edge in G^*, then $\|\tau(\mathbf{a}^*) - \tau(\mathbf{b}^*)\|_p$ could be large and, thus the corresponding pair of points $(\widetilde{\mathbf{a}^*}, \widetilde{\mathbf{b}^*}) \in \widetilde{A} \times \widetilde{B}$, may not be the closest pair. Nevertheless, we are not totally hopeless: if $(\mathbf{a}^*, \mathbf{b}^*)$ is an edge, then we are in good shape and the reduction is correct.

With the above observation in mind, consider picking a random permutation π of $A \cup B$ such that $\pi(A) = A$ and $\pi(B) = B$ and then initiate the above reduction with the map $(\tau \circ \pi)$ instead of τ. Note that $\tau \circ \pi$ is simply a realization of an appropriate permutation G' of G^* (i.e., G' is isomorphic to G^*). Due to this, the probability that we are "lucky" and $(\mathbf{a}^*, \mathbf{b}^*)$ is an edge in G' is $p := |E|/n^2$; when this is the case, solving CP on the resulting instance would give the correct answer for the original BCP instance. If we repeat this $\log n/p = n^{o(1)}$ times, we would find the optimum of the original BCP instance with high probability.

To recap, even when G^* is not a biclique, we can still use it to give a reduction from BCP to CP, except that the reduction produces multiple (i.e. $\widetilde{O}(n^2/|E^*|)$) instances of CP. We remark here that the reduction can be derandomized: we can deterministically (and efficiently) pick the permutations so that the permuted graphs covers $K_{n,n}$ (see Lemma 3.11). As a minor digression, we would like to draw a parallel here with a recent work of Abboud, Rubinstein, and Williams [ARW17b]. The obstacle raised in [DKL18] is about the impossibility of certain kinds of many-one gadget reductions. We overcame it by designing a reduction from BCP to CP which not only increased the number of dimensions but also the number of points (by creating multiple instances of CP). This technique is also utilized in [ARW17b] where they showed the impossibility of Deterministic Distributed PCPs (Theorem I.2 in [ARW17b]) but then overcame that obstacle by using an advice (which is then enumerated over resulting in multiple instances) to build Non-deterministic Distributed PCPs.

Constructing a dense bipartite graph with low contact dimension. We now proceed to construct the desired graph $G^* = (A^* \cup B^*, E^*)$. Note that any construction of a dense bipartite graph with contact dimension $n^{o(1)}$ is non-trivial. This is because it is known that a random graph has contact dimension $\Omega(n)$ in the Euclidean metric with high probability [RRS89, BL05], and therefore our graph construction must be significantly better than a random graph.

Our realization τ^* of G^* will map into a subset of $\{0,1\}^{(\log n)^{\omega(1)}}$. As a result, we can fix $p = 0$, since a realization of a graph with entries in $\{0,1\}$ in the Hamming-metric also realizes the same graph in every ℓ_p-metric for any $p \neq \infty$.

Fix $g = \omega(1)$. We associate $[n]$ with $\mathbb{F}_q^h$ where $q = \Theta\left((\log n)^g\right)$ is a prime and $h = \Theta\left(\frac{\log n}{g \cdot \log \log n}\right)$. Let $\mathcal{P}$ be the set of all univariate polynomials (in x) over $\mathbb{F}_q$ of degree at most $h-1$. We have that $|\mathcal{P}| = q^h = n$ and associate $\mathcal{P}$ with A^*. Let $\mathcal{Q}$ be the set of all univariate monic polynomials (in x) over $\mathbb{F}_q$ of degree h, i.e.,

$$\mathcal{Q} = \{x^h + p(x) \mid p(x) \in \mathcal{P}\}.$$

We associate the polynomials in $\mathcal{Q}$ with the vertices in B^* (note that $|\mathcal{Q}| = n$). In fact, we view the vertices in A^* and B^* as being uniquely labeled by polynomials in $\mathcal{P}$ and $\mathcal{Q}$ respectively. For notational clarity, we write p_a (resp. p_b) to denote the polynomial in $\mathcal{P}$ (resp. $\mathcal{Q}$) that is associated to $a \in A^*$ (resp. $b \in B^*$).

For every $a \in A^*$ and $b \in B^*$, we include (a, b) as an edge in E^* if and only if the polynomial $p_b - p_a$ (which is of degree h) has h distinct roots. This completes the construction of G^*. We have to now show the following two claims about G^*: (i) $|E^*| = n^{2-O(1/g)} = n^{2-o(1)}$ and (ii) there is $\tau : A^*\dot{\cup}B^* \to \{0,1\}^{(\log n)^{O(g)}} = \{0,1\}^{(\log n)^{\omega(1)}}$ that realizes G^*.

To show (i), let $\mathcal{R}$ be the set of all monic polynomials of degree h with h distinct roots. We have that $|\mathcal{R}| = \binom{q}{h}$. Fix a vertex $a \in A^*$. Its degree in G^* is exactly $|\mathcal{R}| = \binom{q}{h}$. This is because, for every polynomial $r \in \mathcal{R}$, $r + a$ belongs to $\mathcal{Q}$, and therefore $(a, r + a) \in E^*$. This implies the following bound on $|E^*|$:

$$|E^*| = q^h \cdot \binom{q}{h} \geq q^h \cdot \frac{q^h}{h^h} > \frac{n^2}{(\log n)^{\Theta((\log n)/(g \cdot \log\log n))}} = n^{2-O(1/g)}.$$

Next, to show (ii), we construct a realization $\tau^* : A^*\dot{\cup}B^* \to \mathbb{F}_q^q$ of G^*. We note that, it is simple to translate the entries to $\{0,1\}$ instead of $\mathbb{F}_q$, by replacing $i \in \mathbb{F}_q$ with the i-th standard basis $\mathbf{e}_i \in \{0,1\}^q$. This would result in a realization $\tau^* : A^*\dot{\cup}B^* \to \{0,1\}^{q^2}$ of G^*; notice that the dimension of τ^* is $q^2 = \Theta((\log n)^{2g})$ as claimed.

We define τ^* as follows.

- For every $a \in A^*$, $\tau^*(a)$ is simply the vector of evaluation of p_a on every element in $\mathbb{F}_q$. More precisely, for every $j \in [q]$, the j-th coordinate of $\tau^*(a)$ is $p_a(j-1)$.
- Similarly, for every $b \in B^*$ and $j \in [q]$, the j-th coordinate of $\tau^*(b)$ is $p_b(j-1)$.

We now show that τ^* is indeed a realization of G^*; specifically, we show that τ^* satisfies (1) and (2) with $\beta = q - h$.

Consider any edge $(a, b) \in E^*$. Notice that $\|\tau^*(a) - \tau^*(b)\|_0$ is the number of $x \in \mathbb{F}_q$ such that $p_b(x) - p_a(x) \neq 0$. By definition of E^*, $p_b - p_a$ is a polynomial with h distinct roots over $\mathbb{F}_q$. Thus, $\|\tau^*(a) - \tau^*(b)\|_0 = q - h = \beta$ as desired.

Next, consider a non-edge $(a, b) \in (A^* \times B^*) \setminus E^*$. Then, we know that $p_b - p_a$ has at most $h - 1$ distinct roots over $\mathbb{F}_q$. Therefore, the polynomial $p_b - p_a$ is non-zero on at least $q - h + 1$ coordinates. This implies that $\|\tau^*(a) - \tau^*(b)\|_0 \geq q - h + 1 > \beta$.

Finally, for any distinct $a, a' \in A^*$, we have $\|\tau^*(a) - \tau^*(a')\|_0 \geq q - h + 1$ because $p_a - p_{a'}$ is a non-zero polynomial of degree at most $h - 1$ and thus can be zero over $\mathbb{F}_q$ in at most $h - 1$ locations. Similarly, $\|\tau^*(b) - \tau^*(b')\|_0 \geq q - h + 1$ for any distinct $b, b' \in B^*$.

This completes the proof sketch for both the claims about G^* and yields Theorem 1.4 for $d = (\log n)^{\omega(1)}$. Finally we remark that in the actual proof of Theorem 1.4, we will set the parameters in the above construction more carefully and achieve the bound $\mathrm{cd}_p(G^*) = (\log n)^{O_\varepsilon(1)}$.

2.2 Abstracting the Construction via Error-Correcting Codes

Before we move on to discuss the proofs of Theorems 1.6 and 1.5, let us give an abstraction of the construction in the previous subsection. This will allow us to easily generalize the construction for the aforemention theorems, and also to explain where our motivation behind the construction comes from in the first place.

Dense Bipartite Graph with Low Contact Dimension from Codes. In order to construct a balanced bipartite graph G^* on $2n$ vertices with $n^{2-o(1)}$ edges such that $\mathrm{cd}_p(G^*) \leq d^*$, it suffices to have a code C^* with the following properties (for code-related definitions, see Section 3.2):

- $C^* \subseteq \mathbb{F}_q^\ell$ of cardinality n is a linear code with block length ℓ over alphabet $\mathbb{F}_q$, and minimum distance Δ.
- There exists a *center* $s^* \in \mathbb{F}_q^\ell$ and $r^* < \Delta$ such that $|C^*|^{1-o(1)}$ codewords are at Hamming distance exactly r^* from s^* and no codeword is at distance less than r^* from s^*.
- $q \cdot \ell = d^*$.

We also require that C^* and s^* can be constructed in $\mathrm{poly}(n)$ time but we shall ignore this requirement for the ease of exposition.

We describe below how to construct G^* from C^*, but first note that the construction of G^* we saw in the previous subsubsection was just showing that Reed Solomon codes [RS60] of block length $q = \Theta((\log n)^g)$ and message length $h = \Theta\left(\frac{\log n}{g \cdot \log \log n}\right)$ over alphabet $\mathbb{F}_q$ with minimum distance $q - h + 1$ has the above properties. The center s^* in that construction was the evaluation of the polynomial x^h over $\mathbb{F}_q$, and r^* was $q - h$.

In general, to construct G^* from C^*, we first define a subset $S^* \subseteq \mathbb{F}_q^\ell$ of cardinality n as follows:

$$S^* = \{\mathbf{s}^* + \mathbf{c} \mid \mathbf{c} \in C^*\}.$$

We associate the vertices in A^* with the codewords of C^* and vertices in B^* with the strings in S^*. For any $(\mathbf{a}, \mathbf{b}) \in A^* \times B^*$, let $(\mathbf{a}, \mathbf{b}) \in E^*$ if and only if $\|\mathbf{b} - \mathbf{a}\|_0 = r^*$. This completes the construction of G^*. We have to now show the following claims about G^*: (i) $|E^*| = n^{2-o(1)}$ and (ii) there is $\tau : A^* \dot\cup B^* \to \{0,1\}^{q \cdot \ell}$ that realizes G^*.

Item (i) follows rather easily from the properties of C^* and s^*. Let T^* be the subset of C^* of all codewords which are at distance exactly equal to r^* from s^*. From the definition of s^*, we have $|T^*| = |C^*|^{1-o(1)}$. Fix $\mathbf{a} \in A^*$. Its degree in G^* is $|T^*| = |C^*|^{1-o(1)}$. This is because for every codeword $\mathbf{t} \in T^*$ we have that $\mathbf{t} - \mathbf{a}$ is a codeword in C^* (from the linearity of C^*) and thus $\mathbf{s}^* - \mathbf{t} + \mathbf{a}$ is in S^*, and therefore $(\mathbf{a}, \mathbf{s}^* - \mathbf{t} + \mathbf{a}) \in E^*$.

For item (ii), consider the identity mapping $\tau^* : A^* \dot\cup B^* \to \mathbb{F}_q^\ell$ that maps each string to itself. It is simple to check that τ^* realizes G^* in the Hamming metric (with $\beta = r^*$).

Recall from the previous subsection that given $\tau^* : A^* \dot\cup B^* \to \mathbb{F}_q^\ell$ that realizes G^* in the Hamming metric, it is easy to construct $\tau : A^* \dot\cup B^* \to \{0,1\}^{q \cdot \ell}$ that realizes G^* in the Hamming metric with a q multiplicative factor blow-up in the dimension. This completes the proof of both the claims about G^* and gives a general way to prove Theorem 1.4 given the construction of C^* and s^*.

Finding Center from Another Code. One thing that might not be clear so far is: where does the center s^* come from? Here we provide a systematic way to produce such an s^*, by looking at another code that contains C^*. More precisely, let $C^* \subseteq \widetilde{C}^* \subseteq \mathbb{F}_q^\ell$ be two linear codes with the same block length and alphabet. Suppose that the distance of C^* is Δ, the distance of $\widetilde{C}^*$ is r^* and that $r^* < \Delta$. It is easy to see that, by taking s^* to be any element of $\widetilde{C}^* \setminus C^*$, it holds that every codeword in C^* is at distance at least r^* from s^*, simply because both s^* and the codewords of C^* are codewords of $\widetilde{C}^*$.

Hence, we are only left to argue that there are many codewords of C^* that is of distance exactly r^* from s^*. While this is not true in general, we can show by an averaging argument that this is true (for some $s^* \in \widetilde{C}^*$) if a large fraction (e.g. $|C^*|^{-o(1)}$ fraction) of codewords of $\widetilde{C}^*$ has Hamming weight exactly r^* (see Lemma 5.1).

Indeed, viewing in this light, our previous choice of center for Reed-Solomon code (i.e. evaluation of x^h) is not coincidental: we simply take $\widetilde{C}^*$ to be another Reed-Solomon code with message length $h + 1$ (whereas the base code C^* is of message length h).

Comparison to Locally Dense Codes. We end this subsection by remarking that the codes that we seek are very similar to locally dense codes [DMS03, CW12, Mic14], which is indeed our inspiration. A *locally dense code* is a linear code of block length ℓ and large minimum distance Δ, admitting a ball centered at s of radius[6] $r < \Delta$ and containing a large (i.e. $\exp(\text{poly}(\ell))$) number of codewords[7]. Such codes are non-trivial to construct and in particular all known constructions of locally dense codes are using codes that beat the Gilbert-Varshamov (GV) bound [Gil52, Var57]; in other words we need to do better than random codes to construct them. This is because (as noted in [DMS03]), for a random code $C \subseteq \mathbb{F}_q^\ell$ (or any code that does not beat the GV bound), a random point in $\mathbb{F}_q^\ell$ acting as the center contains in expectation less than one codeword in a ball of radius Δ. Of course, this is simply an intuition and not a formal proof that a locally dense code needs to beat the GV bound, since there may be more sophisticated ways to pick a center.

Although the codes we require are similar to locally dense codes, there are differences between the two. Below we list four such differences: the first two makes it *harder* for us to construct our codes whereas the latter two makes it *easier* for us.

- We seek a center s^* so that no codewords in C^* lies at distance less than r^*, as opposed to locally dense codes which allows codewords to be close to s^*. This is indeed where our idea of using another code $\widetilde{C}^* \supseteq C^*$ comes in, as picking s^* from $\widetilde{C}^* \setminus C^*$ ensures us that no codeword of C^* is too close to s^*.

- Another difference is that we need the number of codewords at distance r^* from s^* to be very large, i.e., $|C^*|^{1-o(1)}$, whereas locally dense codes allow for much smaller number of codewords. Indeed, the deterministic constructions from [CW12, Mic14] only yield the bound of $2^{O(\sqrt{\log |C^*|})}$. Hence, these do not directly work for us.

- Locally dense codes requires r to be at most $(1 - \varepsilon)\Delta$ for some constant $\varepsilon > 0$, whereas we are fine with any $r^* < \Delta$. In fact, our Reed-Solomon code based construction above only yields $r^* = \Delta - 1$ which would not suffice for locally dense codes. Nevertheless, as we will see later for inapproximability of CP, we will also

[6] Clearly, for the ball to contain more than a single codeword, it must be $r \geq \Delta/2$. Here we are interested in balls with radius not much bigger than that, say $r < \gamma \cdot \Delta$ for some constant $1/2 < \gamma < 1$.

[7] Strictly speaking, a locally dense code also requires an auxiliary matrix T used to index these codewords. However, in previous works, finding T is typically not hard given the center s. Hence, we ignore T in our discussion here for the ease of exposition.

need the ratio r^*/Δ to be a constant bounded away from 1 as well and, since we need a code with these extraordinary properties, they are very hard to find. Indeed, in this case we only manage to prove a weaker lower bound on gap-CP.

- Finally, we remark that locally dense codes are required to be efficiently constructed in $\text{poly}(\log |C^*|)$ time, which is part of why it is hard to find. Specifically, while [DMS03] shows that an averaging argument works for a random center, derandomizing this is a big issue and a few subsequent works are dedicated solely to this issue [CW12, Mic14]. (We also note that it remains open whether a center can be deterministically found for a variant of locally dense codes used in hardness of parameterized version of the minimum distance problem. See [BGKM18] for more details.) On the other hand, brute force search (over all codewords in $\widetilde{C}^*$) suffices to find a center for us, as we are allowed construction time of $\text{poly}(|C^*|)$.

2.3 Inapproximability of Closest Pair and Maximum Inner Product

In this subsection, we sketch our inapproximability results for MIP and CP. Both these results use the same reduction that we had from BCP to CP, except that we now need stronger properties from the gadget, i.e., the previously used notions of contact dimension does not suffice anymore. Below we sketch the required strengthening of the gadget properties and explain how to achieve them.

2.3.1 Approximate Maximum Inner Product

Observe that the gadget we construct for CP in Subsection 2.2 can also be written in terms of inner product as follows: there exists a dense balanced bipartite graph $G^* = (A^* \dot\cup B^*, E^*)$, a mapping $\tau : A^* \dot\cup B^* \to \{0,1\}^{q \cdot \ell}$ such that the following holds.

(i) For all edges $(a, b) \in E^*$, $\langle \tau(a), \tau(b) \rangle = \ell - r^*$.

(ii) For all edges $(a, b) \in (A^* \times B^*) \setminus E^*$, $\langle \tau(a), \tau(b) \rangle < \ell - r^*$.

(iii) For all distinct a, b both from A^* or both from B^*, $\langle \tau(a), \tau(b) \rangle \leq \ell - \Delta$.

Notice that we wrote the conditions above in a slightly different way than in previous subsections; previously in the contact dimension notation, (ii) and (iii) would be simply written together as: for all non-edge (a, b), $\langle \tau(a), \tau(b) \rangle < \ell - r^*$. This change is intentional, since, to get gap in our reductions, we only need a gap between the bounds in (i) and (iii) (but not in (ii)). In particular, to get hardness of approximating MIP, we require $\frac{\ell - r^*}{\ell - \Delta}$ to be at least $(1 + \varepsilon)$ for some $\varepsilon > 0$.

From our Reed-Solomon construction above, $\ell - \Delta$ and $\ell - r^*$ are exactly the message length of C^* minus one and the message length of $\widetilde{C}^*$ minus one respectively. Previously, we selected these two to be h and $h + 1$. Now to obtain the desired gap, we simply take the larger code $\widetilde{C}^*$ to be a Reed-Solomon code with larger (i.e. $(1 + \varepsilon)h$) message length[8].

[8] This approach can in fact give not just $(1 + \varepsilon)$ but arbitrarily large constant gap between the two cases. In the actual reduction, we take this gap to be 3 (Theorem 6.2), which makes some computations simpler.

Finally, we note that even with the above gadget, the reduction only gives a small (i.e. $1 + o(1)$) factor hardness of approximating MIP (Theorem 6.2). To boost the gap to near polynomial, we simply tensor the vectors with themselves (see Section 6).

2.3.2 Approximate Closest Pair

Once again, recall that we have the following gadget from Subsection 2.2: there exists a dense balanced bipartite graph $G^* = (A^* \dot\cup B^*, E^*)$, a mapping $\tau : A^* \dot\cup B^* \to \{0,1\}^{q \cdot \ell}$ such that the following holds.

(i) For all edges $(a,b) \in E^*$, $\|\tau(a) - \tau(b)\|_0 = r^*$.

(ii) For all edges $(a,b) \in (A^* \times B^*) \setminus E^*$, $\|\tau(a) - \tau(b)\|_0 > r^*$.

(iii) For all distinct a, b both from A^* or both from B^*, $\|\tau(a) - \tau(b)\|_0 \geq \Delta$.

Once again, we need an $(1 + \varepsilon)$ gap between the bounds in (iii) and (i), i.e., $\frac{\Delta}{r^*}$. Unfortunately, we cannot construct such codes using any of the Reed-Solomon code families. We turn to another type of codes that beat the Gilbert-Varshamov bound: Algebraic- Geometric (AG) codes. Similar to the Reed-Solomon code based construction, we take C^* as an AG code and $\widetilde{C}^*$ to be a "higher degree" AG code; getting the desired gap simply means that the distance of C^* must be at least $(1 + \varepsilon)$ times the distance of $\widetilde{C}^*$.

Recall from Subsection 2.2 also that, to bound the density of G^*, we need a lower bound on the number of minimum weight codewords of $\widetilde{C}^*$. Such bounds for AG codes are non-trivial and we turn to the bounds from [ABV01, Vlă18]. Unfortunately, this only gives G^* with density $|C^*|^{-1/2-o(1)}$, instead of $|C^*|^{-o(1)}$ as before. This is indeed the reason that our running time lower bound for approximate CP is only $n^{1.5-\varepsilon}$.

We are not aware of any result on the (asymptotic) tightness of the bounds from [ABV01, Vlă18] that we use. However, improving upon such bounds would have other consequences, such as a better bound on the kissing numbers of lattices constructed in [Vlă18]. As a result, it seems likely that more understanding of AG codes (and perhaps even new constructions) are needed in order to improve these bounds.

3 Preliminaries

In this section we define the geometric problems of interest to this paper, give an alternate proof for the conditional lower bound on bichromatic closest pair, and recall the definition of the contact dimension of a graph.

3.1 Notations, Problems and Fine-Grained Hypotheses

Distance Measures. For any two vectors $a, b \in \mathbb{R}^d$, the distance between them in the ℓ_p-metric is denoted by $||a - b||_p = \left(\sum_{i=1}^{d} |a_i - b_i|^p\right)^{1/p}$. Their distance in the ℓ_∞-metric is denoted by $||a - b||_\infty = \max\limits_{i \in [d]} \{|a_i - b_i|\}$, and in the ℓ_0-metric is denoted by $||a - b||_0 = |\{i \in [d] : a_i \neq b_i\}|$, i.e., the number of coordinates on which a and b differ. More generally, for any two vectors $a, b \in \mathbb{R}^d$ in the Δ-metric, we denote by $\Delta(a, b)$ its distance in that metric space. The ℓ_p-metrics that are well studied in literature are the *Hamming metric* (ℓ_0-metric), the *rectilinear metric* (ℓ_1-metric), the *Euclidean metric* (ℓ_2-metric), and the *Chebyshev metric* (ℓ_∞-metric). We denote the inner product (associated with the Euclidean space) of a and b by $\langle a, b\rangle = \sum_{i\in[d]} a_i \cdot b_i$. Finally, for every positive integer d we define the edit metric over Σ to be the space Σ^d endowed with distance function $\mathsf{ed}(a, b)$, which is defined as the minimum number of character substitutions/insertions/deletions to transform a into b.

Problems. Here we give formal definitions of Orthogonal Vectors (OV), Closest Pair (CP) and Bichromatic Closest Pair (BCP) problems, and also Maximum Inner Product (MIP) and Bichromatic Maximum Inner Product (BMIP) problems.

Definition 3.1 (Orthogonal Vectors Problem, OV)**.** *In* OV*, we are given two collections of n points $A, B \subseteq \{0, 1\}^d$, and the goal is to find a pair of points $a \in A$, $b \in B$ such that $\langle a, b\rangle = 0$.*

Definition 3.2 (Closest Pair Problem, CP)**.** *In* CP *in the Δ-metric, we are given a collection of n points $P \subseteq \mathbb{R}^d$ and a positive real α, and the goal is to find a pair of distinct points $a, b \in P$ such that $\Delta(a, b) \le \alpha$.*

Definition 3.3 (Bichromatic Closest Pair Problem, BCP)**.** *In* BCP *in the Δ-metric, we are given two collections of n points $A, B \subseteq \mathbb{R}^d$ and a positive real α, and the goal is to find a pair of points $a \in A$, $b \in B$ such that $\Delta(a, b) \le \alpha$.*

We will also use gap versions of these problems. For any $\delta \ge 0$, we define $(1 + \delta)$-CP (resp. $(1 + \delta)$- BCP) in the Δ-metric to be the problem of distinguishing between the case whether there exist distinct $a, b \in P$ (resp. $a \in A$ and $b \in B$) such that $\Delta(a, b) \le \alpha$ and the case where for all distinct $a, b \in P$ (resp. $a \in A$ and $b \in B$) we have $\Delta(a, b) > (1 + \delta) \cdot \alpha$.

Definition 3.4 (Maximum Inner Product Problem, MIP)**.** *In* MIP*, we are given a collection of n points $P \subseteq \mathbb{R}^d$ and a real α, and the goal is to find a pair of distinct points $a, b \in P$ such that $\langle a, b\rangle \ge \alpha$.*

Definition 3.5 (Bichromatic Maximum Inner Product Problem, BMIP)**.** *In* BMIP*, we are given two collections of n points $A, B \subseteq \mathbb{R}^d$ and a real α, and the goal is to find a pair of points $a \in A$, $b \in B$ such that $\langle a, b\rangle \ge \alpha$.*

Again we define the gap versions of these problems as follows. For any $\gamma \ge 1$, we define γ-MIP (resp. γ-BMIP) to be the problem of distinguishing between the case whether there exist distinct $a, b \in P$ (resp. $a \in A$ and $b \in B$) such that $\langle a, b\rangle \ge \alpha$ and the case where for all distinct $a, b \in P$ (resp. $a \in A$ and $b \in B$) we have $\langle a, b\rangle < {}^{\alpha}/_{\gamma}$.

Hypotheses. Finally, we give formal definitions of the relevant fine-grained hypotheses (see [Wil18b] for a survey on the state-of-the-art conditional lower bounds that are known under these hypotheses).

Definition 3.6 (Strong Exponential Time Hypothesis, SETH [IP01, IPZ01, CIP06])**.** *For every $\varepsilon > 0$, there exists $k = k(\varepsilon) \in \mathbb{N}$ such that no algorithm can solve k-SAT (i.e., satisfiability on a CNF of width k) in $O(2^{(1-\varepsilon)m})$ time where m is the number of variables. Moreover, this holds even when the number of clauses is at most $c(\varepsilon)m$ where $c(\varepsilon)$ denotes a constant that depends only on ε.*

Definition 3.7 (Orthogonal Vector Hypothesis, OVH)**.** *For every $\varepsilon > 0$, no algorithm can solve* OV *in $O(n^{2-\varepsilon})$ time. Moreover, this holds even when the dimension d is at most $c(\varepsilon) \log n$ where $c(\varepsilon)$ denotes a constant that depends only on ε.*

It is known that SETH implies OVH [Wil05], and therefore in the rest of the paper, we base all our conditional lower bounds on OVH.

3.2 Error-Correcting Codes

We recall here a few coding theoretic notations since all of our gadgets are based on error-correcting codes. As is standard in error-correcting codes, we will use $\Delta(\mathbf{a}, \mathbf{b})$ to denote $\|\mathbf{a} - \mathbf{b}\|_0$, the Hamming distance of $\mathbf{a}$ and $\mathbf{b}$, for any $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^N$ and we further define $\Delta(\mathbf{a}, S) := \min_{\mathbf{b} \in S} \Delta(\mathbf{a}, \mathbf{b})$ for any $\mathbf{a} \in \mathbb{F}_q^N$ and $S \subseteq \mathbb{F}_q^N$. The weight of $\mathbf{a} \in \mathbb{F}_q^N$, denoted by $\Delta(\mathbf{a})$, is simply $\|\mathbf{a}\|_0 := |i \in [N] : a_i \neq 0|$. For $\mathbf{a} \in \mathbb{F}_q^N$ and $d \in \mathbb{N}$, we use $\mathcal{B}(\mathbf{a}, d)$ to denote the (closed) Hamming ball of radius d centered at $\mathbf{a}$, i.e., $\mathcal{B}(\mathbf{a}, d) := \{\mathbf{b} \in \mathbb{F}_q^N \mid \Delta(\mathbf{a}, \mathbf{b}) \leq d\}$.

An error correcting code of block length N over alphabet $\mathbb{F}_q$ is simply a collection of codewords $\mathcal{C} \subseteq \mathbb{F}_q^N$. The distance of the code $\mathcal{C}$, denoted by $\Delta(\mathcal{C})$, is defined as $\min_{\mathbf{a} \neq \mathbf{b} \in \mathcal{C}} \Delta(\mathbf{a}, \mathbf{b})$. A code is said to be linear if $\mathcal{C}$ is a subspace of $\mathbb{F}_q^N$. For a linear code $\mathcal{C}$, its message length is defined to be the dimension of $\mathcal{C}$, or equivalently $\log_q |\mathcal{C}|$. We often use the notion $[N, K, D]_q$ to denote a linear code of block length N, message length K, and distance D. The rate and relative distance of a linear $[N, K, D]_q$ code $\mathcal{C}$ are defined as K/N and D/N respectively. Note also that, for a linear code $\mathcal{C}$, $\Delta(\mathcal{C})$ is equal to the minimum weight of a non-zero codeword of $\mathcal{C}$. Finally, for any code $\mathcal{C}$, we use $A_w(\mathcal{C}) := |\{\mathbf{c} \in \mathcal{C} \mid \Delta(\mathbf{c}) = w\}|$ to denote the number of codewords of weight w.

Let us also recall the Singleton bound and the definition of *maximum distance separable* (MDS) codes.

Theorem 3.8 (Singleton bound [Sin64])**.** *For any linear $[N, K, D]_q$ code, $K + D \leq N + 1$.*

Definition 3.9 (MDS Codes)**.** *A linear $[N, K, D]_q$ code is said to be a maximum distance separable (MDS) code if it matches the Singleton bound, i.e., $K + D = N + 1$.*

We note here that the above bound and notation are well-defined (or can be naturally extended) also for non-linear codes, but we will only use them in context of linear codes in this paper.

3.3 Miscellaneous Tools

Covering Biclique by Isomorphic Graphs. A useful fact we use to derandomize our reductions is that the biclique can be covered by any dense bipartite graph G with only a few graphs that are isomorphic to G. To state this more formally, let us first define a few notions.

Definition 3.10. *For any graph $G = (V_G, E_G)$ and any permutation $\pi : V_G \to V_G$, we use G_π to denote the graph (V_{G_π}, E_{G_π}) where the vertex set V_{G_π} is equal to V_G and $E_{G_\pi} = \{(\pi(a), \pi(b)) \mid (a, b) \in E_G\}$.*

For brevity, we say that a permutation $\pi : A\dot{\cup}B \to A\dot{\cup}B$ of vertices of a bipartite graph $G = (A\dot{\cup}B, E_G)$ is *side-preserving* if $\pi(A) = A$ and $\pi(B) = B$.

We can now state the result as follows. The proof, which proceeds via a simple set covering argument, is deferred to Appendix B.

Lemma 3.11. *For any bipartite graph $G(A\dot{\cup}B, E_G)$ where $|A| = |B| = n$ and $E_G \neq \emptyset$, there exist side-preserving permutations $\pi_1, \ldots, \pi_k : A \cup B \to A \cup B$ where $k \leq \frac{2n^2 \ln n}{|E_G|} + 1$ such that*

$$\bigcup_{i\in[k]} E_{G_{\pi_i}} = E_{K_{n,n}}$$

Moreover, such permutations can be found in time $O(n^6 \log n)$.

Translating Finite Fields Vectors to $\{0, 1\}$-Vectors. Another simple fact which was already mentioned in the proof overview (Section 2) is that, we can embed Hamming metric on alphabet of size q to Hamming metric on Boolean alphabet, with only q multiplicative factor blow-up in the dimension:

Proposition 3.12. *For any $q, N \in \mathbb{N}$, and alphabet Σ such that $|\Sigma| = q$, there exists a mapping $\psi : \Sigma^N \to \{0,1\}^{q\cdot N}$ such that, for all $\mathbf{v}_1, \mathbf{v}_2 \in \Sigma^N$, we have $\|\psi(\mathbf{v}_1) - \psi(\mathbf{v}_2)\|_0 = 2 \cdot \Delta(\mathbf{v}_1, \mathbf{v}_2)$ and $\langle\psi(\mathbf{v}_1), \psi(\mathbf{v}_2)\rangle = N - \Delta(\mathbf{v}_1, \mathbf{v}_2)$.*

Proof. The mapping ψ simply replaces each coordinate that is equal to $j \in \Sigma$ by the j-th standard basis in the q-dimensional space. More precisely, for $\mathbf{v} = (v_1, \ldots, v_N) \in \mathbb{F}_q$, we define

$$\psi(\mathbf{v}) = e_{v_1} \circ e_{v_2} \circ \cdots \circ e_{v_N},$$

where $\circ$ denotes concatenation of vectors and e_j denote the j-th standard basis in $\mathbb{R}^q$, i.e., the vector whose j-th coordinate is one and the remaining coordinates are zeroes.

It is simple to check that this satisfies the two requirements. □

3.4 OVH-hardness of Exact Bichromatic Closest Pair

Alman and Williams [AW15] showed the conditional hardness (under OVH) of exact BCP in every ℓ_p-metric even when the point-sets are over $\{0,1\}$ via a Turing reduction from OV. David, Karthik, and Laekhanukit [DKL18] gave an alternate proof of the same result where point-sets were over $\mathbb{R}$ via a many-one reduction from OV. For independent interest, below we give another proof, which is both a many-one reduction and the point-sets are over $\{0,1\}$.

Theorem 3.13. *Assuming* OVH, *for every $\varepsilon > 0$, no algorithm running in time $n^{2-\varepsilon}$ can solve* BCP, *even when the point-sets A, B are subsets of $\{0,1\}^d$ and $d = c_\varepsilon \log n$, for some constant $c_\varepsilon > 1$ (only depending on ε).*

Proof. Let $A, B \subseteq \{0,1\}^d$ where $|A| = |B| = n$ be the input to an OV instance. We build an instance (A', B', α) of BCP where $A', B' \subseteq \{0,1\}^{5d}$, $|A| = |B| = n$, and $\alpha = 2d$, using functions T_A and T_B guaranteed by the following claim.

Claim 3.14. *There are functions $T_A, T_B : \{0,1\} \to \{0,1\}^5$ such that for every $x, y \in \{0,1\}$ we have:*

- $x \cdot y = 0$ *implies* $\|T_A(x) - T_B(y)\|_0 = 2$.
- $x \cdot y = 1$ *implies* $\|T_A(x) - T_B(y)\|_0 = 4$.

For every $i \in [n]$, the i^{th} point of A', say a' is constructed from the i^{th} point of A, say a by simply applying T_A pointwise on each coordinate of a, i.e., $a' = (T_A(a_1), \ldots, T_A(a_d))$. Similarly we apply T_B pointwise on each coordinate of points in B. It is easy to see that there exists $(a'_i, b'_j) \in A' \times B'$ such that $\|a'_i - b'_j\|_0 = 2d$ if and only if $\langle a_i, b_j \rangle = 0$, and otherwise every pair of points in $A' \times B'$ is at Hamming distance at least $2d + 2$. □

Proof of Claim 3.14. We define for all $x, y \in \{0, 1\}$, $T_A(x) = (T_A(x)_{0,0}, T_A(x)_{0,1}, T_A(x)_{1,0}, x, 0)$ and $T_B(y) = (T_B(y)_{0,0}, T_B(y)_{0,1}, T_B(y)_{1,0}, 0, y)$, where for all $i, j \in \{0, 1\}$ such that $i \cdot j = 0$, we have $T_A(x)_{i,j} = 1$ if and only if $x = i$ and $T_B(y)_{i,j} = 1$ if and only if $y = j$. More succinctly, T_A and T_B are described below as strings and the claim follows by a straightforward calculation.

$$T_A(0) = 11000 \qquad T_A(1) = 00110$$
$$T_B(0) = 10100 \qquad T_B(1) = 01001$$

□

3.5 Contact Dimension of a Graph

The central gadget in our reduction from BCP to CP is based on the contact dimension of a graph. Below we reproduce its definition from the proof overview (i.e. Definition 2.1) for convenience.

Definition 3.15 (Contact Dimension [Pac80]). *For any graph $G = (V, E)$, a mapping $\tau : V \to \mathbb{R}^d$ is said to* realize G (in the ℓ_p-metric) *if for some $\beta > 0$, the following holds:*

(i) For all $(u, v) \in E$, $\|\tau(u) - \tau(v)\|_p = \beta$.

(ii) For all $(u, v) \notin E$, $\|\tau(u) - \tau(v)\|_p > \beta$.

The contact dimension *(in the ℓ_p-metric) of G, denoted by $\mathsf{cd}_p(G)$, is the minimum $d \in \mathbb{N}$ such that there exists $\tau : V \to \mathbb{R}^d$ realizing G in the ℓ_p-metric.*

We may also say that τ *β-realizes* G if we wishes to emphasize the value of β.

Note here that we may view points in $\tau(V)$ as centers of spheres of radius $\beta/2$. No two spheres overlap but they may touch, and G has an edge (u, v) if and only if the spheres centered at $\tau(u)$ and $\tau(v)$ touches.

For a summary of the bounds on $\mathsf{cd}(G)$ for various graphs in the Euclidean metric see [Mae85, FM86, FM88, Mae91] and for a summary of the bounds on $\mathsf{cd}(K_{n,n})$ in various metrics see [DKL18]. For this paper, the following bounds are relevant.

Theorem 3.16 (Frankl-Maehara [FM88]). *$(1.286)n - 1 < \mathsf{cd}_2(K_{n,n}) < (1.5)n$.*

Theorem 3.17 (David-Karthik-Laekhanukit [DKL18]). *$\mathsf{cd}_0(K_{n,n}) = n$.*

In particular, the above two theorems are the obstacles of the approach of [DKL18] for the ℓ_2 and Hamming metrics respectively. As discussed in the proof overview, we will overcome these barriers by constructing dense bipartite graphs with low contact dimensions in every ℓ_p metrics.

As discussed in Section 2.3.2, we need a generalization of contact dimension in order to show inapproximability for CP. This is formally defined below; it should be noted that the definition only makes sense for bipartite graphs, whereas the original contact dimension is well-defined for any graphs. Moreover, when $\lambda = 1$, the notion of gap contact dimension coincides with the (non-gap) contact dimension in bipartite graphs.

Definition 3.18 (Gap Contact Dimension). *For any bipartite graph $G = (A \dot\cup B, E)$ and $\lambda \geq 1$, a mapping $\tau : V \to \mathbb{R}^d$ is said to* λ-gap-realize G (in the ℓ_p-metric) *if for some $\beta > 0$, the following holds:*

(i) For all $(u, v) \in E$, $\|\tau(u) - \tau(v)\|_p = \beta$.

(ii) For all $(u, v) \in (A \times B) \setminus E$, $\|\tau(u) - \tau(v)\|_p > \beta$.

(iii) For all distinct u, v both from A or both from B, $\|\tau(u) - \tau(v)\|_p > \lambda \cdot \beta$.

The λ-gap contact dimension *(in the ℓ_p-metric) of G, denoted by $\lambda\text{-cd}_p(G)$, is the minimum $d \in \mathbb{N}$ such that there exists $\tau : V \to \mathbb{R}^d$ λ-gap-realizing G in the ℓ_p-metric.*

Again, we may say that τ (β, λ)-gap-realizes G to emphasize the value of β.

Finally, we define an analogous notion for inner product:

Definition 3.19 (Gap Inner Product Dimension). *For any bipartite graph $G = (A \dot\cup B, E)$ and $\lambda \geq 1$, a mapping $\tau : V \to \mathbb{R}^d$ is said to* λ-gap-IP-realize G *if for some $\beta > 0$, the following holds:*

(i) For all $(u, v) \in E$, $\langle \tau(u), \tau(v) \rangle = \beta$.

(ii) For all $(u, v) \in (A \times B) \setminus E$, $\langle \tau(u), \tau(v) \rangle < \beta$.

(iii) For all distinct u, v both from A or both from B, $\langle \tau(u), \tau(v) \rangle < \beta / \lambda$.

The λ-gap inner product dimension *of G, denoted by $\lambda\text{-ipd}(G)$, is the minimum $d \in \mathbb{N}$ such that there exists $\tau : V \to \mathbb{R}^d$ λ-gap-IP-realizing G.*

We may say that τ (β, λ)-gap-IP-realizes G to emphasize the value of β.

4 Lower Bound on Closest Pair under Orthogonal Vector Hypothesis

In this section, we prove the subquadratic hardness for CP (assuming OVH) using the efficient construction of a realization of a dense bipartite graph. The construction will be be formally stated below and the details will be given in Section 5.2.1. First, we define the notion of a *log-dense* sequence of integers:

Definition 4.1. *A sequence $(n_i)_{i \in \mathbb{N}}$ of increasing positive integers is said to be* log-dense *if there exists a constant $C \geq 1$ such that $\log n_{i+1} \leq C \cdot \log n_i$ for all $i \in \mathbb{N}$.*

As outlined in Section 2.1 , we use Reed-Solomon codes to construct a family of dense bipartite graphs with low contact dimensions. While the construction does not yield a graph for every number of vertices n, it does yield a graph for a log-dense sequence of numbers of vertices, which turns out to be sufficient for the purpose of the reduction. More formally, we will prove the following in Section 5.2.1.

Theorem 4.2. *For every $0 < \delta < 1$, there exists a log-dense sequence $(n_i)_{i \in \mathbb{N}}$ such that, for every $i \in \mathbb{N}$, there is a bipartite graph $G_i = (A_i \dot\cup B_i, E_i)$ where $|A_i| = |B_i| = n_i$ and $|E_i| \geq \Omega(n_i^{2-\delta})$, such that $\mathsf{cd}(G_i) = (\log n_i)^{O(1/\delta)}$. Moreover, for all $i \in \mathbb{N}$, a realization $\tau : A_i \dot\cup B_i \to \{0,1\}^{(\log n_i)^{O(1/\delta)}}$ of G_i can be constructed in time $n_i^{2+o(1)}$.*

Notice that we did not specify any ℓ_p-metric in the notion of contact dimension above. This is intentional, because our point sets $\tau(A_i \dot\cup B_i)$ have coordinate entries in $\{0,1\}$, for which the distances in the Hamming metric are equivalent (up to power of p) to distances in any ℓ_p-metric ($p \neq \infty$). We also adopt this notational convenience below. Specifically, we will prove the following theorem which states that CP is hard even when the points are from $\{0,1\}^d$; clearly, this also implies Theorem 1.4 due to the aforementioned equivalence to other ℓ_p-metrics.

Theorem 4.3 (Subquadratic Hardness of $\{0,1\}$-CP)**.** *Assuming* OVH*, for every $\varepsilon > 0$, there exists $s_\varepsilon > 0$ such that no algorithm running in $O(n^{2-\varepsilon})$ time can solve* CP *in the Hamming metric even when $d = (\log n)^{s_\varepsilon}$ and all points have $\{0,1\}$ entries.*

Proof. For any $\varepsilon > 0$, let $C_{\exp}$ be the constant such that the dimension guarantee for τ in Theorem 4.2 is at most $(\log n_i)^{C_{\exp}/\varepsilon}$ for $\delta = \varepsilon/2$. We define s_ε as $2 \cdot C_{\exp}/\varepsilon + 2$.

Assume that there exists $\varepsilon > 0$ and an algorithm $\mathcal{A}$ that can solve CP in time $n^{2-\varepsilon}$ in the Hamming metric for any input of n points in $\{0,1\}^{(\log n)^{s_\varepsilon}}$. We will construct an algorithm $\mathcal{A}'$ that solves any instance of BCP in time $n^{2-\varepsilon'}$ for some constant $\varepsilon' > 0$ (to be specified below), on n points in dimension $d := c_{\varepsilon'} \cdot \log n$ with coordinate entries in $\{0,1\}$. Together with Theorem 3.13, this implies that OVH is false, arriving at a contradiction.

Let C_ε denote the log-density constant (i.e. $\sup_i \frac{\log n_{i+1}}{\log n_i}$) of the sequence from Theorem 4.2 for $\delta = \varepsilon/2$, and let ε' be $0.01 \cdot \varepsilon / C_\varepsilon$. The algorithm $\mathcal{A}'$ on input (A, B, α) where $A, B \subseteq \{0,1\}^d$, with $|A| = |B| = n$, and $\alpha \in [d]$, works as follows:

1. Let n' be the largest number in the sequence from Theorem 4.2 with $\delta = \varepsilon/2$ s.t. $n' \leq n^{0.1}$.
2. Let $G' = (A' \dot\cup B', E')$ be the graph from Theorem 4.2 with $|A'| = |B'| = n'$, $|E'| \geq \Omega((n')^{2-\delta})$, and $\tau : A' \dot\cup B' \to \{0,1\}^{(\log n')^{C\exp/\varepsilon}}$ be a β-realization of G' where $\beta \in \mathbb{N}$.
3. We use the algorithm from Lemma 3.11 to find $\pi_1, \ldots, \pi_k$ where $k = O((n')^\delta \log n')$ such that the union of $E_{G'_{\pi_1}}, \ldots, E_{G'_{\pi_k}}$ is $E_{K_{n',n'}}$.
4. We assume w.l.o.g.[9] that n is divisible by n'. Partition A and B into $A_1, \ldots, A_{n/n'}$ and $B_1, \ldots, B_{n/n'}$ each of size n'. For each $i, j \in [n/n'], t \in [k]$, do the following:
    (a) Let τ_t be an appropriate permutation of τ that β-realizes G'_{π_t}. Label the vertices of G'_{π_t} with the points in $A_i \dot\cup B_j$.
    (b) Let $\alpha' = \alpha + (d+1) \cdot \beta$, and define A_i^t, B_j^t as
$$A_i^t = \{\mathbf{a} \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{a})) \mid \mathbf{a} \in A_i\}, B_j^t = \{\mathbf{b} \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{b})) \mid \mathbf{b} \in B_j\}$$

[9] This is without loss of generality, since if n is not divisible by n', we can use brute force for the remainder points. This requires only $O(n \cdot n' \cdot) = O(n^{1.1} \log n)$ which does not affect the overall asymptotic running time of the algorithm.

where $\mathbf{1}_{d+1} \otimes \mathbf{v}$ simply denotes $\mathbf{v} \circ \mathbf{v} \circ \cdots \circ \mathbf{v}$, i.e., the concatenation of $d+1$ copies of $\mathbf{v}$.

(c) Run $\mathcal{A}$ on $(A_i^t \dot\cup B_j^t, \alpha')$. If $\mathcal{A}$ outputs YES, then output YES and terminate.

5. If none of the executions of $\mathcal{A}$ returns YES, then output NO.

Observe that the bottleneck in the running time of the algorithm is in the executions of $\mathcal{A}$. The number of executions is $(n/n')^2 \cdot k$ and each execution takes $O((n')^{2-\varepsilon})$ time. Hence, in total the running time of the algorithm $\mathcal{A}'$ is $O((n/n')^2 \cdot k \cdot (n')^{2-\varepsilon}) \leq O(n^2 \log n \cdot (n')^{-\varepsilon/2})$. Now, from the log-density of the sequence from Theorem 4.2, we have $n' \geq n^{0.1/C_\varepsilon} = n^{10\varepsilon'/\varepsilon}$. As a result, the running time of $\mathcal{A}$ is at most $O(n^{2-5\varepsilon'} \log n) \leq O(n^{2-\varepsilon'})$ as desired.

To see the correctness of the algorithm, first observe that the dimensions of vectors in A_i^t, B_j^t are at most $d + (d+1) \cdot (\log n')^{C_{\exp}/\varepsilon}$ which is at most $(\log n)^{s_\varepsilon}$ for any sufficiently large n; that is, the calls to $\mathcal{A}$ are valid. Next, observe that, if (A, B, α) is a YES instance of BCP, there must be $i, j \in [n/n']$ and $\mathbf{a}^* \in A_i, \mathbf{b}^* \in B_j$ such that $\|\mathbf{a}^* - \mathbf{b}^*\|_0$ is at most α. Since $G'_{\pi_1}, \ldots, G'_{\pi_k}$ covers $K_{n',n'}$, there must be $t \in [k]$ such that $\|\tau_t(\mathbf{a}^*) - \tau_t(\mathbf{b}^*)\|_0 = \beta$. As a result, $\|(\mathbf{a}^* \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{a}^*))) - (\mathbf{b}^* \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{b}^*)))\|_0 \leq \alpha + (d+1) \cdot \beta = \alpha'$. Thus, $(A_i^t \cup B_j^t, \alpha')$ is a YES instance for CP and $\mathcal{A}'$ outputs YES as desired.

Finally, assume that (A, B, α) is a NO instance of BCP. Consider any $i, j \in [n/n']$ and $t \in [k]$. To argue that $(A_i^t \cup B_j^t, \alpha')$ is a NO instance for CP, we have to show that any two points in $A_i^t \cup B_j^t$ have distance more than α'. To see this, let us consider two cases.

1. Both points are either from A_i^t or from B_j^t. Assume w.l.o.g. that the two points are from A_i^t; let them be $\mathbf{a} \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{a}))$ and $\mathbf{a}' \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{a}'))$. Recall that, from the definition of β-realization, $\|\tau_t(\mathbf{a}) - \tau_t(\mathbf{a}')\|_0 > \beta$. Since $\|\tau_t(\mathbf{a}) - \tau_t(\mathbf{a}')\|_0$ is an integer, we must have $\|\tau_t(\mathbf{a}) - \tau_t(\mathbf{a}')\|_0 \geq \beta + 1$. As a result, the Hamming distance between the two points is at least $(d+1) \cdot (\beta + 1) > d + (d+1) \cdot \beta = \alpha'$.

2. One of the point is from A_i^t and the other from B_j^t. Let them be $\mathbf{a} \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{a}))$ and $\mathbf{b} \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{b}))$. Since (A, B, α) is a NO instance of BCP, $\|\mathbf{a} - \mathbf{b}\|_0 > \alpha$. Furthermore, from definition of β-realization, we must have $\|\tau_t(\mathbf{a}) - \tau_t(\mathbf{b})\|_0 \geq \beta$. Combining the two implies that the Hamming distance between $\mathbf{a} \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{a}))$ and $\mathbf{b} \circ (\mathbf{1}_{d+1} \otimes \tau_t(\mathbf{b}))$ is more than α'.

Hence, $(A_i^t \dot\cup B_j^t, \alpha')$ must be a NO instance for CP for every $t \in [k]$ and $i, j \in [n/n']$. Thus, $\mathcal{A}'$ outputs NO as desired. □

5 Gadget Constructions

In this section, we construct all the gadgets that are used in our reductions, including the basic gadget (Theorem 4.2) and more advanced gadgets used for MIP and approximate version of CP.

5.1 Finding a Center of a Code via Another Code

At the heart of all our gadgets is the task of finding a code $\mathcal{C}_1$ and a center $\mathbf{s}$ such that there are $|\mathcal{C}_1|^{1-o(1)}$ many codewords at Hamming distance exactly equal to r (for some $r > 0$) from $\mathbf{s}$ but there is no codeword in $\mathcal{C}_1$ at distance less than r from $\mathbf{s}$. The below lemma is useful in finding such an $\mathbf{s}$.

Lemma 5.1. *Let $\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \mathbb{F}_q^N$ be two linear codes with the same block length N and alphabet $\mathbb{F}_q$ such that $\Delta(\mathcal{C}_2) < \Delta(\mathcal{C}_1)$. Then, there exists a center $\mathbf{s} \in \mathbb{F}_q^N$ such that (1) $\Delta(\mathbf{s}, \mathcal{C}_1) \geq \Delta(\mathcal{C}_2)$ and (2) $|\mathcal{B}(\mathbf{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1|/|\mathcal{C}_1| \geq A_{\Delta(\mathcal{C}_2)}(\mathcal{C}_2)/|\mathcal{C}_2|$. Moreover, given $\mathcal{C}_1, \mathcal{C}_2$, such an $\mathbf{s}$ can be found in $O(|\mathcal{C}_1| \cdot |\mathcal{C}_2| \cdot qN)$ time.*

Proof. We show that there exists $\mathbf{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1$ such that (2) holds. Note that (1) immediately holds, because $\mathbf{s} - \mathbf{c}$ must be a non-zero codeword of $\mathcal{C}_2$ which implies that $\Delta(\mathbf{s}, \mathbf{c}) \geq \Delta(\mathcal{C}_2)$.

To show that there exists $\mathbf{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1$ such that $|\mathcal{B}(\mathbf{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1| \geq |\mathcal{C}_1| \cdot A_{\Delta(\mathcal{C}_2)}/|\mathcal{C}_2|$. We will in fact show a stronger statement: for a random $\mathbf{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1$, we have $\mathbb{E}[|\mathcal{B}(\mathbf{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1|] \geq |\mathcal{C}_1| \cdot A_{\Delta(\mathcal{C}_2)}/|\mathcal{C}_2|$. Consider $\mathbb{E}_{\mathbf{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1}[|\mathcal{B}(\mathbf{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1|]$. Due to linearity of expectation, we have

$$\begin{aligned}
\mathbb{E}_{\mathbf{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1}[|\mathcal{B}(\mathbf{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1|] &= \sum_{\mathbf{c} \in \mathcal{C}_1} \Pr_{\mathbf{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1}[\mathbf{c} \in \mathcal{B}(\mathbf{s}, \Delta(\mathcal{C}_2))] \\
&= \sum_{\mathbf{c} \in \mathcal{C}_1} \Pr_{\mathbf{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1}[\Delta(\mathbf{s} - \mathbf{c}) \leq \Delta(\mathcal{C}_2)] \\
&= \sum_{\mathbf{c} \in \mathcal{C}_1} \Pr_{\mathbf{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1}[\Delta(\mathbf{s}) \leq \Delta(\mathcal{C}_2)] \\
&= |\mathcal{C}_1| \cdot \frac{|(\mathcal{C}_2 \setminus \mathcal{C}_1) \cap \mathcal{B}(\mathbf{0}, \Delta(\mathcal{C}_2))|}{|\mathcal{C}_2 \setminus \mathcal{C}_1|}.
\end{aligned}$$

Now, since $\Delta(\mathcal{C}_1) > \Delta(\mathcal{C}_2)$, we have $\mathcal{C}_1 \cap \mathcal{B}(\mathbf{0}, \Delta(\mathcal{C}_2)) = \{\mathbf{0}\}$. That is, $|(\mathcal{C}_2 \setminus \mathcal{C}_1) \cap \mathcal{B}(\mathbf{0}, \Delta(\mathcal{C}_2))| = |(\mathcal{C}_2 \setminus \{\mathbf{0}\}) \cap \mathcal{B}(\mathbf{0}, \Delta(\mathcal{C}_2))| = A_{\Delta(\mathcal{C}_2)}(\mathcal{C}_2)$. Plugging this back into the above equality, we have

$$\mathbb{E}_{\mathbf{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1}[|\mathcal{B}(\mathbf{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1|] = |\mathcal{C}_1| \cdot \frac{A_{\Delta(\mathcal{C}_2)}(\mathcal{C}_2)}{|\mathcal{C}_2 \setminus \mathcal{C}_1|} \geq |\mathcal{C}_1| \cdot \frac{A_{\Delta(\mathcal{C}_2)}(\mathcal{C}_2)}{|\mathcal{C}_2|}.$$

Thus, there must exist a center $\mathbf{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1$ that satisfies (2) (and also (1)) as desired.

Finally, note that $\mathbf{s}$ can be found by a brute force algorithm that tries every $\mathbf{s} \in \mathcal{C}_2$ and check whether (2) is satisfied; this algorithm takes $O(|\mathcal{C}_1| \cdot |\mathcal{C}_2| \cdot qN)$ time. □

5.2 Gadgets based on Reed-Solomon Codes

In this subsection, we construct gadgets based on the Reed Solomon codes, which are defined below.

Theorem 5.2 (Reed-Solomon Codes). *For every prime power q, and every $K \leq N \leq q$, there exists a $[N, K, N - K + 1]_q$ linear code, denoted by $\mathsf{RS}_q[N, K]$. The generator matrix of this code can be computed in time* $\mathrm{poly}(N, K, q)$. *Moreover, for every $q \geq N \geq K_2 > K_1$, we have $\mathsf{RS}_q[N, K_1] \subseteq \mathsf{RS}_q[N, K_2]$.*

In order to find a good center $\mathbf{s}$, we use the following (well-known) bound on the number of minimum weight codewords of Reed Solomon codes (and more generally MDS codes). For a reference of this bound, see e.g. [MS77, Ch. 11, Theorem 6].

Lemma 5.3. *Let $\mathcal{C}$ be any linear $[N, K, D]_q$ code that is MDS. Then, $A_D(\mathcal{C}) = \binom{N}{K-1} \cdot (q-1)$.*

5.2.1 The Basic Gadget: Dense Bipartite Graphs with Low Contact Dimensions

Now we construct a dense bipartite graph with low contact dimension. A proof sketch of this construction was provided in Section 2.1 and was formally stated as Theorem 4.2.

Proof of Theorem 4.2. Let q_i be the i-th prime number and let $n_i = (q_i)^{(\lfloor q_i^\delta \rfloor)}$; it is simple to see that the sequence $(n_i)_{i\in\mathbb{N}}$ is log-dense. For $q = q_i$, consider the Reed-Solomon codes $\mathcal{C}_1 = \mathsf{RS}_q[q, K_1]$ and $\mathcal{C}_2 = \mathsf{RS}_q[q, K_2]$ where $K_1 = \lfloor q^\delta \rfloor$ and $K_2 = K_1 + 1$. Applying Lemma 5.1 with $(\mathcal{C}_1, \mathcal{C}_2)$ implies that there exists a center $\mathbf{s} \in \mathcal{C}_2$ such that

$$\begin{aligned}
\frac{|\mathcal{B}(\mathbf{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1|}{|\mathcal{C}_1|} &\geq \frac{A_{\Delta(\mathcal{C}_2)}}{|\mathcal{C}_2|} \\
\text{(By Lemma 5.3)} &= \frac{\binom{q}{K_2-1} \cdot (q-1)}{q^{K_2}} \\
&\geq \frac{\left(\frac{q}{K_2-1}\right)^{K_2-1} \cdot (q-1)}{q^{K_2}} \\
&= \frac{q-1}{q} \cdot \left(\frac{1}{K_2-1}\right)^{K_2-1} \\
&= \frac{q-1}{q} \cdot \frac{1}{K_1^{K_1}} \\
&\geq \frac{1}{2} \cdot \frac{1}{q^{\delta K_1}} \\
&= \Omega(|\mathcal{C}_1|^{-\delta}),
\end{aligned}$$

where the last equality follows from the fact that $|\mathcal{C}_1| = q^{K_1}$.

We construct the graph $G_i = (A_i, B_i, E_i)$ and a realization τ as follows. Let $A_i = \mathcal{C}_1, B_i = \{\mathbf{s} + \mathbf{c} \mid \mathbf{c} \in \mathcal{C}_1\}$ and $E_i = \{(\mathbf{a}, \mathbf{b}) \in A_i \times B_i \mid \Delta(\mathbf{a}, \mathbf{b}) = \Delta(\mathcal{C}_2)\}$. G_i can be easily realized by applying the mapping $\psi : \mathbb{F}_q^q \to \{0,1\}^{q^2}$ from Proposition 3.12. More precisely, let τ be the restriction of ψ on $A_i \cup B_i$. Below we argue about the density of G_i and that τ is a $2\Delta(\mathcal{C}_2)$-realization of G_i.

- First, notice that $|E_i|$ is exactly $|\mathcal{C}_1| \cdot |\mathcal{B}(\mathbf{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1| \geq \Omega(|\mathcal{C}_1|^{2-\delta}) = \Omega(n_i^{2-\delta})$.

- Second, notice that, for every $\mathbf{v}_1, \mathbf{v}_2$ both from A_i or both from B_i, we have $\mathbf{v}_1 - \mathbf{v}_2 \in \mathcal{C}_1 \setminus \{\mathbf{0}\}$. This implies that $\|\tau(\mathbf{v}_1) - \tau(\mathbf{v}_2)\|_0 = 2\Delta(\mathbf{v}_1, \mathbf{v}_2) \geq 2\Delta(\mathcal{C}_1) > 2\Delta(\mathcal{C}_2)$.

- Third, for every $\mathbf{a} \in A_i$ and $\mathbf{b} \in B_i$, we have $\mathbf{a} - \mathbf{b} \in \mathcal{C}_2 \setminus \{\mathbf{0}\}$. Thus, $\Delta(\mathbf{a}, \mathbf{b}) \geq \Delta(\mathcal{C}_2)$. Hence, $\|\tau(\mathbf{a}) - \tau(\mathbf{b})\|_0 = 2\Delta(\mathbf{a}, \mathbf{b}) \geq 2\Delta(\mathcal{C}_2)$. Moreover, the inequality is an equality if and only if $\Delta(\mathbf{a}, \mathbf{b}) = \Delta(\mathcal{C}_2)$, i.e., $(\mathbf{a}, \mathbf{b}) \in E_i$ as desired.

- Finally, observe that the dimension is $q^2 = (\log n_i)^{O(1/\delta)}$.

As for the running time of constructing G_i and τ, observe that the bottleneck is the running time needed to find the center $\mathbf{s}$; according to Lemma 5.1, $\mathbf{s}$ can be computed in $O(|\mathcal{C}_1| \cdot |\mathcal{C}_2| \cdot q^2) = O(n_i^2 \cdot q^2)$, which is $n_i^{2+o(1)}$ as desired. □

5.2.2 A Gadget for Maximum Inner Product

Now, we build gadgets (stated below) which will be used for proving the inapproximability of MIP.

Theorem 5.4. *For every $0 < \delta < 1$, there exists a log-dense sequence $(n_i)_{i \in \mathbb{N}}$ such that, for every $i \in \mathbb{N}$, there is a bipartite graph $G_i = (A_i \dot{\cup} B_i, E_i)$ where $|A_i| = |B_i| = n_i$ and $|E_i| \geq \Omega(n_i^{2-\delta})$, such that 3-$\mathsf{ipd}(G) = (\log n_i)^{O(1/\delta)}$. Moreover, for all $i \in \mathbb{N}$, a 3-gap-IP-realization $\tau : A_i \dot{\cup} B_i \to \{0,1\}^{(\log n_i)^{O(1/\delta)}}$ of G_i can be constructed in time $n_i^{4+o(1)}$.*

Proof. The proof here is exactly the same as the proof of Theorem 4.2, except that we will not pick $K_2 = K_1 + 1$, but rather pick $K_2 > 3K_1$ (and n_i accordingly).

More precisely, let q_i be the i-th prime number and let $n_i = (q_i)^{(\lfloor q_i^{0.3\delta}/3 \rfloor)}$; it is simple to see that the sequence $(n_i)_{i \in \mathbb{N}}$ is log-dense. For $q = q_i$, consider the Reed-Solomon codes $\mathcal{C}_1 = \mathsf{RS}_q[q, K_1]$ and $\mathcal{C}_2 = \mathsf{RS}_q[q, K_2]$ where $K_1 = \lfloor q^{0.3\delta}/3 \rfloor$ and $K_2 = 3K_1 + 1$. Similar to the proof of Theorem 4.2, applying Lemma 5.1 with $(\mathcal{C}_1, \mathcal{C}_2)$ implies that there exists $\mathbf{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1$ such that

$$\frac{|\mathcal{B}(\mathbf{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1|}{|\mathcal{C}_1|} \geq \frac{q-1}{q} \cdot \left(\frac{1}{K_2 - 1}\right)^{K_2 - 1} = \frac{q-1}{q} \cdot \frac{1}{(3K_1)^{(3K_1)}} \geq \frac{1}{2} \cdot \frac{1}{q^{\delta K_1}} = \Omega(|\mathcal{C}_1|^{-\delta}).$$

We construct the graph $G_i = (A_i, B_i, E_i)$ and a realization τ as follows. Let $A_i = \mathcal{C}_1, B_i = \{\mathbf{s} + \mathbf{c} \mid \mathbf{c} \in \mathcal{C}_1\}$ and $E_i = \{(\mathbf{a}, \mathbf{b}) \in A_i \times B_i \mid \Delta(\mathbf{a}, \mathbf{b}) = \Delta(\mathcal{C}_2)\}$. G_i can be easily 3-gap-IP-realized by applying the mapping $\psi : \mathbb{F}_q^q \to \{0,1\}^{q^2}$ from Proposition 3.12. More precisely, let τ be the restriction of ψ on $A_i \cup B_i$. Below we argue about the density of G_i and that τ is a $(K_2 - 1, 3)$-gap-IP-realization of G_i.

- First, notice that $|E_i|$ is exactly $|\mathcal{C}_1| \cdot |\mathcal{B}(\mathbf{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1| \geq \Omega(|\mathcal{C}_1|^{2-\delta}) = \Omega(n_i^{2-\delta})$.
- Second, for every $\mathbf{v}_1, \mathbf{v}_2$ both from A_i or both from B_i, we have $\mathbf{v}_1 - \mathbf{v}_2 \in \mathcal{C}_1 \setminus \{\mathbf{0}\}$. Thus, $\langle \tau(\mathbf{v}_1), \tau(\mathbf{v}_2) \rangle = q - \Delta(\mathbf{v}_1, \mathbf{v}_2) \leq q - \Delta(\mathcal{C}_1) = K_1 - 1 < (K_2 - 1)/3$.
- Third, for every $\mathbf{a} \in A_i$ and $\mathbf{b} \in B_i$, we have $\mathbf{a} - \mathbf{b} \in \mathcal{C}_2 \setminus \{\mathbf{0}\}$. Thus, $\Delta(\mathbf{a}, \mathbf{b}) \geq \Delta(\mathcal{C}_2)$. Hence, $\langle \tau(\mathbf{a}), \tau(\mathbf{b}) \rangle = q - \Delta(\mathbf{a}, \mathbf{b}) \leq q - \Delta(\mathcal{C}_2) = K_2 - 1$. Moreover, the inequality is an equality if and only if $\Delta(\mathbf{a}, \mathbf{b}) = \Delta(\mathcal{C}_2)$, i.e., $(\mathbf{a}, \mathbf{b}) \in E_i$ as desired.
- Finally, observe that the dimension is $q^2 = (\log n_i)^{O(1/\delta)}$.

Once again, the running time of the construction is $O(|\mathcal{C}_1| \cdot |\mathcal{C}_2| \cdot q^2) \leq n_i^{4+o(1)}$. □

5.3 Gadgets based on AG Codes

In this subsection, we construct gadgets based on algebraic geometric (AG) codes. The definitions of AG Codes are well beyond the scope of this work and we refer the readers to [Sti08, VNT07] for more thorough introductions.

Once again to find a good center, we need a bound on the number of minimum weight codewords. On this front, we use the following bound[10] from [Vlă18]. Throughout this subsection, we follow the notations from [Vlă18].

Theorem 5.5 (Theorem 4.3 of [Vlă18])**.** *Let q be a prime power, X be a curve of genus g over $\mathbb{F}_q$, let $S \subseteq X(\mathbb{F}_q)$ such that $|S| = N$, and let $a \in \mathbb{N}$ with $1 \leq a \leq N-1$. Then, there exists an $\mathbb{F}_q$-positive divisor $D \geq 0, \deg(D) = a$, such that the corresponding AG Code $\mathcal{C} = \mathcal{C}(X, D, S)$ has minimum distance $N - a$ and*

$$A_{N-a}(\mathcal{C}) \geq \frac{\binom{N}{a}}{(\sqrt{q}+1)^{2g}}.$$

We also need the following well-known (central) fact about the parameters of AG codes.

Theorem 5.6. *Let q be a prime power, X be a curve of genus g over $\mathbb{F}_q$, let $S \subseteq X(\mathbb{F}_q)$ such that $|S| = N$, and let $a \in \mathbb{N}$ with $1 \leq a \leq N-1$. Then, the corresponding AG Code $C = C(X, D, S)$ is a linear code over $\mathbb{F}_q$ with block length N, distance at least $N - a$ and message length $k \geq a - g + 1$.*

Recall also the tower of functions of Garcia and Stichtenoth [GS96], whose parameters approach the TVZ bound. We note here that, it suffices for us to have the genus approaching $\Omega(N/\sqrt{q})$ and there are also other curves that satisfy this.

Theorem 5.7 ([GS96])**.** *For any $\zeta > 0$ and any square of prime q, there exists a dense sequence[11] $(N_i)_{i\in\mathbb{N}}$ such that there exists a curve X_i with genus at most $\frac{N_i}{\sqrt{q}-1} + \zeta$ where $|X_i(\mathbb{F}_q)| \geq N_i$.*

Plugging the bound from [Vlă18] into the above family of curves immediately yields the following:

Lemma 5.8. *For any $\zeta > 0$ and any square of prime q, there exists a dense sequence $(N_i)_{i\in\mathbb{N}}$ such that the following holds. For any $i \in \mathbb{N}$ and any $a_1, a_2 \in \mathbb{N}$ such that $1 \leq a_1 < a_2 \leq N_i - 1$, there exists linear codes $\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \mathbb{F}_q^{N_i}$ such that the following holds, where $g_i = \frac{N_i}{\sqrt{q}-1} + \zeta$:*

- *$\mathcal{C}_1$ has message length at least $a_1 - g_i + 1$ and distance at least $N_i - a_1$.*
- *$\mathcal{C}_2$ has message length at least $a_2 - g_i + 1$ and distance exactly $N_i - a_2$ and*

$$A_{N_i-a_2}(\mathcal{C}_2) \geq \frac{\binom{N_i}{a_2}}{(\sqrt{q}+1)^{2g_i}}. \tag{3}$$

Moreover, the generator matrices of $\mathcal{C}_1, \mathcal{C}_2$ can be computed in $O\left(\binom{N+a_2-1}{a_2} \cdot |\mathcal{C}_2| \cdot \text{poly}(N_i)\right)$ time.

Proof. Let $(N_i)_{i\in\mathbb{N}}$ be a dense sequence as in Theorem 5.7. From Theorem 5.5, there exists an $\mathbb{F}_q$-positive divisor D_2 of degree a_2 such that the corresponding code $\mathcal{C}_2 =$

[10] Note that most of the proof of this bound was from [ABV01]; [Vlă18] simply makes the bound more explicit, which is more convenience for us.

[11] A sequence $(N_i)_{i\in\mathbb{N}}$ of increasing positive integers is said to be *dense* if there exists a constant $C \geq 1$ such that $N_{i+1} \leq C \cdot N_i$ for all $i \in \mathbb{N}$.

$C(X_i, D_2, S_i)$ (where $S \subseteq X_i(\mathbb{F}_q)$ of size N_i) satisfies (3) and that its distance is $N_i - a_2$; from Theorem 5.6, its message length must also be at least $a_2 - g_i + 1$. Next, let D_1 be any $\mathbb{F}_q$-positive divisor of degree a_1 such that $D_2 - D_1 \geq 0$. Let $\mathcal{C}_1 = C(X_i, D_1, S_i)$ be the corresponding AG code; once again, Theorem 5.6 yields the desired bounds on its message length and distance. Finally, observe that $D_2 - D_1 \geq 0$ implies that $\mathcal{C}_1 \subseteq \mathcal{C}_2$ as desired.

The main bottleneck to algorithmically construct such codes lies in finding D_2. Nevertheless, the total number of degree-a_2 $\mathbb{F}_q$-positive divisor is only $\binom{N_i + a_2 - 1}{a_2}$. We can use brute force to enumerate all of them and check whether the corresponding code satisfies (3), which further takes $|\mathcal{C}_2|$ time. This results in the claimed running time. □

Finally, we can now construct our gadgets, by an appropriate setting of parameters. In particular, a_1 and a_2 will be selected to be close to each other and to both be slightly larger than $N/\sqrt{q}$. This results in the graphs whose degrees are roughly square root of the number of vertices.

Theorem 5.9. *For every $0 < \delta < 1$, there exist $\mu > 0$ and a log-dense sequence $(n_i)_{i \in \mathbb{N}}$ such that, for every $i \in \mathbb{N}$, there is a bipartite graph $G_i = (A_i \dot\cup B_i, E_i)$ where $|A_i| = |B_i| = n_i$ and $|E_i| \geq \Omega(n_i^{2-\delta})$, such that $(1+\mu)$-$\mathsf{cd}(G) = O(\log n_i)$. Moreover, for all $i \in \mathbb{N}$, a $(\beta, 1+\mu)$-gap-realization $\tau : A_i \dot\cup B_i \to \{0,1\}^{O(\log n_i)}$ of G_i can be constructed in time $O(n_i^3)$ for some $\beta = \Theta(\log n_i)$.*

Proof. Once again, the proof here is similar to those of Theorems 4.2 and 5.4, except that we use the (pairs of) AG codes from Lemma 5.8 instead of Reed-Solomon codes.

Let $q \geq 49$ be any sufficiently large square of prime and $\zeta > 0$ be any sufficiently small positive real number (both to be precisely specified later).

Let $(N_i)_{i \in \mathbb{N}}$ be the sequence guarantee by Lemma 5.8. Let $a_1 = N_i \cdot \left(\frac{1}{q^{0.5(1-\delta)}} - \frac{1}{q}\right)$ and $a_2 = \frac{N_i}{q^{0.5(1-\delta)}}$. For convenience, we assume that a_1 and a_2 are integers[12]. Let $\mathcal{C}_1, \mathcal{C}_2$ be the codes given by Lemma 5.8. The sequence $(n_i)_{i \in \mathbb{N}}$ is defined as $n_i = |\mathcal{C}_1|$.

Applying Lemma 5.1 to $(\mathcal{C}_1, \mathcal{C}_2)$ implies that there exists $\mathbf{s} \in \mathcal{C}_2 \setminus \mathcal{C}_1$ such that

$$\begin{aligned}
\frac{|\mathcal{B}(\mathbf{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1|}{|\mathcal{C}_1|} &\geq \frac{A_{\Delta(\mathcal{C}_2)}(\mathcal{C}_2)}{|\mathcal{C}_2|} \\
\text{(From Lemma 5.8)} &\geq \frac{\binom{N_i}{a_2}}{(\sqrt{q}+1)^{2g_i} \cdot |C_2|} \\
\text{(Singleton Bound)} &\geq \frac{\binom{N_i}{a_2}}{(\sqrt{q}+1)^{2g_i} \cdot q^{a_2+1}} \\
&\geq \frac{(N_i/a_2)^{a_2}}{(\sqrt{q}+1)^{2g_i} \cdot q^{a_2+1}} \\
&= \frac{q^{0.5(1-\delta)a_2}}{(\sqrt{q}+1)^{2g_i} \cdot q^{a_2+1}} \\
&= \frac{1}{(\sqrt{q}+1)^{2g_i} \cdot q^{(0.5+0.5\delta)a_2+1}}
\end{aligned}$$

[12] Note that, for sufficiently large N_i, one can take the ceilings (or floors) of the specified values to get integers with negligible affect to the calculations.

$$
\begin{aligned}
&= \frac{1}{q^{(0.5+0.5\delta+o(1))a_2}} \\
&= \frac{1}{q^{(0.5+0.5\delta+o(1))(a_1+o(1))}} \\
&= \frac{1}{|\mathcal{C}_1|^{(0.5+0.5\delta+o(1))}} \\
&\geq \Omega(|\mathcal{C}_1|^{-0.5-0.5\delta-o(1)}) \qquad (4)
\end{aligned}
$$

where $o(1)$ terms above denote the terms that go to zero as $q \to \infty$ and $\zeta \to 0$. As a result, by picking q sufficiently large and ζ sufficiently small, the term in (4) is at least $\Omega(|\mathcal{C}_1|^{-0.5-\delta})$.

We construct the graph $G_i = (A_i, B_i, E_i)$ and a realization τ as follows. Let $A_i = \mathcal{C}_1, B_i = \{\mathbf{s} + \mathbf{c} \mid \mathbf{c} \in \mathcal{C}_1\}$ and $E_i = \{(\mathbf{a}, \mathbf{b}) \in A_i \times B_i \mid \Delta(\mathbf{a}, \mathbf{b}) = \Delta(\mathcal{C}_2)\}$. G_i can be easily realized by applying the mapping $\psi : \mathbb{F}_q^q \to \{0,1\}^{q^2}$ from Proposition 3.12. More precisely, let τ be the restriction of ψ on $A_i \cup B_i$. Below we argue about the density of G_i and that τ is a $(2\Delta(\mathcal{C}_2), 1+\mu)$-gap-realization of G_i where $\mu = \frac{\Delta(\mathcal{C}_1)-1}{\Delta(\mathcal{C}_2)} - 1$. Note that

$$\mu \geq \frac{a_2 - a_1 - 1}{N_i - a_2} = \Omega(1/q).$$

Let us now check that G_i and τ satisfy all the claimed properties:

- First, notice that $|E_i|$ is exactly $|\mathcal{C}_1| \cdot |\mathcal{B}(\mathbf{s}, \Delta(\mathcal{C}_2)) \cap \mathcal{C}_1| \geq \Omega(|\mathcal{C}_1|^{1.5-\delta}) = \Omega(n_i^{1.5-\delta})$.
- For any $\mathbf{v}_1 = \psi(\mathbf{c}_1), \mathbf{v}_2 = \psi(\mathbf{c}_2)$ both from X_i or both from Y_i, we have $\mathbf{c}_1 - \mathbf{c}_2 \in \mathcal{C}_1 \setminus \{\mathbf{0}\}$. Hence, $\|\mathbf{v}_1 - \mathbf{v}_2\|_0 = 2 \cdot \Delta(\mathbf{v}_1, \mathbf{v}_2) \geq 2 \cdot \Delta(\mathcal{C}_1) > (1+\mu) \cdot (2\Delta(\mathcal{C}_2))$.
- Next, for every $\mathbf{a} \in A_i$ and $\mathbf{b} \in B_i$, we have $\mathbf{a} - \mathbf{b} \in \mathcal{C}_2 \setminus \{\mathbf{0}\}$. Thus, $\Delta(\mathbf{a}, \mathbf{b}) \geq \Delta(\mathcal{C}_2)$. Hence, $\|\tau(\mathbf{a}) - \tau(\mathbf{b})\|_0 = 2\Delta(\mathbf{a}, \mathbf{b}) \geq 2\Delta(\mathcal{C}_2)$. Moreover, the inequality is an equality if and only if $\Delta(\mathbf{a}, \mathbf{b}) = \Delta(\mathcal{C}_2)$, i.e., $(\mathbf{a}, \mathbf{b}) \in E_i$ as desired.

Given $\mathcal{C}_1, \mathcal{C}_2$, the running time of constructing (X_i, Y_i) is $O(|\mathcal{C}_1| \cdot |\mathcal{C}_2| \cdot q^2) = O(n_i^3)$. Moreover, the running time to construct $\mathcal{C}_1$ and $\mathcal{C}_2$, as given by Lemma 5.8, is

$$
\begin{aligned}
O\left(\binom{N + a_2 - 1}{a_2} \cdot |\mathcal{C}_2| \cdot \mathrm{poly}(N_i)\right) &\leq O\left((e(N+a_2)/a_2)^{a_2} \cdot |\mathcal{C}_2| \cdot \mathrm{poly}(N_i)\right) \\
&\leq O\left((2e\sqrt{q})^{a_2} \cdot |\mathcal{C}_2| \cdot \mathrm{poly}(N_i)\right) \\
&\leq O\left(|\mathcal{C}_1| \cdot |\mathcal{C}_2| \cdot \mathrm{poly}(N_i)\right) \\
&\leq O(n_i^3),
\end{aligned}
$$

where the last two inequalities are true for any sufficiently large q. □

6 Inapproximability of Maximum Inner Product

In this section, we prove the hardness of approximating MIP. Once again, we show a stronger version (than Theorem 1.6) where every point has Boolean coordinates, as stated below.

Theorem 6.1. *Assuming* OVH*, for every $\varepsilon > 0$, there is no algorithm running in $O(n^{2-\varepsilon})$ time for γ-*MIP *even for points in $\{0,1\}^{n^{o(1)}}$, for any $\gamma \leq 2^{(\log n)^{1-o(1)}}$.*

The proof proceeds in two steps: first, we show hardness of approximating MIP in low dimension but with a small $(1+o(1))$ approximation factor. Second, we use tensor product operation to amplify the gap to be almost polynomial, as stated in Theorem 6.1. More specifically, in the first step, we prove the following:

Theorem 6.2. *Assuming* OVH*, for every $\varepsilon > 0$, there exists $s_\varepsilon > 0$ such that no algorithm running in $O(n^{2-\varepsilon})$ time can solve $\left(1 + \frac{1}{\log\log n}\right)$-*MIP *even for points in $\{0,1\}^{(\log n)^{s_\varepsilon}}$.*

Note that the factor $\frac{1}{\log\log n}$ is not significant, and this can be replaced by any $o(1)$ factor; we use this just to make the calculations more concrete. Before we move on to the proof of Theorem 6.2, let us first show how it implies Theorem 6.1.

Proof of Theorem 6.1 from Theorem 6.2. Let (P,α) be an instance of $\left(1 + \frac{1}{\log\log n}\right)$-MIP where $P \subseteq \{0,1\}^{(\log n)^{s_\varepsilon}}$. For $t = \frac{\log n}{(\log\log n)^2}$, define $P' = \{\mathbf{x}^{\otimes t} \mid \mathbf{x} \in P\}, \alpha' = \alpha^t$ and $\gamma = \left(1 + \frac{1}{\log\log n}\right)^t = 2^{(\log n)^{1-o(1)}}$. The dimension of points in P' is $(\log n)^{s_\varepsilon \cdot t} = n^{o(1)}$. Moreover, it is easy to check, based on the identity $\langle \mathbf{x}^{\otimes t}, \mathbf{y}^{\otimes t}\rangle = \langle \mathbf{x}, \mathbf{y}\rangle^t$, that (P', α') is a YES (resp. no) instance of γ-MIP iff (P,α) is a YES (resp. NO) instance of $\left(1 + \frac{1}{\log\log n}\right)$-MIP.

In other words, if there is an $O(n^{2-\varepsilon})$ time algorithm for γ-MIP in $n^{o(1)}$ dimension, then there also exist an $O(n^{2-\varepsilon})$ subquadratic time algorithm for $\left(1 + \frac{1}{\log\log n}\right)$-MIP in $(\log n)^{s_\varepsilon}$ dimension. Thus, Theorem 6.1 follows from Theorem 6.2. □

The rest of this section is devoted to proving Theorem 6.2. To do so, we consider the gap-Additive-BMIP problem.

Definition 6.3 (γ-Additive-BMIP problem)**.** *Let $\gamma \geq 0$. In the γ-*Additive-BMIP *problem we are given two sets A, B each of n points in $\{0,1\}^d$ and an integer $\alpha \in [d]$ as input, and the goal is to distinguish between the following two cases.*

- ***Completeness.*** *There exists $(a,b) \in A \times B$ such that $\langle a,b\rangle \geq \alpha$.*
- ***Soundness.*** *For every $(a,b) \in A \times B$ we have $\langle a,b\rangle < \alpha - \gamma$.*

We need the below hardness result from [Rub18]. Note that the result is stated differently in [Rub18]; for how the result in [Rub18] implies the one below, see Section 3.2 of [Che18a].

Theorem 6.4 ([Rub18])**.** *Assuming* OVH*, for every $\varepsilon > 0$, there is no algorithm running in $O(n^{2-\varepsilon})$ time for the γ-*Additive-BMIP *problem, for any $d = \omega(\log n)$ and $\gamma = o(d)$.*

Proof of Theorem 6.2. For any $\varepsilon > 0$, let $C_{\exp}$ be the constant such that the dimension of τ in Theorem 5.4 is at most $(\log n_i)^{C_{\exp}/\varepsilon}$ for $\delta = \varepsilon/2$. We define s_ε as $2 \cdot C_{\exp}/\varepsilon + 2$.

Suppose contrapositively that there exists $\varepsilon > 0$ and an algorithm $\mathcal{A}$ that can solve $\left(1 + \frac{1}{\log\log n}\right)$-MIP of dimension $(\log n)^{s_\varepsilon}$ in time $n^{2-\varepsilon}$. We will construct an algorithm $\mathcal{A}'$

that solves $(\log n)$-Additive-BMIP in time $n^{2-\varepsilon'}$ for some constant $\varepsilon' > 0$ (to be specified below) for $d = (\log n\sqrt{\log\log n})$ dimensions. Together with Theorem 6.4, this implies that OVH is false, as desired.

Let C_ε denote the constant of the log-dense sequence from Theorem 5.4 for $\delta = \varepsilon/2$, and let ε' be $0.01 \cdot \varepsilon/C_\varepsilon$. The algorithm $\mathcal{A}'$ on input (A, B, α) where $A, B \subseteq \{0,1\}^d, \alpha \in [d]$ works as follows:

1. Let n' be the largest number in the sequence from Theorem 5.4 with $\delta = \varepsilon/2$ s.t. $n' \leq n^{0.1}$.
2. Let $G' = (A'\dot\cup B', E')$ be the graph from Theorem 5.4 with $|A'| = |B'| = n'$, $|E'| \geq \Omega((n')^{2-\delta})$, and $\tau : A'\dot\cup B' \to \{0,1\}^{(\log n')^{C\mathrm{exp}/\varepsilon}}$ be a $(\beta, 3)$-gap-IP-relization of G' where $\beta \in \mathbb{N}$.
3. We use the algorithm from Lemma 3.11 to find $\pi_1, \dots, \pi_k$ where $k = O((n')^\delta \log n')$ such that the union of $E_{G'_{\pi_1}}, \dots, E_{G'_{\pi_k}}$ is $E_{K_{n',n'}}$
4. We assume w.l.o.g. that n is divisible by n'. Partition A and B into $A_1, \dots, A_{n/n'}$ and $B_1, \dots, B_{n/n'}$ each of size n'. For each $i, j \in [n/n'], t \in [k]$, do the following:
   (a) Let τ_t be an appropriate permutation of τ that $(\beta, 3)$-gap-IP-realizes G'_{π_t}.
   (b) Let $\alpha' = \beta \cdot \alpha + 3d \cdot \beta$, and define A_i^t, B_j^t as
   $$A_i^t = \{(\mathbf{1}_\beta \otimes \mathbf{a}) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{a})) \mid \mathbf{a} \in A_i\}, B_j^t = \{(\mathbf{1}_\beta \otimes \mathbf{b}) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{b})) \mid \mathbf{b} \in B_j\}.$$
   (c) Run $\mathcal{A}$ on $(A_i^t \dot\cup B_j^t, \alpha')$. If $\mathcal{A}$ outputs YES, then output YES and terminate.
5. If none of the executions of $\mathcal{A}$ returns with YES, then output NO.

Observe that the bottleneck in the running time of the algorithm is in the executions of $\mathcal{A}$. The number of executions is $(n/n')^2 \cdot k$ and each execution takes $O((n')^{2-\varepsilon})$ time. Hence, in total the running time of the algorithm $\mathcal{A}'$ is $O((n/n')^2 \cdot k \cdot (n')^{2-\varepsilon}) \leq O(n^2 \log n \cdot (n')^{-\varepsilon/2})$. Now, from the log-density of the sequence from Theorem 5.4, we have $n' \geq n^{0.1/C_\varepsilon} = n^{10\varepsilon'/\varepsilon}$. As a result, the running time of $\mathcal{A}$ is at most $O(n^{2-5\varepsilon'} \log n) \leq O(n^{2-\varepsilon'})$ as desired.

To see the correctness of the algorithm, first observe that the dimensions of vectors in A_i^t, B_j^t are at most $\beta \cdot d + 3d \cdot (\log n')^{C_{\exp}/\varepsilon}$ which is at most $(\log n)^{s_\varepsilon}$ for any sufficiently large n; that is, the calls to $\mathcal{A}$ are valid. Next, observe that, if (A, B, α) is a YES instance of Additive-BMIP, there must be $i, j \in [n/n']$ and $\mathbf{a}^* \in A_i, \mathbf{b}^* \in B_j$ such that $\langle \mathbf{a}^*, \mathbf{b}^* \rangle$ is at least α. Since $G'_{\pi_1}, \dots, G'_{\pi_k}$ covers $K_{n',n'}$, there must be $t \in [k]$ such that $\langle \tau_t(\mathbf{a}^*), \tau_t(\mathbf{b}^*) \rangle \geq \beta$. As a result, $\big\langle (\mathbf{1}_\beta \otimes \mathbf{a}^*) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{a}^*), (\mathbf{1}_\beta \otimes \mathbf{b}^*) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{b}^*)) \big\rangle \geq \beta \cdot \alpha + 3d \cdot \beta = \alpha'$. Thus, $(A_i^t \cup B_j^t, \alpha')$ is a YES instance for MIP and $\mathcal{A}'$ outputs YES as desired.

Finally, let us assume that (A, B, α) is a NO instance of $(\log n)$-Additive-BMIP. Consider any $i, j \in [n/n']$ and $t \in [k]$. To argue that $(A_i^t \cup B_j^t, \alpha')$ is a NO instance for $\left(1 + \frac{1}{\log\log n'}\right)$-MIP, we have to show that any two points in $A_i^t \cup B_j^t$ have inner product less than $\alpha' / \left(1 + \frac{1}{\log\log n'}\right)$. To see this, let us consider two cases.

1. The two points are either both from A_i^t or both from B_j^t. Assume w.l.o.g. that the two points are from A_i^t; let them be $(\mathbf{1}_\beta \otimes \mathbf{a}) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{a}))$ and $(\mathbf{1}_\beta \otimes \mathbf{a}') \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{a}'))$. Recall that, from Theorem 5.4, we must have $\langle \tau_t(\mathbf{a}), \tau_t(\mathbf{a}') \rangle < \beta/3$. Moreover, since $\mathbf{a}, \mathbf{a}' \in \{0,1\}^d$, we have $\langle \mathbf{a}, \mathbf{a}' \rangle \leq d$. Thus, we can conclude that
$$\begin{aligned} \big\langle (\mathbf{1}_\beta \otimes \mathbf{a}) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{a})), (\mathbf{1}_\beta \otimes \mathbf{a}') \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{a}')) \big\rangle &< \beta \cdot d + 3d \cdot (\beta/3) \\ &< (2/3) \cdot \alpha', \end{aligned}$$
which is less than $\alpha' / \left(1 + \frac{1}{\log\log n'}\right)$ for any sufficiently large n.

2. One of the point is from A_i^t and the other from B_j^t. Let them be $(\mathbf{1}_\beta \otimes \mathbf{a}) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{a}))$ and $(\mathbf{1}_\beta \otimes \mathbf{b}) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{b}))$. Since (A, B, α) is a NO instance of $(\log n)$-Additive-BMIP, we must have $\langle \mathbf{a}, \mathbf{b} \rangle < \alpha - \log n$. Furthermore, from Theorem 5.4, we must have $\langle \tau_t(\mathbf{a}), \tau_t(\mathbf{b}) \rangle \leq \beta$. Combining the two implies that
$$\begin{aligned} \big\langle (\mathbf{1}_\beta \otimes \mathbf{a}) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{a})), (\mathbf{1}_\beta \otimes \mathbf{b}) \circ (\mathbf{1}_{3d} \otimes \tau_t(\mathbf{v})) \big\rangle &< \beta \cdot (\alpha - \log n) + 3d \cdot \beta \\ &= \alpha' - \beta \cdot (\log n) \\ (\text{Since } \alpha' \leq 4d\beta) &\leq \alpha' \left(1 - \frac{1}{4\sqrt{\log\log n}}\right) \\ &\leq \alpha' \left(1 - \frac{1}{\log\log n'}\right) \\ &\leq \alpha' / \left(1 + \frac{1}{\log\log n'}\right), \end{aligned}$$
where the second-to-last inequality holds for any sufficiently large n.

Hence, $(A_i^t \dot\cup B_j^t, \alpha')$ must be a NO instance for $\left(1 + \frac{1}{\log\log n'}\right)$-MIP for every $t \in [k]$ and $i, j \in [n/n']$. Thus, $\mathcal{A}'$ outputs NO as desired. □

7 Inapproximability of Closest Pair

In this section, we prove the hardness of approximating CP (Theorem 1.5). As usual, we reduce from the bichromatic version of the problem, and the lower bound for the bichromatic version is stated below:

Theorem 7.1 (Rubinstein [Rub18]). *Assuming* OVH, *for every $\varepsilon > 0$ there exists $\kappa > 0$ such that there is no algorithm running in $n^{2-\varepsilon}$ time for $(1+\kappa)$-*BCP *in the Hamming metric. Moreover, this holds even for instances (A, B, α) of $(1+\kappa)$-*BCP *when $d = \Theta_\varepsilon(\log n), \alpha = \Theta_\varepsilon(\log n)$ and $A, B \subseteq \{0,1\}^d$.*

Again, we prove below the inapproximability of the gap-CP problem for Boolean vectors. Clearly, this immediately implies Theorem 1.5.

Theorem 7.2. *Assuming* OVH, *for every $\varepsilon > 0$, there exists $\theta > 0$ and $c > 0$ such that there is no algorithm running in $n^{1.5-\varepsilon}$ time for $(1+\theta)$-*CP *in the Hamming metric for point-set in $\{0,1\}^{c \cdot \log n}$.*

Proof. Assume towards a contradiction that there exists an $\varepsilon > 0$ and an algorithm $\mathcal{A}$ that, for every $\theta > 0$ solves $(1+\theta)$-CP of dimension $c \cdot \log n$ in time $O(n^{1.5-\varepsilon})$, where $c := c(\varepsilon)$ is a constant that will be specified later. Let $\varepsilon' > 0$ be a small constant (depending on ε) that we will specify below and let $\kappa = \kappa(\varepsilon')$ be as in Theorem 7.1. We construct below an algorithm $\mathcal{A}'$ that solves $(1+\kappa)$-BCP in time $O(n^{2-\varepsilon'})$ for any instance (A, B, α) such that $A, B \subseteq \{0,1\}^{O(\log n)}$ and $\alpha = \Theta(\log n)$. Together with Theorem 7.1, this implies that OVH is false, as desired.

Let C_ε denote the constant of the log-dense sequence from Theorem 5.9 for $\delta = \varepsilon/2$, and let ε' be $0.01 \cdot \varepsilon / C_\varepsilon$. Let μ be the constant from Theorem 5.9. Select $\theta > 0$ be a sufficiently small constant such that $\frac{\mu-\theta}{1+\theta} > \frac{\theta}{\kappa-\theta}$.

The algorithm $\mathcal{A}'$ on (A, B, α) where $A, B \subseteq \{0,1\}^{O(\log n)}, \alpha = \Theta(\log n)$ works as follows:

1. Let n' be the largest number in the sequence from Theorem 5.9 with $\delta = \varepsilon/2$ s.t. $n' \leq n^{0.1}$.
2. Let $G' = (A' \dot\cup B', E')$ be the graph from Theorem 5.9 with $|A'| = |B'| = n'$, $|E'| \geq \Omega((n')^{1.5-\delta})$, and $\tau : A' \dot\cup B' \to \{0,1\}^{O(\log n')}$ be a $(\beta, 1+\mu)$-gap-relization of G' where $\beta \in \mathbb{N}$ and $\beta = \Theta(\log n')$.
3. We use the algorithm from Lemma 3.11 to find $\pi_1, \dots, \pi_k$ where $k = O((n')^{0.5+\delta} \log n')$ such that the union of $E_{G'_{\pi_1}}, \dots, E_{G'_{\pi_k}}$ is $E_{K_{n',n'}}$
4. We assume w.l.o.g. that n is divisible by n'. Partition A and B into $A_1, \dots, A_{n/n'}$ and $B_1, \dots, B_{n/n'}$ each of size n'. For each $i, j \in [n/n'], t \in [k]$, do the following:
   (a) Let τ_t be an appropriate permutation of τ that $(\beta, 1+\mu)$-gap-realizes G'_{π_t}.
   (b) Pick r_1, r_2 such that
   $$\frac{\theta}{\kappa-\theta} \cdot \frac{\beta}{\alpha} \leq \frac{r_1}{r_2} \leq \frac{\mu-\theta}{1+\theta} \cdot \frac{\beta}{\alpha}. \tag{5}$$
   Notice that the upper and lower bounds are $\Theta(1)$ and they are also $\Theta(1)$ apart. Hence, we can pick these r_1, r_2 so that $r_1, r_2 = \Theta(1)$.
   (c) Let $\alpha' = r_1 \cdot \alpha + r_2 \cdot \beta$ and define A_i^t, B_j^t as
   $$A_i^t = \{(\mathbf{1}_{r_1} \otimes \mathbf{a}) \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{a})) \mid \mathbf{a} \in A_i\}, B_j^t = \{(\mathbf{1}_{r_1} \otimes \mathbf{b}) \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{b})) \mid \mathbf{b} \in B_j\}.$$
   (d) Run $\mathcal{A}$ on $(A_i^t \cup B_j^t, \alpha')$. If $\mathcal{A}$ outputs YES, then output YES and terminate.
5. If none of the executions of $\mathcal{A}$ returns with YES, then output NO.

Observe that the bottleneck in the running time of the algorithm is in the executions of $\mathcal{A}$. The number of executions is $(n/n')^2 \cdot k$ and each execution takes $O((n')^{1.5-\varepsilon})$ time. Hence, in total the running time of the algorithm $\mathcal{A}'$ is $O((n/n')^2 \cdot k \cdot (n')^{1.5-\varepsilon}) \leq O(n^2 \log n \cdot (n')^{-\varepsilon/2})$. Now, from the log-density of the sequence from Theorem 5.9, we have $n' \geq n^{0.1/C_\varepsilon} = n^{10\varepsilon'/\varepsilon}$. As a result, the running time of $\mathcal{A}$ is at most $O(n^{2-5\varepsilon'} \log n) \leq O(n^{2-\varepsilon})$ as desired.

To see the correctness of the algorithm, first observe that the dimensions of vectors in A_i^t, B_j^t are at most $r_1 \cdot \alpha + r_2 \cdot \beta$ which is $O(\log n')$; that is, the calls to $\mathcal{A}$ are valid.

Next, observe that, if (A, B, α) is a YES instance of BCP, there must be $i, j \in [n/n']$ and $\mathbf{a}^* \in A_i, \mathbf{b}^* \in B_j$ such that $\|\mathbf{a}^* - \mathbf{b}^*\|_0$ is at most α. Since $G'_{\pi_1}, \ldots, G'_{\pi_k}$ covers $K_{n',n'}$, there must be $t \in [k]$ such that $\|\tau_t(\mathbf{a}^*) - \tau_t(\mathbf{b}^*)\|_0 \leq \beta$. As a result, $\|((\mathbf{1}_{r_1} \otimes \mathbf{a}^*) \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{a}^*)) - ((\mathbf{1}_{r_1} \otimes \mathbf{b}^*) \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{b}^*)))\|_0 \leq r_1 \cdot \alpha + r_2 \cdot \beta = \alpha'$. Thus, $(A_i^t \cup B_j^t, \alpha')$ is a YES instance for CP and $\mathcal{A}'$ outputs YES as desired.

Finally, let us assume that (A, B, α) is a NO instance of $(1 + \kappa)$-BCP. Consider any $i, j \in [n/n']$ and $t \in [k]$. To argue that $(A_i^t \cup B_j^t, \alpha')$ is a NO instance for $(1 + \theta)$-CP, we have to show that any two points in $A_i^t \cup B_j^t$ have distance more than α'. To see this, let us consider two cases.

1. Both points are either from A_i^t or from B_j^t. Assume w.l.o.g. that they are from A_i^t; let them be $(\mathbf{1}_{r_1} \otimes \mathbf{a}) \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{a}))$ and $(\mathbf{1}_{r_1} \otimes \mathbf{a}') \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{a}'))$. Recall that, from the definition of X'_t and Theorem 5.9, we must have $\|\tau_t(\mathbf{a}) - \tau_t(\mathbf{a}')\|_0 > (1 + \mu) \cdot \beta$. Thus, the Hamming distance between the two points is more than $r_2 \cdot (1 + \mu) \cdot \beta \geq (1 + \theta) \cdot \alpha'$, where the inequality comes from our choice of r_1, r_2.
2. One of the point is from A_i^t and the other from B_j^t. Let them be $(\mathbf{1}_{r_1} \otimes \mathbf{a}) \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{a}))$ and $(\mathbf{1}_{r_1} \otimes \mathbf{b}) \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{b}))$. Since (A, B, α) is a NO instance of $(1 + \kappa)$-BCP, $\|\mathbf{a} - \mathbf{b}\|_0 > (1 + \kappa) \cdot \alpha$. Moreover, from definition of τ_t, we must have $\|\tau_t(\mathbf{a}) - \tau_t(\mathbf{b})\|_0 \geq \beta$. Combining the two implies that the distance between $(\mathbf{1}_{r_1} \otimes \mathbf{a}) \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{a}))$ and $(\mathbf{1}_{r_1} \otimes \mathbf{b}) \circ (\mathbf{1}_{r_2} \otimes \tau_t(\mathbf{b}))$ is more than $r_1 \cdot (1 + \kappa) \cdot \alpha + r_2 \cdot \beta \geq (1 + \theta) \cdot \alpha'$, where the inequality is once again from our choice of r_1, r_2.

Hence, $(A_i^t \dot\cup B_j^t, \alpha')$ must be a NO instance for $(1 + \theta)$-CP for every $t \in [k]$ and $i, j \in [n/n']$. Thus, $\mathcal{A}'$ outputs NO as desired. □

8 Discussion and Open Questions

It remains open to completely resolve Open Questions 1.1 and 1.2. It is still possible that our framework can be used to resolve these problems: we just need to construct gadgets with better parameters! In particular, to resolve Question 1.1, we have to improve the dimension bound in Theorem 4.2 to $O_\delta(\log n_i)$. For Question 1.2, we just have to improve the bound on the number of pairs in (3) of Theorem 5.9 to $\Omega(n_i^{2-\delta})$. Following our observation from Lemma 5.1, this motivates us to ask the following purely coding theoretic question:

Open Question 8.1. *For every $0 < \delta < 1$, are there linear codes $\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \mathbb{F}_q^N$ both of block length N over alphabet $\mathbb{F}_q$ such that the following holds:*

- $\Delta(\mathcal{C}_1) \geq (1 + f(\delta)) \cdot \Delta(\mathcal{C}_2)$, *for some $f : (0, 1) \to (0, 1)$.*
- $|A_{\Delta(\mathcal{C}_2)}(\mathcal{C}_2)|/|\mathcal{C}_2| \geq |\mathcal{C}_1|^{-\delta}$.

Apart from the aforementioned questions, Rubinstein [Rub18] pointed out an interesting obstacle, aptly dubbed the “triangle inequality barrier”, to obtain fine-grained lower bounds against 3-approximation algorithms for BCP (see Open Question 3 in [Rub18]). In the case of CP, this barrier turns out to be against 2-approximation algorithms as noted in [DKL18]. We reiterate this below as an open problem to be resolved:

Open Question 8.2. *Can we show that assuming* SETH*, for some constant* $\varepsilon > 0$*, no algorithm running in time* $n^{1+\varepsilon}$ *can solve* 2-CP *in* any *metric when the points are in* $\omega(\log n)$ *dimensions?*

Another interesting direction is to extend the hardness of MIP to the k-vector generalization of the problem, called k-MIP. In k-MIP, we are given a set of n points $P \subseteq \mathbb{R}^d$ and we would like to select k distinct points $\mathbf{a}_1, \ldots, \mathbf{a}_k \in P$ that maximizes

$$\langle \mathbf{a}_1, \ldots, \mathbf{a}_k \rangle := \sum_{j \in [d]} (\mathbf{a}_1)_j \cdots (\mathbf{a}_k)_j.$$

It is known that the k-chromatic variant of k-MIP is hard to approximate (see Appendix B of [KLM18]) but this is not known to be true for k-MIP itself. Our approach seems quite compatible to tackling this problem as well; in particular, if we can construct a certain (natural) generalization of our gadget for MIP, then we would immediately arrive at the inapproximability of k-MIP even for $\{0,1\}$-entries vectors. The issue in constructing this gadget is that we are now concerned about agreements of more than two vectors, which does not correspond to error-correcting codes anymore and some additional tools are needed to argue for this more general case.

It should be noted that the hardness of approximating k-MIP for $\{0,1\}$-entry vectors is equivalent to the *one-sided k-biclique* problem [Lin15], in which a bipartite graph is given and the goal is to select k vertices on the right that maximize the number of their common neighbors. The equivalence can be easily seen by viewing the coordinates as the left-hand-side vertices and the vectors as the right-hand-side vertices. The one-sided k-biclique is shown to be W[1]-hard to approximate by Lin [Lin15] who also showed a lower bound of $n^{\Omega(\sqrt{k})}$ for the problem assuming ETH. If the generalization of our gadget for k-MIP works as intended, then this lower bound can be improved to $n^{\Omega(k)}$ under ETH and even $n^{k-o(1)}$ under SETH.

The one-sided k-biclique is closely related to the (two-sided) k-biclique problem, where we are given a bipartite graph and we wish to decide whether it contains $K_{k,k}$ as a subgraph. The k-biclique problem was consider a major open problem in parameterized complexity (see e.g., [DF13]) until it was shown by Lin to be W[1]-hard [Lin15]. Nevertheless, the running time lower bound known is still not tight: currently, the best lower bound known for this problem is $n^{\Omega(\sqrt{k})}$ both for the exact version (under ETH) [Lin15] and its approximate variant (under Gap-ETH) [CCK+17]. It remains an interesting open question to close the gap between the above lower bounds and the trivial upper bound of $n^{O(k)}$. Progresses on the one-sided k-biclique problem could lead to improved lower bounds for k-biclique problem too, although several additional steps have to be taken care of.

Acknowledgements

We are grateful to Madhu Sudan for extremely helpful and informative discussion about AG codes; in particular, Madhu pointed us to [Vlă18]. We thank Bundit Laekhanukit and Or Meir for general discussions, and the Simons Institute for their wonderful workspace. Finally, we would like to thank Lijie Chen for sharing [CW19], and Orr Paradise for useful comments on an earlier draft of this manuscript.

References

[ABV01] Alexei E. Ashikhmin, Alexander Barg, and Serge G. Vladut. Linear codes with exponentially many light vectors. *J. Comb. Theory, Ser. A*, 96(2):396–399, 2001.

[AC09] Nir Ailon and Bernard Chazelle. The fast johnson–lindenstrauss transform and approximate nearest neighbors. *SIAM J. Comput.*, 39(1):302–322, 2009. Preliminary version in STOC'06.

[ACW16] Josh Alman, Timothy M. Chan, and R. Ryan Williams. Polynomial representations of threshold functions and algorithmic applications. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 467–476, 2016.

[AESW91] Pankaj K. Agarwal, Herbert Edelsbrunner, Otfried Schwarzkopf, and Emo Welzl. Euclidean minimum spanning trees and bichromatic closest pairs. *Discrete & Computational Geometry*, 6:407–422, 1991. Preliminary version in SoCG'90.

[Alp10] Ethem Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2nd edition, 2010.

[ARW17a] Amir Abboud, Aviad Rubinstein, and Ryan Williams. Distributed PCP theorems for hardness of approximation in P. *CoRR*, abs/1706.06407, 2017.

[ARW17b] Amir Abboud, Aviad Rubinstein, and Ryan Williams. Distributed PCP theorems for hardness of approximation in P. In *FOCS*, pages 25–36, 2017.

[AW15] Josh Alman and Ryan Williams. Probabilistic polynomials and hamming nearest neighbors. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 136–150, 2015.

[Ben80] Jon Louis Bentley. Multidimensional divide-and-conquer. *Commun. ACM*, 23(4):214–229, 1980.

[Ben83] Michael Ben-Or. Lower bounds for algebraic computation trees (preliminary report). In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 80–86, 1983.

[BGKM18] Arnab Bhattacharyya, Suprovat Ghoshal, Karthik C. S., and Pasin Manurangsi. Parameterized intractability of even set and shortest vector problem from gap-eth. In *ICALP*, pages 17:1–17:15, 2018.

[BL05] Yonatan Bilu and Nathan Linial. Monotone maps, sphericity and bounded second eigenvalue. *J. Comb. Theory, Ser. B*, 95(2):283–299, 2005.

[BS76] Jon Louis Bentley and Michael Ian Shamos. Divide-and-conquer in multidimensional space. In *Proceedings of the 8th Annual ACM Symposium on Theory of Computing, May 3-5, 1976, Hershey, Pennsylvania, USA*, pages 220–230, 1976.

[CCK$^+$17] Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-eth to fpt-inapproximability: Clique, dominating set, and more. In *FOCS*, pages 743–754, 2017.

[Che18a] Lijie Chen. On the hardness of approximate and exact (bichromatic) maximum inner product. In *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, pages 14:1–14:45, 2018.

[Che18b] Lijie Chen. Toward super-polynomial size lower bounds for depth-two threshold circuits. *CoRR*, abs/1805.10698, 2018.

[CIP06] Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. A duality between clause width and clause density for SAT. In *21st Annual IEEE Conference on Computational Complexity (CCC 2006), 16-20 July 2006, Prague, Czech Republic*, pages 252–260, 2006.

[CL99] Edith Cohen and David D. Lewis. Approximating matrix multiplication for pattern recognition tasks. *J. Algorithms*, 30(2):211–252, 1999.

[CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.

[CW12] Qi Cheng and Daqing Wan. A deterministic reduction for the gap minimum distance problem. *IEEE Trans. Information Theory*, 58(11):6935–6941, 2012.

[CW19] Lijie Chen and Ryan Williams. An equivalence class for orthogonal vectors. *To appear in SODA*, 2019.

[DF13] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

[DKL18] Roee David, Karthik C. S., and Bundit Laekhanukit. On the complexity of closest pair via polar-pair of point-sets. In *34th International Symposium on Computational Geometry, SoCG 2018, June 11-14, 2018, Budapest, Hungary*, pages 28:1–28:15, 2018.

[DMS03] Ilya Dumer, Daniele Micciancio, and Madhu Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Trans. Information Theory*, 49(1):22–37, 2003.

[FM86] Peter Frankl and Hiroshi Maehara. Embedding the n-cube in lower dimensions. *Eur. J. Comb.*, 7(3):221–225, 1986.

[FM88] Peter Frankl and Hiroshi Maehara. On the contact dimensions of graphs. *Discrete & Computational Geometry*, 3:89–96, 1988.

[Gal14] François Le Gall. Powers of tensors and fast matrix multiplication. In *International Symposium on Symbolic and Algebraic Computation, ISSAC'14, Kobe, Japan, July 23-25, 2014*, pages 296–303, 2014.

[Gil52] E. N. Gilbert. A comparison of signalling alphabets. *Bell System Technical Journal*, 31:504 – 522, 1952.

[GS96] Arnaldo Garcia and Henning Stichtenoth. On the asymptotic behaviour of some towers of function fields over finite fields. *Journal of Number Theory*, 61(2):248 – 273, 1996.

[GS16] Omer Gold and Micha Sharir. Dominance products and faster algorithms for high-dimensional closest pair under l_∞. *CoRR*, abs/1605.08107, 2016.

[Hen06] Tomislav Hengl. Finding the right pixel size. *Computers & Geosciences*, 32(9):1283 – 1298, 2006.

[HNS88] Klaus H. Hinrichs, Jürg Nievergelt, and Peter Schorn. Plane-sweep solves the closest pair problem elegantly. *Inf. Process. Lett.*, 26(5):255–261, 1988.

[ILLP04] Piotr Indyk, Moshe Lewenstein, Ohad Lipsky, and Ely Porat. Closest pair problems in very high dimensions. In *Automata, Languages and Programming: 31st International Colloquium, ICALP 2004, Turku, Finland, July 12-16, 2004. Proceedings*, pages 782–792, 2004.

[IM98] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 604–613, 1998.

[Ind00] Piotr Indyk. Dimensionality reduction techniques for proximity problems. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, January 9-11, 2000, San Francisco, CA, USA.*, pages 371–378, 2000.

[IP01] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. Preliminary version in CCC'99.

[IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. Preliminary version in FOCS'98.

[JL84] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

[Kle97] Jon M. Kleinberg. Two algorithms for nearest-neighbor search in high dimensions. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 599–608, 1997.

[KLM18] Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. In *STOC*, 2018. To appear.

[KLN99] Drago Krznaric, Christos Levcopoulos, and Bengt J. Nilsson. Minimum spanning trees in d dimensions. *Nord. J. Comput.*, 6(4):446–461, 1999.

[KM95] Samir Khuller and Yossi Matias. A simple randomized sieve algorithm for the closest-pair problem. *Inf. Comput.*, 118(1):34–37, 1995.

[Kop13] Swastik Kopparty. *Lecture 5: k-wise independent hashing and applications*. Lecture notes for Topics in Complexity Theory and Pseudorandomness. Rutgers University, 2013.

[KT05] Jon Kleinberg and Éva Tardos. *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[Lin15] Bingkai Lin. The parameterized complexity of k-biclique. In *SODA*, pages 605–615, 2015.

[Lue09] George S. Lueker. Improved bounds on the average length of longest common subsequences. *J. ACM*, 56(3):17:1–17:38, 2009.

[Mae85] Hiroshi Maehara. Contact patterns of equal nonoverlapping spheres. *Graphs and Combinatorics*, 1(1):271–282, 1985.

[Mae91] Hiroshi Maehara. Dispersed points and geometric embedding of complete bipartite graphs. *Discrete & Computational Geometry*, 6:57–67, 1991.

[Man89] Udi Manber. *Introduction to Algorithms: A Creative Approach*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[McD89] Colin McDiarmid. *On the method of bounded differences*. London Mathematical Society Lecture Note Series. Surveys in Combinatorics: Invited Papers at the Twelfth British Combinatorial Conference, Cambridge University Press, 1989.

[Mic14] Daniele Micciancio. Locally dense codes. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 90–97, 2014.

[MNP07] Rajeev Motwani, Assaf Naor, and Rina Panigrahy. Lower bounds on locality sensitive hashing. *SIAM J. Discrete Math.*, 21(4):930–935, 2007.

[MS77] F. J. MacWilliams and N. J. A. Sloane. *The theory of error correcting codes.* North-Holland mathematical library: v. 16. Amsterdam ; New York : North-Holland Pub. Co. ; New York : sole distributors for the U.S.A. and Canada, Elsevier/North Holland, 1977., 1977.

[OWZ14] Ryan O'Donnell, Yi Wu, and Yuan Zhou. Optimal lower bounds for locality-sensitive hashing (except when q is tiny). *TOCT*, 6(1):5:1–5:13, 2014.

[Pac80] Janos Pach. Decomposition of multiple packing and covering. *Diskrete Geometrie*, 2 Kolloq. Math. Inst. Univ. Salzburg:169–178, 1980.

[PS85] Franco P. Preparata and Michael I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag New York, Inc., New York, NY, USA, 1985.

[Rab76] Michael O. Rabin. Probabilistic algorithms. In *Proceedings of a Symposium on New Directions and Recent Results in Algorithms and Complexity, Computer Science Department, Carnegie-Mellon University, April 7-9, 1976*, pages 21–39, 1976.

[Raz17] Ilya Razenshteyn. High-dimensional similarity search and sketching: Algorithms and hardness. *PhD Thesis, MIT*, 2017.

[RRS89] Jan Reiterman, Vojtech Rödl, and Edita Sinajová. Embeddings of graphs in euclidean spaces. *Discrete & Computational Geometry*, 4:349–364, 1989.

[RS60] Irving S. Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics (SIAM)*, 8(2):300 – 304, 1960.

[Rub18] Aviad Rubinstein. Hardness of approximate nearest neighbor search. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1260–1268, 2018.

[SH75] Michael Ian Shamos and Dan Hoey. Closest-point problems. In *16th Annual Symposium on Foundations of Computer Science, Berkeley, California, USA, October 13-15, 1975*, pages 151–162, 1975.

[Sin64] Richard C. Singleton. Maximum distance q -nary codes. *IEEE Trans. Information Theory*, 10(2):116–118, 1964.

[Sti08] Henning Stichtenoth. *Algebraic Function Fields and Codes*. Springer Publishing Company, Incorporated, 2nd edition, 2008.

[Val15] Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and the closest pair problem. *J. ACM*, 62(2):13:1–13:45, 2015.

[Var57] R. R. Varshamov. Estimate of the number of signals in error correcting codes. *Dokl. Akad. Nauk SSSR*, 117:739 – 741, 1957.

[Vlă18] Serge Vlăduţ. Lattices with exponentially large kissing numbers. *arXiv preprint arXiv:1802.00886*, 2018.

[VNT07] Serge Vladut, Dmitry Nogin, and Michael Tsfasman. *Algebraic Geometric Codes: Basic Notions*. American Mathematical Society, Boston, MA, USA, 2007.

[Wil05] Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 348(2-3):357–365, 2005.

[Wil18a] Ryan Williams. On the difference between closest, furthest, and orthogonal pairs: Nearly-linear vs barely-subquadratic complexity. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1207–1215, 2018.

[Wil18b] Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proc. Int. Cong. of Math.*, volume 3, pages 3431–3472, 2018.

[WTFX07] Raymond Chi-Wing Wong, Yufei Tao, Ada Wai-Chee Fu, and Xiaokui Xiao. On efficient spatial matching. In *Proceedings of the 33rd International Conference on Very Large Data Bases, University of Vienna, Austria, September 23-27, 2007*, pages 579–590, 2007.

[Yao91] Andrew Chi-Chih Yao. Lower bounds for algebraic computation trees with integer inputs. *SIAM J. Comput.*, 20(4):655–668, 1991. Preliminary version in FOCS'89.

[Zah71] Charles T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Computers*, 20(1):68–86, 1971.

A Lower Bound on Gap Closest Pair in Edit Distance Metric

In this section we prove Theorem 1.7. The proof is almost identical to Rubinstein's [Rub18] proof for the OVH-hardness of gap-BCP in the edit distance metric and uses the following technical tool established in [Rub18].

Lemma A.1 (Rubinstein [Rub18]). *For large enough $d \in \mathbb{N}$, there is a function $\zeta : \{0,1\}^d \to \{0,1\}^{d'}$, where $d' = O(d \log d)$, such that for all $a, b \in \{0,1\}^d$ the following holds for some constant $\lambda > 0$:*

$$|\mathsf{ed}(\zeta(a), \zeta(b)) - \lambda \cdot \log d \cdot \|a - b\|_0| = o(d').$$

At a high level, ζ picks a random $O(\log d)$-bit string $s_{i,x}$ uniformly and independently for every $(i, x) \in [d] \times \{0,1\}$, and for every vector $u \in \{0,1\}^d$, replaces the i^{th} coordinate u_i by s_{i,u_i}. The claims in the lemma statement follow by the known concentration bounds on the edit distance of random strings [McD89, Lue09]. This construction is further efficiently derandomized by using $\log \log n$-wise independent strings [Kop13].

Proof of Theorem 1.7. We show that if there exists an algorithm $\mathcal{A}$ running in time $O(n^{1.5-\varepsilon})$ for some $\varepsilon > 0$ that can solve $(1+\delta)$-CP in the edit distance metric for some $\delta > 0$ over point-sets in $\{0,1\}^{d'}$, then $\mathcal{A}$ can be used to solve $(1+\delta-o(1))$-CP in the Hamming metric in time $O(n^{1.5-\varepsilon})$ over point-sets in $\{0,1\}^d$, where $d' = O(d \log d)$. Together with Theorem 7.2, this implies that OVH is false, as desired.

Let (P, α) be an instance of $(1+\delta)$-CP in the Hamming metric over point-sets in $\{0,1\}^d$. It is clear[13] from the proofs of Theorem 7.1 and Theorem 7.2 that $\alpha = \Omega(d)$. We now define an instance of $(P', \alpha' := (1+o(1)) \cdot \lambda \log d \cdot \alpha)$ of $(1+\delta-o(1))$-CP in the edit distance metric as follows. Recall the function ζ from Lemma A.1 and define the set $P' = \{\zeta(p) \mid p \in P\}$. Notice that for every pair of distinct points $p, q \in P$, we have $|\mathsf{ed}(\zeta(p), \zeta(q)) = \lambda \cdot \log d \cdot \|p - q\|_0| = o(d')$. In other words if we had a pair of distinct points p, q in P such that $\|p - q\|_0 \leq \alpha$ then, $\mathsf{ed}(\zeta(p), \zeta(q)) \leq \lambda \log d \cdot \alpha + o(d') = (1 + o(1)) \cdot \lambda \log d \cdot \alpha$ and suppose for all pairs of distinct points $p, q \in P$ we had $\|p - q\|_0 > (1+\delta) \cdot \alpha$ then $\mathsf{ed}(\zeta(p), \zeta(q)) > \lambda \log d \cdot (1+\delta) \cdot \alpha - o(d') > (1+\delta-o(1))\lambda \log d \cdot \alpha$, since $\alpha = \Omega(d)$. This completes the analysis of the completeness and soundness cases, and we can conclude that running $\mathcal{A}$ on input (P', α') solves the instance (P, α) of $(1+\delta)$-CP in the Hamming metric. □

B Covering Biclique By Isomorphic Graphs: Proof of Lemma 3.11

Below we prove Lemma 3.11. The proof strategy is similar to how the greedy approximation algorithms for the set cover problem are analyzed: we show that at each step, we can pick a graph isomorphic to G that covers at least $|E_G|/n^2$ fraction of the remaining edges of the biclique. By doing so, we guarantee that the process ends in $O(\log n) \cdot n^2/|E_G|$ steps. Note however that, there are exponential number of isomorphisms and thus we cannot simply enumerate all isomorphisms to find one that covers the desired fraction of uncovered edges. Nevertheless, it is not hard to see that we can use the method of conditional expectation to find one such isomorphism in polynomial time. This is formalized below.

[13] In fact, one can design a $2^\alpha \cdot n \log n$ time algorithm for CP in the Hamming metric, and therefore to assume OVH, we require $\alpha = \Omega(d)$.

Lemma B.1. *For any two bipartite graphs $G = (A \dot{\cup} B, E_G)$ and $H = (A \dot{\cup} B, E_H)$, there exists a side-preserving permutation $\pi : A \dot{\cup} B \to A \dot{\cup} B$ such that*

$$|E_H \cap E_{G_\pi}| \geq \frac{|E_G| \cdot |E_H|}{|A| \cdot |B|}.$$

Moreover, such a permutation π can be found (deterministically) in $O((|A| + |B|)^4)$ time.

Proof. Notice that, if we pick $\pi|_A$ and $\pi|_B$ randomly among all permutations of A and B respectively, then, for a fixed $(a, b) \in E_H$, the probability that (a, b) belongs to E_{G_π} is $\frac{|E_G|}{|A| \cdot |B|}$. Thus,

$$\mathbb{E}_\pi \left[|E_H \cap E_{G_\pi}| \right] = \frac{|E_G| \cdot |E_H|}{|A| \cdot |B|}.$$

This proves the existence part of the claim. To deterministically find such a π, we use the method of conditional expectation. Suppose $A \dot{\cup} B = \{1, \ldots, n\}$. The algorithm works as follows:

1. Let $V_{\text{assigned}} \leftarrow \emptyset$.
2. For $i = 1, \ldots, n$:
   (a) If $i \in A$, let $V_{\text{candidate}} = A \setminus V_{\text{assigned}}$. Otherwise, if $i \in B$, let $V_{\text{candidate}} = B \setminus V_{\text{assigned}}$.
   (b) For each $k \in V_{\text{candidate}}$, compute the conditional expectation:

   $$\mathbb{E}_\pi \left[|E_H \cap E_{G_\pi}| \,\middle|\, \pi(i) = k \wedge \left(\bigwedge_{j=1}^{i-1} \pi(j) = \pi^*(j) \right) \right].$$

   Let k^* be the maximizer for the above conditional expectation. We set $\pi^*(i) = k^*$.
3. Output π^*.

It is simple to see that the conditional expectation never decreases as we fill in the permutation. As a result, we must have $|E_H \cap E_{G_\pi}| \geq \frac{|E_G| \cdot |E_H|}{|A| \cdot |B|}$ as desired. Moreover, it is easy to see that the conditional expectation can be computed in time $O(|A| \cdot |B|)$ because, for each edge $(a, b) \in E_H$, we can compute the probability that $(a, b) \in E_{G_\pi}$ in $O(1)$ time. As a result, the overall running time of the algorithm is $O((|A| + |B|)^4)$. □

Finally using Lemma B.1, we prove Lemma 3.11 using the strategy outlined earlier in this section.

Proof of Lemma 3.11. We describe below an algorithm for finding $\pi_1, \ldots, \pi_k$. It works as follows.

1. Let $k \leftarrow 0$.
2. While $E_H := E_{K_{n,n}} \setminus \bigcup_{i \in [k]} E_{G_{\pi_i}}$ is non-empty, do the following:

(a) Let $k \leftarrow k + 1$.

(b) Let $H = (A \dot\cup B, E_H)$.

(c) Use the algorithm from Lemma B.1 to find π_k such that $|E_H \cap E_{G_{\pi_k}}| \geq |E_H| \cdot \frac{|E_G|}{n^2}$.

3. Output $\pi_1, \ldots, \pi_k$.

It is obvious that the permutations are all side-preserving permutations and that the union of $E_{G_{\pi_i}}$ over $i \in [k]$ is equal to $E_{K_{n,n}}$. To see that $k \leq \frac{2n^2 \ln n}{|E_G|} + 1$, observe that due to the guarantee of Lemma B.1, $|E_H|$ decreases by a multiplicative factor of (at most) $(1 - |E_G|/n^2) \leq e^{-|E_G|/n^2}$ for each permutation picked. Since the set E_H remains non-empty after $k - 1$ permutations are picked, we have $e^{-(k-1)\cdot|E_G|/n^2} \cdot n^2 \geq 1$, which implies that $k \leq 2n^2 \ln n / |E_G| + 1$ as desired. Finally, the bottleneck in the running time is Step 2c; we execute this step k times and each execution takes $O(n^4)$ time. Thus, the total running time is $O(nk) = O(n^6 \log n)$. □