Finer-Grained Reductions in Fine-Grained Hardness of Approximation

Elie Abboud
Department of Computer Science, University of Haifa, Israel

Noga Ron-Zewi
Department of Computer Science, University of Haifa, Israel

Abstract

We investigate the relation between δ and ϵ required for obtaining a $(1+\delta)$-approximation in time $N^{2-\epsilon}$ for closest pair problems under various distance metrics, and for other related problems in fine-grained complexity.

Specifically, our main result shows that if it is impossible to (exactly) solve the (bichromatic) inner product (IP) problem for vectors of dimension $c \log N$ in time $N^{2-\epsilon}$, then there is no $(1+\delta)$-approximation algorithm for (bichromatic) Euclidean Closest Pair running in time $N^{2-2\epsilon}$, where $\delta \approx (\epsilon/c)^2$ (where $\approx$ hides polylog factors). This improves on the prior result due to Chen and Williams (SODA 2019) which gave a smaller polynomial dependence of δ on ϵ, on the order of $\delta \approx (\epsilon/c)^6$. Our result implies in turn that no $(1+\delta)$-approximation algorithm exists for Euclidean closest pair for $\delta \approx \epsilon^4$, unless an algorithmic improvement for IP is obtained. This in turn is very close to the approximation guarantee of $\delta \approx \epsilon^3$ for Euclidean closest pair, given by the best known algorithm of Almam, Chan, and Williams (FOCS 2016). By known reductions, a similar result follows for a host of other related problems in fine-grained hardness of approximation.

Our reduction combines the hardness of approximation framework of Chen and Williams, together with an MA communication protocol for IP over a small alphabet, that is inspired by the MA protocol of Chen (Theory of Computing, 2020).

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases Fine-grained complexity, conditional lower bound, fine-grained reduction, Approximation algorithms, Analysis of algorithms, Computational geometry, Computational and structural complexity theory

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.7

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2311.00798`

Funding *Elie Abboud*: Research supported in part by ISF grant 735/20, and by the European Union (ERC, ECCC, 101076663). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.
Noga Ron-Zewi: Research supported in part by ISF grant 735/20, and by the European Union (ERC, ECCC, 101076663). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

1 Introduction

Traditionally, the approach to determine whether a computational problem is tractable was to find out whether it has a polynomial-time algorithm. Finding such an algorithm implies that the problem is in P, and thus it was considered efficiently computable. Otherwise, if one is interested in proving that the problem is intractable, we usually lack the tools to

51st International Colloquium on Automata, Languages, and Programming (ICALP 2024).
Editors: Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson;
Article No. 7; pp. 7:1–7:17


Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

prove lower bounds; instead one relies on hardness assumptions which allow us to prove conditional lower-bounds. In the classical theory of NP-hardness, the hardness assumption is that P $\neq$ NP, which is known to imply that no polynomial-time algorithm exists for many central computational problems.

In fine-grained complexity, one is interested in pinning down the *precise* complexity of *tractable* computational problems. In particular, a central objective in fine-grained complexity is to determine the exact exponent in the time complexity of problems already known to be in P. More concretely, given a problem with input length n known to be solvable in $t(n)$-time, is it possible to solve the problem in time $t(n)^{1-\epsilon}$ for some $\epsilon > 0$? This is motivated by the fact that despite rigorous study of many central computational problems in P, we have failed to improve on the running time of their best-known algorithms (see for example the survey [21] for a list of such problems). This motivates the question of whether there is an inherent difficulty in the problem that prevents us from finding faster algorithms.

Once more, we typically lack the tools to prove lower bounds, and we thus instead rely on hardness assumptions to obtain conditional lower bounds for problems in P. One popular such conjecture has been the *Strong Exponential Time Hypothesis* (SETH), which postulates that for any $\epsilon > 0$, there exists an integer $k = k(\epsilon)$ so that it is impossible to solve k-SAT on n variables in time $2^{(1-\epsilon)n}$ [13].

Another popular conjecture is the *Orthogonal Vector Conjecture* (OVC) which in the low-dimensional regime posits that for any $\epsilon > 0$, there exists a $c_{\text{ov}} = c_{\text{ov}}(\epsilon)$ such that given a pair of sets $A, B \subseteq \{0,1\}^d$ of cardinality N each and of dimension $d = c_{\text{ov}} \cdot \log N$, it is impossible to determine whether there exists a pair $(a, b) \in A \times B$ satisfying that $\langle a, b\rangle = 0$ in $N^{2-\epsilon}$ time [11]. It is known that SETH implies OVC [20], and so OVC is at least as plausible as SETH. In terms of algorithms, it is known how to solve the OV problem in time $N^{2-\epsilon}$ with $c = \exp(1/\epsilon)$ [3, 7], which implies that $c_{\text{ov}} \geq \exp(1/\epsilon)$.

A related assumption is the *inner product* (IP) *assumption* which postulates that for any $\epsilon > 0$, there exists a $c_{\text{ip}} = c_{\text{ip}}(\epsilon)$ such that given a pair of sets $A, B \subseteq \{0,1\}^d$ of cardinality N each and of dimension $d = c_{\text{ip}} \cdot \log N$, and an integer $\sigma \in \{0, 1, \ldots, d\}$, it is impossible to determine whether there exists a pair $(a, b) \in A \times B$ satisfying that $\langle a, b\rangle = \sigma$ in $N^{2-\epsilon}$ time. Once more, since the OV problem is a special case of the IP problem, the IP assumption is at least as plausible as OVC[1]. Indeed, the best known algorithms for the IP problem are only able to solve this problem in time $N^{2-\epsilon}$ with $c \approx 1/\epsilon$ [5], and this only imposes that $c_{\text{ip}} \gtrapprox 1/\epsilon$.[2]

In recent years, there has been a flurry of work showing fine-grained lower bounds for many central computational problems in P, based on the above assumptions. A main challenge in showing such fine-grained lower bounds based on these assumptions is that one must carefully design the reductions so that they run fast enough as not to supersede the lower bound assumptions.

One fundamental problem for which such fine-grained reductions were shown is the *Closest Pair* (CP) *problem*. In this problem, given a distance metric $\text{dist} : \{0,1\}^d \times \{0,1\}^d \to \mathbb{R}^+$, and given a pair of sets $A, B \subseteq \{0,1\}^d$, the goal is to find a pair $(a, b) \in A \times B$ which minimizes $\text{dist}(a, b)$. This problem was studied for various metrics such as Hamming, ℓ_p, and edit distance, and it has many applications, for example in computational geometry, geographic information systems [12], clustering [24, 6], and matching problems [23], to name a few. For concreteness, in what follows we restrict our attention only to the Euclidean ℓ_2 metric, though many of the results we mention hold also for other metrics.

---

[1] The IP assumption is at least as plausible as OVC if we allow an arbitrary dependence of c_{ip} on epsilon.
[2] We use $\approx$, $\gtrsim$, $\lesssim$ to hide polylog factors.

One can naïvely solve the (Euclidean) Closest Pair problem in $O(N^2 d)$ time. On the other hand, algorithms have been developed which solve the problem in time $\approx N^{O(c)}$ in the low-dimensional regime $d = c \log N$, [16, 14]; Thus, a truly sub-quadratic algorithm is only known for smaller values of c. On the other hand, in [5] it was shown that assuming OVC, for any $\epsilon > 0$ there exists $c = c(\epsilon)$ so that no algorithm can solve this problem in time $N^{2-\epsilon}$.

1.1 Fine-grained hardness of approximation

Given the above state of affairs, it is natural to ask whether relaxing the requirements and settling for an approximate "close-enough" answer can help in designing faster fine-grained algorithms. For example, it is known that for the (Euclidean) CP problem, one can obtain a $(1+\delta)$-approximation with running time $N^{2-\epsilon}$ for $\delta \approx \epsilon^3$ (for any dimension $d \leq N^{1-\epsilon}$) [4], which is much faster than the best-known exact algorithm.

In terms of impossibility results, known fine-grained reductions can typically be adapted to the approximate setting, based on appropriate *gap assumptions*, such as Gap-SETH.[3] In the theory of NP-hardness, it is often possible to base hardness of gap-problems on hardness of exact problems using PCPs. However, a major barrier in applying this approach in the fine-grained setting (for example for the purpose of reducing SETH to Gap-SETH) is the large (super-constant) blow-up in the length of existing PCPs, which translates into a large (super-constant) blow-up in the number of variables n in the reduction.

Nevertheless, in a recent breakthrough, Abboud, Rubinstein, and Williams [2] have shown how to utilize PCP machinery (specifically, the sumcheck protocol) for showing fine-grained hardness of approximation results based on *non-gap assumptions*. Since then, many works have utilized this framework for showing fine-grained hardness of approximation results for many central problems in P, based on non-gap assumptions such as SETH or OVC (see the recent surveys [18, 10] for a description of this line of work).

In particular, for the CP problem, Rubinstein [17] has shown that assuming OVC, for any $\epsilon > 0$ there exists $\delta = \delta(\epsilon)$ such that there is no $(1+\delta)$-approximation algorithm for (Euclidean) CP running in time $N^{2-\epsilon}$. This rules out truly sub-quadratic approximation algorithms, running, say, in time $f(\delta) \cdot N^{1.99}$. However, the obtained dependence of δ on ϵ is far from optimal, specifically $\delta = \exp(-c_{\text{ov}}/\epsilon)$, where $c_{\text{ov}} = c_{\text{ov}}(\epsilon) \geq \exp(1/\epsilon)$ is the constant guaranteed by the OVC conjecture.

In a follow-up work, Chen and Williams [9] have shown an improved hardness of approximation result for CP in which δ only depends polynomially on ϵ. Specifically, they showed that if the IP assumption holds, then for any $\epsilon > 0$ there is no $(1+\delta)$-approximation algorithm for (Euclidean) CP running in time $N^{2-\epsilon}$, where $\delta = \text{poly}(\epsilon/c_{\text{ip}})$ and $c_{\text{ip}} = c_{\text{ip}}(\epsilon) \gtrapprox 1/\epsilon$ is the constant guaranteed by the IP assumption. However, the obtained dependence of δ on ϵ was still quite small, on the order of $\delta \approx (\epsilon/c_{\text{ip}})^6 \lessapprox \epsilon^{12}$. This is still quite far from the dependence obtained by the best known approximation algorithm for CP which gives an $(1+\delta)$-approximation in time $N^{2-\epsilon}$ for $\delta \approx \epsilon^3$.

In this work, we investigate the question of whether the dependence of δ on ϵ can even be further improved, potentially to match the best known approximation algorithm.

[3] The Gap-SETH assumption asserts that for any $\epsilon > 0$, there are k and $\delta > 0$, so that no $2^{(1-\epsilon)n}$-time algorithm can, given a k-CNF on n variables, distinguish between the case that it is satisfiable, and the case that any assignment satisfies at most an $(1-\delta)$-fraction of its clauses.

1.2 Our results

Recall that by the discussion above, the best known approximation algorithm for (Euclidean) CP gives an $(1+\delta)$-approximation in time $N^{2-\epsilon}$ for $\delta \approx \epsilon^3$, while the best-known hardness of approximation result shows that if the IP assumption holds, then no $(1+\delta)$-approximation algorithm running in time $N^{2-\epsilon}$ exists for $\delta \approx (\epsilon/c_{\text{ip}})^6 \lessapprox \epsilon^{12}$, where $c_{\text{ip}} = c_{\text{ip}}(\epsilon) \gtrapprox 1/\epsilon$ is the constant guaranteed by the IP assumption. Thus there remains a large polynomial gap between the upper and lower bounds, and our main result narrows this gap.

▶ **Theorem 1.1.** *Suppose that the IP assumption holds, i.e., for any $\epsilon' > 0$, there exists a $c_{\text{ip}} = c_{\text{ip}}(\epsilon')$ such that given a pair of sets $A, B \subseteq \{0,1\}^d$ of cardinality N each and of dimension $d = c_{ip}(\epsilon') \cdot logN$, and an integer $\sigma \in \{0, 1, \ldots, d\}$, it is impossible to find a pair $(a, b) \in A \times B$ satisfying that $\langle a, b\rangle = \sigma$ in $N^{2-\epsilon'}$ time.*

Then for any $\epsilon > 0$, there is $\delta = \tilde{\Theta}((\frac{\epsilon}{c_{\text{ip}}(\epsilon/2)})^2)$, so that any algorithm running in time $N^{2-\epsilon}$ cannot $(1+\delta)$-approximate Euclidean CP.

Recall that it is known how to solve the IP problem in time $N^{2-\epsilon}$ for dimension $d = c \log N$ with $c \approx 1/\epsilon$, and so it must hold that $c_{\text{ip}} \gtrapprox 1/\epsilon$. If we assume that $c_{\text{ip}} \approx 1/\epsilon$, then the above theorem gives a dependence of δ on ϵ of the form $\delta \approx \epsilon^4$, which is very close to the dependence of $\delta \approx \epsilon^3$ given by the best known algorithm. Moreover, improving the dependence in the above theorem to $\delta \approx \frac{\epsilon}{c_{\text{ip}}}$ would imply an algorithmic improvement on the IP problem. We leave the question of determining the exact dependence of δ on ϵ as an interesting open problem for future research.

By known reductions, the above theorem gives a similar improvement for a host of other problems in fine-grained hardness of approximation such as closest pair with respect to other metrics such as Hamming, ℓ_p-norm for any constant $p > 0$, and edit distance, Furthest Pair and approximate nearest neighbor in these metrics, and additive approximations to Max-IP and Min-IP, see Appendix A for more details.

Finally, we remark that the above theorem also holds under OVC (or SETH), but is less meaningful, since as discussed above, for the OV problem we have that $c_{\text{ov}} \geq \exp(1/\epsilon)$.

1.3 Proof overview

Next we give an overview of our proof method, and how it improves on prior work. To this end, we first describe the general framework presented in [2] for obtaining fine-grained hardness of approximation results based on MA communication protocols. Then we discuss the work of Rubinstein [17] who relied on this framework to give the first fine-grained hardness of approximation result for CP, albeit with an exponential dependence of δ on ϵ, and the work of Chen and Williams [9] who improved this dependence to polynomial. Following this, we turn to discuss our proof method that obtains a tighter polynomial relation.

Fine-grained hardness of approximation via MA communication [2]. In a Merlin-Arthur (MA) communication protocol for a function $f : \{0,1\}^d \times \{0,1\}^d \to \{0,1\}$, two players Alice and Bob wish to compute $f(a, b)$, where Alice is given as input only $a \in \{0,1\}^d$, and Bob is given as input only $b \in \{0,1\}^d$. To this end, Alice and Bob engage in a randomized (public coin) communication protocol, where their goal is to use as little communication as possible. To aid them with this task, there is also a (potentially malicious) prover Merlin who sees Alice's and Bob's inputs, and before any communication begins Merlin sends Alice a short message m, which can be thought of as a "proof" or "advice". The requirement is that if

$f(a,b)=1$, then there must exist some message m from Merlin on which Alice accepts with probability 1. Otherwise, if $f(a,b)=0$, then for any possible message $\tilde{m}$ from Merlin, Alice accepts with probability at most $\frac{1}{2}$ on $\tilde{m}$.[4]

In [2], it was shown that an efficient MA communication protocol for *set disjointness*[5] implies a fine-grained reduction from OV to an approximate version of Max-IP in which given two sets $A,B\subseteq\{0,1\}^d$ of cardinality N each, the goal is to output a number sufficiently close to $M:=\max_{a\in A,b\in B}\langle a,b\rangle$.

To see how a reduction as above can be constructed, suppose that there exists an MA communication protocol for set disjointness with Merlin's message length L, communication complexity cc between Alice and Bob, and randomness complexity R. Suppose furthermore that we are given an instance $A,B\subseteq\{0,1\}^d$ of OV, where $|A|=|B|=N$. Then for each possible Merlin's message $m\in\{0,1\}^L$, we construct an instance $A_m,B_m\subseteq\{0,1\}^{2^{\mathrm{cc}+R}}$ of Max-IP, where $|A_m|=|B_m|=N$.

Fix $m\in\{0,1\}^L$. Then the set A_m is obtained from A by mapping each element $a\in A$ to a binary vector a_m that contains an entry for each possible transcript $\Gamma\in\{0,1\}^{\mathrm{cc}}$ and randomness string $r\in\{0,1\}^R$ (so a_m has length $2^{\mathrm{cc}+R}$), and whose (Γ,r)-entry equals 1 if and only if Γ is consistent with r and a, and Alice accepts on input a, randomness string r, and transcript Γ. The set B_m is obtained analogously from B.

Then the main observation is that for some $m\in\{0,1\}^L$, we have that the (Γ,r)-entry of both a_m and b_m equals 1 if and only if Alice accepts on Merlin's message m, inputs a and b, and randomness string r. Consequently, if there exists $(a,b)\in A\times B$ so that $\langle a,b\rangle=0$ (i.e., $f(a,b)=1$), then there exists a Merlin's message m on which Alice accepts with probability 1 on inputs a and b, and consequently we have that the corresponding vectors $(a_m,b_m)\in A_m\times B_m$ satisfy that $\langle a_m,b_m\rangle=2^R$. On the other hand, if $\langle a,b\rangle\neq 0$ (i.e., $f(a,b)=0$) for any $(a,b)\in A\times B$, then for any Merlin's message $\tilde{m}$, and on any inputs $(a,b)\in A\times B$, Alice accepts with probability at most $\frac{1}{2}$, and so $\langle a_{\tilde{m}},b_{\tilde{m}}\rangle\leq\frac{1}{2}\cdot 2^R$ for any pair $(a_{\tilde{m}},b_{\tilde{m}})\in A_{\tilde{m}}\times B_{\tilde{m}}$. This gives the desired gap, showing that OV reduces to 2^L instances of approximate Max-IP.

To obtain a fine-grained reduction, one must make sure that cc, R and L are not too large, so that the total construction time of the reduction is at most N^ϵ. To achieve this, one can use the MA communication protocol of Aaronson and Wigderson [1] for set disjointness in which all these quantities are upper bounded by $\approx\sqrt{d}$.

For $d=c\log N$, this gives that 2^{cc}, 2^R, and 2^L are all upper bounded by $2^{\tilde{O}(\sqrt{\log N})}\ll N^\epsilon$, and so the reduction can be constructed in time N^ϵ. Next we describe the MA communication protocol of [1], as hardness of approximation results for CP (including ours) crucially rely on its properties.

MA communication protocol for set disjointness [1]. The MA communication protocol for set disjointness of Aaronson and Wigderson [1] relies on the influential sumcheck protocol of [15], and it proceeds as follows.

Let $a\in\{0,1\}^d$ be Alice's input. Slightly abusing notation, we view a as a $\sqrt{d}\times\sqrt{d}$ binary matrix in the natural way, and we let $\hat{a}$ denote the $p\times\sqrt{d}$ matrix obtained by encoding each column of a with a systematic Reed-Solomon code $\mathrm{RS}_{\sqrt{d},p}:\mathbb{F}_p^{\sqrt{d}}\to\mathbb{F}_p^p$ of degree $\sqrt{d}$ over a prime field of size $p\approx 4\sqrt{d}$.[6] Let $\hat{b}$ be defined analogously.

[4] The accept probability can be increased by executing the communication phase between Alice and Bob independently for multiple times and accepting if and only if all invocations accept.

[5] Recall that *set disjointness* is the function $\mathrm{disj}:\{0,1\}^d\times\{0,1\}^d\to\{0,1\}$ which satisfies that $\mathrm{disj}(a,b)=1$ if and only if the supports of a and b are disjoint, i.e., if $\langle a,b\rangle=0$.

[6] The systematic Reed-Solomon code $\mathrm{RS}_{d,p}:\mathbb{F}_p^d\to\mathbb{F}_p^p$ of degree d over a prime field of size $p>d$ is a linear

In the protocol, Merlin first computes the pointwise product $\hat{a} \star \hat{b} \in \mathbb{F}_p^{p \times \sqrt{d}}$, and then sends Alice the sum $m \in \mathbb{F}_p^p$ of the columns of $\hat{a} \star \hat{b}$ (where arithemtic is performed mod p). Alice first checks that m is a codeword of $\mathrm{RS}_{2\sqrt{d},p}$, and that the first $\sqrt{d}$ entries of m are all zero, otherwise she rejects and aborts. Then Alice and Bob jointly sample a random index $i \in [p]$, Bob sends Alice the i'th row of $\hat{b}$, Alice computes its inner product with the i'th row of $\hat{a}$, and accepts if and only if this product equals $m(i)$ (where once more, arithmetic is performed mod p).

To see that the protocol is complete, note first that if a and b are disjoint, then $a \star b$ is the all-zero matrix. Consequently, by the systematic property of the Reed-Solomon encoding, the first $\sqrt{d}$ rows of $\hat{a} \star \hat{b}$ are also identically zero, which implies in turn that the first $\sqrt{d}$ entries of m are identically zero. Furthermore, since the product of two polynomials of degree at most $\sqrt{d}$ is a polynomial of degree at most $2\sqrt{d}$, it follows that m is a codeword of $\mathrm{RS}_{2\sqrt{d},p}$. Thus, both Alice's checks will clearly pass. It can also be verified that by construction, the inner product of the i'th rows of $\hat{a}$ and $\hat{b}$ equals $m(i)$, and so Alice accepts with probability 1.

To show soundness, suppose that a and b intersect, and let $\tilde{m}$ denote Merlin's message. We may assume that $\tilde{m}$ is a codeword of $\mathrm{RS}_{2\sqrt{d},p}$, and that the first $\sqrt{d}$ entries of $\tilde{m}$ are all zero, since otherwise Alice clearly rejects. But on the other hand, since a and b intersect, then $a \star b$ has a 1-entry, say in the j-th row, and since $p > \sqrt{d}$, the sum of entries in the j'th row of $a \star b$ is non-zero mod p, which implies in turn that $m(j) \neq 0$. Thus, we conclude that $\tilde{m}$ and m are distinct codewords of $\mathrm{RS}_{2\sqrt{d},p}$ – a code of distance at least $\frac{p}{2}$ – and so they must differ by at least $\frac{1}{2}$ of their entries. But this implies in turn that with probability at least $\frac{1}{2}$ over the choice of i, it holds that $\tilde{m}(i) \neq m(i)$, in which case the inner product of the i'th row of $\hat{a}$ and $\hat{b}$ will be different than $\tilde{m}(i)$, which will cause Alice to reject.

Finally, it can also be verified that in this protocol, cc, R, and L are all upper bounded by $\approx \sqrt{d}$.

Hardness of approximation for CP with exponential dependence [17]. In [17], Rubinstein utilized the above framework to show fine-grained hardness of approximation for CP. The starting point of [17] is a simple linear-time reduction from δ-additive approximation for Max-IP[7] to an $(1+\Theta(\delta))$-approximation for (Euclidean) CP. Thus, to show that no algorithm can find an $(1+\Theta(\delta))$-approximation for (Euclidean) CP in time $N^{2-\epsilon}$, it suffices to show that no algorithm can find a δ-additive approximation for Max-IP in time $N^{2-\epsilon}$.

The [2] framework discussed above generates instances of Max-IP of dimension $2^{\mathrm{cc}+R}$ and additive gap of $\frac{1}{2} \cdot 2^R$, which gives $\delta := \Theta(2^{-\mathrm{cc}})$. However, the MA protocol of [1] described above only gives cc $\approx \sqrt{d}$ which is super-constant for a super-constant dimension d, and consequently only yields a *sub-constant* δ.

To deal with this, [17] first utilized the fact (previously utilized also in [2]) that the [1] protocol described above works equally well on skewed matrices of dimensions $\frac{d}{T} \times T$, in which case we have that $L = \frac{d}{T} \cdot \log p$ and $\mathrm{cc} = T \cdot \log p$. Thus, assuming $d = c \log N$, to achieve $2^L \leq N^\epsilon$, one can set $T = \frac{c}{\epsilon} \cdot \log p$, which gives in turn $\mathrm{cc} = \frac{c}{\epsilon} \cdot \log^2 p$.

---

map, defined as follows. To encode a message $m = (m(0), \ldots, m(d-1)) \in \mathbb{F}_p^d$, one finds the (unique) degree $d-1$ polynomial $P_m(X) \in \mathbb{F}_p[X]$ which satisfies that $P_m(i) = m(i)$ for any $i = 0, \ldots, d-1$, and lets $\mathrm{RS}_{d,p}(m) = (P_m(0), \ldots, P_m(p-1))$. The code is called *systematic* since the message is a prefix of its encodings. The code has distance at least $p-d+1$ since any pair of distinct degree $d-1$ polynomials can agree on at most $d-1$ points.

[7] In a *δ-additive approximation* for Max-IP, given $A, B \subseteq \{0,1\}^d$ of cardinality N each, the goal is to output a number in $[M - \delta \cdot d, M]$, where $M := \max_{a \in A, b \in B} \langle a, b \rangle$.

However, this is still not quite enough since the MA protocol of [1] requires setting $p > \sqrt{d}$ because of the use of Reed-Solomon codes that are only defined over a large alphabet, and consequently the communication complexity is still super-constant. However, the main observation in [1] is that the protocol can actually be executed using any error-correcting code with a *multiplication property*[8]. Relying on this observation, Rubinstein replaced the Reed-Solomon codes in the protocol of [1] with algebraic-geometric (AG) codes that satisfy the multiplication property over a constant-size alphabet. This reduced the communication complexity to $\approx c/\epsilon$, yielding in turn an approximation factor of $\delta = 2^{-\tilde{\Theta}(c/\epsilon)}$.

Polynomial dependence [9]. While [17] gave the first non-trivial hardness of approximation result for CP, a downside of this result was that the approximation factor δ depended exponentially on the running time parameter ϵ. In the follow-up work [9], Chen and Williams showed how to reduce this dependence to just polynomial.

The main observation of Chen and Williams was that instead of thinking of the output of the MA protocol of [17] as being just accept or reject, one can view the output as being *short vectors* $a', b' \in \mathbb{F}_p^T$ (namely, the i'th row of $\hat{a}, \hat{b}$, respectively), and $\sigma' \in \{0, 1, \ldots, T \cdot p^2\}$ (namely, the i'th entry of Merlin's message m), where a' only depends on Alice's input a and the randomness string, b' only depends on Bob's input b and the randomness string, and σ' only depends on Merlin's message and the randomness string. The requirement then is that if $\langle a, b\rangle = 0$ for some $(a, b) \in A \times B$, then for some Merlin's message m, $\langle a', b'\rangle = \sigma'$ with probability 1, while if $\langle a, b\rangle \neq 0$ for any $(a, b) \in A \times B$, then for any Merlin's message $\tilde{m}$, then $\langle a', b'\rangle \neq \sigma'$ with probability at least $\frac{1}{2}$.

Chen and Williams then suggested to create an instance A_m, B_m for any Merlin's message m, where the set A_m is obtained from A by simply mapping each element $a \in A$ to a vector $a_m \in \mathbb{F}_p^{T \times 2^R}$ that is the concatenation of all possible output vectors a' for all possible randomness strings, and analogously for B_m. The advantage is that now the dimension of the vectors in A_m and B_m is much shorter than in [17]. However, a disadvantage is that now the alphabet is not binary anymore, and an even more serious problem is that the soundness guarantee is only that $\langle a', b'\rangle \neq \sigma'$, so the reduction does not seem to produce any gap.

To deal with these issues, Chen and Williams use an *encoding lemma* which gives mappings g, h and a value Γ, where g, h, and Γ only depend on p and T, so that $g(a', \sigma')$ and $h(b', \sigma')$ are binary vectors of length $\text{poly}(p, T)$ satisfying that if $\langle a', b'\rangle = \sigma'$ then $\langle g(a', \sigma'), h(b', \sigma')\rangle = \Gamma$, while if $\langle a', b'\rangle \neq \sigma'$ then $\langle g(a', \sigma'), h(b', \sigma')\rangle < \Gamma$ (see Lemma 4.2 for a formal statement). This produces the desired additive gap, on the order of $\Omega(2^R)$. Since the encoding lemma increases the dimension of the vectors only by a factor of $\text{poly}(p, T)$, this yields an approximation factor of $\delta = \frac{1}{\text{poly}(p,T)} = \text{poly}(\frac{\epsilon}{c})$.

We note that a delicate issue that should be dealt with in the reduction is that the encoding lemma works over the integers, while the protocol works over finite fields, and in particular, over non-prime fields, as AG codes are only known to exist over non-prime fields. Additionally, Chen and Williams show that the reduction works equally well when using the IP problem instead of OV as its starting point, and using an MA communication protocol for IP similar to that of [1]. This can potentially allow for a smaller value of c as it is only known how to solve (exact) IP in time $N^{2-\epsilon}$ up to a dimension of $c \log N$ for $c \approx 1/\epsilon$.

[8] Informally, we say that a linear code $C : \mathbb{F}^k \to \mathbb{F}^n$ has a *multiplication property* if the set $\text{span}\{C(m) \star C(m') \mid m, m' \in \mathbb{F}^k\}$ has sufficiently large distance.

This work – tighter polynomial dependence. While [9] obtained a polynomial dependence of δ on ϵ, the dependence was quite small, on the order of $\delta \approx (\frac{\epsilon}{c})^6$, and in the current work we show how to improve the dependence to $\delta \approx (\frac{\epsilon}{c})^2$.

To this end, we first observe that one reason for the small polynomial dependence obtained in [9] was the large polynomial dependence of the dimension of the resulting vectors in the encoding lemma on the alphabet size p. While in the protocol of [17] the field size p can be made constant using AG codes, the field still needs to be of characteristic at least T, since otherwise the sum of entries in a non-zero row of $a \star b$ may sum to zero over $\mathbb{F}_p$, and consequently the soundness analysis will not go through.

To reduce the alphabet size, we first design a new MA protocol in which the alphabet size is only *polylogarithmic* in T (see Theorem 3.1). This protocol is inspired by the MA protocol of [8] for IP which achieved communication complexity $O(\sqrt{d \log d \log \log d})$, improving on the communication complexity of $O(\sqrt{d} \log d)$ of [1]. In a nutshell, Chen's idea was to execute the original MA protocol of [1] multiple times over different small prime fields, hoping that if $\langle a, b \rangle \neq 0$, then $\langle a, b \rangle$ is also non-zero modulo many of the primes, and so the protocol will be executed correctly. Chen showed that this is indeed possible to achieve using $O(\log d)$ distinct primes of cardinality at most polylog(d) each.

We observe that for skewed matrices of dimensions $\frac{d}{T} \times T$, it in fact suffices to execute the protocol with $O(\log T)$ distinct primes of cardinality at most polylog(T) each. While this choice does not necessarily guarantee the property above that if $\langle a, b \rangle \neq 0$, then $\langle a, b \rangle$ is also non-zero modulo many of the primes, this turns out to still suffice for a correct execution of the protocol.

We then further observe that such a protocol can be used in the framework of [9] to obtain an improved hardness of approximation result for Max-IP. Once more, a delicate issue is how to use the encoding lemma in the presence of many different non-prime fields.

To the best of our knowledge, this is the first use of the techniques underlying the improved MA protocol of [8] for showing a fine-grained hardness of approximation result.[9]

Paper organization. The rest of the paper is organized as follows. We begin in Section 2 below with the required notation and terminology with respect to fine-grained complexity problems and error-correcting codes. Then in Section 3 we present our improved MA protocol over a small alphabet, while in Section 4 we show how to use this protocol for obtaining an improved reduction from IP to approximate Max-IP. Finally, in Appendix A we show implications of our latter result to showing hardness of approximation results for closest pair, as well as other related problems in fine-grained complexity.

2 Preliminaries

We start by setting some general notation. For a positive integer d, we let $[d] := \{1, 2, \ldots, d\}$. For convenience, we often view a vector $a \in \Sigma^d$ as a function $a : [d] \to \Sigma$, and we let $a(i)$ denote the i-th entry of a. For a pair of vectors $a, b \in \mathbb{N}^d$, we let $\langle a, b \rangle := \sum_{i=1}^{d} a(i) \cdot b(i)$ denote their inner product, and we let $a \star b \in \mathbb{N}^d$ denote their pointwise product, given by $(a \star b)(i) = a(i) \cdot b(i)$ for $i \in [d]$. For $a, b \in \Sigma^d$, we let $\Delta(a, b) := |\{i \in [d] \mid a(i) \neq b(i)\}|$ denote their Hamming distance. For an $n \times k$ matrix A and $i \in [n]$ ($j \in [k]$, respectively), we let $\mathrm{row}_i(A)$ ($\mathrm{col}_j(A)$, respectively) denote the i-th row (j-th column, respectively) of A.

[9] The paper [8] contains various hardness of approximation results for Max-IP, as well as the improved MA protocol for IP. To the best of our knowledge, the improved MA protocol presented in this paper was not used in this paper or in any subsequent work as the basis for hardness of approximation results.

2.1 Problems in fine-grained complexity

Below we list the main fine-grained problems that we will be concerned with in this paper.

▶ **Definition 2.1** (Inner Product (IP)). *In the inner product* $\mathrm{IP}_{N,d}$ *problem, given two sets* $A, B \subseteq \{0,1\}^d$ *of cardinality* N *each, and an integer* $\sigma \in \{0,1,\ldots,d\}$*, the goal is to determine whether there exists a pair* $(a,b) \in A \times B$ *satisfying that* $\langle a,b\rangle = \sigma$*.*

▶ **Definition 2.2** (Maximum Inner Product (Max-IP)). *In the maximum inner product* Max-$\mathrm{IP}_{N,d}$ *problem, given two sets* $A, B \subseteq \{0,1\}^d$ *of cardinality* N *each, the goal is to compute* $M := \max_{a\in A, b\in B} \langle a,b\rangle$ *.*

For the approximate version of Max-IP, defined next, we will consider the less standard *additive* approximation version that will be useful for obtaining hardness of approximation for the closest pair problem.

▶ **Definition 2.3** (Approximate Maximum Inner Product (Apx-Max-IP)). *Let* $\delta > 0$ *be a parameter. In the (additive) approximate maximum inner product* δ-Apx-Max-$\mathrm{IP}_{N,d}$ *problem, given two sets* $A, B \subseteq \{0,1\}^d$ *of cardinality* N *each, the goal is to output a number in* $[M - \delta \cdot d, M]$*, where* $M := \max_{a\in A, b\in B} \langle a,b\rangle$*.*

▶ **Definition 2.4** (Closest Pair (CP)). *Let* $\mathrm{dist} : \{0,1\}^d \times \{0,1\}^d \to \mathbb{R}^+$ *be a distance function. In the closest pair* $\mathrm{CP}_{N,d,\mathrm{dist}}$ *problem, given two sets* $A, B \subseteq \{0,1\}^d$ *of cardinality* N *each, the goal is to compute* $M := \min_{a\in A, b\in B} \mathrm{dist}(a,b)$*.*

▶ **Definition 2.5** (Approximate Closest Pair (Apx-CP)). *Let* $\mathrm{dist} : \{0,1\}^d \times \{0,1\}^d \to \mathbb{R}^+$ *be a distance function, and let* $\delta > 0$ *be a parameter. In the approximate closest pair* δ-Apx-$\mathrm{CP}_{N,d,\mathrm{dist}}$ *problem, given two sets* $A, B \subseteq \{0,1\}^d$ *of cardinality* N *each, the goal is to output a number in* $[M, (1+\delta)M]$*, where* $M := \min_{a\in A, b\in B} \mathrm{dist}(a,b)$*.*

2.2 Error-correcting codes

Our reduction will make use of error-correcting codes. In what follows, we first present some general notation and terminology with respect to error-correcting codes, and then describe the kind of codes we shall use for our reduction.

Let Σ be a finite alphabet, and k, n be positive integers (the message length and the codeword length, respectively). An (error-correcting) code is an injective map $C : \Sigma^k \to \Sigma^n$. The elements in the domain of C are called messages, and the elements in the image of C are called codewords. We say that C is systematic if the message is a prefix of the corresponding codeword, i.e., for every $x \in \Sigma^k$ there exists $z \in \Sigma^{n-k}$ such that $C(x) = (x,z)$. The rate of a code $C : \Sigma^k \to \Sigma^n$ is the ratio $\rho := \frac{k}{n}$. The relative distance $\mathrm{dist}(C)$ of C is the maximum $\delta > 0$ such that for every pair of distinct messages $x, y \in \Sigma^k$ it holds that $\Delta(C(x), C(y)) \geq \delta$.

If $\Sigma = \mathbb{F}$ for some finite field $\mathbb{F}$, and C is a linear map between the vector spaces $\mathbb{F}^k$ and $\mathbb{F}^n$ then we say that C is linear. The generating matrix of a linear code $C : \mathbb{F}^k \to \mathbb{F}^n$ is a matrix $G \in \mathbb{F}^{n\times k}$ such that $C(x) = G \cdot x$ for any $x \in \mathbb{F}^k$. We say that a linear code C is explicit if G can be generated in time $\mathrm{poly}(n)$.

For our reduction, we shall require linear codes over a small (constant-size, independent of the codeword length) alphabet, satisfying the *multiplication property*, which informally says that all pointwise products of pairs of codewords span a code of large distance. Such codes can be obtained from the AG codes of [19] (see also [17, Theorem 2.4]).

▶ **Theorem 2.6** ([19]; [17], Theorem 2.4). *There exists a constant integer* p_0 *so that for any prime* $p \geq p_0$*, there exist two explicit code families* $\mathcal{C} = \{C_k\}_{k\in\mathbb{N}}$ *and* $\mathcal{C}_\star = \{(C_\star)_k\}_{k\in\mathbb{N}}$ *so that the following hold for any* $k \in \mathbb{N}$*:*

- *$C_k, (C_\star)_k$ are systematic linear codes over $\mathbb{F}_{p^2}$ of relative distance at least* 0.1 *and rate at least* 0.1.
- *C_k has message length k.*
- *For any $x, y \in (\mathbb{F}_{p^2})^k$, $C_k(x) \star C_k(y)$ is a codeword of $(C_\star)_k$.*

3 MA protocol for IP over a small alphabet

In this section, we will provide an MA protocol for IP over a small alphabet. The protocol will be later used in Section 4 below to show a reduction from IP to Apx-Max-IP.

▶ **Theorem 3.1** (MA Protocol for IP over a small alphabet). *For any sufficiently large integer T, there is an integer $q = O(\log^2 T)$, so that for any integer d which is a multiple of T there is an* MA *Protocol which satisfies the following:*

1. *Alice is given as input a vector $a \in \{0,1\}^d$ and an integer $\sigma \in \{0,1,\ldots,d\}$, Bob is given as input a vector $b \in \{0,1\}^d$, and Merlin is given as input a, b, and σ.*
2. *Merlin sends Alice a message m of (bit) length $L = O(\frac{d}{T} \cdot \log^2 T)$. Alice reads Merlin's message, and based on this message and σ, decides whether to reject and abort, or continue.*
3. *Alice and Bob sample a joint random string r of (bit) length $R = \log(\frac{d}{T}) + \log\log T + O(1)$.*
4. *Alice outputs a string $a' \in \{0,1,\ldots,q\}^T$ and an integer $\sigma' \in \{0,1,\ldots,T \cdot q^2\}$, where a' only depends on Alice's input a and the randomness string r, and σ' only depends on Merlin's message m and r, and Bob outputs a string $b' \in \{0,1,\ldots,q\}^T$, which only depends on Bob's input b and r, so that the following hold:*
   - *(Completeness) If $\langle a, b\rangle = \sigma$, then on Merlin's message m, Alice and Bob output a', b', and σ' so that $\langle a', b'\rangle = \sigma'$ with probability* 1.
   - *(Soundness) If $\langle a, b\rangle \neq \sigma$, then for any Merlin's message $\tilde{m}$, Alice and Bob output a', b', and σ' so that $\langle a', b'\rangle = \sigma'$ with probability at most* 0.98.

   Moreover, the running time of both Alice and Bob is $\mathrm{poly}(d)$.

The main difference between the above protocol and that of [9], is that instead of working over a field of characteristic $\Theta(T)$, we perform the protocol of [9] simultaneously over $O(\log T)$ different fields of size $O(\log^2 T)$ each.

To this end, we start by fixing some notation. Let t be an integer such that $t^t = T$. By Lemma 2.4 in [8], for a large enough integer t, there exist $10t$ distinct primes $p_1 < p_2 < \cdots < p_{10t}$, where the value of each prime is bounded in the interval $[t, t^2]$. Let $q := t^2$, and note that $t = O(\log T)$ and $q = O(\log^2 T)$. For each $\ell \in [10t]$, let $C^{(\ell)}, C^{(\ell)}_\star$ be the systematic linear codes over $\mathbb{F}_{p_\ell^2}$ guaranteed by Theorem 2.6, where $C^{(\ell)}$ has message length $\frac{d}{T}$ and codeword length $n_\ell := O(\frac{d}{T})$. Finally, recall that the elements of $\mathbb{F}_{p_\ell^2}$ can be viewed as degree 1 polynomials over $\mathbb{F}_{p_\ell}$, where multiplication is performed modulo an irreducible polynomial Q_ℓ of degree 2 over $\mathbb{F}_{p_\ell}$.

Let $a \in \{0,1\}^d$ be Alice's input. Slightly abusing notation, we view a as a $\frac{d}{T} \times T$ binary matrix in the natural way. For $\ell \in [10t]$, let $a^{(\ell)}$ denote the $n_\ell \times T$ matrix over $\mathbb{F}_{p_\ell^2}$ obtained by encoding each column of a with the code $C^{(\ell)}$. View each entry of $a^{(\ell)}$ as a degree 1 polynomial over $\mathbb{F}_{p_\ell}$, and let $a^{(\ell,0)}, a^{(\ell,1)}$ denote the $n_\ell \times T$ matrices over $\mathbb{F}_{p_\ell}$, obtained from $a^{(\ell)}$ by keeping in each entry only the free coefficient and linear coefficient, respectively. Let $b, b^{(\ell)}, b^{(\ell,0)}, b^{(\ell,1)}$ be defined analogously for $\ell \in [10t]$. In what follows, all arithmetic operations are performed over the reals, unless otherwise stated.

The protocol. The protocol proceeds as follows:

1. **a.** Merlin sends

$$m_0 := \sum_{j=1}^{T} \mathrm{col}_j(a) \star \mathrm{col}_j(b) \in \{0,1,\ldots,T\}^{d/T}.$$

   b. For $\ell = 1,\ldots,10t$ and $\alpha,\beta \in \{0,1\}$, Merlin sends

$$m_{\ell,\alpha,\beta} := \sum_{j=1}^{T} \mathrm{col}_j(a^{(\ell,\alpha)}) \star \mathrm{col}_j(b^{(\ell,\beta)}) \in \{0,1,\ldots,T\cdot q^2\}^{n_\ell}.$$

2. **a.** Alice checks that $\sum_{i=1}^{d/T} m_0(i) = \sigma$.
   b. Alice checks that $m_{\ell,0,0}(i) = m_0(i)$ and $m_{\ell,0,1}(i) = m_{\ell,1,0}(i) = m_{\ell,1,1}(i) = 0$ for $\ell = 1,\ldots,10t$ and $i = 1,\ldots,\frac{d}{T}$.
   c. For $\ell = 1,\ldots,10t$, let $m_\ell \in (\mathbb{F}_{p_\ell^2})^{n_\ell}$ given by

$$m_\ell = m_{\ell,0,0} + (m_{\ell,0,1} + m_{\ell,1,0}) \cdot X + m_{\ell,1,1} \cdot X^2 \pmod{Q_\ell}.$$

   Alice checks that m_ℓ is a codeword of $C_\star^{(\ell)}$ for $\ell = 1,\ldots,10t$.
   If any of the checks is unsatisfied, then Alice rejects and aborts.
3. Alice and Bob jointly sample $\ell_* \in [10t]$, $i_* \in [n_{\ell_*}]$, and $\alpha_*,\beta_* \in \{0,1\}$.
4. Alice outputs $a' := \mathrm{row}_{i_*}(a^{(\ell_*,\alpha_*)}) \in \{0,1,\ldots,q\}^T$ and $\sigma' := m_{\ell_*,\alpha_*,\beta_*}(i_*) \in \{0,1,\ldots,T\cdot q^2\}$, and Bob outputs $b' := \mathrm{row}_{i_*}(b^{(\ell_*,\beta_*)}) \in \{0,1,\ldots,q\}^T$.

It can be verified that the protocol has the required structure, and that the running times of Alice and Bob are as claimed. Next we show completeness and soundness.

Completeness. Suppose that $\langle a,b\rangle = \sigma$, we shall show that in this case Alice and Bob output a', b', and σ' so that $\langle a',b'\rangle = \sigma'$ with probability 1.

We first show that in this case all of Alice's checks on Step 2 always pass.

To this end, first note that by assumption that $\langle a,b\rangle = \sigma$, we have that

$$\sum_{i=1}^{d/T} m_0(i) = \sum_{i=1}^{d/T}\sum_{j=1}^{T} a(i,j)\cdot b(i,j) = \langle a,b\rangle = \sigma, \tag{1}$$

so Alice's check on Step 2a will pass.

We now show that Alice's check on Step 2b passes. Fix $\ell \in [10t]$, and recall that $a^{(\ell)}$ is obtained by encoding each column of the matrix $a \in \{0,1\}^{\frac{d}{T}\times T}$ with a systematic linear code. Consequently, a is the restriction of $a^{(\ell)}$ to the first $\frac{d}{T}$ rows, and similarly for b. This implies in turn that for any $i \in [\frac{d}{T}]$, we have that

$$m_{\ell,0,0}(i) = \left\langle \mathrm{row}_i(a^{(\ell,0)}), \mathrm{row}_i(b^{(\ell,0)}) \right\rangle = \langle \mathrm{row}_i(a), \mathrm{row}_i(b)\rangle = m_0(i), \tag{2}$$

and

$$m_{\ell,0,1}(i) = m_{\ell,1,0}(i) = m_{\ell,1,1}(i) = 0. \tag{3}$$

So Alice's check on Step 2b will pass as well.

Finally, we show that Alice's check on Step 2c passes. Fix $\ell \in [10t]$, and note that

$$\begin{aligned} m_\ell &= m_{\ell,0,0} + (m_{\ell,0,1} + m_{\ell,1,0}) \cdot X + m_{\ell,1,1} \cdot X^2 \pmod{Q_\ell} \\ &= \sum_{j=1}^{T} \Big[\mathrm{col}_j(a^{(\ell,0)}) \star \mathrm{col}_j(b^{(\ell,0)}) \\ &\quad + \Big(col_j(a^{(\ell,0)}) \star \mathrm{col}_j(b^{(\ell,1)}) + col_j(a^{(\ell,1)}) \star \mathrm{col}_j(b^{(\ell,0)}) \Big) \cdot X \\ &\quad + col_j(a^{(\ell,1)}) \star \mathrm{col}_j(b^{(\ell,1)}) \cdot X^2 \Big] \pmod{Q_\ell} \\ &= \sum_{j=1}^{T} (\mathrm{col}_j(a^{(\ell,0)}) + \mathrm{col}_j(a^{(\ell,1)}) \cdot X) \star (\mathrm{col}_j(b^{(\ell,0)}) + \mathrm{col}_j(b^{(\ell,1)}) \cdot X) \pmod{Q_\ell} \\ &= \sum_{j=1}^{T} \mathrm{col}_j(a^{(\ell)}) \star \mathrm{col}_j(b^{(\ell)}) \pmod{Q_\ell}. \end{aligned} \tag{4}$$

Now, since each column of $a^{(\ell)}$ and $b^{(\ell)}$ is a codeword of $C^{(\ell)}$, we have that $\mathrm{col}_j(a^{(\ell)}) \star \mathrm{col}_j(b^{(\ell)}) \pmod{Q_\ell}$ is a codeword of $C_\star^{(\ell)}$ for any $j \in [T]$. By linearity of $C_\star^{(\ell)}$, this implies in turn that m_ℓ is a codeword of $C_\star^{(\ell)}$, and so Alice's check on Step 2c will also pass.

Thus, we conclude that all of Alice's checks on Step 2 pass. Furthermore, we clearly have that

$$\langle a', b' \rangle = \Big\langle \mathrm{row}_{i_*}(a^{(\ell_*,\alpha_*)}), \mathrm{row}_{i_*}(b^{(\ell_*,\beta_*)}) \Big\rangle = m_{\ell_*,\alpha_*,\beta_*}(i) = \sigma'.$$

We conclude that in the case that $\langle a, b \rangle = \sigma$, we have that $\langle a', b' \rangle = \sigma'$ with probability 1, as required.

Soundness. Assume that $\langle a, b \rangle \neq \sigma$, and let $\tilde{m}_0$ and $\tilde{m}_{\ell,\alpha,\beta}$ for $\ell = 1, \ldots, 10t$ and $\alpha, \beta \in \{0,1\}$ be Merlin's messages on Step 1. We shall show that in this case Alice and Bob output a', b', and σ' so that $\langle a', b' \rangle = \sigma'$ with probability at most 0.98.

To this end, first note that we may assume that $\sum_{i=1}^{d/T} \tilde{m}_0(i) = \sigma$, since otherwise Alice clearly rejects on Step 2a. On the other hand, by (1) and by assumption that $\langle a, b \rangle \neq \sigma$, we have that $\sum_{i=1}^{d/T} m_0(i) = \langle a, b \rangle \neq \sigma$. Consequently, there exists $i \in [\frac{d}{T}]$ so that $m_0(i) \neq \tilde{m}_0(i)$.

Moreover, since $m_0(i), \tilde{m}_0(i) \in \{0, 1, \ldots, T\}$, we have that $|m_0(i) - \tilde{m}_0(i)| \leq T$. Recalling that $t^t = T$, and that $p_\ell \geq t$ for any $\ell \in [10t]$, we conclude that at most t of the p_ℓ's can divide $|m_0(i) - \tilde{m}_0(i)|$. Thus, with probability at least 0.9 over the choice of ℓ_*, it holds that $p_{\ell *}$ does not divide $|m_0(i) \neq \tilde{m}_0(i)|$, and so $m_0(i) \neq \tilde{m}_0(i) \pmod{p_{\ell_*}}$. In what follows, assume that this event holds.

Let $\tilde{m}_{\ell_*} \in (\mathbb{F}_{p_{\ell_*}^2})^{n_{\ell *}}$ be given by

$$\tilde{m}_{\ell_*} = \tilde{m}_{\ell_*,0,0} + (\tilde{m}_{\ell_*,0,1} + \tilde{m}_{\ell^*,1,0}) \cdot X + \tilde{m}_{\ell^*,1,1} \cdot X^2 \pmod{Q_{\ell_*}}.$$

Next observe that we may assume that for any $i \in [\frac{d}{T}]$,

$$\tilde{m}_{\ell *}(i) = \tilde{m}_{\ell_*,0,0}(i) + (\tilde{m}_{\ell_*,0,1}(i) + \tilde{m}_{\ell^*,1,0}(i)) \cdot X + \tilde{m}_{\ell^*,1,1}(i) \cdot X^2 \pmod{Q_{\ell_*}} = \tilde{m}_0(i) \pmod{p_{\ell_*}},$$

since otherwise Alice clearly rejects on Step 2b. On the other hand, by (2) and (3) we have that for any $i \in [\frac{d}{T}]$,

$$m_{\ell *}(i) = m_{\ell_*,0,0}(i) + (m_{\ell_*,0,1}(i) + m_{\ell^*,1,0}(i)) \cdot X + m_{\ell^*,1,1}(i) \cdot X^2 \pmod{Q_{\ell_*}} = m_0(i) \pmod{p_{\ell_*}}.$$

Consequently, by assumption that $m_0(i) \neq \tilde{m}_0(i) \pmod{p_{\ell_*}}$ for some $i \in [\frac{d}{T}]$, we have that $\tilde{m}_{\ell_*}(i) \neq m_{\ell_*}(i)$.

Finally, note that we may assume that $\tilde{m}_{\ell_*}$ is a codeword of $C_\star^{(\ell_*)}$, since otherwise Alice clearly rejects on Step 2c. Moreover, by (4) we also have that m_{ℓ^*} is a codeword of $C_\star^{(\ell_*)}$. Since $C_\star^{(\ell_*)}$ has relative distance at least 0.1, and by assumption that $\tilde{m}_{\ell_*} \neq m_{\ell_*}$, we have that $\tilde{m}_{\ell^*}$ and m_{ℓ^*} differ on at least a 0.1-fraction of their entries, and so with probability at least 0.1 over the choice of i_* it holds that $\tilde{m}_{\ell_*}(i_*) \neq m_{\ell_*}(i_*)$. In what follows, assume that this event holds as well.

By assumption that $\tilde{m}_{\ell_*}(i_*) \neq m_{\ell_*}(i_*)$, there exist $\alpha, \beta \in \{0,1\}$ so that $\tilde{m}_{\ell_*,\alpha,\beta}(i_*) \neq m_{\ell_*,\alpha,\beta}(i_*)$. Consequently, with probability at least 0.25 over the choice of α_*, β_*, it holds that $\tilde{m}_{\ell_*,\alpha_*,\beta_*}(i_*) \neq m_{\ell_*,\alpha_*,\beta_*}(i_*)$. But assuming that this latter event holds, we have that

$$\langle a', b' \rangle = \left\langle \mathrm{row}_{i_*}(a^{(\ell_*,\alpha_*)}), \mathrm{row}_{i_*}(b^{(\ell_*,\beta_*)}) \right\rangle = m_{\ell_*,\alpha_*,\beta_*}(i_*) \neq \tilde{m}_{\ell_*,\alpha_*,\beta_*}(i_*) = \sigma'.$$

We conclude that in the case that $\langle a, b \rangle \neq \sigma$, for any Merlin's message, we have that Alice either rejects or $\langle a', b' \rangle \neq \sigma'$ with probability at least $0.9 \cdot 0.1 \cdot 0.25 \geq 0.02$ over the choice of ℓ_*, i_*, α_*, and β_*. So $\langle a', b' \rangle = \sigma'$ with probability at most 0.98 over the choice of ℓ_*, i_*, α_*, and β_*.

4 From IP to Apx-Max-IP

In this section we use Theorem 3.1 from the previous section which gives an MA protocol for IP over a small alphabet to give a fine-grained reduction from IP to Apx-Max-IP with a tighter polynomial dependence of the approximation parameter δ on the running time parameter ϵ.

▶ **Lemma 4.1** (From IP to Apx-Max-IP). *The following holds for any $\epsilon > 0$ and integer $c \geq 1$. Suppose that $\mathrm{IP}_{N,d}$ cannot be solved in time $N^{2-\epsilon}$ for $d = c \log N$. Then there exists d' such that δ-Apx-Max-$\mathrm{IP}_{N,d'}$ cannot be solved in time $N^{2-2\epsilon}$ for $\delta = \tilde{\Theta}((\frac{\epsilon}{c})^2)$.*

To prove the above lemma, we shall use the following encoding lemma from [9], which can be used to turn the (non-binary) vectors a', b' from the protocol given in Theorem 3.1 into (binary) vectors, whose inner product exhibits a gap.

▶ **Lemma 4.2** (Encoding Lemma, [9]). *For any non-negative integers T and q, there exist mappings $g, h : \{0,1,\ldots,q\}^T \times \{0,1,\ldots,T \cdot q^2\} \to \{0,1\}^{O(T^2 q^4)}$ and an integer $\Gamma \leq O(T^2 \cdot q^4)$, so that for any $a, b \in \{0,1,\ldots,q\}^T$ and $\sigma \in \{0,1,\ldots,T \cdot q^2\}$:*

- *If $\langle a, b \rangle = \sigma \Rightarrow \langle g(a,\sigma), h(b,\sigma) \rangle = \Gamma$.*
- *If $\langle a, b \rangle \neq \sigma \Rightarrow \langle g(a,\sigma), h(b,\sigma) \rangle < \Gamma$.*

Moreover, g, h can be computed in time $\mathrm{poly}(T, q)$.

The reduction. We shall show a reduction from IP to many instances of Apx-Max-IP, based on our MA protocol for IP over a small alphabet given in Theorem 3.1, and the above encoding Lemma 4.2.

Let $A, B \subseteq \{0,1\}^d$ and $\sigma \in \{0,1,\ldots,d\}$ be an instance of $\mathrm{IP}_{N,d}$. Let T be a sufficiently large integer, to be determined later on, and let π be the protocol guaranteed by Theorem 3.1 for T, $q = O(\log^2 T)$, and d (without loss of generality assume that T divides d). Let g, h, and Γ be the mappings and the integer guaranteed for T and q by Lemma 4.2.

Let $\mathrm{Rej} \subseteq \{0,1\}^L$ denote the subset of Merlin's messages $m \in \{0,1\}^L$ in π on which Alice rejects on input σ. For $b \in B$ and $r \in \{0,1\}^R$, let b'_r denote the string output by Bob in the protocol π on input b and randomness string r. Similarly, for $a \in A$ and $r \in \{0,1\}^R$, let a'_r denote the string output by Alice in the protocol π on input a and randomness string r. For $m \in \{0,1\}^L \setminus \mathrm{Rej}$ and $r \in \{0,1\}^R$, let $\sigma'_{m,r}$ denote the integer output by Alice on Merlin's message m and randomness string r.

For any $m \in \{0,1\}^L \setminus \text{Rej}$, we create an instance A_m, B_m of Apx-Max-IP$_{N,d'}$, given by

$$A_m := \{(g(a'_r, \sigma'_{m,r}))_{r\in\{0,1\}^R} \mid a \in A\},$$

and

$$B_m := \{(h(b'_r, \sigma'_{m,r}))_{r\in\{0,1\}^R} \mid b \in B\},$$

where

$$d' = O(2^R \cdot T^2 \cdot q^4).$$

Let $\delta := \frac{0.01\cdot 2^R}{d'}$. Given an algorithm $\mathcal{A}$ for δ-Apx-Max-IP$_{N,d'}$, we show an algorithm $\mathcal{A}'$ for IP$_{N,d}$: Given an instance A, B, σ for IP$_{N,d}$, the algorithm $\mathcal{A}'$ generates all instances A_m, B_m for $m \in \{0,1\}^L \setminus \text{Rej}$, and runs $\mathcal{A}$ on any of these instances. If on any of the instances the algorithm $\mathcal{A}$ outputs a value at least $2^R \cdot (\Gamma - 0.01)$ then the algorithm $\mathcal{A}'$ accepts, otherwise it rejects.

Correctness. Correctness relies on the following claim.

▷ Claim.

- If there exists $(a, b) \in A \times B$ so that $\langle a, b\rangle = \sigma$, then there exist $m \in \{0,1\}^L \setminus \text{Rej}$ and $(a'', b'') \in A_m \times B_m$ so that $\langle a'', b''\rangle = 2^R \cdot \Gamma$.
- If $\langle a, b\rangle \neq \sigma$ for any $(a, b) \in A \times B$, then $\langle a'', b''\rangle \leq 2^R \cdot (\Gamma - 0.02)$ for any $m \in \{0,1\}^L \setminus \text{Rej}$ and $(a'', b'') \in A_m \times B_m$.

Proof. For the first item, suppose that there exists $(a, b) \in A \times B$ so that $\langle a, b\rangle = \sigma$. Let m be Merlin's message in the protocol π on inputs a, b, and σ, and let $(a'', b'') \in A_m \times B_m$ be given by $a'' = (g(a'_r, \sigma'_{m,r}))_{r\in\{0,1\}^R}$ and $b'' = (h(b'_r, \sigma'_{m,r}))_{r\in\{0,1\}^R}$. By the completeness property of π, we have that $m \notin \text{Rej}$, and $\langle a'_r, b'_r\rangle = \sigma'_{m,r}$ for any $r \in \{0,1\}^R$. Consequently, by Lemma 4.2, $\left\langle g(a'_r, \sigma'_{m,r}), h(b'_r, \sigma'_{m,r})\right\rangle = \Gamma$ for any $r \in \{0,1\}^R$. But this implies in turn that

$$\langle a'', b''\rangle = \sum_{r\in\{0,1\}^R} \left\langle g(a'_r, \sigma'_{m,r}), h(b'_r, \sigma'_{m,r})\right\rangle = 2^R \cdot \Gamma.$$

For the second item, suppose that $\langle a, b\rangle \neq \sigma$ for any $(a, b) \in A \times B$. Fix $m \in \{0,1\}^L \setminus \text{Rej}$ and $(a'', b'') \in A_m \times B_m$. Then by construction, $a'' = (g(a'_r, \sigma'_{m,r}))_{r\in\{0,1\}^R}$ and $b'' = (h(b'_r, \sigma'_{m,r}))_{r\in\{0,1\}^R}$. By the soundness property of π, for at least a 0.02-fraction of the randomness strings $r \in \{0,1\}^R$, it holds that $\langle a'_r, b'_r\rangle \neq \sigma'_{m,r}$. Consequently, by Lemma 4.2 for at least a 0.02-fraction of the randomness strings $r \in \{0,1\}^R$, it holds that $\left\langle g(a'_r, \sigma'_{m,r}), h(b'_r, \sigma'_{m,r})\right\rangle \leq \Gamma - 1$. But this implies in turn that

$$\langle a'', b''\rangle = \sum_{r\in\{0,1\}^R} \left\langle g(a'_r, \sigma'_{m,r}), h(b'_r, \sigma'_{m,r})\right\rangle \leq 0.98\cdot 2^R\cdot\Gamma + 0.02\cdot 2^R\cdot(\Gamma-1) = 2^R\cdot(\Gamma-0.02).\triangleleft$$

Now, if there exists $(a, b) \in A \times B$ so that $\langle a, b\rangle = \sigma$, then by the above claim there exists $m \in \{0,1\}^L \setminus \text{Rej}$ so that $\max_{a''\in A_m, b''\in B_m} \langle a'', b''\rangle \geq 2^R \cdot \Gamma$. Consequently, the algorithm $\mathcal{A}$ will output a value greater than $2^R \cdot \Gamma - \delta \cdot d' = 2^R \cdot (\Gamma - 0.01)$ on the instance A_m, B_m, and so the algorithm $\mathcal{A}'$ will accept.

If on the other hand, $\langle a, b\rangle \neq \sigma$ for any $(a, b) \in A \times B$, then by the above claim $\max_{a''\in A_m, b''\in B_m} \langle a'', b''\rangle \leq 2^R \cdot (\Gamma - 0.02)$ for any $m \in \{0,1\}^L \setminus \text{Rej}$. Consequently, the algorithm $\mathcal{A}$ will output a value at most $2^R \cdot (\Gamma - 0.02) < 2^R \cdot (\Gamma - 0.01)$ on any of the instances, an so the algorithm $\mathcal{A}'$ will reject.

Running time. Suppose that the algorithm $\mathcal{A}$ for δ-Apx-Max-IP$_{N,d'}$ runs in time $N^{2-2\epsilon}$, we shall show that for an appropriate choice of T, the running time of the algorithm $\mathcal{A}'$ for IP$_{N,d}$ is at most $N^{2-\epsilon}$.

The algorithm $\mathcal{A}'$ enumerates over all possible Merlin's messages $m \in \{0,1\}^L$, and for each such message checks whether Alice rejects m in π, which takes time poly(d), and if she does not reject, it generates the instance A_m, B_m which takes time $N \cdot 2^R \cdot \text{poly}(d) \cdot \text{poly}(T,q) \leq N \cdot \text{poly}(d)$, and runs the algorithm $\mathcal{A}$ on A_m, B_m which takes time $N^{2-2\epsilon}$.

Hence the total running time of the algorithm $\mathcal{A}'$ is at most

$$\begin{aligned} 2^L \cdot (N \cdot \text{poly}(d) + N^{2-2\epsilon}) &\leq 2^{O(\frac{d}{T} \cdot \log^2 T)} \cdot (N \cdot \text{poly}(d) + N^{2-2\epsilon}) \\ &= 2^{O(\frac{c \log N}{T} \cdot \log^2 T)} \cdot (N \cdot \text{poly}(c \log N) + N^{2-2\epsilon}). \\ &\leq 2^{O(\frac{c \log N}{T} \cdot \log^2 T)} \cdot N^{2-2\epsilon}. \end{aligned}$$

Finally, it can be verified that the latter expression is at most $N^{2-\epsilon}$ for choice of $T = \tilde{\Theta}(c/\epsilon)$ which divides d.

Approximation parameter. By choice of $\delta = \frac{0.01 \cdot 2^R}{d'}$, $d' = O(2^R \cdot T^2 \cdot q^4)$, $T = \tilde{\Theta}(c/\epsilon)$, and $q = O(\log^2 T)$, we have that

$$\delta = \Theta\left(\frac{1}{T^2 q^4}\right) = \tilde{\Theta}\left(\left(\frac{\epsilon}{c}\right)^2\right).$$

References

1. Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *ACM Trans. Comput. Theory*, 1(1):2:1–2:54, 2009.
2. Amir Abboud, Aviad Rubinstein, and Ryan Williams. Distributed pcp theorems for hardness of approximation in p. In *FOCS*, pages 25–36. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.12.
3. Amir Abboud, Richard Ryan Williams, and Huacheng Yu. More applications of the polynomial method to algorithm design. In Piotr Indyk, editor, *SODA*, pages 218–230. SIAM, 2015.
4. Josh Alman, Timothy Chan, and Ryan Williams. Polynomial representations of threshold functions and algorithmic applications. In *FOCS*, pages 467–476. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.57.
5. Josh Alman and Ryan Williams. Probabilistic polynomials and hamming nearest neighbors. In Venkatesan Guruswami, editor, *FOCS*, pages 136–150. IEEE Computer Society, 2015. URL: http://www.computer.org/csdl/proceedings/focs/2015/8191/00/index.html.
6. Ethem Alpaydin. *Introduction to Machine Learning*. MIT Press, Cambridge, Massachusetts, 2014.
7. Timothy M. Chan and R. Ryan Williams. Deterministic apsp, orthogonal vectors, and more: Quickly derandomizing razborov-smolensky. *ACM Trans. Algorithms*, 17(1):2:1–2:14, 2021.
8. Lijie Chen. On the hardness of approximate and exact (bichromatic) maximum inner product. *Theory of Computing*, 16:1–50, 2020.
9. Lijie Chen and Ryan Williams. An equivalence class for orthogonal vectors. In *SODA*, pages 21–40. SIAM, 2019.
10. Andreas Emil Feldmann, C. S. Karthik, Euiwoong Lee, and Pasin Manurangsi. A survey on approximation in parameterized complexity: Hardness and algorithms. *Algorithms*, 13(6):146, 2020.
11. Jiawei Gao, Russell Impagliazzo, Antonina Kolokolova, and Ryan Williams. Completeness for first-order properties on sparse structures with algorithmic applications. *ACM Trans. Algorithms*, 15(2):23:1–23:35, 2019.

12 Tomislav Hengl. Finding the right pixel size. *Computers and Geosciences*, 32(9):1283–1298, 2006.

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

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

15 C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, 1992.

16 M. O. Rabin. *Probabilistic Algorithms*, pages 21–39. Academic Press, NY, 1976.

17 Aviad Rubinstein. Hardness of approximate nearest neighbor search. In *STOC*, pages 1260–1268. ACM, 2018. URL: `http://dl.acm.org/citation.cfm?id=3188745`.

18 Aviad Rubinstein and Virginia Vassilevska Williams. Seth vs approximation. *SIGACT News*, 50(4):57–76, 2019.

19 Kenneth W. Shum, Ilia Aleshnikov, P. Vijay Kumar, Henning Stichtenoth, and Vinay Deolalikar. A low-complexity algorithm for the construction of algebraic-geometric codes better than the gilbert-varshamov bound. *IEEE Transactions on Information Theory*, 47(6):2225–2241, 2001. `doi:10.1109/18.945244`.

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

21 Virginia Vassilevska Williams. Some open problems in fine-grained complexity. *SIGACT News*, 49(4):29–35, 2018.

22 Virginia Vassilevska Williams and R. Ryan Williams. Subcubic equivalences between path, matrix, and triangle problems. *J. ACM*, 65(5):27:1–27:38, 2018.

23 Raymond Chi-Wing Wong, Yufei Tao, Ada Wai-Chee Fu, and Xiaokui Xiao. On efficient spatial matching. In *VLDB*, pages 579–590. ACM, 2007. URL: `http://www.vldb.org/conf/2007/papers/research/p579-wong.pdf`.

24 Charles Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, 20(1):68–86, January 1971.

A Applications

In this section we show a couple of consequences of Lemma 4.1 to obtaining tighter fine-grained hardness of approximation results based on the IP assumption.

Closest pair in Hamming metric. The following reduction from Max-IP to CP in the Hamming metric Δ is implicit in [17].

▶ **Lemma A.1** (From Apx-Max-IP to Apx-CP$_\Delta$, [17]). *Suppose that δ-Apx-Max-IP$_{N,d}$ cannot be solved in time $N^{2-\epsilon}$. Then δ'-Apx-CP$_{N,d',\Delta}$ cannot be solved in time $N^{2-2\epsilon}$ for $\delta' = \frac{\delta}{2}$.*

The above lemma and Lemma 4.1 readily imply the following.

▶ **Corollary A.2** (From IP to Apx-CP$_\Delta$). *Suppose that IP$_{N,d}$ cannot be solved in time $N^{2-\epsilon}$ for $d = c \log N$. Then δ-Apx-CP$_{N,d',\Delta}$ cannot be solved in time $N^{2-2\epsilon}$ for $\delta = \tilde{\Theta}((\frac{\epsilon}{c})^2)$.*

In contrast, it is known how to obtain an $(1+\delta)$-approximation for CP over the Hamming metric in time $N^{2-\epsilon}$ for $\delta = \tilde{\Theta}(\epsilon^3)$ [4].

Closest pair in ℓ_p metric. The following reduction from CP in the Hamming metric to CP in the ℓ_p metric is also implicit in [17].

▶ **Lemma A.3** (From Apx-CP$_\Delta$ to Apx-CP$_{\ell_p}$, [17]). *Suppose that δ-Apx-CP$_{N,d,\Delta}$ cannot be solved in time $N^{2-\epsilon}$. Then for any $p > 0$, δ'-Apx-CP$_{N,d,\ell_p}$ cannot be solved in time $N^{2-2\epsilon}$ for $\delta' = \Theta_p(\delta)$.*

The following corollary is a consequence of the above lemma and Corollary A.2, and implies Theorem 1.1.

▶ **Corollary A.4** (From IP to Apx-CP$_{\ell_p}$). *Suppose that* IP$_{N,d}$ *cannot be solved in time $N^{2-\epsilon}$ for $d = c \log N$. Then for any $p > 0$, δ-Apx-CP$_{N,d',\ell_p}$ cannot be solved in time $N^{2-2\epsilon}$ for $\delta = \tilde{\Theta}_p((\frac{\epsilon}{c})^2)$.*

In contrast, it is known how to obtain an $(1+\delta)$-approximation for CP over the ℓ_p metric in time $N^{2-\epsilon}$ for $\delta = \tilde{O}(\epsilon^3)$ and $p \in \{1, 2\}$ [4].

Closest pair in edit distance metric. For $a, b \in \Sigma^d$, we let $ED(a, b)$ denote their edit distance which is the minimum number of character deletion, insertion, and substitution operations needed to transform a into b. The following Lemma is also implicit in [17].

▶ **Lemma A.5** (From Apx-CP$_\Delta$ to Apx-CP$_{ED}$, [17]). *Suppose that δ-Apx-CP$_{N,d,\Delta}$ cannot be solved in time $N^{2-\epsilon}$. Then δ'-Apx-CP$_{N,d',ED}$ cannot be solved in time $N^{2-2\epsilon}$ for $\delta' = \Theta(\delta)$.*

The above lemma and Corollary A.2 imply the following corollary.

▶ **Corollary A.6** (From IP to Apx-CP$_{ED}$). *Suppose that* IP$_{N,d}$ *cannot be solved in time $N^{2-\epsilon}$ for $d = c \log N$. Then δ-Apx-CP$_{N,d',ED}$ cannot be solved in time $N^{2-2\epsilon}$ for $\delta' = \tilde{\Theta}((\frac{\epsilon}{c})^2)$.*

To the best of our knowledge, it is not known how to solve $(1+\delta)$-Apx-CP$_{\text{ED}}$ in sub-quadratic time.

▶ Remark (Apx-Min-IP and Furthest-Pair). It is not hard to show (see e.g., [9], Lemma 5.3) that there is a simple linear-time reduction from δ-Apx-Max-IP$_{N,d}$ to δ-Apx-Min-IP$_{N,d}$ (and vice versa), and so the same result as in Lemma 4.1 also holds for δ-Apx-Min-IP$_{N,d}$ (where the goal is to output a number in $[M, M + \delta \cdot d]$, where $M := \min_{a \in A, b \in B} \langle a, b \rangle$).

Using Apx-Min-IP as the starting point for the reductions cited above instead of Apx-Max-IP implies the same results as in Corollaries A.2, A.4, and A.6 for Furthest Pair (where the goal is to output a number in $[(1-\delta)M, M]$, where $M := \max_{a \in A, b \in B} dist(a, b)$).

Data structure setting. Our results extend to the data structure setting.

▶ **Definition A.7** (Approximate Nearest Neighbor (Apx-NN)). *Let* dist : $\{0,1\}^d \times \{0,1\}^d \to \mathbb{R}^+$ *be a distance function, and let $\delta > 0$ be a parameter. In the Approximate Nearest Neighbor δ-Apx-NN$_{N,d,dist}$ problem, given a set $A \subseteq \{0,1\}^d$ of cardinality N, the goal is to pre-process the set, so that given a vector $b \in \{0,1\}^d$ it is possible to quickly output a number in $[M, (1+\delta)M]$, where $M := \min_{a \in A}$* dist(a, b).

It is known that Apx-CP can be reduced to Apx-NN [22] (see also proof of Corollary 1.4 in [2]).

▶ **Lemma A.8** (From Apx-CP$_\Delta$ to Apx-NN, [22]). *Let* dist : $\{0,1\}^d \times \{0,1\}^d \to \mathbb{R}^+$ *be a distance function. Suppose that δ-Apx-CP$_{N,d,dist}$ cannot be solved in $N^{2-\epsilon}$ time. Then for any $r > 0$, δ-Apx-NN$_{N,d,dist}$ cannot be solved with N^r preprocessing time and $N^{1-2r\epsilon}$ time.*

▶ **Corollary A.9** (From IP to Apx-NN). *Suppose that* IP$_{N,d}$ *cannot be solved in time $N^{2-\epsilon}$ for $d = c \log N$. Then for any distance function* dist $\in \{\Delta, \ell_p, \text{ED}\}$ *and $r > 0$, δ-Apx-NN$_{N,d',dist}$ cannot be solved with N^r preprocessing time and $N^{1-3r\epsilon}$ query time for $\delta = \tilde{\Theta}((\frac{\epsilon}{c})^2)$.*