# An Equivalence Class for Orthogonal Vectors*

Lijie Chen
MIT

Ryan Williams
MIT

**Abstract**

The Orthogonal Vectors problem (OV) asks: *given $n$ vectors in $\{0,1\}^{O(\log n)}$, are two of them orthogonal?* OV is easily solved in $O(n^2 \log n)$ time, and it is a central problem in fine-grained complexity: dozens of conditional lower bounds are based on the popular hypothesis that OV cannot be solved in (say) $n^{1.99}$ time. However, unlike the APSP problem, few other problems are known to be non-trivially *equivalent* to OV.

We show OV is truly-subquadratic equivalent to several fundamental problems, all of which (a priori) look harder than OV. A partial list is given below:

1. (Min-IP/Max-IP) Find a red-blue pair of vectors with minimum (respectively, maximum) inner product, among $n$ vectors in $\{0,1\}^{O(\log n)}$.
2. (Exact-IP) Find a red-blue pair of vectors with inner product equal to a given target integer, among $n$ vectors in $\{0,1\}^{O(\log n)}$.
3. (Apx-Min-IP/Apx-Max-IP) Find a red-blue pair of vectors that is a 100-approximation to the minimum (resp. maximum) inner product, among $n$ vectors in $\{0,1\}^{O(\log n)}$.
4. (Approximate Bichrom.-$\ell_p$-Closest-Pair) Compute a $(1+\Omega(1))$-approximation to the $\ell_p$-closest red-blue pair (for a constant $p \in [1,2]$), among $n$ points in $\mathbb{R}^d$, $d \leq n^{o(1)}$.
5. (Approximate $\ell_p$-Furthest-Pair) Compute a $(1+\Omega(1))$-approximation to the $\ell_p$-furthest pair (for a constant $p \in [1,2]$), among $n$ points in $\mathbb{R}^d$, $d \leq n^{o(1)}$.

Therefore, quick constant-factor approximations to maximum inner product imply quick *exact* solutions to maximum inner product, in the $O(\log n)$-dimensional setting. Another consequence is that the ability to find vectors with zero inner product suffices for finding vectors with maximum inner product.

Our equivalence results are robust enough that they continue to hold in the data structure setting. In particular, we show that there is a $\mathrm{poly}(n)$ space, $n^{1-\varepsilon}$ query time data structure for *Partial Match* with vectors from $\{0,1\}^{O(\log n)}$ if and only if such a data structure exists for $1+\Omega(1)$ *Approximate Nearest Neighbor Search* in Euclidean space.

To establish the equivalences, we introduce two general frameworks for reductions to OV: one based on $\Sigma_2$ communication protocols, and another based on locality-sensitive hashing families.

In addition, we obtain an $n^{2-1/O(\log c)}$ time algorithm for Apx-Min-IP with $n$ vectors from $\{0,1\}^{c \log n}$, matching state-of-the-art algorithms for OV and Apx-Max-IP. As an application, we obtain a faster algorithm for approximating "almost solvable" MAX-SAT instances.

## 1 Introduction

Fine-grained complexity asks: *what is the "correct" exponent in the running time of a given problem?* For a problem known to be solvable in time $t(n)$, can it be solved in time $t(n)^{1-\varepsilon}$, for a constant $\varepsilon > 0$? If not, can

*Supported by NSF CCF-1741615 (CAREER: Common Links in Algorithms and Complexity). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

we give evidence that such an improvement is impossible? In recent years, based on several conjectures such as the Orthogonal Vectors Conjecture (OVC) (implied by the Strong Exponential Time Hypothesis, a.k.a. SETH[1]), the APSP Conjecture and the $k$-Sum Conjecture, tight conditional polynomial-time lower bounds have been established for problems in P from many areas of computer science.

In a nutshell, results in the *Fine-Grained Complexity* program begin with the conjecture that it is *hard to improve the runtime exponent* of some problem $\Pi_{\mathsf{hard}}$, and show it is also hard to improve the exponent of another problem $\Pi$, by constructing a "fine-grained" reduction from $\Pi_{\mathsf{hard}}$ to $\Pi$. This is similar to the situation with NP-completeness, where one shows a problem $\Pi$ is "hard" by giving a polynomial-time reduction from another NP-complete problem to $\Pi$.

A crucial conceptual difference between the Fine-Grained Complexity program and NP-hardness is that *all of the thousands of known* NP*-complete problems form an equivalence class*: there is either a polynomial-time algorithm for all of them, or no polynomial-time algorithm for any of them. In contrast, with Fine-Grained Complexity, few equivalence classes are known, especially for those numerous problems whose hardnesses are based on the SETH/OVC (a notable exception is the equivalence class for APSP [VW10, Vas18]; see the related works section for more details).

To give three (out of many examples), it is known that Edit Distance [BI15], Frechet Distance [Bri14], and computing the diameter of a sparse graph [RW13] cannot be done in $n^{2-\delta}$ time for any $\delta > 0$, assuming the following problem is not in $n^{2-\varepsilon}$ time for a universal $\varepsilon > 0$:

> **Orthogonal Vectors** (OV): Given $n$ vectors in $\{0,1\}^d$ where $d = O(\log n)$, are there two vectors with inner product zero?

However, it is not known if Edit Distance, Frechet Distance, or Diameter are *equivalent* to OV, in any interesting sense.

Prior work has established an equivalence class for "moderate-dimensional OV", where the vector dimension $d = n^\delta$ for a constant $\delta > 0$ [GIKW17]. In particular, this version of OV is equivalent to various sparse graph and hypergraph problems. It seems likely that "moderate-dimensional OV" is much more difficult to solve than the "low-dimensional" setting of $d = O(\log n)$ as defined above, and the SETH already implies that the low-dimensional case is difficult [Wil05, WY14]. Thus the problem of establishing an equivalence class for "low-dimensional" OV is an interesting one.

### 1.1 An Equivalence Class for Sparse Orthogonal Vectors

Our first result is an interesting equivalence class for Orthogonal Vectors in the $O(\log n)$-dimensional setting. To formally state our results, we begin with some notation.

- For a problem $\Pi$ on Boolean vectors, we say $\Pi$ *is in truly subquadratic time* if there is an $\varepsilon > 0$ such that for all constant $c$, $\Pi$ is solvable in $O(n^{2-\varepsilon})$ time on $n$ vectors in $c \log n$ dimensions. Note the Orthogonal Vectors Conjecture (OVC) is equivalent to saying "OV is not in truly subquadratic time."
- For a problem $\Pi$ on real-valued points, we say $\Pi$ *can be approximated in truly subquadratic time*, if there is a $\delta > 0$ such that for all $\varepsilon > 0$, a $(1+\varepsilon)$ approximation to $\Pi$ is computable in $O(n^{2-\delta})$ time.
- For a problem $\Pi$ with output in $[0, L]$ (for a parameter $L$), we say $\Pi$ *can be additively approximated in truly subquadratic time*, if there is a $\delta > 0$ such that for all $\varepsilon > 0$, an $\varepsilon \cdot L$ additive approximation to $\Pi$ is computable in $O(n^{2-\delta})$ time.

[1]The Strong Exponential Time Hypothesis (SETH) states that for every $\varepsilon > 0$ there is a $k$ such that $k$-SAT cannot be solved in $O((2-\varepsilon)^n)$ time [IP01].

**Theorem 1.1.** *The following problems are either all in (or can be approximated in) truly subquadratic time, or none of them are:*[2]

1. *(OV) Finding an orthogonal pair among $n$ vectors.*
2. *(Min-IP/Max-IP) Finding a red-blue pair of vectors with minimum (respectively, maximum) inner product, among $n$ vectors.*
3. *(Exact-IP) Finding a red-blue pair of vectors with inner product exactly equal to a given integer, among $n$ vectors.*
4. *(Apx-Min-IP/Apx-Max-IP) Finding a red-blue pair of vectors that is a 100-approximation to the minimum (resp. maximum) inner product, among $n$ vectors.*[3]
5. *(Approximate Bichrom. $\ell_p$-Closest Pair) Approximating the $\ell_p$-closest red-blue pair (for a constant $p \in [1, 2]$), among $n$ points.*
6. *(Approximate $\ell_p$-Furthest Pair) Approximating the $\ell_p$-furthest pair (for a constant $p \in [1, 2]$), among $n$ points.*
7. *(Approximate Additive Max-IP) Additively approximating the maximum inner product of all red-blue pairs, among $n$ vectors.*
8. *(Approximate Jaccard-Index-Pair) Additively approximating the maximum Jaccard index*[4] *between $a \in A$ and $b \in B$, where $A$ and $B$ are two collections of $n$ sets.*

For approximate additive Max-IP, $L$ (the additive approximation parameter) is simply the dimensions of the vectors, while for approximate Jaccard-Index-Pair, $L$ is 1. For $\Pi$ among the first four problems listed above, we use the notation $\Pi_{n,d}$ to denote $\Pi$ with $n$ vectors from $\{0,1\}^d$. [5] For the last four problems, we assume the dimensions (or the size of the sets) and the bit complexity of the points are $n^{o(1)}$ throughout the paper.

Prior work showed OV is equivalent to Dominating Pair[6] [Cha17] and other simple set problems [BCH16]; our results add several interesting new members into the equivalence class. All problems listed above were already known to be OV-hard [Wil05, AW15, Rub18]. Our main contribution here is to show that ***they can all be reduced back to OV***. For example, detecting an orthogonal *Boolean* pair (OV) is equivalent to approximating the distance between two sets of points in $\mathbb{R}^{n^{o(1)}}$ (Bichrom.-Closest-Pair)!

In previous works [GIKW17, ABDN18], several general techniques are given for constructing reductions to OV. These papers focus on the "moderate-dimensional" setting, and their reductions can not be used directly in the "sparse" $O(\log n)$ dimensional setting here.

**Our Techniques: Two Reduction Frameworks for OV.** In order to construct reductions to $O(\log n)$ dimensional OV, we propose the following two general frameworks.

[2] A list of formal definitions of the these problems can be found in Definition 2.1.

[3] The constant 100 can be replaced by any fixed constant $\kappa > 1$.

[4] see Theorem 2.3 for a formal definition

[5] In the paper we will consider red-blue version for all the above problems, and $\Pi_{n,d}$ denotes $\Pi$ with two sets of $n$ vectors from $\{0,1\}^d$.

[6] Given two sets $A, B$ of vectors from $\mathbb{R}^{O(\log n)}$, find $(a, b) \in A \times B$ such that $b$ *dominates* $a$ (that is, $b_i > a_i$ for all $i$).

- $\Sigma_2^{\mathsf{cc}}$ **Protocols.** Inspired by previous works on the connections between communication complexity and fine-grained complexity [ARW17, Rub18, KLM18, AR18, Che18, CGL$^+$18], we draw another connection along this line, showing that an efficient $\Sigma_2^{\mathsf{cc}}$ protocol[7] for a function $F$ implies a reduction from a related problem to OV. We use this technique to establish the equivalences among the first four problems in Theorem 1.1.

- **Locality-sensitive Hashing Families (LSH).** To show equivalences between OV and the last four approximation problems, we apply known tools from *locality-sensitive hashing*. In particular, we show that for any metric admitting an efficient LSH family, finding the closest bichromatic pair or the furthest pair w.r.t. this metric can be reduced to Apx-Max-IP, which can in turn be reduced to OV.

We remark that there are no non-trivial lower bounds known against $\Sigma_2^{\mathsf{cc}}$ protocols [GPW18], which suggests that $\Sigma_2^{\mathsf{cc}}$ protocols ***could be very powerful***, and the first approach (Theorem 1.14) may be applicable to many other problems. This is not the case for $\mathsf{MA}^{\mathsf{cc}}$ protocols which were used in several previous works [ARW17, Rub18, KLM18, Che18]: for example, there is an essentially tight $\Omega(\sqrt{n})$ $\mathsf{MA}^{\mathsf{cc}}$ lower bound for Set-Disjointness [Kla03, AW09, Che18]. These two frameworks are discussed in Section 1.3 in detail.

**Equivalence Between Partial Match and Approximate Nearest Neighbor Search.** Our reductions are robust enough that they also hold in the data structure setting. In particular, consider the following two fundamental data structure problems:

- **Partial Match:** Preprocess a database $\mathcal{D}$ of $n$ points in $\{0,1\}^d$ such that, for all query of the form $q \in \{0,1,\star\}^d$, either report a point $x \in D$ matching all non-$\star$ characters in $q$ or report that no $x$ exists.

- **Approximate Nearest Neighbor Search (NNS) in $\ell_p$ space:** Preprocess a database $\mathcal{D}$ of $n$ points from $\mathbb{R}^m$ such that, for all query point $x \in \mathbb{R}^m$, one can find a point $y \in \mathcal{D}$ such that $\|x-y\|_p \leq (1+\varepsilon) \cdot \min_{z\in\mathcal{D}} \|x-z\|_p$.

**Remark 1.2.** *We remark that Partial Match is known to be equivalent to an online version of OV [AWY15] (see also Section 7), and NNS in $\ell_p$ space is simply the online version of Bichrom.-$\ell_p$-Closest-Pair.*

Partial Match has been studied extensively for decades (see e.g. Rivest's PhD thesis [Riv74]). However, the algorithmic progress beyond trivial solutions (building a look-up table of size $2^{\Omega(d)}$, or trying all $n$ points on each single query) have been quite limited. It is generally believed that it is intractable when $d$ is large enough. Many unconditional lower bounds are known in the cell-probe model [MNSW98, BOR99, JKKR04, PTW08, PT09], but the gap between the best data structures [CIP02, CGL04] and known lower bounds remains very large.

Approximate Nearest Neighbor Search has a wide range of applications in computing, including machine learning, computer vision, databases and others (see [AI08, Mor08] for an overview). Tremendous research effort has been devoted to this problem (see e.g. the recent survey of [AIR18] and Razenshteyn's PhD thesis [Raz17]). Yet all known algorithms exhibit a query time of at least $n^{1-O(\varepsilon)}$ when the approximation ratio is $1+\varepsilon$, approaching the brute-force query time $n$ when $\varepsilon$ goes to 0.

In general, whether there is a *polynomial* space, $n^{1-\delta}$ query time data structure for Partial Match for all $d = O(\log n)$, or Approximate NNS for all constant approximation ratio $> 1$ are two long-standing open questions.[8] We show these two questions are ***equivalent***.

[7] see Definition 1.13 for a formal definition

[8] Under SETH, it is shown that there is no such data structure with *polynomial pre-processing time* [APRS16, Wil05, Rub18].

**Theorem 1.3.** *The following are equivalent:*

- *There is a $\delta > 0$ such that for all constant $c$, there is a data structure for Partial Match with string length $d = c \log n$ that uses $\mathrm{poly}(n)$ space and allows $n^{1-\delta}$ query time.*
- *There is a $\delta > 0$ such that for all $\varepsilon > 0$, there is an data structure for Approximate NNS in $\ell_p$ with approximation ratio $(1+\varepsilon)$ that uses $\mathrm{poly}(n)$ space and allows $n^{1-\delta}$ query time, for some constant $p \in [1, 2]$.*

**Tighter Connection Between Max-IP, Bichrom. $\ell_p$-Closest Pair and $\ell_p$-Furthest Pair.** For a subset of problems in Theorem 1.1, we can establish even tighter reductions.

The state-of-the-art algorithm for $(1+\varepsilon)$ approximation to Bichrom.-$\ell_p$-Closest-Pair runs in $n^{2-\widetilde{O}(\varepsilon^{1/3})}$ time, and for Max-IP$_{n,c\log n}$, the best running time $n^{2-\widetilde{O}(1/\sqrt{c})}$. Both algorithms are presented in [ACW16], and relied on probabilistic threshold functions.

Comparing to the $n^{2-1/O(\log c)}$ time algorithm for OV$_{n,c\log n}$ [AWY15, CW16], the dependence on $c$ or $\varepsilon$ in these two algorithms are much worse, rendering them useless when $\varepsilon^{-1}$ or $c$ are $\log^{\omega(1)} n$. So it is natural to ask whether the dependence can be improved to at least sub-polynomial in $\varepsilon$ and $c$, i.e. $n^{2-1/c^{o(1)}}$ or $n^{2-\varepsilon^{o(1)}}$.

We show that a modest improvement on the running time dependence on $\varepsilon$ or $c$ for any of the following problems directly implies similar improvements for other problems as well.

**Theorem 1.4.** *The following are equivalent:*

- *An $\varepsilon \cdot d$ additive approximation to Max-IP$_{n,d}$ is computable in $n^{2-\varepsilon^{o(1)}}$ time.*
- *Max-IP$_{n,c\log n}$ is solvable in $n^{2-1/c^{o(1)}}$ time.*
- *Exact-IP$_{n,c\log n}$ is solvable in $n^{2-1/c^{o(1)}}$ time.*
- *A $(1+\varepsilon)$ approximation to Bichrom.-$\ell_p$-Closest-Pair is computable in $n^{2-\varepsilon^{o(1)}}$ time (for a constant $p \in [1, 2]$).*
- *A $(1+\varepsilon)$ approximation to $\ell_p$-Furthest-Pair is computable in $n^{2-\varepsilon^{o(1)}}$ time (for a constant $p \in [1, 2]$).*

In [Rub18] (Theorem 4.1), it is implicitly shown that Exact-IP$_{n,c\log n}$ can be reduced to $(1 + 1/\exp(c))$ approximating Bichrom.-$\ell_p$-Closest-Pair. This suffices for the case when $c$ is a constant (which is needed for Theorem 1.1), but falls short of proving the above tighter connections.

In a nutshell, [Rub18]'s reduction applies a very efficient MA protocol for Set-Disjointness using AG-codes, and it uses "brute-force" gadgets to simulate an inner product between two short vectors in $\mathbb{F}_{q^2}$. We improve [Rub18]'s reduction by carefully modifying its MA protocol, and replacing its brute-force gadgets by a more efficient one. Informally, our theorem shows Exact-IP$_{n,c\log n}$ can be reduced to $(1 + 1/\mathrm{poly}(c))$ approximating Bichrom.-Closest-Pair (see Lemma 6.4 and Lemma 6.7), which is an exponential improvement over the old reduction.

**Equivalence Results in the Moderate Dimensional Setting.** Theorem 1.1 establishes an equivalence class for the sparse $O(\log n)$ dimensional setting. It is natural to ask whether the equivalence continues to hold in the moderate dimensional case as well.

Unfortunately, an unusual (and interesting) property of our reduction used in Theorem 1.1 is that it blows up $c$ (the constant before $\log n$) exponentially, and creates multiple instances. That is, an Exact-IP instance with $c \log n$ dimensions is reduced to *many* OV instances with $\exp(c) \log n$ dimensions (see the proof of

Lemma 4.2). This renders the reduction useless in the moderate-dimensional setting, where $c$ could be as large as $n^{\delta}$.

Still, using different techniques, we obtain some additional equivalence results in the moderate dimensional setting. For a problem $\Pi$ on Boolean vectors, we say that *moderate dimensional* $\Pi$ *is in truly subquadratic time*, if there are two constants $\varepsilon, \delta > 0$ such that $\Pi$ is solvable in $n^{2-\varepsilon}$ time on $n$ vectors with $n^{\delta}$ dimensions.

**Theorem 1.5.** *Moderate dimensional* OV *is in truly subquadratic time if and only if moderate dimensional* Apx-Min-IP *is.*

**Theorem 1.6.** *For moderate dimensional* Max-IP*,* Min-IP*, and* Exact-IP*, either all of them are in truly subquadratic time, or none of them are.*

To show moderate dimensional OV and Apx-Min-IP are equivalent, we use a sophisticated reduction which is partially inspired by the classical Goldwasser-Sipser AM protocol for approximate counting [GS89] (see the proof of Lemma 5.1 for details). For Max-IP, Min-IP and Exact-IP, we apply some folklore encoding tricks.

It is an interesting open question that whether these two separate equivalence classes can be merged into one. In particular, *is moderate dimensional* OV *equivalent to moderate dimensional* Max-IP*?*

An immediate corollary of Theorem 1.5 is that it adds Apx-Min-IP as a new member to the equivalence class of moderate dimensional OV established in [GIKW17].

## 1.2 New Algorithms for Apx-Min-IP and Apx-Max-IP

It was recently shown in [Che18] that Apx-Max-IP can be solved in $n^{2-1/O(\log c)}$ time, while the best known algorithm for solving Apx-Min-IP just applies the $n^{2-1/\widetilde{O}(\sqrt{c})}$ time algorithm for Min-IP [ACW16]. We show that in fact we can derive an algorithm with similar running time for Apx-Min-IP as well.

**Theorem 1.7.** *There are $n^{2-1/O(\log c)}$ time randomized algorithms for* Apx-Min-IP$_{n,c\log n}$ *and* Apx-Max-IP$_{n,c\log n}$.

**Remark 1.8.** *Our new algorithm works equally well for* Apx-Max-IP*. Hence, we provide a different $n^{2-1/O(\log c)}$ time algorithm for* Apx-Max-IP *than [Che18]. One caveat here is that our algorithms are* randomized*, while the algorithms in [Che18] are* deterministic*.*

The algorithms are based on the polynomial method: we construct a low-degree probabilistic polynomial over $\mathbb{F}_2$ for functions closely related to Apx-Min-IP and Apx-Max-IP, and the rest follows from the framework of [AWY15].

**Application: A Fast Algorithm for Approximating "Almost Solvable" MAX-SAT Instances.** Here we give an application of our Apx-Min-IP algorithm. For a MAX-SAT instance $\varphi$ with $m$ clauses, we denote $\mathsf{OPT}(\varphi)$ to be the maximum number of the clauses that can be satisfied, and $\mathsf{sat}(\varphi) := \mathsf{OPT}(\varphi)/m$.

**Theorem 1.9.** *Let $\varphi$ be a* MAX-SAT *instance on $n$ variables with $m$ clauses, and $\varepsilon = 1 - \mathsf{sat}(\varphi)$. There is a $2^{n(1-1/O(\log \varepsilon^{-1}))}$ time algorithm to find an assignment $x$ satisfying at least $(1-2\varepsilon)\cdot m$ clauses*[9].

That is, when $\varphi$ is "almost solvable" ($\mathsf{sat}(\varphi)$ is very close to 1), we have a fast algorithm to compute an approximate solution $x$, which is only a "little" worse than the optimal solution. The following corollary is immediate.

**Corollary 1.10.** *Let $\varphi$ be a* MAX-SAT *instance on $n$ variables and $\varepsilon \in (0, 1/10)$. Given the promise that either $\mathsf{sat}(\varphi) \geq 1-\varepsilon$ or $\mathsf{sat}(\varphi) < 1-2\varepsilon$, there is a $2^{n(1-1/O(\log \varepsilon^{-1}))}$ time algorithm for deciding which is the case.*

[9] $(1-2\varepsilon)$ can be replaced by $(1-\kappa\varepsilon)$ for any constant $\kappa > 1$.

The best known previous algorithm for the above problem requires at least $2^{n(1-\varepsilon^{1/3})}$ time [ACW16], in which case the dependence on $\varepsilon$ is exponentially worse than our new algorithm. In particular, it fails to give any improvement when $\varepsilon < 1/n^3$, while our algorithm is faster than brute-force even if $\varepsilon = 1/2^{n^{0.99}}$.

## 1.3 Techniques: Two General Frameworks for Establishing OV Equivalence

In the following we discuss two general frameworks for reductions to OV. To state our results formally, we first define the $F$-Satisfying-Pair problem for a problem $F$.[10]

**Definition 1.11** ([AHWW16])**.** *Let $F : \{0,1\}^d \times \{0,1\}^d \to \{0,1\}$, $F$-Satisfying-Pair$_n$ is the problem: given two sets $A$ and $B$ of $n$ vectors from $\{0,1\}^d$, determine whether there is a pair $(a,b) \in A \times B$ such that $F(a,b) = 1$.*

**Remark 1.12.** *For example, let $F_{\mathsf{OV}}$ be the function checking whether two vectors from $\{0,1\}^d$ are orthogonal. Then, $F_{\mathsf{OV}}$-Satisfying-Pair$_n$ is simply OV$_{n,d}$.*

### 1.3.1 $\Sigma_2$ Communication Protocols and Reductions to Orthogonal Vectors

Our first framework is based on $\Sigma_2$ communication protocols ($\Sigma_2^{\mathsf{cc}}$ protocols). We begin with a formal definition of such protocols.

**Definition 1.13** ($\Sigma_2^{\mathsf{cc}}$ Protocol [BFS86])**.** *Let $F : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a function. A $\Sigma_2^{\mathsf{cc}}$ protocol $\Pi$ for $F$ is specified as follows:*

- *There are two players, Alice holds input $x \in \mathcal{X}$ and Bob holds input $y \in \mathcal{Y}$.*
- *There are two provers Merlin and Megan.*
- *Merlin sends a string $a \in \{0,1\}^{m_1}$ and Megan sends a string $b \in \{0,1\}^{m_2}$ (which are functions of both $x$ and $y$) to both Alice and Bob. Then Alice and Bob communicate $\ell$ bits with each other, and Alice decides whether to accept or reject the pair $(a,b)$.*
- *$F(x,y) = 1$ if and only if there exists a string $a$ from Merlin, such that for all strings $b$ from Megan, Alice accepts $(a,b)$ after communications with Bob.*

*We say the protocol $\Pi$ is computationally-efficient, if both Alice and Bob's response functions can be computed in polynomial time with respect to their input length.*

We show that for any function $F$, if $F$ admits a certain efficient $\Sigma_2^{\mathsf{cc}}$ protocol, then $F$-Satisfying-Pair can be efficiently reduced to OV. Formally, we have:

**Theorem 1.14.** *Let $F : \{0,1\}^d \times \{0,1\}^d \to \{0,1\}$ and $n \in \mathbb{N}$, suppose $F$ has a computationally-efficient $\Sigma_2^{\mathsf{cc}}$ protocol, in which Merlin sends $m_1$ bits, Megan sends $m_2$ bits, and Alice and Bob communicate $\ell$ bits. Then there is a reduction from every $F$-Satisfying-Pair$_n$ instance $I$ to OV$_{n,2^{(m_2+\ell)}}$ instances $J_1, J_2, \ldots, J_{2^{m_1}}$, such that $I$ is a yes instance if and only if there is a $j$ such that $J_j$ is a yes instance. The reduction takes $n \cdot 2^{O(m_1+m_2+\ell)} \cdot \mathrm{poly}(d)$ time.*

**Applications.** We use Theorem 1.14 to establish the equivalence between OV, Min-IP / Max-IP, Apx-Max-IP / Apx-Min-IP and Exact-IP. Previous works have established that OV can be reduced to all these problems, and that these problems can be reduced to Exact-IP. So it suffices for us to construct a reduction from Exact-IP to OV. Let the IP$_{d,m} : \{0,1\}^d \times \{0,1\}^d \to \{0,1\}$ be the function that checks whether $\langle x, y\rangle = m$, Exact-IP is IP$_{d,m}$-Satisfying-Pair, so we can apply Theorem 1.14 with an efficient $\Sigma_2^{\mathsf{cc}}$ protocol for IP$_{d,m}$. More applications can be found in the full version of the paper.

[10] This notation is borrowed from [AHWW16], which studied the Satisfying Pair problem for Branching Programs.

### 1.3.2 Locality-sensitive Hashing (LSH) Families and Reductions to Additive Approximation to Max-IP

To establish equivalence between OV and other approximation problems, we make use of a connection with LSH families. We begin with a generalized definition of an LSH family for a partial function. In the following, let $\mathcal{X}$ be an arbitrary set.

**Definition 1.15.** *Let $f : \mathcal{X} \times \mathcal{X} \to \{0, 1, \perp\}$[11]. We say $f$ admits a $(p_1, p_2)$-sensitive LSH family, if there is a family $\mathcal{F}$ of functions $h : \mathcal{X} \to \mathcal{S}$, such that for any $x, y \in \mathcal{X}$, a uniformly random function $h \in \mathcal{F}$ satisfies:*

- *If $f(x, y) = 1$, then $h(x) = h(y)$ with probability at least $p_1$.*
- *If $f(x, y) = 0$, then $h(x) = h(y)$ with probability at most $p_2$.*

*In addition, we require that $h$ can be efficiently drawn from $\mathcal{F}$, and $h(p)$ can be efficiently computed.*[12]

The usual LSH families for a metric space are special cases of the above generalized definition.

**Definition 1.16.** *For a function $\mathsf{dist} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$, we say $\mathsf{dist}$ admits an LSH family, if for all $\varepsilon > 0$ and real $R > 0$, there are two reals $p_1 = p_1(\varepsilon)$ and $p_2 = p_2(\varepsilon)$ such that the function $f^{\mathsf{dist}}_{R,(1+\varepsilon)R} : \mathcal{X} \times \mathcal{X} \to \{0, 1, \perp\}$ defined as*

$$f^{\mathsf{dist}}_{R,(1+\varepsilon)R}(x, y) = \begin{cases} 1 & \mathsf{dist}(x, y) \leq R, \\ 0 & \mathsf{dist}(x, y) \geq (1 + \varepsilon) \cdot R, \\ \perp & \text{otherwise,} \end{cases}$$

*admits a $(p_1, p_2)$-sensitive LSH family and $p_1 > p_2$.*

In particular, we show that an LSH family for a function implies a reduction to additively approximating Max-IP, which can in turn be reduced to OV. To formally state our reduction, we need to define $\mathcal{F}$-Satisfying-Pair for a partial function $\mathcal{F}$.

**Definition 1.17.** *For a partial function $\mathcal{F} : \mathcal{X} \times \mathcal{X} \to \{0, 1, \perp\}$, $\mathcal{F}$-Satisfying-Pair$_n$ is the problem: given two sets $A, B \subseteq \mathcal{X}$ of size $n$, distinguish between the two cases:*

- *There is an $(x, y) \in A \times B$ such that $\mathcal{F}(x, y) = 1$.*
- *For all $(x, y) \in A \times B$, $\mathcal{F}(x, y) = 0$.*

**Remark 1.18.** *Let $\mathcal{X}$ be $\mathcal{R}^d$, and set $\mathcal{F}(x, y) = 1$ for $\|x - y\| \leq R$, $\mathcal{F}(x, y) = 0$ for $\|x - y\| \geq (1 + \varepsilon) \cdot R$ and undefined otherwise. Then $\mathcal{F}$-Satisfying-Pair distinguishes between the cases that the minimum distance between $A$ and $B$ is $\leq R$ and $\geq (1 + \varepsilon) \cdot R$, which is the decision version of $(1 + \varepsilon)$-approximation to Bichrom.-Closest-Pair.*

Now we are ready to state our general reduction.

**Theorem 1.19.** *Suppose $f : \mathcal{X} \times \mathcal{X} \to \{0, 1, \perp\}$ admits a $(p_1, p_2)$-sensitive LSH family. Let $\varepsilon = p_1 - p_2$.*

*Then there is a randomized reduction from $f$-Satisfying-Pair$_n$ to computing an $\varepsilon/8 \cdot d$ additive approximation to Max-IP$_{n,d}$ with $d = O(\varepsilon^{-2} \log n)$, which succeeds with probability at least $1 - 1/n$.*

From Theorem 1.19, reductions from Bichrom.-$\ell_2$-Closest-Pair and Furthest-Pair to OV follows:

**Corollary 1.20.** *For a distance function $\mathsf{dist} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ which admits an LSH family, Bichrom.-Closest-Pair$_{n,\mathsf{dist}}$ and Furthest-Pair$_{n,\mathsf{dist}}$ can be approximated in truly subquadratic time if OV is in truly subquadratic time.*

[11] $f(x, y) = \perp$ means $f$ is "undefined" on $(x, y)$.

[12] Being efficient here means the running time is polynomial in the bit complexity of the input.

**Applications.** We use Theorem 1.19 and Corollary 1.20 to establish the equivalence between OV and all approximation problems listed in Theorem 1.1. In particular, the $\ell_p$ metric and Jaccard Index admit efficient LSH families via $p$-stable distributions and the minHash method, which implies that they can be reduced to OV by Theorem 1.19.

## 1.4 Related Works

**Equivalence Classes in Fine-Grained Complexity.** It is known that the All-Pairs Shortest Paths problem is sub-cubic time equivalent to many other problems [VW10, BDT16, AGW15, LWW18]. A partial list includes: Negative Triangle, Triangle Listing, Shortest Cycle, 2nd Shortest Path, Max Subarray, Graph Median, Graph Radius and Wiener Index (see [Vas18] for more details on the APSP equivalence class).

In [GIKW17], it is shown that "moderate-dimensional" OV (i.e., OV with $n^\delta$ dimensions for some $\delta > 0$) is equivalent to High-dimension Sparse OV, High-dimension 2-Set Cover, and High-dimension Sperner Family. It is also shown that for every $(k+1)$-quantifier first-order property, its model-checking problem can be reduced to Sparse $k$-OV. In [CGL$^+$18], an equivalence class for Closest-LCS-Pair[13] is established, in particular, it shows Closest-LCS-Pair and its (constant factor) approximate version are equivalent. In [CMWW17], the authors present an equivalence class for $(\min, +)$-convolution, including some variants of the classical knapsack problem and problems related to subadditive sequences.

**Faster-Than-Brute-Force Algorithms for Problems in the Equivalence Class.** Most of the problems listed in Theorem 1.1 have algorithms with some non-trivial speed-up depending on $c$ (when the dimension is $c \log n$) or $\varepsilon$ (when the approximation ratio is $1+\varepsilon$). Table 1 gives the state-of-the-art runtime bounds for these problems.

| Problem | $n^{2-\delta}$ time, $\delta = f(c)$ or $f(\varepsilon)$ |
|---|---|
| OV | $1/O(\log c)$ [AWY15, CW16] |
| Min-IP & Max-IP | $1/\widetilde{O}(\sqrt{c})$ [ACW16] |
| Exact-IP | $1/\widetilde{O}(c)$ [AW15] |
| Apx-Max-IP | $1/O(\log c)$ [Che18] [This paper] |
| Apx-Min-IP | $1/O(\log c)$ [This paper] |
| B.-$\ell_2$-Closest-Pair | $\widetilde{O}(\varepsilon^{1/3})$ [ACW16] |
| $\ell_p$-Furthest-Pair | $\widetilde{O}(\varepsilon^{1/3})$ [ACW16][14] |

Table 1: The best known running-time exponents for the problems shown (in this paper) to be equivalent to OV.

**Fine-Grained Complexity and Communication Complexity.** The connection between communication complexity and Fine-Grained Complexity dates back at least to [PW10], in which it is shown that a sub-linear, computational efficient protocol for 3-party Number-On-Forehead Set-Disjointness problem would refute SETH. The work of [ARW17] shows hardness for approximate version for a host of important problems in P, using the $\widetilde{O}(\sqrt{n})$ MA communication protocol for Set-Disjointness [AW09].

Using Algebraic Geometry codes, [Rub18] obtains a better MA protocol, which in turn improves the efficiency of the previous "distributed PCP" construction of [ARW17]. He then shows $n^{2-o(1)}$-time hardness

[13] Closest-LCS-Pair is: given two sets $A, B$ of strings, compute $\max_{(a,b)\in A\times B} \mathsf{LCS}(a,b)$.

[14] [ACW16] only discussed Bichrom.-$\ell_p$-Closest-Pair when $p \in \{1,2\}$, but one can observe that their algorithm in fact works equally well with Bichrom.-$\ell_p$-Closest-Pair and $\ell_p$-Furthest-Pair for $p \in [1,2]$.

for $1 + o(1)$-approximations to Bichromatic Closest Pair and $o(d)$-additive approximations to $\mathsf{Max\text{-}IP}_{n,d}$ with this new technique. [KLM18] use the Distributed PCP framework to derive inapproximability results for $k$-Dominating Set under various assumptions. In particular, building on the techniques of [Rub18], it is shown that under SETH, $k$-Dominating Set has no $(\log n)^{1/\operatorname{poly}(k,e(\varepsilon))}$ approximation in $n^{k-\varepsilon}$ time[15].

[AR18] make use of the $\widetilde{O}(\log n)$ $\mathsf{IP}$ communication protocol for Set-Disjointness in [AW09], and shows a fast deterministic approximation algorithm to Longest Common Subsequence has interesting circuit lower bound consequences. Making use of the $\mathsf{IP}$ communication protocol for low-space computation, [CGL$^+$18] establish an equivalence class for Closest-LCS-Pair.

[Che18] establishes a connection between hardness of the furthest pair problem in low dimensional Euclidean space and $\mathsf{NP} \cdot \mathsf{UPP}$ communication protocols for Set-Disjointness. He also shows the $\mathsf{BQP}$ communication protocol for Set-Disjointness [BCW98] can be used to derive an inapproximability result for $\{-1,1\}$-Max-IP.[16]

# 2 Preliminaries

In this paper, we use $\mathbb{R}^+$ to denote the set of all positive reals. For notational convenience, we first give the formal definitions of the problem we study in this paper.

## 2.1 Problem List

**Definition 2.1** (Boolean Vector Problem List)**.** *For $n, d \in \mathbb{N}$, we define several problems. For all of them, the input is the same: we are given sets $A$ and $B$ of $n$ vectors from $\{0,1\}^d$.*

1. $\mathsf{OV}_{n,d}$[17]*:* Given $A, B \subseteq \{0,1\}^d$ with $|A| = |B| = n$, determine whether there exists $(a,b) \in A \times B$ such that $a \cdot b = 0$.
2. $\mathsf{Exact\text{-}IP}_{n,d}$*:* Given $A, B$ as before, and an integer $0 \leq m \leq d$, determine whether there exists $(a,b) \in A \times B$ such that $a \cdot b = m$.
3. $\mathsf{Max\text{-}IP}_{n,d}$*:* Given $A, B$ as before, compute
$$\mathsf{Max}(A,B) := \max_{a \in A, b \in B} a \cdot b.$$
4. $\mathsf{Min\text{-}IP}_{n,d}$*:* Given $A, B$ as before, compute
$$\mathsf{Min}(A,B) := \min_{a \in A, b \in B} a \cdot b.$$
5. $\mathsf{Apx\text{-}Max\text{-}IP}_{n,d}$*:* Given $A, B$ as before, output a number $\widetilde{\mathsf{Max}}(A,B) \in [\mathsf{Max}(A,B)/2, \mathsf{Max}(A,B)]$.
6. $\mathsf{Apx\text{-}Min\text{-}IP}_{n,d}$*:* Given $A, B$ as before, output a number $\widetilde{\mathsf{Min}}(A,B) \in [\mathsf{Min}(A,B), 2 \cdot \mathsf{Min}(A,B)]$.

**Remark 2.2.** *The constant factor 2 in the definitions of Apx-Min-IP and Apx-Max-IP is only chosen for convenience, it can be replaced by any constant $\kappa > 1$ (such as $1.001$, or $100$).*

**Definition 2.3** (Other Problems)**.** *We define the following problems.*

[15] where $e$ is a certain function from $\mathbb{R}^+ \to \mathbb{N}$

[16] the variant of Max-IP with vectors from $\{-1,1\}^d$ instead of $\{0,1\}^d$

[17] Note that we consider the red-blue version of OV in this paper for convenience, and it is equivalent to the original monochromatic version.

1. *Bichrom.-$\ell_p$-Closest-Pair$_n$: For a fixed real $p \in [1,2]$, given two sets $A, B$ of $n$ points in $\mathbb{R}^d$ where $d = n^{o(1)}$, compute* $\min_{(a,b)\in A\times B} \|a-b\|_p$.

2. *$\ell_p$-Furthest-Pair$_n$: For a fixed real $p \in [1,2]$, given a set $A$ of $n$ points in $\mathbb{R}^d$ where $d = n^{o(1)}$, compute* $\max_{(a,b)\in A\times A} \|a-b\|_p$.

3. *Jaccard-Index-Pair$_n$: Given $A, B$ as two collections of $n$ sets of size $n^{o(1)}$, compute* $\max_{(S,T)\in A\times B} J(S,T)$, *where* $J(S,T) := \frac{|S\cap T|}{|S\cup T|}$.

### 2.2 Locality-sensitive Hashing

In this paper we apply some well-known results from the theory of *locality-sensitive hashing* (LSH) (See [WSSJ14, AIR18] for excellent recent references on LSH families and their applications).

**$\ell_p$ Norm.** From the theory of $p$-stable distributions, LSH families for $\ell_p$ norm when $p \in [1,2]$ have been constructed.

**Lemma 2.4** ([DIIM04])**.** *For a constant $p \in [1,2]$, the $\ell_p$ distance $\mathsf{dist}_p(x,y) := \|x-y\|_p$ admits a LSH family. Moreover, for all real $\varepsilon \in (0, 0.1)$ and real $R > 0$, $f^{\mathsf{dist}_p}_{R,(1+\varepsilon)R}$ admits a $(p_1,p_2)$-sensitive LSH family, such that $p_1 - p_2 \geq \Omega(\varepsilon)$.*

**Jaccard Index.** For two sets $A, B$, recall that their Jaccard index is defined as $J(A,B) := \frac{|A\cap B|}{|A\cup B|}$. It is well-known that this measure admits a LSH family by the MinHash method.

**Lemma 2.5** ([Bro97])**.** *Let $0 \leq p_2 < p_1 \leq 1$ be two reals, and $f$ be the function on two sets such that $f(A,B) = 1$ when $J(A,B) \geq p_1$, $f(A,B) = 0$ when $J(A,B) \leq p_2$ and undefined otherwise. $f$ admits a $(p_1,p_2)$-sensitive LSH family.*

## 3 General Reduction Frameworks with $\Sigma_2$ Communication Protocols and LSH Families

In this section we present two general reduction frameworks for showing equivalence to OV.

### 3.1 $\Sigma_2$ Communication Protocols and Reductions to OV

We first show that an efficient $\Sigma_2^{\mathsf{cc}}$ protocol for a function $f$ implies a reduction from $f$-Satisfying-Pair to OV.

**Reminder of Theorem 1.14** *Let $F : \{0,1\}^d \times \{0,1\}^d \to \{0,1\}$ and $n \in \mathbb{N}$, suppose $F$ has a computationally-efficient $\Sigma_2^{\mathsf{cc}}$ protocol, in which Merlin sends $m_1$ bits, Megan sends $m_2$ bits, and Alice and Bob communicate $\ell$ bits. Then there is a reduction from every $F$-Satisfying-Pair$_n$ instance $I$ to OV$_{n,2^{(m_2+\ell)}}$ instances $J_1, J_2, \ldots, J_{2^{m_1}}$, such that $I$ is a yes instance if and only if there is a $j$ such that $J_j$ is a yes instance. The reduction takes $n \cdot 2^{O(m_1+m_2+\ell)} \cdot \mathrm{poly}(d)$ time.*

*Proof of Theorem 1.14.* Let $F$ and $\Pi$ be the given function and $\Pi$ be its $\Sigma_2$ protocol. Fix $a \in \{0,1\}^{m_1}$ and $b \in \{0,1\}^{m_2}$ as the proofs from Merlin and Megan. Let $w_1, w_2, \ldots, w_{2^\ell}$ be an enumeration of all possible communication transcripts between Alice and Bob (note they communicate $\ell$ bits). We define two binary vectors $R_x(a,b), R_y(a,b) \in \{0,1\}^{2^\ell}$ as follows: for all $a, b$, $R_x(a,b)_i = 1$ ($R_y(a,b)_i = 1$) if and only if the

transcript $w_i$ is consistent with Alice's input $x$ (Bob's input $y$), and $w_i$ makes Alice reject. Note that since the transcript is uniquely determined by $x$, $y$, $a$ and $b$, only one $w_i$ is consistent with both $x$ and $y$ given the pair $(a, b)$. It follows that $\langle R_x(a, b), R_y(a, b)\rangle = 0$ if and only if Alice accepts the pair $(a, b)$.

Now, suppose we are given an $F$-Satisfying-Pair$_n$ instance $I$ with sets $A$ and $B$ of $n$ vectors from $\{0, 1\}^d$. We first enumerate Merlin's possible string $a \in \{0, 1\}^{m_1}$, and use $R_x(a, \cdot)$ to denote the string obtained by concatenating all $R_x(a, b)$'s for $b \in \{0, 1\}^{m_2}$. $R_y(a, \cdot)$ is defined similarly. For each $a$, let $A_a$ be the set of $R_x(a, \cdot) \in \{0, 1\}^{m_2+\ell}$ for all $x \in A$, and $B_a$ be the set of $R_y(a, \cdot) \in \{0, 1\}^{m_2+\ell}$ for all $y \in B$.

We claim $I$ is a yes instance if and only if some pair $(A_a, B_a)$ is a yes instance for OV.

- Suppose $I$ is a yes instance. Then there is an $(x, y) \in A \times B$ such that $F(x, y) = 1$. By the definition of $\Sigma_2^{\mathsf{cc}}$ protocols and our constructions, there is an $a \in \{0, 1\}^{m_1}$ such that for all $b \in \{0, 1\}^{m_2}$ we have $\langle R_x(a, b), R_y(a, b)\rangle = 0$. Hence, for such an $a$, $\langle R_x(a, \cdot), R_y(a, \cdot)\rangle = 0$, and therefore $(A_a, B_a)$ is a yes instance for OV.
- Suppose $I$ is a no instance. Then for all $(x, y) \in A \times B$, $F(x, y) = 0$. Hence, for all $a \in \{0, 1\}^{m_1}$ and all $(x, y) \in A \times B$, we have $\langle R_x(a, \cdot), R_y(a, \cdot)\rangle \neq 0$, which means all $(A_a, B_a)$'s are no instances for OV.

Finally, since $\Pi$ is computationally-efficient, the above reduction takes $O(n \cdot 2^{O(m_1+m_2+\ell)} \cdot \mathrm{poly}(d))$ time, which completes the proof. □

### 3.2 LSH Families and Reductions to Additive Approximate Max-IP

Next, we show that an efficient LSH family implies a reduction to additively approximating Max-IP.

**Reminder of Theorem 1.19** *Suppose $f : \mathcal{X} \times \mathcal{X} \to \{0, 1, \perp\}$ admits a $(p_1, p_2)$-sensitive LSH family. Let $\varepsilon = p_1 - p_2$.*

*Then there is a randomized reduction from $f$-Satisfying-Pair$_n$ to computing an $\varepsilon/8 \cdot d$ additive approximation to Max-IP$_{n,d}$ with $d = O(\varepsilon^{-2} \log n)$, which succeeds with probability at least $1 - 1/n$.*

*Proof.* Let $\mathcal{F}$ be the corresponding $(p_1, p_2)$-sensitive LSH family, and $\mathcal{S}$ be the co-domain for hash functions from $\mathcal{F}$. Consider the following process: draw $h$ from $\mathcal{F}$ uniformly at random, then map each item in $\mathcal{S}$ independently to the string $(0, 1)$ or $(1, 0)$, each with probability $0.5$. Let this map be $\varphi$. Composing $h$ and $\varphi$, we obtain a function $g(x) = \varphi(h(x))$ such that:

- If $f(x, y) = 1$, then $\langle g(x), g(y)\rangle = 1$ with probability at least $p_1 + (1 - p_1)/2 \geq \frac{1}{2} + \frac{1}{2} \cdot p_1$.
- If $f(x, y) = 0$, then $\langle g(x), g(y)\rangle = 1$ with probability at most $p_2 + (1 - p_2)/2 \leq \frac{1}{2} + \frac{1}{2} \cdot p_2$.

Repeat the above process for $N = c \log n$ times, independently drawing functions $g_1, g_2, \ldots, g_N$, where $c$ is a parameter to be specified later. We set our reduction $w(x)$ to be the concatenation of all $g_i(x)$'s. Let $\tau_1 = \frac{1}{2} + \frac{1}{2} \cdot (p_1 - \varepsilon/4)$ and $\tau_2 = \frac{1}{2} + \frac{1}{2} \cdot (p_2 + \varepsilon/4)$. By a simple Chernoff bound, there is a real $c_1 = \Theta(\varepsilon^2)$ such that

- If $f(x, y) = 1$, then $\langle w(x), w(y)\rangle > \tau_1 \cdot N$ with probability at least $1 - 2^{c_1 \cdot N}$.
- If $f(x, y) = 0$, then $\langle w(x), w(y)\rangle < \tau_2 \cdot N$ with probability at least $1 - 2^{c_1 \cdot N}$.

Set $c := 3/c_1$, and let $A_{\mathsf{new}}$ (respectively, $B_{\mathsf{new}}$) be the set of $w(a)$'s for all $a \in A$ (the set of $w(b)$'s for all $b \in B$). It follows that with probability at least $1 - 1/n$, if there is an $(x, y) \in A \times B$ with $f(x, y) = 1$ then $\mathsf{Max}(A_{\mathsf{new}}, B_{\mathsf{new}}) > \tau_1 \cdot N$, and if $f(x, y) = 0$ for all $(x, y) \in A \times B$, then $\mathsf{Max}(A_{\mathsf{new}}, B_{\mathsf{new}}) < \tau_2 \cdot N$. Observe this reduction satisfies the desired approximation property. □

# 4 An Equivalence Class for Orthogonal Vectors

In this section we apply our two general frameworks to prove Theorem 1.1.

## 4.1 Equivalence Between Boolean Vectors Problem

We first show that all Boolean vectors problems listed in Theorem 1.1 can be trivially reduced to Exact-IP, and OV can be reduced to all of them.

**Lemma 4.1.** *The following holds:*

- *If Exact-IP is in truly subquadratic time, then so are OV, Apx-Min-IP (Apx-Max-IP) and Max-IP (Min-IP).*
- *If any of Apx-Min-IP (Apx-Max-IP), Max-IP (Min-IP) and Exact-IP is in truly subquadratic time, then so is OV.*

*Proof.* For the first item, Apx-Min-IP (Apx-Max-IP) and Max-IP (Min-IP) can all be trivially reduced to Exact-IP, and OV can be reduced to Max-IP by [Wil05].

For the second item, the case of Apx-Max-IP follows from Theorem 4.1 in [Rub18], and it is easy to see that OV can be trivially reduced to Min-IP or Apx-Min-IP (OV is equivalent to asking whether the minimum inner product is zero). □

Therefore, all we need is a reduction from Exact-IP to OV. We provide it by constructing a good $\Sigma_2$ communication protocol, and applying Theorem 1.14.

**Lemma 4.2.** *If OV is in truly subquadratic time, then so is Exact-IP.*

**Proposition 4.3.** *Let $\mathsf{IP}_{n,k} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be the function that checks whether $\langle x, y\rangle = k$. For all $n, k \in \mathbb{Z}^+$, and a parameter $1 \le \ell \le n$, there is a $\Sigma_2^{\mathsf{cc}}$ computationally-efficient protocol for $\mathsf{IP}_{n,k}$ in which Merlin sends $\ell \cdot \lceil \log(\lceil n/\ell \rceil + 1)\rceil$ bits, Megan sends $\lceil \log \ell \rceil$ bits and Alice and Bob communicate $\lceil n/\ell \rceil$ bits.*

*Proof.* We assume $\ell$ divides $n$ for simplicity. Let $x, y$ be the inputs of Alice and Bob, respectively. We partition $x$ into $\ell$ equally-sized groups of length $n/\ell$, let them be $x_1, x_2, \ldots, x_\ell$. Similarly, we partition $y$ into groups $y_1, y_2, \ldots, y_\ell$. Clearly, $\langle x, y\rangle = \sum_{i=1}^{\ell} \langle x_i, y_i\rangle$.

Merlin's message is a vector $\psi \in \{0, 1, \ldots, n/\ell\}^\ell$, where $\psi_i$ is intended to be $\langle x_i, y_i\rangle$.

Alice rejects immediately if $\sum_{i=1}^{\ell} \psi_i \neq k$, regardless of Megan's message. Otherwise, Megan's message is an index $i$ in $[\ell]$. Bob sends $y_i$ to Alice, and Alice accepts if and only if $\langle x_i, y_i\rangle = \psi_i$.

We argue the protocol correctly decides $\mathsf{IP}_{n,k}$. If $\langle x, y\rangle = k$, it is easy to see that for the correct $\psi$, Alice accepts all messages from Megan (and Bob). When $\langle x, y\rangle \neq k$, for all $\psi$ such that $\sum_{i=1}^{\ell} \psi_i = k$ (otherwise Alice always rejects), there must be an $i$ such that $\langle x_i, y_i\rangle \neq \psi_i$, which means Alice rejects on the pair $\psi$ and $i$. Finally, it is easy to see that the protocol satisfies the requirements of computational efficiency, which completes the proof. □

Now we are ready to prove Lemma 4.2.

*Proof of Lemma 4.2.* Suppose there is a universal constant $\delta > 0$ such that for all constants $c'$, $\mathsf{OV}_{n,c'\log n}$ can be solved in $n^{2-\delta}$ time. Let $c$ be an arbitrary constant.

Observe that an $\mathsf{Exact\text{-}IP}_{n,c\log n}$ instance with target integer $m$, is simply a $\mathsf{IP}_{c\log n,m}$-$\mathsf{Satisfying\text{-}Pair}_n$ instance. Set $\ell := \varepsilon \cdot \log n$ for an $\varepsilon > 0$ to be specified later. By Proposition 4.3, there is a $\Sigma_2^{\mathsf{cc}}$ protocol for

$\mathsf{IP}_{c\log n,m}$ such that Merlin sends $\varepsilon \cdot \log(c/\varepsilon) \cdot \log n$ bits, Megan sends $\log(\varepsilon \log n)$ bits and Alice and Bob communicate $c/\varepsilon$ bits.

By Theorem 1.14, there is a reduction from an $\mathsf{Exact\text{-}IP}_{n,c\log n}$ instance to $2^{\varepsilon \log(c/\varepsilon)\log n} = n^{\varepsilon \log(c/\varepsilon)}$ many $\mathsf{OV}_{n,O(2^{c/\varepsilon}\log n)}$ instances. We can set $\varepsilon$ so that $\varepsilon \log(c/\varepsilon) < \delta/2$. Note that $\varepsilon$ only depends on $c$ and $\delta$, so it is still a fixed constant, which means (by assumption) that $\mathsf{OV}_{n,O(2^{c/\varepsilon}\log n)}$ can be solved in $n^{2-\delta}$ time. Applying the algorithm for OV, we get an $n^{2-\delta/2}$ time algorithm for $\mathsf{Exact\text{-}IP}_{n,c\log n}$, which completes the proof. □

## 4.2 Equivalences Between OV and Approximation Problems

Now we deal with approximation problems in Theorem 1.1.

### Bichrom.-$\ell_p$-Closest-Pair and $\ell_p$-Furthest-Pair

We first show OV is equivalent to approximate Bichrom.-$\ell_p$-Closest-Pair, $\ell_p$-Furthest-Pair and additive approximate Max-IP. One direction is already established in [Rub18].

**Lemma 4.4** (Theorem 4.1 of [Rub18])**.** *If Bichrom.-$\ell_p$-Closest-Pair or $\ell_p$-Furthest-Pair can be approximated in truly subquadratic time for any $p \in [1,2]$ or Max-IP can be additively approximated in truly subquadratic time, then OV is in truly subquadratic time.*[18]

In the following we show the reverse also holds.

**Lemma 4.5.** *If OV is in truly-subquadratic time, then for all $p \in [1,2]$, Bichrom.-$\ell_p$-Closest-Pair and $\ell_p$-Furthest-Pair can be approximated in truly subquadratic time, and Max-IP can be additively approximated in truly subquadratic time.*

We are going to apply Theorem 1.19 and will actually prove a much stronger result. We show that for any metric $\mathsf{dist} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ which admits a *Locality-sensitive hashing* (LSH) family, approximate Bichrom.-Closest-Pair and Furthest-Pair with respect to dist can be efficiently reduced to OV.

In the following, we use $\mathsf{Bichrom.\text{-}Closest\text{-}Pair}_{n,\mathsf{dist}}$ and $\mathsf{Furthest\text{-}Pair}_{n,\mathsf{dist}}$ to denote the corresponding problems with respect to the metric dist. Now we are ready to give the reduction.

**Reminder of Corollary 1.20** *For a distance function $\mathsf{dist} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ which admits an LSH family, $\mathsf{Bichrom.\text{-}Closest\text{-}Pair}_{n,\mathsf{dist}}$ and $\mathsf{Furthest\text{-}Pair}_{n,\mathsf{dist}}$ can be approximated in truly subquadratic time if OV is in truly subquadratic time.*

*Proof.* Suppose OV is in truly subquadratic time. By Lemma 4.1 and Lemma 4.2, Max-IP and Min-IP are also in truly-subquadratic time. In the following we only discuss $\mathsf{Bichrom.\text{-}Closest\text{-}Pair}_{n,\mathsf{dist}}$; the reduction for $\mathsf{Furthest\text{-}Pair}_{n,\mathsf{dist}}$ is analogous (with Min-IP in place of Max-IP).

Let $\varepsilon > 0$ be an arbitrary constant. We want to approximate the minimum distance between two sets $A$ and $B$ of $n$ elements from $\mathcal{X}$ within a $(1+\varepsilon)$ multiplicative factor. By a standard (simple) search to decision reduction that incurs only a negligible factor in the running time, we only have to consider the decision version, in which you are given a real $R$, and want to distinguish the following two cases: (1) $\min_{(a,b)\in A\times B} d(a,b) \leq R$; (2) $\min_{(a,b)\in A\times B} d(a,b) \geq (1+\varepsilon)\cdot R$.

By Theorem 1.19, this decision problem can be reduced to additive approximation to $\mathsf{Max\text{-}IP}_{n,O(\log n)}$, which is in truly-subquadratic time by Lemma 4.2. This completes the proof. □

[18] [Rub18] only discussed Bichrom.-$\ell_p$-Closest-Pair and additive approximation to Max-IP, but it is easy to see that the proof also works for $\ell_p$-Furthest-Pair.

Now, from the LSH families for $\ell_p$-metric, Lemma 4.5 follows directly.

*Proof of Lemma 4.5.* Assume OV is in truly-subquadratic time. It follows directly from Corollary 1.20 and Lemma 2.4 that for all $p \in [1, 2]$, Bichrom.-$\ell_p$-Closest-Pair and $\ell_p$-Furthest-Pair can be approximated in truly subquadratic time.

Also, by a simple random sampling method and a Chernoff bound (see e.g. Lemma 3.6 of [Che18]), computing an $\varepsilon \cdot d$ additive approximation to Max-IP$_{n,d}$ can be reduced to Max-IP$_{n,O(\varepsilon^{-2}\log n)}$, which can be solved in truly-subquadratic time by Lemma 4.2 and Lemma 4.1. □

### Jaccard-Index-Pair

Finally, we show the equivalence between OV and approximate Jaccard-Index-Pair.

**Lemma 4.6.** *OV is in truly-subquadratic time if and only if Jaccard-Index-Pair can be additively approximated in truly-subquadratic time.*

*Proof.* For one direction, suppose OV is in truly subquadratic time. Using a similar argument as in Corollary 1.20, from Lemma 2.5 and Theorem 1.19 it follows that Jaccard-Index-Pair can be additively approximated in truly-subquadratic time.

For the other direction, suppose Jaccard-Index-Pair can be additively approximated in truly subquadratic time. By Lemma 4.4, it suffices to show that Max-IP can be additively approximated in truly-subquadratic time. Given a Max-IP$_{n,d}$ instance with sets $A, B$ consisting of $n$ vectors from $\{0,1\}^d$, suppose we want to compute an $\varepsilon \cdot d$ approximation to it. In the following we show how to reduce it to a Jaccard-Index-Pair instance.

We begin by setting up some notation. For $t \in [d]$, we use $e^{[t]}$ to denote the Boolean vector $1^t 0^{d-t}$ from $\{0,1\}^d$ (that is, the first $t$ coordinates are 1, and the rest are 0). For two vectors $a, b$, we use $a \circ b$ to denote their concatenation.

For each $x \in A \subseteq \{0,1\}^d$ and $y \in B \subseteq \{0,1\}^d$, we create two vectors $\widehat{x}, \widehat{y} \in \{0,1\}^{3d}$, as follows:

$$\widehat{x} = x \circ e^{[d-\|x\|_1]} \circ e^{[0]}, \widehat{y} = y \circ e^{[0]} \circ e^{[d-\|y\|_1]}.$$

Interpreting $\widehat{x}$ and $\widehat{y}$ as indicator vectors, we create their corresponding sets $S_x, T_y \subseteq [3d]$. That is, for $i \in [3d]$, $\widehat{x}_i = 1$ if and only if $i \in S_x$ (the same holds for $\widehat{y}$ and $T_y$). Observe that

$$J(S_x, T_y) = \frac{|S_x \cap T_y|}{|S_x \cup T_y|} = \frac{\langle x, y\rangle}{2d - \langle x, y\rangle}. \tag{1}$$

Now we create $\widehat{A}$ and $\widehat{B}$ as the sets of all $S_x$ for $x \in A$ and $T_y$ for $y \in B$. Let $t = \max_{(S,T)\in\widehat{A}\times\widehat{B}} J(S,T)$ and $w = \max_{(a,b)\in A\times B}\langle a, b\rangle$. From Equation (1), we can see $t = \frac{w}{2d-w}$ and $w = d \cdot 2 \cdot \frac{t}{t+1}$. Therefore, an $\varepsilon/3$ approximation to $t$ is enough to obtain an $\varepsilon \cdot d$ approximation to $w$, which completes the reduction. □

And Theorem 1.1 follows from Lemma 4.1, Lemma 4.2, Lemma 4.4, Lemma 4.5 and Lemma 4.6.

## 5 Equivalences for Moderate Dimensional Problems

In this section we prove our equivalence theorems for moderate dimensional Boolean vectors problems.

## 5.1 OV and Apx-Min-IP

We first show moderate dimensional OV and Apx-Min-IP are equivalent.

**Reminder of Theorem 1.5** *Moderate dimensional OV is in truly subquadratic time if and only if moderate dimensional Apx-Min-IP is.*

To prove Theorem 1.5, we construct the following reduction.

**Lemma 5.1.** *For all integers $n, d$ and a parameter $\varepsilon > 0$, an $\mathsf{Apx\text{-}Min\text{-}IP}_{n,d}$ instance can be reduced to $n^{O(\varepsilon)}$ $\mathsf{OV}_{n,d^{O(1/\varepsilon)}\log n}$ instances. The reduction is randomized and succeeds with probability at least $2/3$, and it takes $n^{1+O(\varepsilon)} \cdot d^{O(1/\varepsilon)}$ time.*

Before proving Lemma 5.1, we show it implies Theorem 1.5.

*Proof of Theorem 1.5.* Recall that $\mathsf{Min}(A, B) := \min_{(a,b)\in A\times B}\langle a, b\rangle$. For the first direction, note that OV with two sets $A$ and $B$ essentially asks whether $\mathsf{Min}(A, B) = 0$, and a 2-approximation to $\mathsf{Min}(A, B)$ is already enough to answer that question. Therefore, if moderate dimensional Apx-Min-IP is in truly subquadratic time, then so is OV.

For the second direction, suppose there are constants $\varepsilon_1, \delta_1 > 0$ such that $\mathsf{OV}_{n,n^{\delta_1}}$ can be solved in $n^{2-\varepsilon_1}$ time. Let $\varepsilon$ be a parameter to be set later, by Lemma 5.1, there are constants $c_1, c_2$ such that all $\mathsf{Apx\text{-}Min\text{-}IP}_{n,n^\delta}$ instance can be efficiently reduced to $n^{c_1\varepsilon}$ $\mathsf{OV}_{n,n^{\delta c_2/\varepsilon}}$ instances.

We set $\varepsilon$ such that $c_1\varepsilon = \varepsilon_1/2$, and $\delta$ such that $\delta \cdot c_2/\varepsilon < \delta_1$. Then applying the algorithm for OV, $\mathsf{Apx\text{-}Min\text{-}IP}_{n,n^\delta}$ can be solved in $n^{2-\varepsilon_1/2}$ time, which completes the proof. □

The following probability inequality will be useful in the proof of Lemma 5.1.

**Lemma 5.2.** *Letting $\varepsilon \in (0, 0.1)$, and $\mathcal{D}$ be a distribution on $\{0, 1\}$ such that $\mathbb{E}_{X\sim\mathcal{D}}[X] = \varepsilon$, there is a universal constant $c$ such that for any integer $m$ and any $cm$ independent random variables $X_1, X_2, \ldots, X_{cm}$ from $\mathcal{D}$, we have*

$$\Pr\left[\sum_{i=1}^{cm} X_i \geq \frac{1}{2} \cdot cm\right] \leq \varepsilon^{-m}.$$

The proof of Lemma 5.2 can be found in the appendix.

Finally, we prove Lemma 5.1.

*Proof of Lemma 5.1.* Before presenting the reduction, we first introduce some notation. For a vector $x \in \{0, 1\}^d$, and a subset $S \subset [d]$, $x_{|S} \in \{0, 1\}^{|S|}$ denotes the projection of $x$ onto the coordinates of $S$. Similarly, for a sequence $T$ of integers from $[d]$, let $x_{|T} \in \{0, 1\}^{|T|}$ denote the projection of $x$ on $T$, such that $\left(x_{|T}\right)_i := x_{T_i}$ for each $i \in [|T|]$. We also use the Iverson bracket notation: for a predicate $P$, $[P]$ takes value 1 when $P$ is true, and 0 otherwise.

**Reduction to a Decision Problem.** Our reduction will focus on a corresponding decision problem: given two sets $A, B$ of $n$ vectors from $\{0, 1\}^d$ and an integer $\tau \leq d/2$, we want to distinguish the following two cases: $\mathsf{Min}(A, B) \geq 2\tau$ or $\mathsf{Min}(A, B) \leq \tau$ (the algorithm can output anything when $\tau < \mathsf{Min}(A, B) < 2\tau$). It is easy to see that via a binary search, $\log d$ calls to this decision problem can be used to solve the original Apx-Min-IP problem, and a factor of $\log d \leq \log n$ can be ignored here.

**One Step Reduction with $\mathcal{D}_T$.** Now, suppose we pick a sequence of $d/\tau$ uniform random numbers from $[d]$ and let $\mathcal{D}_T$ be its distribution. Then for $x, y \in \{0,1\}^d$, we have:

- If $\langle x, y\rangle \le \tau$:
$$\Pr_{T\leftarrow\mathcal{D}_T}[\langle x_{|T}, y_{|T}\rangle = 0] \ge (1-\tau/d)^{d/\tau} \ge \left(1-\frac{1}{2}\right)^2 > 0.25.$$
- If $\langle x, y\rangle \ge 2\tau$:
$$\Pr_{T\leftarrow\mathcal{D}_T}[\langle x_{|T}, y_{|T}\rangle = 0] \le (1-2\tau/d)^{d/\tau} \le e^{-2} < 0.14.$$

The important observation is that there is a constant probability gap between the above two cases.

**A Micro Reduction to OV.** Now, let $N$ be an integer and $\mathcal{D}_T^{\otimes N}$ be the joint distribution of $N$ independent samples from $\mathcal{D}_T$. We write $\{T_i\} \leftarrow \mathcal{D}_T^{\otimes N}$ to denote that $(T_1, T_2, \ldots, T_N)$ is a random sample from $\mathcal{D}_T^{\otimes N}$. By a standard Chernoff bound, when $\{T_i\} \leftarrow \mathcal{D}_T^{\otimes N}$, there is a constant $c_1$ such that:

- If $\langle x, y\rangle \le \tau$:
$$\Pr\left[\sum_{i=1}^{N}\left[\langle x_{|T_i}, y_{|T_i}\rangle = 0\right] > 0.2N\right] \ge 1 - 2^{-c_1 N}.$$
- If $\langle x, y\rangle \ge 2\tau$:
$$\Pr\left[\sum_{i=1}^{N}\left[\langle x_{|T_i}, y_{|T_i}\rangle = 0\right] < 0.2N\right] \ge 1 - 2^{-c_1 N}.$$

Now, for a fixed $\{T_i\}$, we can distinguish the above two cases via a reduction to a "micro" OV instance.

Note that $\sum_{i=1}^{N}\left[\langle x_{|T_i}, y_{|T_i}\rangle = 0\right] > 0.2N$ is equivalent to the condition that there is are $t = 0.8N$ pairs $(i_1, j_1), (i_2, j_2), \ldots, (i_t, j_t) \in [N] \times [d/\tau]$ such that all $i_k$'s are distinct, and for all $k \in [t]$, $\left(x_{|T_{i_k}}\right)_{j_k} \cdot \left(y_{|T_{i_k}}\right)_{j_k} = 1$.

With this observation, we can construct our reduction. There are
$$L = \binom{N}{t} \cdot (d/\tau)^t = (d/\tau)^{O(N)}$$
possible $t$-tuples of pairs. We sort them in an arbitrary but consistent order. Now we construct a mapping $\phi_{\{T_i\}} : \{0,1\}^d \to \{0,1\}^L$ as follows:

For each $\ell \in [L]$, let $(i_1, j_1), (i_2, j_2), \ldots, (i_t, j_t)$ be the $\ell$-th $t$-tuple of pairs. For a vector $z \in \{0,1\}^d$, we set $\phi_{\{T_i\}}(z)_\ell = 1$, iff $\left(z_{|T_{i_k}}\right)_{j_k} = 1$ for all $k \in [t]$.

Then for all $x, y \in \{0,1\}^d$, we have $\sum_{i=1}^{N}\left[\langle x_{|T_i}, y_{|T_i}\rangle = 0\right] > 0.2N$ is further equivalent to $\langle \phi_{\{T_i\}}(x), \phi_{\{T_i\}}(y)\rangle = 0$. For convenience, we let $\mathcal{D}_\phi$ denote the distribution of $\phi_{\{T_i\}}$ when $\{T_i\}$ is drawn from $\mathcal{D}_T^{\otimes N}$ and we set $N = \varepsilon^{-1}/c_1$.

To summarize, we have:

- If $\langle x, y\rangle \le \tau$:
$$\Pr_{\phi\leftarrow\mathcal{D}_\phi}[\langle \phi(x), \phi(y)\rangle = 0] \ge 1 - 2^{-\varepsilon^{-1}}.$$
- If $\langle x, y\rangle \ge 2\tau$:
$$\Pr_{\phi\leftarrow\mathcal{D}_\phi}[\langle \phi(x), \phi(y)\rangle > 0] \ge 1 - 2^{-\varepsilon^{-1}}.$$

**The Final Reduction.** Finally, letting $c_2$ be the universal constant in Lemma 5.2, we pick $m = 3c_2 \cdot \varepsilon \log n$ i.i.d. mappings $\phi_1, \phi_2, \ldots, \phi_m$ from $\mathcal{D}_\phi$. Applying Lemma 5.2, we have:

- If $\langle x, y\rangle \le \tau$:
$$\Pr_{\{\phi_i\}\leftarrow\mathcal{D}_\phi^{\otimes m}}\left[\sum_{i=1}^{m}[\langle\phi_i(x),\phi_i(y)\rangle = 0] > \frac{1}{2}\cdot m\right] \ge 1 - n^{-3}.$$
- If $\langle x, y\rangle \ge 2\tau$:
$$\Pr_{\{\phi_i\}\leftarrow\mathcal{D}_\phi^{\otimes m}}\left[\sum_{i=1}^{m}[\langle\phi_i(x),\phi_i(y)\rangle = 0] < \frac{1}{2}\cdot m\right] \ge 1 - n^{-3}.$$

Now, we use our final reduction to distinguish the above two cases. Note that $\sum_{i=1}^{m}[\langle\phi_i(x),\phi_i(y)\rangle = 0] > \frac{1}{2}\cdot m$ is equivalent to the condition that there is a subset $S \subseteq [m]$ with $|S| > \frac{1}{2}\cdot m$ such that $\langle\phi_i(x),\phi_i(y)\rangle = 0$ for all $i \in S$.

We enumerate all possible such subsets $S$. For a vector $z \in \{0,1\}^d$, we define $\phi_S(z)$ to be the concatenation of $\phi_i(z)$'s for all $i \in S$. We set $A_S$ as the set of all $\phi_S(x)$'s for $x \in A$, and $B_S$ as the set of all $\phi_S(y)$'s for $y \in B$.

Then we can see that $\sum_{i=1}^{m}[\langle\phi_i(x),\phi_i(y)\rangle = 0] > \frac{1}{2}\cdot m$ is further equivalent to whether there is a subset $S$ with $|S| > \frac{1}{2}\cdot m$ and $(A_S, B_S)$ is a yes instance for OV.

**Summary.** Putting everything together, we have a randomized reduction to $T = 2^{O(\varepsilon\log n)} = n^{O(\varepsilon)}$ $OV_{n,(d/\tau)^{O(1/\varepsilon)}\log n}$ instances with set-pairs $(A_1, B_1), (A_2, B_2), \ldots, (A_T, B_T)$ such that, with probability at least $1 - 1/n$:

- If $\mathsf{Min}(A, B) \le \tau$, then one of the $(A_i, B_i)$ is a yes instance for OV.
- If $\mathsf{Min}(A, B) \ge 2\tau$, all $(A_i, B_i)$'s are no instance for OV.

The above completes the proof. □

## 5.2 Exact-IP, Max-IP and Min-IP

Now we proceed to show moderate dimensional Exact-IP, Max-IP and Min-IP are equivalent.

**Reminder of Theorem 1.6** *For moderate dimensional Max-IP, Min-IP and Exact-IP, either all of them are in truly subquadratic time, or none of them are.*

To prove the above theorem, we need the following two simple reductions, whose proofs can be found in the appendix.

**Lemma 5.3.** *There are functions $\psi^x_{\mathsf{rev}}, \psi^y_{\mathsf{rev}} : \{0,1\}^* \to \{0,1\}^*$ such that for all integer $d$ and $x, y \in \{0,1\}^d$, we have $\psi^x_{\mathsf{rev}}(x), \psi^y_{\mathsf{rev}}(y) \in \{0,1\}^{2d}$ and $\langle\psi^x_{\mathsf{rev}}(x), \psi^y_{\mathsf{rev}}(y)\rangle = d - \langle x, y\rangle$.*

**Lemma 5.4.** *For all integers $d$ and $0 \le m \le d$, there are mappings $\varphi^x_{d,m}, \varphi^y_{d,m} : \{0,1\}^d \to \{0,1\}^{O(d^2)}$ and an integer $M_d$, such that for all $x, y \in \{0,1\}^d$:*

- *If $\langle x, y\rangle = m$, then $\langle\varphi^x_{d,m}(x), \varphi^y_{d,m}(y)\rangle = M_d$.*
- *Otherwise, $\langle\varphi^x_{d,m}(x), \varphi^y_{d,m}(y)\rangle > M_d$.*

*Proof of Theorem 1.6.* By Lemma 5.3, one can easily reduce a Max-IP$_{n,d}$ instance to a Min-IP$_{n,2d}$ and vice versa. Therefore, moderate dimensional Max-IP and Min-IP are truly-subquadratic equivalent. We only need to show that moderate dimensional Min-IP and Exact-IP are equivalent.

Assuming moderate dimensional Exact-IP is in truly subquadratic time, so there are two constants $\varepsilon$ and $\delta$ such that Exact-IP$_{n,n^\delta}$ can be solved in $n^{2-\varepsilon}$ time. Let $\delta' = \min(\varepsilon, \delta)/2$. Given a Min-IP$_{n,n^{\delta'}}$ instance, by enumerating all possible inner products between 0 and $n^{\delta'}$, we can reduce the instance to $n^{\delta'}$ instances of Exact-IP$_{n,n^{\delta'}}$. Applying the algorithm for Exact-IP, we then have an $n^{2-\varepsilon+\delta'} \leq n^{2-\delta'}$ time algorithm for Min-IP$_{n,n^{\delta'}}$. Hence, moderate dimensional Min-IP is also in truly-subquadratic time.

Finally, assume moderate dimensional Min-IP is in truly subquadratic time. Note that by Lemma 5.4, an Exact-IP$_{n,d}$ instance can be reduced to a Min-IP$_{n,O(d^2)}$ instance, which immediately implies that moderate dimensional Exact-IP is also in truly subquadratic time. □

# 6 Tighter Connection Between Max-IP, Bichrom.-$\ell_p$-Closest-Pair and $\ell_p$-Furthest-Pair

In this section we establish the tighter connections between Max-IP, Bichrom.-$\ell_p$-Closest-Pair and $\ell_p$-Furthest-Pair.

In Section 6.1, we show tighter connections for Max-IP, Exact-IP and additive approximation to Max-IP. And in Section 6.2, we show similar connections for additive approximation to Max-IP, Bichrom.-$\ell_p$-Closest-Pair and $\ell_p$-Furthest-Pair.

## 6.1 Tighter Connection between Exact-IP, Max-IP and Additive Approximation to Max-IP

The following lemma is implicit in [Rub18], which is used to show Bichrom.-$\ell_p$-Closest-Pair can not be approximated in truly-subquadratic time under SETH. [Rub18] only states a reduction from OV. However, the MA protocol in [Rub18] works equally well for the Inner Product problem, so it actually gives a reduction from Exact-IP.

**Lemma 6.1** (Implicit in Theorem 4.1 of [Rub18])**.** *For all sufficiently large integers $n, c$ and a parameter $\varepsilon > 0$, an Exact-IP$_{n,c\log n}$ instance can be reduced to $n^{O(\varepsilon \log(c/\varepsilon))}$ instances of computing $\Omega(1/\exp\{\widetilde{O}(c/\varepsilon)\})\cdot d$ additive approximation to Max-IP$_{n,d}$ for $d = n^{o(1)}$.*

In order to prove our tighter connection, our goal here is to improve the additive approximation ratio from $\Omega(1/\exp\{\widetilde{O}(c/\varepsilon)\})$ to $\Omega(1/\operatorname{poly}(c/\varepsilon))$.

### 6.1.1 A New MA Protocol for Inner Product

For that purpose, we need to modify the MA protocol from [Rub18]. In the following, we first describe the MA protocol for Inner Product in [Rub18] based on AG codes. Below we only summarize the relevant properties we need; readers can refer to [Rub18] for the details of the protocol.

**Lemma 6.2** (Theorem 3.1 [Rub18])**.** *For every $T \in [2, N]$, there is a computationally-efficient MA protocol for Inner Product such that*

1. *Alice and Bob hold input $x, y \in \{0,1\}^N$ respectively, and want to decide whether $\langle x, y\rangle = m$ for a target integer $m$.*
2. *Set $q$ to be the first prime larger than $T$ and a universal constant $c_1$, and set $R = \log(N/T) + O(1)$.*
3. *Merlin sends Alice a vector $z \in \mathbb{F}_{q^2}^{2^R}$, Alice rejects $z$ immediately if it doesn't satisfy some conditions.*

4. *Alice and Bob then toss $R$ coins to get $r \in [2^R]$. Based on $x$ (or $y$) and $r$, Alice and Bob generate two vectors in $\mathbb{F}_{q^2}^T$, $\vec{a}(x,r)$ and $\vec{b}(y,r)$ respectively,*

5. *Bob sends Alice $\vec{b}(y,r)$, and Alice calculates $u(x,y,r) = \langle \vec{a}(x,r), \vec{b}(y,r) \rangle$. Alice accepts if and only if $u(x,y,r) = z_r$.*

*The protocol satisfies the following conditions:*

- *If $\langle x, y \rangle = m$, then there is a proof (the vector $z$) from Merlin such that Alice always accepts.*
- *If $\langle x, y \rangle \neq m$, then for all proofs from Merlin, Alice accepts with probability at most $1/2$.*

**Our Modified Protocol.** We make some minor modifications to the above protocol. First, note that an element from $\mathbb{F}_{q^2}$ can be treated as an element in $\mathbb{F}_q[x]/(P_{\mathsf{irred}}(x))$, where $P_{\mathsf{irred}}(x) \in \mathbb{F}_q[x]$ is an irreducible polynomial of degree 2. In this way, we can interpret all elements in $\vec{a}(x,r)$ and $\vec{b}(y,r)$ as degree 1 polynomials in $\mathbb{F}_q[x]$, which can in turn be interpreted as degree 1 polynomials in $\mathbb{Z}[x]$. We denote these vectors of polynomials by $\vec{U}(x,r), \vec{V}(y,r) \in \mathbb{Z}[x]^T$, with coefficients from $\{0, 1, \ldots, q-1\}$.

Next, we set $W(x,y,r) = \langle \vec{U}(x,r), \vec{V}(y,r) \rangle$, which is a degree 2 polynomial in $\mathbb{Z}[x]$. Note that the coefficients of $W(x,y,r)$ are between 0 and $O(q^2 \cdot T) = O(T^3)$.

Now, in the message from Merlin, for all possible $r \in [2^R]$, we also add a claimed description of $W(x,y,r)$. This takes $O\left(\frac{N \log T}{T}\right)$ bits, so it doesn't affect the message complexity from Merlin. Then, after Alice receives $\vec{b}(y,r)$ from Bob (from which she can obtain $\vec{V}(y,r)$), Alice computes $W(x,y,r)$ instead of $u(x,y,r)$, and rejects immediately if this $W(x,y,r)$ does not match the one given by Merlin. After that, she knows that $u(x,y,r) = W(x,y,r)/(P_{\mathsf{irred}}(x))$, and proceeds as in the original protocol.

It is easy to see that, when $\langle x, y \rangle = m$, if Merlin provides the correct $W(x,y,r)$'s, then Alice still always accepts (regardless of $r$). And when $\langle x, y \rangle \neq m$, since these $W(x,y,r)$'s only provide additional checks, Alice still accepts with probability at most $1/2$ for all proofs.

We use $\Pi_{\mathsf{orig}}$ to denote the protocol from [Rub18] (Lemma 6.2), and $\Pi_{\mathsf{new}}$ to denote our new protocol. In the following we utilize $\Pi_{\mathsf{new}}$ to give an improved reduction from Exact-IP to additive approximation to Max-IP.

Before that, we need the following encoding trick, whose proof can be found in the appendix.

**Lemma 6.3.** *For all integers $d, r$ and $0 \leq m \leq dr^2$, there are mappings $\varphi^x, \varphi^y : \{0, 1, \ldots, r\}^d \to \{0,1\}^{O(dr^2)^2}$ and an integer $0 \leq M \leq O(dr^2)^2$, such that for all $x, y \in \{0, 1, \ldots, r\}^d$:*

- *If $\langle x, y \rangle = m$, then $\langle \varphi^x(x), \varphi^y(y) \rangle = M$.*
- *Otherwise, $\langle \varphi^x(x), \varphi^y(y) \rangle < M$.*
- *Moreover, $M$ only depends on $d$ and $r$.*

**Lemma 6.4.** *For all sufficiently large integers $n, c$ and a parameter $\varepsilon > 0$, every Exact-IP$_{n, c \log n}$ instance can be reduced to $n^{O(\varepsilon \log(c/\varepsilon))}$ instances of computing an $\Omega((\varepsilon/c)^6) \cdot d$ additive approximation to Max-IP$_{n,d}$ for $d = n^{o(1)}$.*

*Proof.* Consider an Exact-IP$_{n, c \log n}$ instance with sets $A$ and $B$, and integer $m$. Using our protocol $\Pi_{\mathsf{new}}$ for checking whether $\langle x, y \rangle = m$, we only need to figure out whether there is a pair $(x, y) \in A \times B$ and a proof from Merlin such that Alice always accepts.

Let $N = c \log n$, and set $T = c/\varepsilon$. Then the message complexity from Merlin is $O(\varepsilon \log n \log(c/\varepsilon))$ and the total number of random bits is $R = \log(N/T) + O(1) \leq \log(\varepsilon \log n) + O(1)$.

We first enumerate all valid proofs $\psi$, which is a pair of $z \in \mathbb{F}_{q^2}^{2^R}$ and $W \in \mathbb{Z}[x]^{2^R}$ such that for all $r \in [2^R]$, we have $z_r = W_r / P_{\mathsf{irred}}(x)$.

Next, we want to determine whether there is a pair $(x, y) \in A \times B$, such that this proof $\psi$ makes Alice always accepts. Note we only need to distinguish the following two cases:

- For all $r \in [2^R]$, $\langle \vec{U}(x, r), \vec{V}(y, r) \rangle = W_r$.
- For at most half of $r \in [2^R]$, $\langle \vec{U}(x, r), \vec{V}(y, r) \rangle = W_r$.

Recall that $\vec{U}(x, r)$ and $\vec{V}(y, r)$ are vectors of $T$ degree 1 polynomials from $\mathbb{Z}[x]$, with coefficients in $\{0, 1, \ldots, q-1\}$, and $W_r$ is a degree 2 polynomial in $\mathbb{Z}[x]$, with coefficients in $\{0, 1, \ldots, O(q^3)\}$. For a polynomial $P(x)$ in $\mathbb{Z}[x]$ and an integer $t$, let $[t]P(x)$ denote the coefficient of $x^t$ in $P(x)$. Then we can see $\langle \vec{U}(x, r), \vec{V}(y, r) \rangle = W_r$ is equivalent to the condition: for all $0 \le t \le 2$,

$$\sum_{i=0}^{t} \sum_{k=1}^{T} [i]\vec{U}(x, r)_k \cdot [t-i]\vec{V}(y, r)_k = [t]W_r. \tag{2}$$

Note that the left side of Equation (2) is an inner product between two vectors from $\{0, 1, \ldots, q-1\}^{3T}$. By Lemma 6.3, we can construct three Boolean vectors $u_0, u_1, u_2 \in \{0,1\}^{O(q^6)}$ from $\vec{U}(x, r)$ and also $v_0, v_1, v_2 \in \{0,1\}^{O(q^6)}$ from $\vec{V}(y, r)$ and an integer $M$ (which only depends on $T$), such that:

- If Equation (2) holds for all $t$, then $\sum_{i=0}^{2} \langle u_i, v_i \rangle = M$.
- Otherwise, $\sum_{i=0}^{2} \langle u_i, v_i \rangle < M$.

Now, we concatenate all these $u_0, u_1, u_2$ for all possibles $r$'s to form a single vector $u_x$, and construct $v_y$ similarly. We have:

- If for all $r \in [2^R]$, $\langle \vec{U}(x, r), \vec{V}(y, r) \rangle = W_r$, then $\langle u_x, v_y \rangle \ge 2^R \cdot M$.
- If for at most half of $r \in [2^R]$, $\langle \vec{U}(x, r), \vec{V}(y, r) \rangle = W_r$, then $\langle u_x, v_y \rangle \le 2^R \cdot (M - 1/2)$.

Now, let $A_\psi$ and $B_\psi$ be the collections of $u_x$ and $v_y$ with the proof $\psi$ respectively. Then we want to distinguish between the following two cases:

- There is a $\psi$ such that $\mathsf{Max}(A_\psi, B_\psi) \ge 2^R \cdot M$.
- For all $\psi$, $\mathsf{Max}(A_\psi, B_\psi) \le 2^R \cdot (M - 1/2)$.

Note that vectors in $A_\psi$ and $B_\psi$ are of dimension $d = O(q^6 \cdot 2^R)$, so the above can be solved by $2^{O(\varepsilon \log n \log(c/\varepsilon))} = n^{O(\varepsilon \log(c/\varepsilon))}$ calls to $\Omega(1/q^6) \cdot d = \Omega((\varepsilon/c)^6) \cdot d$ additive approximation to $\mathsf{Max\text{-}IP}_{n,d}$, which completes the proof. □

Now we are ready to prove Theorem 6.5.

**Theorem 6.5.** *The following are equivalent:*

1. *An $\varepsilon \cdot d$ additive approximation to $\mathsf{Max\text{-}IP}_{n,d}$ is computable in $n^{2-\varepsilon^{o(1)}}$ time.*
2. *$\mathsf{Max\text{-}IP}_{n,c\log n}$ is solvable in $n^{2-1/c^{o(1)}}$ time.*
3. *$\mathsf{Exact\text{-}IP}_{n,c\log n}$ is solvable in $n^{2-1/c^{o(1)}}$ time.*

*Proof.* We only need to show that Item (1) implies Item (3). By Lemma 6.4, there are constants $c_1, c_2$ such that for any constant $\varepsilon_1 > 0$, every Exact-IP$_{n,c\log n}$ instance can be reduced to $n^{c_1\varepsilon_1 \log(c/\varepsilon_1)}$ instances of $c_2 \cdot (\varepsilon_1/c)^6 \cdot d$ additive approximations to Max-IP$_{n,d}$ for $d = n^{o(1)}$.

Suppose Item (1) holds, we set $\varepsilon_1 = 1/c$, then Exact-IP$_{n,c\log n}$ can be solved in

$$n^{c_1 \log(c^2)/c + 2 - (c_2 \cdot c^{-12})^{o(1)}} = n^{2-1/c^{o(1)}}$$

time, which completes the proof. □

## 6.2 Tighter Connection Between Additive Approximation to Max-IP and Some Geometric Problems

Now we are ready to establish a similar connection between additive approximation to Max-IP and some geometric problems.

**Theorem 6.6.** *The following are equivalent:*

1. *An $\varepsilon \cdot d$ additive approximation to Max-IP$_{n,d}$ is computable in $n^{2-\varepsilon^{o(1)}}$ time.*
2. *An $\varepsilon \cdot d$ additive approximation to Min-IP$_{n,d}$ is computable in $n^{2-\varepsilon^{o(1)}}$ time.*
3. *A $(1+\varepsilon)$ approximation to Bichrom.-$\ell_p$-Closest-Pair is computable in $n^{2-\varepsilon^{o(1)}}$ time (for a constant $p \in [1,2]$).*
4. *A $(1+\varepsilon)$ approximation to $\ell_p$-Furthest-Pair is computable in $n^{2-\varepsilon^{o(1)}}$ time (for a constant $p \in [1,2]$).*

One direction is simple, and already implicit in previous work.

**Lemma 6.7** (Theorem 4.1 [Rub18])**.** *For any $p \in [1,2]$, if Bichrom.-$\ell_p$-Closest-Pair or $\ell_p$-Furthest-Pair can be approximated in $n^{2-\varepsilon^{o(1)}}$ time, then there is an algorithm computing $\varepsilon \cdot d$ additive approximation to Max-IP in $n^{2-\varepsilon^{o(1)}}$ time.*

So it suffices to prove the other direction, we are going to apply Theorem 1.19.

*Proof of Theorem 6.6.* The equivalence between Item (1) and (2) follows directly from Lemma 5.3. By Lemma 6.7, Item (3) and (4) both imply Item (1). So it suffices to show Item (1) implies Item (3) and Item (4).

We only consider Bichrom.-$\ell_p$-Closest-Pair here; the case for $\ell_p$-Furthest-Pair are symmetric. Note that by a binary search (which incurs a negligible factor in the running time), we only need to consider the decision version, in which we are given a real $R$, and want to distinguish the two cases: (1) $\min_{(a,b)\in A\times B} \|a-b\|_p \le R$; (2) $\min_{(a,b)\in A\times B} \|a-b\|_p \ge (1+\varepsilon)\cdot R$.

By Theorem 1.19 and Lemma 2.4, this decision problem can be reduced to computing an $\Omega(\varepsilon \cdot d)$ approximation to Max-IP$_{n,O(\varepsilon^{-2}\log n)}$, which by assumption can be solved in $n^{2-\varepsilon^{o(1)}}$ time. □

Finally, Theorem 1.4 is a simple corollary of Theorem 6.5 and Theorem 6.6.

# 7 Equivalence in the Data Structure Setting

In this section, we generalize our equivalence results to the data structure setting.

We first introduce the data structure versions of OV and Max-IP, which are used as intermediate problems for the reductions.

- **Online OV:** Preprocess a database $\mathcal{D}$ of $n$ points in $\{0,1\}^d$ such that, for all query of the form $q \in \{0,1\}^d$, either report a point $x \in \mathcal{D}$ which is orthogonal to $q$ or report that no $x$ exists.
- **Online Max-IP:** Preprocess a database $\mathcal{D}$ of $n$ points in $\{0,1\}^d$ such that, for all query of the form $q \in \{0,1\}^d$, find a point $x \in \mathcal{D}$ maximizing $\langle x, q\rangle$.

**Theorem 7.1.** *The following are equivalent:*

- *There is a $\delta > 0$ such that for all constant $c$, there is a data structure for Online OV with $d = c\log n$ uses $\mathrm{poly}(n)$ space and allows $n^{1-\delta}$ query time.*
- *There is a $\delta > 0$ such that for all constant $c$, there is a data structure for Online Max-IP with $d = c\log n$ uses $\mathrm{poly}(n)$ space and allows $n^{1-\delta}$ query time.*
- *There is a $\delta > 0$ such that for all $\varepsilon > 0$, there is a data structure for approximate NNS in $\ell_p$ with approximation ratio $(1+\varepsilon)$ uses $\mathrm{poly}(n)$ space and allows $n^{1-\delta}$ query time for a constant $p \in [1,2]$.*

Note that by [AWY15], Online OV is equivalent to Partial Match, so the above theorem implies Theorem 1.3.

We also need the following two important observations from the proof of Lemma 4.2 and Lemma 6.4.

**Lemma 7.2** (Implicit in Lemma 4.2)**.** *Let $n$ be an integer, $c$ be a constant, $\varepsilon > 0$ and $0 \le k \le c\log n$. There are two families of functions $f_1, f_2, \ldots, f_m$ and $g_1, g_2, \ldots, g_m$ from $\{0,1\}^{c\log n}$ to $\{0,1\}^{2^{O(c/\varepsilon)}\log n}$ where $m = n^{O(\varepsilon\log(c/\varepsilon))}$, such that for all $x, y \in \{0,1\}^{c\log n}$, $\langle x, y\rangle = k$ if and only if there is an $i \in [m]$ such that $\langle f_i(x), g_i(y)\rangle = 0$. Moreover, functions $f_i$'s and $g_i$'s can be evaluated in $\mathrm{polylog}(n)$ time.*

**Lemma 7.3** (Implicit in Lemma 6.4 and 6.7)**.** *Let $p \in [1,2]$, $n$ be an integer, $c$ be a constant, $\varepsilon > 0$ and $0 \le k \le c\log n$. There are two families of functions $f_1, f_2, \ldots, f_m$ and $g_1, g_2, \ldots, g_m$ from $\{0,1\}^{c\log n}$ to $\mathbb{R}^{n^{o(1)}}$ where $m = n^{O(\varepsilon\log(c/\varepsilon))}$, such that for all $x, y \in \{0,1\}^{c\log n}$,*

- *If $\langle x, y\rangle = k$, then there is an $i \in [m]$ such that $\|f_i(x) - g_i(y)\|_p \le 1 - \Omega((\varepsilon/c)^6)$.*
- *Otherwise, for all $i \in [m]$, $\|f_i(x) - g_i(y)\|_p \ge 1$.*

*Moreover, functions $f_i$'s and $g_i$'s can be evaluated in $n^{o(1)}$ time.*

*Proof of Theorem 7.1.* In the below we first show the equivalence between Online OV and Online Max-IP, the equivalence between Online Max-IP and NNS is proved similarly, so we only sketch the main ideas.

**Online OV $\Leftrightarrow$ Online Max-IP.** The reduction from Online OV to Online Max-IP is trivial. For the other direction, suppose there is a $\delta > 0$ such that for all constant $c$, there is an algorithm for Online OV with $d = c\log n$ such that it uses $\mathrm{poly}(n)$ space and allows $n^{1-\delta}$ query time.

Let $d = c\log n$ for a constant $c$, and $c_1$ be the constant hiding in the big-$O$ of $m = 2^{O(\varepsilon\log(c/\varepsilon))}$ in Lemma 7.2. Suppose we are given a set $\mathcal{D}$ of $n$ points from $\{0,1\}^d$.

We set $\varepsilon$ such that $c_1 \cdot \varepsilon\log(c/\varepsilon) = \delta/2$ and apply Lemma 7.2. Now, for each $0 \le k \le d$, we build $n^{c_1\cdot\varepsilon\log(c/\varepsilon)} = n^{\delta/2}$ data structures for Online OV, the $i$-th data structure consists of the $f_i(x)$'s for all $x \in \mathcal{D}$. Note that the $f_i(x)$'s have length $2^{O(c/\varepsilon)} \cdot \log n$, which is still $O(\log n)$ as $\varepsilon$ is a constant.

For each query $q \in \{0,1\}^d$, note that there is an $x \in \mathcal{D}$ such that $\langle x, q\rangle = k$ if and only if there is an $i$ such that the $i$-th Online OV structure contains an orthogonal point to $g_i(q)$. Therefore, by enumerating $k$ from $d$ down to 0, $i$ from $\lceil n^{\delta/2} \rceil$, and making corresponding queries to the Online OV data structures, one can answer queries for Online Max-IP in $n^{1-\delta/2} \cdot d$ time.

**Online Max-IP $\Leftrightarrow$ Approximate NNS (Sketch).** Using Lemma 7.3, the reduction from Online Max-IP to Approximate NNS can be proved similarly as from Online Max-IP to Online OV.

For the direction from approximate NNS to Online Max-IP: suppose the approximation ratio is $(1+\varepsilon)$. It suffices, for all $R$ of the form $(1+\varepsilon/3)^k$ for an integer $k$, to construct a data structure which finds a point with distance smaller than $R \cdot (1+\varepsilon/3)$ if the minimum distance is smaller than $R$, and reports a failure if the minimum distance is greater than $R \cdot (1+\varepsilon/3)$ (its behavior can be arbitrary if neither case holds). Using the reduction implicit in proof of Theorem 1.19, this can be reduced to Online Max-IP with $d = O(\log n)$. □

# 8 Algorithms for Apx-Min-IP and Apx-Max-IP

In this section we give fast algorithms for Apx-Min-IP and Apx-Max-IP. Our algorithms make use of the polynomial method [AWY15]. For simplicity of exposition, we set the approximation factors in Apx-Min-IP and Apx-Max-IP to be 2, but our algorithms can be extended to work for any constant approximation factor $\kappa > 1$ easily.

## 8.1 Low Degree Probabilistic Polynomial Implies Fast Algorithms

Abboud, Williams, and Yu [AWY15], show that for a Boolean vector problem, a "sparse" probabilistic polynomial for the problem implies a fast algorithm. To state their result formally, we first introduce some notations.

For our purposes, we will think of a *probabilistic polynomial* $\mathcal{P}$ as a distribution over $\mathbb{F}_2$-polynomials (polynomials over the field $\mathbb{F}_2$), and the *degree* of a probabilistic polynomial is the maximum degree of all polynomials in its support. For a function $f : D \to \{0,1\}$, we say $\mathcal{P}$ is an $\varepsilon$-error probabilistic polynomial for $f$, if for every $x \in D$, $\Pr_{P\sim\mathcal{P}}[P(x) \neq f(x)] \leq \varepsilon$.

Let us abstract out a key result from [AWY15], for our use here:

**Theorem 8.1** ([AWY15])**.** *Let $c$ be an integer and $d = c \log n$, let $f : D \to \{0,1\}$ with $D \subseteq \{0,1\}^d \times \{0,1\}^d$ be a function. Suppose that:*

- *For any $\varepsilon > 0$, there is an $\varepsilon$-error probabilistic polynomial $\mathcal{P}$ for $f$ with degree $t = O(\log \varepsilon^{-1})$.*
- *A sample from $\mathcal{P}$ can be generated in $\mathrm{poly}(\binom{d}{\leq t})$ time.*[19]

*Then there is an algorithm $\mathbb{A}$ such that:*

- *Given two sets $A$ and $B$ of $n$ vectors from $\{0,1\}^d$, $\mathbb{A}$ runs in $n^{2-1/O(\log c)}$ time.*
- *If for every $(a,b) \in A \times B$, $f(a,b) = 0$, then $\mathbb{A}$ outputs 0 with probability at least $1 - 1/n$.*
- *If there is an $(a,b) \in A \times B$, $f(a,b) = 1$, then $\mathbb{A}$ outputs 1 with probability at least $1 - 1/n$.*[20]

[19] $\binom{n}{\leq m}$ denotes $\sum_{i=0}^{m} \binom{n}{i}$.

[20] If neither of the above two cases hold, the algorithm can output anything.

## 8.2 $n^{2-1/O(\log c)}$ Time Algorithms for Apx-Min-IP and Apx-Max-IP

In order to apply the theorem above, we need to switch from Apx-Min-IP and Apx-Max-IP to their closely related decision problems Gap-Min-IP and Gap-Max-IP.

**Definition 8.2.** *For $n, d \in \mathbb{N}$, we define the problems:*

- *Gap-Min-IP$_{n,d}$ :* Given sets $A$ and $B$ of $n$ vectors from $\{0,1\}^d$ and an integer $\tau$, and the promise that either $\mathsf{Min}(A,B) \le \tau$ or $\mathsf{Min}(A,B) \ge 2\tau$, the task is to decide which.
- *Gap-Max-IP$_{n,d}$ :* Given sets $A$ and $B$ of $n$ vectors from $\{0,1\}^d$ and an integer $\tau$, and the promise that either $\mathsf{Max}(A,B) \le \tau$ or $\mathsf{Max}(A,B) \ge 2\tau$, the task is to decide which.

*Moreover, for two vectors $x, y \in \{0,1\}^d$ and an integer $\tau$, we define the corresponding gap-deciding function:*

$$f^{\mathsf{gap}}_{d,\tau}(x,y) = \begin{cases} 1 & \langle x,y\rangle \ge 2\tau, \\ 0 & \langle x,y\rangle \le \tau, \\ \textit{undefined} & \textit{otherwise.} \end{cases}$$

*When $d$ and $\tau$ are clear from the context, we omit them for simplicity.*

**Remark 8.3.** *Gap-Max-IP$_{n,d}$ (Gap-Min-IP$_{n,d}$) is equivalent to determine whether there is an $(a,b) \in A \times B$ such that $f^{\mathsf{gap}}(a,b) = 1$ ($f^{\mathsf{gap}}(a,b) = 0$) or for all $(a,b) \in A \times B$ we have $f^{\mathsf{gap}}(a,b) = 0$ ($f^{\mathsf{gap}}(a,b) = 1$).*

The following lemma is the key technical ingredient of this section.

**Lemma 8.4.** *For all $d, \tau \in \mathbb{N}$ and $\varepsilon \in (0, 1/10)$, there is a $t = O(\log \varepsilon^{-1})$-error probabilistic polynomial $\mathcal{P}$ for $f^{\mathsf{gap}}_{d,\tau}$. Moreover, a sample from $\mathcal{P}$ can be generated in $\mathrm{poly}(\binom{d}{\le t})$ time.*

Before proving Lemma 8.4, we first show it implies Theorem 1.7 (restated below) together with Theorem 8.1.

**Reminder of Theorem 1.7.** *There are $n^{2-1/O(\log c)}$ time randomized algorithms for Apx-Min-IP$_{n,c\log n}$ and Apx-Max-IP$_{n,c\log n}$.*

*Proof of Theorem 1.7.* We only consider Apx-Min-IP here; the case for Apx-Max-IP is symmetric. By Lemma 8.4, Theorem 8.1, and Remark 8.3, there is a randomized algorithm $\mathbb{A}$ for Gap-Min-IP$_{n,c\log n}$ in $n^{2-1/O(\log c)}$ time.

Now we turn $\mathbb{A}$ into an algorithm for Min-IP. We say $\mathbb{A}$ outputs 1 if it decides $\mathsf{Min}(A,B) \le \tau$, and 0 otherwise. We enumerate $\tau$ from 0 to $d$, and let $\tau_{\mathsf{min}}$ be the smallest $\tau$ such that $\mathbb{A}$ outputs 1. Note that such $\tau$ exists, as $\mathbb{A}$ must output 1 when $\tau = d$.

With probability at least $1 - (d+1)/n \ge 2/3$, $\mathbb{A}$ operates correctly on all enumerated $\tau$'s. We condition on that event in the following. Since $\mathbb{A}$ outputs 1 with $\tau_{\mathsf{min}}$, we have $\mathsf{Min}(A,B) < 2\tau_{\mathsf{min}}$ (otherwise it must output 0). Similarly, as $\mathbb{A}$ outputs 0 with $\tau_{\mathsf{min}} - 1$ ($\tau_{\mathsf{min}}$ is the smallest), we have $\mathsf{Min}(A,B) > \tau_{\mathsf{min}} - 1$ (otherwise it must output 1). Therefore, we can see $2\tau_{\mathsf{min}} \in [\mathsf{Min}(A,B), 2 \cdot \mathsf{Min}(A,B)]$ with probability at least $2/3$, and we obtain an $n^{1-1/O(\log c)}$ algorithm for Apx-Min-IP. □

Finally, we devote the rest of this section to the proof of Lemma 8.4.

*Proof of Lemma 8.4.* In the following, we assume $\tau \leq d/2$ as the function becomes trivial otherwise.

We begin by introducing some notation. Let $T$ be a sequence of integers from $[d]$, we use $x_{|T} \in \{0,1\}^{|T|}$ to denote the projection of $x$ on $T$, such that $(x_{|T})_i := x_{T_i}$ for $i \in [|T|]$. We also use the Iverson bracket notation: for a predicate $P$, $[P]$ takes value 1 when $P$ is true, and 0 otherwise.

**Construction of "Micro" Probabilistic Polynomial $\mathcal{P}_{\mathsf{micro}}$.** The first step is to construct a probabilistic polynomial $\mathcal{P}_{\mathsf{micro}}$ of degree 1, such that for $x, y \in \{0,1\}^d$:

- If $\langle x, y\rangle \geq 2\tau$: $\Pr_{P\sim\mathcal{P}_{\mathsf{micro}}}[P(x,y)=1] \geq c_1$ for a universal constant $c_1$.
- If $\langle x, y\rangle \leq \tau$: $\Pr_{P\sim\mathcal{P}_{\mathsf{micro}}}[P(x,y)=1] \leq c_2$ for a universal constant $c_2$.
- $c_1 > c_2$.

Let $k = \frac{d}{\tau}$. By our assumption, we have $k \geq 2$. Now a sample from $\mathcal{P}_{\mathsf{micro}}$ is generated as follows:

- We pick a sequence $T$ of $k$ uniform random numbers from $[d]$ and a uniform random vector $z \in \{0,1\}^k$.
- We set $P(x,y) := \sum_{i=1}^{k} z_i \cdot (x_{|T})_i \cdot (y_{|T})_i$ (which is an $\mathbb{F}_2$ polynomial).

First, we make the following observations:

- If $\langle x, y\rangle \geq 2\tau$:

$$\begin{aligned}\Pr_T[\langle x_{|T}, y_{|T}\rangle > 0] &\geq 1 - \left(1 - \frac{2\tau}{d}\right)^k \\ &= 1 - \left(1 - \frac{2}{k}\right)^k \geq 1 - e^{-2} > 0.86.\end{aligned}$$

- If $\langle x, y\rangle \leq \tau$:

$$\begin{aligned}\Pr_T[\langle x_{|T}, y_{|T}\rangle > 0] &\leq 1 - \left(1 - \frac{\tau}{d}\right)^k \\ &= 1 - \left(1 - \frac{1}{k}\right)^k \leq 1 - \frac{1}{4} = 0.75.\end{aligned}$$

Note that when $\langle x_{|T}, y_{|T}\rangle = 0$, $P(x,y)$ is always 0, and when $\langle x_{|T}, y_{|T}\rangle > 0$, $P(x,y) = 1$ with probability $1/2$. Therefore, we have:

- If $\langle x, y\rangle \geq 2\tau$:
$$\Pr_{P\sim\mathcal{P}_{\mathsf{micro}}}[P(x,y)=1] \geq (0.86)/2 = 0.43.$$
- If $\langle x, y\rangle \leq \tau$:
$$\Pr_{P\sim\mathcal{P}_{\mathsf{micro}}}[P(x,y)=1] \leq (0.75)/2 = 0.375.$$

The above completes our construction of the "micro" probabilistic polynomial $P_{\mathsf{micro}}$.

**Construction of the Probabilistic Polynomial $\mathcal{P}_{\mathsf{final}}$.** Now, let $m = c_1 \cdot \log \varepsilon^{-1}$ for a sufficiently large constant $c_1$. And let $P_1, P_2, \ldots, P_m$ be $m$ i.i.d. samples from $\mathcal{P}_{\mathsf{micro}}$. By a simple Chernoff bound, we have:

- If $\langle x, y\rangle \geq 2\tau$:

$$\Pr_{P_1,P_2,\ldots,P_m\sim\mathcal{P}_{\mathsf{micro}}}\left[\sum_{i=1}^{m}[P_i(x,y)=1] > 0.4\cdot m\right] \geq 1-\varepsilon.$$

- If $\langle x, y\rangle \leq \tau$:

$$\Pr_{P_1,P_2,\ldots,P_m\sim\mathcal{P}_{\mathsf{micro}}}\left[\sum_{i=1}^{m}[P_i(x,y)=1] < 0.4\cdot m\right] \geq 1-\varepsilon.$$

Finally, we set

$$P_{\mathsf{final}}(x,y) := \sum_{S\subseteq[m],|S|>0.4m}\prod_{i\in S}P_i(x,y)\cdot\prod_{i\notin S}(1-P_i(x,y)).$$

Clearly, $P_{\mathsf{final}}(x,y)=1$ if and only if $\sum_{i=1}^{m}[P_i(x,y)=1] > 0.4\cdot m$, and therefore its distribution $\mathcal{P}_{\mathsf{final}}$ is the $\varepsilon$-error probabilistic polynomial we want. And it is easy to see a polynomial from $\mathcal{P}_{\mathsf{final}}$ can be sampled in the stated time.

□

### 8.3 A Fast Algorithm for Approximating "Almost Satisfiable" MAX-SAT Instances

Finally, we give an application of the algorithm for Apx-Min-IP by proving Theorem 1.9 (restated below).

**Reminder of Theorem 1.9** *Let $\varphi$ be a MAX-SAT instance on $n$ variables with $m$ clauses, and $\varepsilon = 1-\mathsf{sat}(\varphi)$. There is a $2^{n(1-1/O(\log\varepsilon^{-1}))}$ time algorithm to find an assignment $x$ satisfying at least $(1-2\varepsilon)\cdot m$ clauses.*

*Proof.* We use the reduction from CNF-SAT to OV, from [Wil05]. For simplicity, suppose 2 divides $n$. For an assignment $x$ to $\varphi$, we use $\mathsf{val}(\varphi, x)$ to denote the number of satisfied clauses of $\varphi$ by $x$, divided by $m$.

First, we do a "sparsification" step: we pick $M = c_1\cdot\varepsilon^{-2}\cdot n$ clauses from $\varphi$ at uniformly random. Let $\psi$ be the MAX-SAT instance with these randomly chosen clauses.

By a standard Chernoff bound, with a sufficiently large universal constant $c_1$, for every assignment $x\in\{0,1\}^n$, we have

$$\Pr\left[|\mathsf{val}(\varphi,x)-\mathsf{val}(\psi,x)|\leq\varepsilon/3\right]\leq 1/2^{2n}.$$

Therefore, by a union bound, with probability at least $1-1/2^n$, for all $x\in\{0,1\}^n$ we have $|\mathsf{val}(\varphi,x)-\mathsf{val}(\psi,x)|\leq\varepsilon/3$, and it follows that $|\mathsf{sat}(\varphi)-\mathsf{sat}(\psi)|\leq\varepsilon/3$. So it suffices to consider $\psi$ now.

Next, we split these $n$ variables into two groups

$$x_L := \{x_1,\ldots,x_{n/2}\} \quad \text{and} \quad x_R := \{x_{n/2+1},\ldots,x_n\}.$$

Let $\mathcal{C}_1,\mathcal{C}_2,\ldots,\mathcal{C}_M$ be all clauses in $\psi$. For each $a\in\{0,1\}^{n/2}$, interpreted as an assignment to variables in $x_L$, we construct a vector $u_a\in\{0,1\}^M$, such that $(u_a)_i=1$ iff $\mathcal{C}_i$ is not satisfied when setting variables in $x_L$ according to $a$. Similarly, for each $b\in\{0,1\}^{n/2}$, we interpret it as an assignment to variables in $x_R$, and construct a vector $v_b\in\{0,1\}^M$ in the same way.

Next, for $a,b\in\{0,1\}^{n/2}$, $\langle(u_a)_i,(v_b)_i\rangle=1$ if and only if $\mathcal{C}_i$ is not satisfied by the joint assignment $(a,b)$. Therefore, $\langle u_a,v_b\rangle$ is the number of clauses that are not satisfied by the joint assignment $(a,b)$.

Let $A$ be the set of all $u_a$'s for $a\in\{0,1\}^{n/2}$, and $B$ be the set of all $v_b$'s for $b\in\{0,1\}^{n/2}$. By Theorem 1.7, there is an algorithm which finds a $(u_a,v_b)\in A\times B$ such that $\langle u_a,v_b\rangle\in[\mathsf{Min}(A,B),1.1\cdot\mathsf{Min}(A,B)]$.

From the definition, we have $\mathsf{Min}(A, B) := (1 - \mathsf{sat}(\psi)) \cdot M \leq \frac{4}{3} \cdot \varepsilon \cdot M$ (recall that $\varepsilon = 1 - \mathsf{sat}(\varphi)$). Therefore, we have $\langle u_a, v_b \rangle \leq 1.1 \cdot \frac{4}{3} \cdot \varepsilon \cdot M \leq 1.5 \cdot \varepsilon \cdot M$.

Let $x$ be the joint assignment $(a, b)$. We have $\mathsf{val}(\psi, x) \geq (1-1.5\varepsilon)$. Since $|\mathsf{val}(\psi, x) - \mathsf{val}(\varphi, x)| \leq \varepsilon/3$, $\mathsf{val}(\varphi, x) \geq (1 - 2\varepsilon)$, which means $x$ is a valid answer.

Finally, as $M = O(\varepsilon^{-2}) \cdot n$, the algorithm runs in $\left(2^{n/2}\right)^{2-1/O(\log \varepsilon^{-1})} = 2^{n \cdot (1-1/O(\log \varepsilon^{-1}))}$ time, which completes the proof. □

## Acknowledgments

We thank Virginia Vassilevska Williams for many comments on an early draft of this paper. We are grateful to Josh Alman, Jiawei Gao, Ofer Grossman, Kaifeng Lyu, Karthik C. S., Ruosong Wang, Virginia Vassilevska Williams, Grigory Yaroslavtsev and Peilin Zhong for helpful discussions during the work. We are especially grateful to Aviad Rubinstein for several helpful discussions which inspired the discovery of the equivalence between Bichrom.-Closest-Pair and OV.

## References

[ABDN18] Amir Abboud, Karl Bringmann, Holger Dell, and Jesper Nederlof. More consequences of falsifying SETH and the orthogonal vectors conjecture. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 253–266, 2018.

[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.

[AGW15] Amir Abboud, Fabrizio Grandoni, and Virginia Vassilevska Williams. Subcubic equivalences between graph centrality problems, APSP and diameter. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1681–1697, 2015.

[AHWW16] Amir Abboud, Thomas Dueholm Hansen, Virginia Vassilevska Williams, and Ryan Williams. Simulating branching programs with edit distance and friends: or: a polylog shaved is a lower bound made. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 375–388, 2016.

[AI08] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122, 2008.

[AIR18] Alexandr Andoni, Piotr Indyk, and Ilya Razenshteyn. Approximate nearest neighbor search in high dimensions. *To appear in the proceedings of ICM*, 2018.

[APRS16] Thomas Dybdahl Ahle, Rasmus Pagh, Ilya P. Razenshteyn, and Francesco Silvestri. On the complexity of inner product similarity join. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 151–164, 2016.

[AR18] Amir Abboud and Aviad Rubinstein. Fast and deterministic constant factor approximation algorithms for LCS imply new circuit lower bounds. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 35:1–35:14, 2018.

[ARW17] Amir Abboud, Aviad Rubinstein, and R. Ryan Williams. Distributed PCP Theorems for Hardness of Approximation in P. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 25–36, 2017.

[AW09] Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *TOCT*, 1(1):2:1–2:54, 2009.

[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.

[AWY15] Amir Abboud, Richard Ryan Williams, and Huacheng Yu. More applications of the polynomial method to algorithm design. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 218–230, 2015.

[BCH16] Michele Borassi, Pierluigi Crescenzi, and Michel Habib. Into the square: On the complexity of some quadratic-time solvable problems. *Electr. Notes Theor. Comput. Sci.*, 322:51–67, 2016.

[BCW98] Harry Buhrman, Richard Cleve, and Avi Wigderson. Quantum vs. classical communication and computation. In *Proc. of the Thirtieth Annual ACM Symposium on the Theory of Computing*, pages 63–68, 1998.

[BDT16] Arturs Backurs, Nishanth Dikkala, and Christos Tzamos. Tight hardness results for maximum weight rectangles. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 81:1–81:13, 2016.

[BFS86] László Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory (preliminary version). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 337–347, 1986.

[BI15] Arturs Backurs and Piotr Indyk. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 51–58, 2015.

[BOR99] Allan Borodin, Rafail Ostrovsky, and Yuval Rabani. Lower bounds for high dimensional nearest neighbor search and related problems. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 312–321, 1999.

[Bri14] Karl Bringmann. Why walking the dog takes time: Frechet distance has no strongly subquadratic algorithms unless SETH fails. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 661–670, 2014.

[Bro97] Andrei Z Broder. On the resemblance and containment of documents. In *Compression and complexity of sequences*, pages 21–29. IEEE, 1997.

[CGI+16] Marco L. Carmosino, Jiawei Gao, Russell Impagliazzo, Ivan Mihajlin, Ramamohan Paturi, and Stefan Schneider. Nondeterministic extensions of the strong exponential time hypothesis and consequences for non-reducibility. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 261–270, 2016.

[CGL04] Richard Cole, Lee-Ad Gottlieb, and Moshe Lewenstein. Dictionary matching and indexing with errors and don't cares. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 91–100, 2004.

[CGL+18] Lijie Chen, Shafi Goldwasser, Kaifeng Lyu, Guy Rothblum, and Aviad Rubinstein. Fine-grained complexity meets IP = PSPACE. *To appear in the proceedings of SODA 2019*, 2018.

[Cha17] Timothy M. Chan. Orthogonal range searching in moderate dimensions: k-d trees and range trees strike back. In *33rd International Symposium on Computational Geometry, SoCG 2017, July 4-7, 2017, Brisbane, Australia*, pages 27:1–27:15, 2017.

[Che18] 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.

[CIP02] Moses Charikar, Piotr Indyk, and Rina Panigrahy. New algorithms for subset query, partial match, orthogonal range searching, and related problems. In *Automata, Languages and Programming, 29th International Colloquium, ICALP 2002, Malaga, Spain, July 8-13, 2002, Proceedings*, pages 451–462, 2002.

[CMWW17] Marek Cygan, Marcin Mucha, Karol Wegrzycki, and Michal Wlodarczyk. On problems equivalent to (min, +)-convolution. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, pages 22:1–22:15, 2017.

[CW16] Timothy M. Chan and Ryan Williams. Deterministic APSP, Orthogonal Vectors, and More: Quickly Derandomizing Razborov-Smolensky. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1246–1255, 2016.

[DIIM04] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the 20th ACM Symposium on Computational Geometry, Brooklyn, New York, USA, June 8-11, 2004*, pages 253–262, 2004.

[GIKW17] Jiawei Gao, Russell Impagliazzo, Antonina Kolokolova, and Ryan Williams. Completeness for first-order properties on sparse structures with algorithmic applications. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2162–2181. SIAM, 2017.

[GPW18] Mika Göös, Toniann Pitassi, and Thomas Watson. The landscape of communication complexity classes. *Computational Complexity*, 27(2):245–304, 2018.

[GS89] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. *Advances in Computing Research*, 5:73–90, 1989.

[IP01] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

[JKKR04] T. S. Jayram, Subhash Khot, Ravi Kumar, and Yuval Rabani. Cell-probe lower bounds for the partial match problem. *J. Comput. Syst. Sci.*, 69(3):435–447, 2004.

[Kla03] Hartmut Klauck. Rectangle size bounds and threshold covers in communication complexity. In *18th Annual IEEE Conference on Computational Complexity (Complexity 2003), 7-10 July 2003, Aarhus, Denmark*, pages 118–134, 2003.

[KLM18] Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1283–1296, 2018.

[LWW18] Andrea Lincoln, Virginia Vassilevska Williams, and R. Ryan Williams. Tight hardness for shortest cycles and paths in sparse graphs. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1236–1252, 2018.

[MNSW98] Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. *J. Comput. Syst. Sci.*, 57(1):37–49, 1998.

[Mor08] Jorge Moraleda. Gregory shakhnarovich, trevor darrell and piotr indyk: Nearest-neighbors methods in learning and vision. theory and practice. *Pattern Anal. Appl.*, 11(2):221–222, 2008.

[PT09] Mihai Patrascu and Mikkel Thorup. Higher lower bounds for near-neighbor and further rich problems. *SIAM J. Comput.*, 39(2):730–741, 2009.

[PTW08] Rina Panigrahy, Kunal Talwar, and Udi Wieder. A geometric approach to lower bounds for approximate near-neighbor search and partial match. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 414–423, 2008.

[PW10] Mihai Patrascu and Ryan Williams. On the possibility of faster SAT algorithms. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1065–1075, 2010.

[Raz17] Ilya P. Razenshteyn. *High-dimensional similarity search and sketching: algorithms and hardness*. PhD thesis, Massachusetts Institute of Technology, Cambridge, USA, 2017.

[Riv74] Ronald Linn Rivest. Analysis of associative retrieval algorithms. PhD thesis, Standford University, 1974.

[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.

[RW13] Liam Roditty and Virginia Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 515–524, 2013.

[Vas18] Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. *To appear in the proceedings of ICM*, 2018.

[VW10] Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 645–654, 2010.

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

[Wil16] Richard Ryan Williams. Strong ETH breaks with merlin and arthur: Short non-interactive proofs of batch evaluation. In *31st Conference on Computational Complexity, CCC*, pages 2:1–2:17, 2016.

[Wil18] 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.

[WSSJ14] Jingdong Wang, Heng Tao Shen, Jingkuan Song, and Jianqiu Ji. Hashing for similarity search: A survey. *arXiv preprint arXiv:1408.2927*, 2014.

[WY14] Ryan Williams and Huacheng Yu. Finding orthogonal vectors in discrete structures. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1867–1877, 2014.

# A Missing Proofs

Here we give some missing proofs in the paper.

*Proof of Lemma 5.2.* Let $X = \sum_{i=1}^{cm} X_i$ and $\mu = \mathbb{E}[X] = \varepsilon \cdot cm$. Set $\delta = \varepsilon^{-1}/3$. By the multiplicative Chernoff bound, we have

$$\Pr[X > (1+\delta)\cdot\mu] < \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu}.$$

Note that $(1+\delta)\cdot\mu = (1+\varepsilon^{-1}/3)\cdot\varepsilon\cdot cm < \frac{1}{2}\cdot cm$. Also, we have

$$\begin{aligned}\left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu} &= e^{-\mu\cdot[(1+\delta)\log(1+\delta)-\delta]}\\ &\le e^{-\mu\cdot[\delta\ln\delta-\delta]}\\ &\le e^{-\varepsilon\cdot cm\cdot\left[\varepsilon^{-1}/3\ln(\varepsilon^{-1}/3)\right]/2}\\ &\le e^{-\varepsilon\cdot cm\cdot\left[\varepsilon^{-1}\ln(\varepsilon^{-1})\right]/12}\\ &\le e^{-\ln\varepsilon^{-1}\cdot cm/12} = \varepsilon^{-cm/12}.\end{aligned}$$

Therefore, we can set $c = 12$, and the proof is completed. □

*Proof of Lemma 5.3.* We define two functions $\varphi_x, \varphi_y : \{0,1\} \to \{0,1\}^2$ such that:

$$\varphi_x(0) := (1,0), \quad \varphi_x(1) := (0,1), \quad \varphi_y(0) := (1,1), \quad \varphi_y(1) := (1,0).$$

It is easy to check that for $a, b \in \{0,1\}$, $a \cdot b = 1 - \langle\varphi_x(a), \varphi_y(b)\rangle$. Then, for $x, y \in \{0,1\}^d$, we define $\psi^x_{\mathsf{rev}}(x) \in \{0,1\}^{2d}$ as the concatenation of $\varphi_x(x_i)$ for each $i \in [d]$, and similarly $\psi^y_{\mathsf{rev}}(y) \in \{0,1\}^{2d}$ as the concatenation of $\varphi_y(y_i)$ for each $i \in [d]$.

Then we can see $\langle\psi^x_{\mathsf{rev}}(x), \psi^y_{\mathsf{rev}}(y)\rangle = \sum_{i=1}^{d}\langle\varphi_x(x_i), \varphi_y(y_i)\rangle = d - \langle x, y\rangle$. □

*Proof of Lemma 5.4.* We remark the reduction here is essentially the same as the trick used in [Wil18]. For a vector $v \in \{0,1\}^*$, we use $v^{\otimes k}$ to denote the concatenation of $k$ copies of $v$.

Consider the following polynomial $P(x,y) := (\langle x,y\rangle - m)^2$, we have

$$P(x,y) = \langle x,y\rangle^2 - 2m\langle x,y\rangle + m^2 = \langle x,y\rangle^2 + 2m(d - \langle x,y\rangle) + m^2 - 2dm.$$

For convenience, for a vector $z \in \{0,1\}^{d^2}$, we use $z_{(i,j)}$ to denote the $(i-1)\cdot d + j$-th coordinate of $z$. For $x, y \in \{0,1\}^d$, we construct $\widetilde{x}, \widetilde{y} \in \{0,1\}^{d^2}$ such that $\widetilde{x}_{(i,j)} = x_i \cdot x_j$ and $\widetilde{y}_{(i,j)} = y_i \cdot y_j$. Then we can see

$$\langle \widetilde{x}, \widetilde{y}\rangle = \sum_{i=1}^{d}\sum_{j=1}^{d} x_i x_j \cdot y_i y_j = \left(\sum_{i=1}^{d} x_i y_i\right)^2 = \langle x,y\rangle^2.$$

Let $\psi^x_{\mathsf{rev}}$ and $\psi^y_{\mathsf{rev}}$ be the two functions from Lemma 5.3. For $x, y \in \{0,1\}^d$, we define

$$\varphi^x_{d,m}(x) := (\widetilde{x}, \psi^x_{\mathsf{rev}}(x)^{\otimes(2m)}) \qquad \text{and} \qquad \varphi^y_{d,m}(y) := (\widetilde{y}, \psi^y_{\mathsf{rev}}(y)^{\otimes(2m)}).$$

Then we have $\langle \varphi^x_{d,m}(x), \varphi^y_{d,m}(y)\rangle = \langle x,y\rangle^2 + 2m(d - \langle x,y\rangle) = P(x,y) + 2dm - m^2$. And we set $M_{d,m} = 2dm - m^2$.

Now, if $\langle x,y\rangle = m$, we have $P(x,y) = 0$, and therefore $\langle \varphi^x_{d,m}(x), \varphi^y_{d,m}(y)\rangle = M_{d,m}$. Otherwise, $\langle x,y\rangle \neq m$ and we have $P(x,y) > 0$, and hence $\langle \varphi^x_{d,m}(x), \varphi^y_{d,m}(y)\rangle > M_{d,m}$. Note that $\varphi^x_{d,m}(x), \varphi^y_{d,m}(y) \in \{0,1\}^{d^2+4dm}$, we add $5d^2 - M_{d,m}$ dummy ones to the end of $\varphi^x_{d,m}(x)$ and $\varphi^y_{d,m}(y)$ and set $M_d = 5d^2$, which completes the proof. □

*Proof of Lemma 6.3.* We begin by the construction of two embeddings $\psi_x, \psi_y : \{0,1,\ldots,r\} \to \{0,1\}^{r^2}$ such that for any $x, y \in \{0,1,\ldots,r\}$, $\langle \psi_x(x), \psi_y(y)\rangle = x \cdot y$.

For convenience, in the following we use $z_{(i,j)}$ to denote the $(i-1)\cdot r + j$-th coordinate of $z$. Then we define $\psi_x(x)_{(i,j)}$ as 1 when $i \le x$, and 0 otherwise; similarly, we define $\psi_y(y)_{(i,j)}$ as 1 when $j \le y$, and 0 otherwise. We have

$$\langle \psi_x(x), \psi_y(y)\rangle = \sum_{i=1}^{r}\sum_{j=1}^{r} \psi_x(x)_{(i,j)} \cdot \psi_y(y)_{(i,j)} = \sum_{i=1}^{x}\sum_{j=1}^{y} 1 \cdot 1 = \langle x,y\rangle.$$

Slightly abusing notations, for $x, y \in \{0,1,\ldots,r\}^d$, we define $\psi_x(x)$ and $\psi_y(y)$ as the concatenation of $\psi_x$ or $\psi_y$ applying on all coordinates of $x$ and $y$. Then we have $\psi_x(x), \psi_y(y) \in \{0,1\}^{dr^2}$, and $\langle \psi_x(x), \psi_y(y)\rangle = \langle x,y\rangle$. Then applying Lemma 5.4 and Lemma 5.3 completes the proof. □

# B More Applications of the $\Sigma_2^{\mathsf{cc}}$ Reduction Framework

To demonstrate the potential power of our $\Sigma_2^{\mathsf{cc}}$ framework. In the following we discuss some of its applications other than establishing our equivalence class. The first one is a very simple reduction from Hopcroft's problem in *constant dimensions* to OV in *polylogarithmic dimensions*. And the second one is a reduction from 3-SUM to 3-OV.

## B.1 Integer Inner Product and Hopcroft's Problem

The Hopcroft's problem is defined as follows: you are given two sets $A, B$ of $n$ vectors from $\mathbb{Z}^d$, and want to determine whether there is an $(a,b) \in A \times B$ such that $\langle a,b\rangle = 0$. In other words, it is the same as OV except for now vectors consist of integer entries.

We use $\mathbb{Z}$-$\mathsf{OV}_{n,d}$ to denote this problem in $d$ dimensions for simplicity, and assume the integers in $\mathbb{Z}$-$\mathsf{OV}_{n,d}$ belong to $[-n^c, n^c]$ for a constant $c$, which is the most interesting case. Now we formally state our reduction.

**Theorem B.1.** *Let $c, d$ be two constants, a $\mathbb{Z}$-$\mathsf{OV}_{n,d}$ instance $I$ with entries in $[-n^c, n^c]$ can be reduced to an $\mathsf{OV}_{n,O(\log n)^{d+1}}$ instance $J$ in $n^{1+o(1)}$ time, such that $I$ is a yes instance if and only if $J$ is a yes instance.*

An immediate corollary is that if moderate dimensional $\mathsf{OV}$ is in truly subquadratic time, then $\mathbb{Z}$-$\mathsf{OV}_{n,d}$ is also in truly subquadratic time for all constant $d$.

Let $d$ be an integer, we define $\mathsf{Z\text{-}IP}_d : \mathbb{Z}^d \times \mathbb{Z}^d \to \{0,1\}$ be the function that checks whether two $d$ dimensional vectors in $\mathbb{Z}^d$ are orthogonal. Note that $\mathbb{Z}$-$\mathsf{OV}_{n,d}$ is equivalent to $\mathsf{Z\text{-}IP}_d$-$\mathsf{Satisfying\text{-}Pair}_n$.

Theorem B.1 is just a direct corollary of the following fast $\Sigma_2$ communication protocol for $\mathsf{Z\text{-}IP}_d$ (it is in fact a $\mathsf{coNP}$ communication protocol, as Merlin sends nothing) and Theorem 1.14.

**Lemma B.2.** *Let $c, d$ be two constants, there is a $\Sigma_2^{\mathsf{cc}}$ protocol for $\mathsf{Z\text{-}IP}_d$ with entries in $[-n^c, n^c]$, in which Merlin sends nothing, Megan sends $\log\log(n) + O(1)$ bits and Alice and Bob communicate $d \cdot \log\log(n) + O(d)$ bits.*

*Proof.* Let $x, y$ be two vectors from $[-n^c, n^c]^d$, we have $|\langle x, y\rangle| \le d \cdot n^{2c}$.

Let $t$ be the smallest number such that the first $t$ primes $p_1, p_2, \ldots, p_t$ satisfy $\prod_{i=1}^{t} p_i > d \cdot n^{2c}$. We first bound $t$ and the largest prime $p_t$. Clearly, $t \le \log(d \cdot n^{2c}) = O(\log n)$. Recall that $n\#$ denotes the product of all primes less than or equal to $n$ (the primordial function), and we have $n\# = e^{(1+o(1))n}$. By the definition of $t$, it follows that $(p_t - 1)\# = e^{(1+o(1))\cdot(p_t-1)} \le d \cdot n^{2c}$ and $p_t = O(\log n)$.

From our choice of $t$, we have $\langle x, y\rangle = 0$ if and only if $\langle x, y\rangle \equiv 0 \pmod{p_i}$ for all $i \in [t]$. So in the protocol, Merlin sends nothing. Megan sends an index $i \in [t]$, which takes $\log t = \log\log n + O(1)$ bits. After that, for each $j \in [d]$, Bob sends $y_j \bmod p_i$ to Alice, which takes $d \cdot \log p_i \le d \cdot \log\log n + O(d)$ bits, and Alice accepts if and only if $\langle x, y\rangle \equiv 0 \pmod{p_i}$. □

## B.2 Sum-Check and 3-Sum

Next we discuss a reduction from 3-SUM to 3-$\mathsf{OV}$. 3-$\mathsf{OV}$ is a generalized version of $\mathsf{OV}$, in which you are given three sets $A, B, C$, each of $n$ vectors from $\{0,1\}^d$, and want to determine whether there is an $(a, b, c) \in A \times B \times C$ such that $\sum_{i=1}^{d} a_i \cdot b_i \cdot c_i = 0$ (the generalized inner product of $a, b$, and $c$ is zero). We use 3-$\mathsf{OV}_{n,d}$ to denote the 3-$\mathsf{OV}$ problem with sets of $n$ vectors of $d$ dimensions.

**Theorem B.3.** *If 3-$\mathsf{OV}$ is in truly-subquadratic time[21], then so is 3-$\mathsf{SUM}$.*

We remark that this reduction is not optimal, as it is conjectured that 3-$\mathsf{OV}$ requires $n^{3-o(1)}$ time (also implied by $\mathsf{SETH}$). We include it here only as an illustration of the applicability of our reduction framework. It would be very interesting to improve it to a reduction from 3-SUM to $\mathsf{OV}$[22].

Note that 3-$\mathsf{OV}$ is actually a *Satisfying-Triple* problem[23], and Theorem 1.14 only works for Satisfying-Pair problems. Still, we can generalize Theorem 1.14 easily to get the same connection between a 3-party $\Sigma_2$ communication protocol and a reduction from a satisfying-triple problem to 3-$\mathsf{OV}$.

Let $F : \left(\{0,1\}^d\right)^3 \to \{0,1\}$ be a function, $F$-$\mathsf{Satisfying\text{-}Triple}_n$ is the problem that you are given sets $A, B, C$ of $n$ vectors from $\{0,1\}^d$, and want to determine whether there is an $(a, b, c) \in A \times B \times C$ such that $F(a, b, c) = 1$. A 3-party $\Sigma_2$ communication protocol can be defined similarly as in Definition 1.13 with the third player named Charles, we omit it here. We have the following analogous theorem of Theorem 1.14.

[21] This means there is an $\varepsilon > 0$ such that for all constant $c$, 3-$\mathsf{OV}_{n,c\log n}$ can be solved in $n^{2-\varepsilon}$ time.

[22] In [CGI+16], it is shown that under the $\mathsf{NSETH}$ (which is controversial, see [Wil16]), there is no fine-grained reduction from $\mathsf{OV}$ to 3-SUM. But there is no formal evidence against the other direction.

[23] It is also called a *Product Space Problem* in [KLM18].

**Theorem B.4.** *Let $F : \left(\{0,1\}^d\right)^3 \to \{0,1\}$ and $n$ be an integer, suppose $F$ admits a computationally-efficient $\Sigma_2^{\mathsf{cc}}$ protocol. In which Merlin sends $m_1$ bits, Megan sends $m_2$ bits, Alice, Bob and Charles communicate $\ell$ bits.*

*Then there is a reduction from an $F$-Satisfying-Triple$_n$ instance $I$ to $2^{m_1}$ 3-OV$_{n,2^{(m_2+\ell)}}$ instances $J_1, J_2, \ldots, J_{2^{m_1}}$, such that $I$ is a yes instance if and only if one of the reduced instances is a yes instance. And the reduction takes $O(n \cdot 2^{O(m_1+m_2+\ell)} \cdot \mathrm{poly}(d))$ time.*

We omit its proof here as it is identical to that of Theorem 1.14.

Note that we can assume the integers in the 3-SUM instance are in $[-n^4/2, n^4/2)$ without loss of generality. In order to apply Theorem B.4, we need an efficient $\Sigma_2^{\mathsf{cc}}$ protocol for checking whether 3 numbers sum to zero.

**Theorem B.5.** *For an integer $n$, let $F_{\mathsf{zero}} : \left(\{0,1\}^{4\log n}\right)^3 \to \{0,1\}$ be the function that treats its inputs as three numbers in $[-n^4/2, n^4/2)$ (via a natural encoding), and checks whether they sum to zero. For any $1 \le T \le 4\log n$, $F_{\mathsf{zero}}$ admits a $\Sigma_2^{\mathsf{cc}}$ protocol in which Merlin sends $O(\log n/T)$ bits, Megan sends $\lceil \log(n/T) \rceil$ bits and Alice, Bob and Charles communicate $O(T)$ bits.*

*Proof.* Let $x, y, z$ be three input numbers. Suppose Alice holds $x$, Bob holds $y$ and Charles holds $z$. We add $n^4/2$ to each of them so that they now belong to $[0, n^4)$, and we want to check whether they sum up to $t = n^4/2 \cdot 3$. Assuming $T$ divides $4\log n$ for simplicity, we treat $x, y, z$ as numbers in $2^T$ base. Let $\ell = 4\log n/T$, and $x_1, x_2, \ldots, x_\ell$ be the digits of $x$ (from the least significant one to the most significant one, $y_i$'s, $z_i$'s, and $t_i$'s are defined similarly).

Suppose we add $x, y, z$ together as numbers in $2^T$ base. Let $c \in \{0,1,2\}^\ell$ be a sequence of carries. We can see $x + y + z = t$ for $t = n^4/2 \cdot 3$ with respect to the carry sequence $c$ if and only if

$$x_i + y_i + z_i + c_{i-1} = t_i + c_i \cdot 2^T \quad \text{for } i \in [\ell + 1].$$

In the above we set $c_0 = x_{\ell+1} = y_{\ell+1} = z_{\ell+1} = 0$.

Therefore, in the protocol, Merlin sends the carry sequence $c$, which takes $O(\ell) = O(\log n/T)$ bits. Megan sends an index $i \in [\ell + 1]$. After that, Bob and Charles send $y_i$ and $z_i$ to Alice, respectively, and Alice accepts if and only if the above equality holds. It is straightforward to verify the protocol works. □

Finally, we are ready to prove Theorem B.3.

*Proof of Theorem B.3.* Suppose 3-OV is in truly-subquadratic time. That is, there is a constant $\varepsilon$ such that for all constant $c$, 3-OV$_{n,c\log n}$ can be solved in $n^{2-\varepsilon}$ time.

Given a 3-SUM instance with integer entries in $[-n^4/2, n^4/2)$, it is just a $F_{\mathsf{zero}}$-Satisfying-Triple$_n$ instance. Let $c_1$ be the constant hiding in $O(\log n/T)$ of Theorem B.4, then $F_{\mathsf{zero}}$ admits a $\Sigma_2^{\mathsf{cc}}$ protocol in which Merlin sends $c_1 \log n/T$ bits, Megan sends $\lceil \log(n/T) \rceil$ bits and Alice, Bob and Charles communicate $O(T)$ bits.

Set $T = c_1 \cdot 2/\varepsilon$, Theorem B.4 implies this 3-SUM instance can be reduced to $n^{\varepsilon/2}$ 3-OV$_{n,2^{O(1/\varepsilon)}\log n}$ instances. Applying the algorithm for 3-OV, 3-SUM can be solved in $n^{2-\varepsilon/2}$ time, which completes the proof. □