Removing Additive Structure in 3SUM-Based Reductions*

Ce Jin
cejin@mit.edu
Massachusetts Institute of Technology
Cambridge, USA

Yinzhan Xu
xyzhan@mit.edu
Massachusetts Institute of Technology
Cambridge, USA

ABSTRACT

Our work explores the hardness of 3SUM instances without certain additive structures, and its applications. As our main technical result, we show that solving 3SUM on a size-$n$ integer set that avoids solutions to $a + b = c + d$ for $\{a, b\} \neq \{c, d\}$ still requires $n^{2-o(1)}$ time, under the 3SUM hypothesis. Such sets are called Sidon sets and are well-studied in the field of additive combinatorics.

Combined with previous reductions, this implies that the All-Edges Sparse Triangle problem on $n$-vertex graphs with maximum degree $\sqrt{n}$ and at most $n^{k/2}$ $k$-cycles for every $k \geq 3$ requires $n^{2-o(1)}$ time, under the 3SUM hypothesis. This can be used to strengthen the previous conditional lower bounds by Abboud, Bringmann, Khoury, and Zandir [STOC’22] of 4-Cycle Enumeration, Offline Approximate Distance Oracle and Approximate Dynamic Shortest Path. In particular, we show that no algorithm for the 4-Cycle Enumeration problem on $n$-vertex $m$-edge graphs with $n^{o(1)}$ delays has $O(n^{1+\epsilon})$ or $O(m^{1/3-\epsilon})$ pre-processing time for $\epsilon > 0$. We also present a matching upper bound via simple modifications of the known algorithms for 4-Cycle Detection.

A slight generalization of the main result also extends the result of Dudek, Gawrychowski, and Starikovskaya [STOC’20] on the 3SUM hardness of nontrivial 3-Variate Linear Degeneracy Testing (3-LDTs): we show 3SUM hardness for all nontrivial 4-LDTs. The proof of our main technical result combines a wide range of tools: Balog-Szemerédi-Gowers theorem, sparse convolution algorithm, and a new almost-linear hash function with almost 3-universal guarantee for integers that do not have small-coefficient linear relations.

CCS CONCEPTS

• Theory of computation → Design and analysis of algorithms.

KEYWORDS

fine-grained complexity, 3SUM, additive combinatorics, graph algorithms

1INTRODUCTION

Fine-grained complexity theory provides conditional lower bounds for a wide range of problems, by designing fine-grained reductions from a few central problems that are hypothesized to be hard (see e.g. [63]); specifically, a fine-grained reduction from a central problem $A$ to some problem $B$ of interest would establish a conditional lower bound for $B$ based on the hardness of $A$. Sometimes, certain structured classes of inputs already capture the full hardness of problem $A$, formally shown by a fine-grained reduction from arbitrary instances of $A$ to structured instances of $A$. Such kind of results would be extremely productive for proving conditional lower bounds: having structures in $A$ makes it much easier to design reductions from $A$ to $B$.

A famous example is the equivalence between 3SUM and the (seemingly easier) 3SUM Convolution problem [25, 54]. In the 3SUM problem, we need to determine if a set of $n$ integers contains three integers that sum up to 0. The 3SUM Convolution problem essentially can be thought of as 3SUM with the additional property that the $n$ input integers have distinct remainders modulo $n$. The results in [25, 54] established that 3SUM Convolution requires essentially quadratic time, under the hypothesis that 3SUM requires essentially quadratic time (which is a central hypothesis in fine-grained complexity called the 3SUM hypothesis). It turned out that this extra structure makes it easier to design fine-grained reductions from 3SUM Convolution, leading to tight conditional lower bounds for problems such as Triangle Listing [54] and Exact Triangle [64], under the 3SUM hypothesis.

Dudek, Gawrychowski and Starikovskaya [39] showed that 3SUM is subquadratically equivalent to all nontrivial 3-Variate Linear Degeneracy Testing (3-LDT). In particular, they showed that 3SUM is equivalent to the AVERAGE problem, in which one needs to determine whether a given set of $n$ integers contains a 3-term arithmetic progression involving distinct numbers, i.e., three distinct numbers $a, b, c$ where $a - 2b + c = 0$. In more details, the reduction from AVERAGE to 3SUM was known earlier [41]. The reduction from 3SUM to AVERAGE first goes through a structured version of tripartite AVERAGE, in which there are three given arrays $A, B, C$, and we

*Full version of this paper is available at https://arxiv.org/abs/2211.07048. Ce Jin is supported by NSF Grant CCF-2129139; Yinzhan Xu is supported by NSF Grant CCF-2129139.

1Another equivalent variant of 3SUM is its tripartite version, in which we are given three sets and need to determine if there are three numbers, one from each set, that sum up to 0. In this work, we use 3SUM to refer to the one set version by default.

2In a more popular definition of 3SUM Convolution, we are given an integer array $A$ indexed by, say, $\{0, \ldots, n-1\}$, and the goal is to decide whether there exist $i, j$ such that $A[i] + A[j] + A[i+j] = 0$. Clearly, it is equivalent to 3SUM on the set $\{2n \cdot A[i] + i \mod n\}$, and this set has the aforementioned property.
must have \( a \in A, b \in B, c \in C \); their structure is that all of \( A, B, C \) are 3-AP free (i.e., do not contain 3-term arithmetic progressions involving distinct numbers). It is simple to show that this problem is equivalent to the tripartite version of 3SUM where each array is 3-AP free, by scaling appropriately. This is thus another example of (tripartite) 3SUM that is hard on structured inputs, and it (implicitly) helps showing the equivalence between 3SUM and AVERAGE.

In light of generalizing this result, it is natural to ask whether 3SUM is still hard on inputs without a certain equation involving 4 numbers. One particularly interesting equation involving 4 numbers is \( a + b − c − d = 0 \), and a set without nontrivial solutions to \( a + b − c − d = 0 \) is called a Sidon set (also known as Golomb ruler). Here, a solution is nontrivial if \( \{a, b\} \neq \{c, d\} \). Sidon sets are extensively studied in the field of additive combinatorics (e.g., see the survey [53]) and are also mentioned explicitly in the conference talk [38] as a barrier for generalizing [39]’s results.

As mentioned, [39]’s reduction from 3SUM to AVERAGE goes through a version of tripartite AVERAGE in which all three arrays are 3-AP free. They achieve this by partitioning each input array of an unstructured tripartite AVERAGE to a subpolynomial of sizes \( U^{1-o(1)} \) [18]. On the contrary, all Sidon subsets of \( |U| \) have sizes at most \( \sqrt{U} + O(U^{1/4}) \) [40], which is too small to apply [39]’s technique. Thus, previously there was no answer for the following natural question:

**Question 1:** Does 3SUM on Sidon sets require \( n^{2-o(1)} \) time under the 3SUM hypothesis?

Recently, a work by Abboud, Bringmann, Khoury, and Zimard [4] shows another example of fine-grained hardness on structured problems. Using their “short cycle removal technique”, they were able to show hardness of certain Triangle Detection problems in graphs with few k-cycles. In particular, they showed that detecting whether an m-edge 4-cycle free graph has a triangle requires \( \Omega(m^{1-\frac{1}{198}}) \) time, assuming Triangle Detection on n-vertex graphs with maximum degree at most \( \sqrt{n} \) requires \( n^{2-o(1)} \) time. Here, the structure of the input is 4-cycle freeness. Their technique is also able to provide conditional lower bounds under more standard hypotheses. In particular, they showed that detecting whether each edge is in a triangle (All-Edges Sparse Triangle) on a graph with maximum degree \( \sqrt{n} \) and \( O(n^{2.344}) \) 4-cycles (or more precisely, \( O(n^{2.27+\epsilon}) \) 4-cycles for any \( \epsilon > 0 \), where \( \omega < 2.37286 \) [8] is the square matrix multiplication exponent) requires \( n^{2-o(1)} \) time, assuming that All-Edges Sparse Triangle on a graph with maximum degree \( \sqrt{n} \) requires \( n^{2-o(1)} \) time. As the assumption is known to hold under either the 3SUM hypothesis or the APSP hypothesis [54, 65], the lower bound holds under these two central hypotheses in fine-grained complexity as well. This lower bound (and its more general version for k-cycle) has a variety of applications, including the hardness for Approximate Offline Distance Oracles, Approximate Dynamic Shortest Path, and k-Cycle Enumeration.

However, it is hard to imagine that \( n^{2.344} \) 4-cycles are indeed the smallest amount of 4-cycles on a graph with maximum degree \( \sqrt{n} \), so that All-Edges Sparse Triangle still requires \( n^{2-o(1)} \) time. For instance, consider random graphs with maximum degree \( \sqrt{n} \). It is unclear how the current best \( O(n^{2.37286}/(\omega+1)) \) time algorithm for All-Edges Sparse Triangle [11], or the brute-force \( O(n^5) \) time algorithm that enumerates all pairs of neighbors of each vertex, can exploit the randomness of the graph. To the best of our knowledge, \( O(n^2) \) time is still the best running time for such random graphs, even if perfect matrix multiplication exists (i.e. \( \omega = 2 \)). However, in such random graphs, the expected number of 4-cycles is only \( O(n^2) \) \( \ll n^{2.344} \). It is thus natural to ask whether All-Edges Sparse Triangle on graphs with maximum degree \( \sqrt{n} \) and fewer than \( n^{2.344} \) 4-cycles is hard.

**Question 2:** Does All-Edges Sparse Triangle on a graph with maximum degree \( \sqrt{n} \) and fewer \( n^{2-o(1)} \) 4-cycles still require \( n^{2-o(1)} \) time?

We affirmatively answer both Question 1 and Question 2. Though not obvious, Question 1 and Question 2 are actually strongly related. As we will show later, an affirmative answer to Question 1 actually implies an affirmative answer to Question 2. Our work thus also connects the previous two seemingly unrelated directions of research [4, 39].

### 1.1 Our Results

3SUM on Sidon Set. As our main result, we show that 3SUM on Sidon sets is indeed hard, resolving Question 1.

**Theorem 1.1.** Under the 3SUM hypothesis, for all constants \( \delta > 0 \), 3SUM on size-\( n \) Sidon sets of integers bounded by \( [-n^{2+\delta}, n^{2+\delta}] \) requires \( n^{2-o(1)} \) time.

Our techniques differ from [39]’s techniques in significant ways. As a high level overview, our reduction combines the celebrated Balog-Szemerédi-Gowers Theorem [14, 45] and efficient sparse convolution algorithm [13, 19−21, 26, 30, 44, 52, 56] to solve 3SUM instances on sets with very high additive energy (many tuples \( (a, b, c, d) \) with \( a + b = c + d \)) in truly subquadratic time. On the other hand, for 3SUM instances on sets with moderately low additive energy, we modify known self-reductions of 3SUM [16] by designing hash functions with better universality guarantee, to self-reduce such 3SUM instances to 3SUM instances on Sidon sets. See Section 1.2 for a more detailed overview.

Given Theorem 1.1, it is not difficult to obtain the following corollary using techniques in [39].

**Corollary 1.2.** Under the 3SUM hypothesis, for all \( \delta > 0 \), determining whether a given set of \( n \) integers bounded by \( [-n^{3+\delta}, n^{3+\delta}] \) is a Sidon set requires \( n^{2-o(1)} \) time.

More generally, we are able to show that all nontrivial 4-LDTs are 3SUM-hard, generalizing [39]’s result on 3-LDTs. A 4-LDT is parameterized by integers \( \beta_1, \beta_2, \beta_3, \beta_4, t \), and asks to determine whether a given integer set \( A \) contains a solution to \( \sum_{i=1}^{4} \beta_i a_i = 0 \) for distinct \( a_i \in A \). Following [39]’s notation, a 4-LDT parameterized by \( \beta_1, \beta_2, \beta_3, \beta_4, t \) is called trivial if either

1. Any of \( \beta_1, \beta_2, \beta_3, \beta_4 \) is 0, or
2. \( t \neq 0 \) and \( \gcd(\beta_1, \beta_2, \beta_3, \beta_4) = t \).

For the first case, the 4-LDT is degenerated to a 3-LDT; for the second case, the answer is always NO. All other 4-LDTs are called nontrivial. Using their techniques, it is easy to show that a nontrivial...
4-LDT parameterized by $\beta_1, \beta_2, \beta_3, \beta_4, t$ requires $n^{2-o(1)}$ time under the 4SUM hypothesis (which is a weaker hypothesis than the 3SUM hypothesis) if $t \neq 0$ or $\beta_1 + \beta_2 + \beta_3 + \beta_4 \neq 0$. Therefore, we will focus on the remaining cases, i.e., $t = 0$ and $\beta_1 + \beta_2 + \beta_3 + \beta_4 = 0$.

**Theorem 1.3.** Fix any non-zero integers $\beta_1, \beta_2, \beta_3, \beta_4$ where $\sum_{i=1}^{4} \beta_i = 0$ and any real number $\delta > 0$. Determining whether a size-$n$ set of integers bounded by $[-n^{3+\delta}, n^{3+\delta}]$ avoids solutions $\sum_{i=1}^{4} \beta_i a_i = 0$ for distinct $a_i$ in the set requires $n^{2-o(1)}$ time, assuming the 3SUM hypothesis.

We remark that this definition of 4-LDT does not perfectly fit the definition of Sidon sets. When $\beta_1 = \beta_2 = 1$ and $\beta_3 = \beta_4 = -1$, a solution $2a = b + c$ is not allowed in Sidon sets as $(a, a) \neq (b, c)$, but is allowed in the definition in Theorem 1.3, as $a, b, c$ are not all distinct. However, it would still make sense to count $2a = b + c$ as a solution to the 4-LDT, as such solutions do not trivially exist in all sets. Thus, we define a slight variant of 4-LDT, in which we need to determine whether a size-$n$ set $A$ of integers avoids nontrivial solutions to $\sum_{i=1}^{4} \beta_i a_i = 0$ for $a_i \in A$. Here, a solution is trivial if for every $a \in \{a_1, \ldots, a_4\}$ it holds that $\sum_{i: a_i = a} \beta_i = 0$. We will show that Theorem 1.3 still works under this alternative definition (and all nontrivial 4-LDTs with $t \neq 0$ or $\beta_1 + \beta_2 + \beta_3 + \beta_4 \neq 0$ are still 4SUM hard under this definition, following previous techniques).

Quasirandom graph. Question 2 concerns graphs with few, say $n^2$, $4$-cycles. As mentioned, in a random $n$-vertex graph with maximum degree at most $\sqrt{n}$, we expect to see $\Theta(n^4)$ 4-cycles. Graphs in which the numbers of 4-cycles are close to those of random graphs with the same edge density are actually well-studied in additive combinatorics, and such graphs are called pseudorandom graphs (see e.g. [69]). In additive combinatorics, a sequence of graphs $(G_n)$ with $G_n$ having $n$ vertices and $(p + o(1))(\binom{n}{2})$ edges are called (sparse) pseudorandom graphs if the number of labeled 4-cycles in $G_n$ is at most $(1 + o(1))p n^4$. We adapt this terminology as follows:

**Definition 1.4.** (Quasirandom Graph). An undirected unweighted $n$-vertex graph is called a quasirandom graph if it has maximum degree at most $\sqrt{n}$ and has at most $n^2$ 4-cycles.

As an application of Theorem 1.1, we show that All-Edges Sparse Triangle is still hard even on quasirandom graphs, answering Question 2 affirmatively.

**Theorem 1.5.** Under the 3SUM hypothesis, All-Edges Sparse Triangle on $n$-vertex quasirandom graphs requires $n^{2-o(1)}$ time.

Theorem 1.5 actually implies hardness of All-Edges Sparse Triangle on certain graphs with few $k$-cycles for any $k \geq 3$.

**Corollary 1.6.** Under the 3SUM hypothesis, All-Edges Sparse Triangle on $n$-vertex graphs which has maximum degree at most $\sqrt{n}$ and has at most $n^{k/2}$ $k$-cycles for every $k \geq 3$ requires $n^{2-o(1)}$ time.

4-Cycle Enumeration. As Theorem 1.5 improves a result of [4], we naturally obtain improved conditional lower bounds for several problems they consider. In particular, we achieve tight conditional lower bound for the 4-Cycle Enumeration problem.

In the 4-Cycle Enumeration problem, we need to first pre-process a given simple graph, and then enumerate all the 4-cycles in this graph with subpolynomial time delay for every 4-cycle enumerated. This problem was first studied by [4], inspired by both the classic 4-Cycle Detection problem [11, 68] and the recent trend of enumeration algorithms [23, 24, 42, 60].

[4] showed an $n^{4 \pm o(1)}$ pre-processing time lower bound for 4-Cycle Enumeration on $m$-edge graphs, under either the 3SUM hypothesis or the APSP hypothesis. This lower bound is only $m^{2/4-o(1)}$ even if $\omega = 2$. Using Theorem 1.5, we show the following improved lower bound.

**Theorem 1.7.** (4-Cycle Enumeration, lower bound). Assuming the 3SUM hypothesis, there is no algorithm with $O(n^{2-\varepsilon})$ pre-processing time and $n^{o(1)}$ delay that solves 4-Cycle Enumeration on $n$-node graphs with $m = \lfloor 0.49n^{1.5} \rfloor$ edges, for any constant $\varepsilon > 0$.

In terms of $m$, this lower bound is $m^{6/3-o(1)}$. By the same reasoning as [4], Theorem 1.7 and the known 3-SUM hardness of Triangle Listing [54] imply an $n^{4/3-o(1)}$ pre-processing lower bound for $k$-Cycle Enumeration for any $k \geq 3$, under the 3SUM hypothesis.

Note that $m = n^{1.5-o(1)}$ is the hardest density for 4-Cycle Enumeration. See the full version for more details.

It is known how to solve 4-Cycle Detection in $O(\min\{n^2, m^{4/3}\})$ time [11, 68]. We show, by simple modifications of the existing 4-Cycle Detection algorithms, that 4-Cycle Enumeration can also be solved in $O(\min\{n^2, m^{4/3}\})$ pre-processing time. Thus, the conditional lower bound in Theorem 1.7 is indeed tight.

**Theorem 1.8.** (4-Cycle Enumeration, upper bound). Given an $n$-vertex $m$-edge undirected graph, we can enumerate $4$-cycles in $O(1)$ delay after an $O(\min\{n^2, m^{4/3}\})$ time pre-processing. The algorithm is deterministic.

Offline Approximate Distance Oracle and Dynamic Approximate Shortest Paths. Another result obtained by [4] is the hardness of Offine Approximate Distance Oracle. A Distance Oracle needs to pre-process a given graph, and then answer (approximate) distance between two query vertices. There exist Distance Oracles that can pre-process an $n$-vertex $m$-edge undirected weighted graph in $O(mn/k)^{4}$ time and then answer $(2k - 1)$-approximate distance queries in $O(1)$ time, for any constant integer $k \geq 1$ [27, 57, 62], and it remains the current best trade-off between pre-processing time and approximate factor. It is thus a natural question to ask whether near-linear pre-processing time is possible for some constant approximation factors.

Pătraşcu, Roditty, and Thorup [55] showed that all Distance Oracles with $(3 - \varepsilon)$-approximation factor and constant query time must use $O(m^{1+\varepsilon})$ space, under a set intersection conjecture. This implies that near-linear pre-processing time is impossible for $(3 - \varepsilon)$ approximation. [4] ruled out this for all $k \geq 4$ as well. More specifically, they showed that, for any $k \geq 4, \delta > 0$, any algorithm for returning a $(k - \delta)$-approximation of the distances between $m$ pairs of vertices (given at once) in an $m$-edge undirected unweighted graph requires $m^{1+\frac{\delta}{2(k-\delta)}} \cdot n^{o(1)}$ time, under either the 3SUM hypothesis or the APSP hypothesis. When $\omega = 2$, the
lower bound becomes $m^{1+\frac{1}{2\varepsilon^2}+\omega(1)}$. This essentially establishes that $m^{1+\frac{1}{2\varepsilon^2}}$ pre-processing time is the correct answer, for achieving $k$-approximation and near-constant query time. However, the constant factor hidden in $\Theta(k)$ still does not match.

Combining Theorem 1.5 and [4]'s technique, we make progress towards closing this gap.

Theorem 1.9 (Offline Distance Oracles, I). Assuming the 3SUM hypothesis, for any constant integer $k \geq 2$ and $\varepsilon, \delta > 0$, there is no $O(m^{1+\frac{1}{2\varepsilon^2}-\varepsilon})$ time algorithm that can achieve $(k-\delta)$-approximate the distances between $m$ given pairs of vertices in a given $n$-vertex $m$-edge undirected unweighted graph, where $m = \Theta(n^{1+\frac{1}{2\varepsilon^2}})$.

Theorem 1.10 (Offline Distance Oracles, II). Assuming the 3SUM hypothesis, for any constant integer $k \geq 3$ and $\varepsilon, \delta > 0$, there is no $(k-\delta)$-approximate distance oracle with $O(n^{1+\frac{1}{2\varepsilon^2}-\varepsilon})$-processing time and $n^{o(1)}$ query time for an $n$-vertex $O(n)$-edge undirected unweighted graph.

Note that Theorem 1.10 has a higher lower bound (in terms of the number of edges) for pre-processing time, while Theorem 1.9 applies to distance oracles with possibly slower query time $m^{1/(2k-1)-\varepsilon}$. Compared to the Thorup-Zwick distance oracle [62] with $(2k-1)$ approximation, constant query time and $O(n^{1+\frac{1}{k}})$-pre-processing time on $O(n)$-edge graphs, our lower bound in Theorem 1.10 loses a factor of 2 on the exponent for large constant $k$, while the previous bound by [4] loses a factor of $\delta$ when $\varepsilon = 2$. Dalirrooyfard, Jin, Vassilevska Williams and Wein [32] studied a similar question called n-Pair Shortest Paths, which is the Offline Distance Oracle problem with $n$ queries, and obtained close-to-optional combinatorial lower bounds for algorithms achieving $(1 + 1/k)$-approximation.

Another problem similar in nature to Distance Oracle is Dynamic Shortest Paths. Here, the difference is that we also need to support updates that can insert or delete an edge in the given graph. For its decremental version where only edge deletions are allowed, data structures with $O(n/k)$ amortized update time and $O(1)$ query time for $O(k)$ approximation are known for weighted undirected graphs [28]. For the fully dynamic version, the best known data structure with $O(n/k)$ amortized update time and $O(1)$ query time provides $O(\log n)O(k)$-approximation [43].

For $(k-\delta)$-approximations, [4] shows that when $\varepsilon = 2$, no algorithms for Decremental Dynamic Approximate Shortest Path on undirected unweighted graphs can have both $O(m^{1+\frac{1}{2\varepsilon^2}})$ total update time and $O(m^{\frac{1}{2\varepsilon^2}-\varepsilon})$ query time for positive $\varepsilon$, under either the 3SUM hypothesis or the APSP hypothesis. As mentioned in [4], this bound follows immediately from their lower bound for Offline Distance Oracle. Thus, our Theorem 1.9 implies improved lower bounds under the 3SUM hypothesis: no algorithms can have both $O(m^{1+\frac{1}{2\varepsilon^2}})$ total update time and $O(m^{\frac{1}{2\varepsilon^2}})$ query time.

They also provided a conditional lower bound for Fully Dynamic Approximate Shortest Path. More specifically, they showed that when $\varepsilon = 2$, no algorithm can pre-process an $n$-vertex undirected unweighted graph in $O(n^3)$ time and supports fully dynamic updates and queries in $O(m^{\frac{1}{2\varepsilon^2}})$ time, where the queries need to be approximated within $(k-\delta)$ factor, for $\delta, \varepsilon > 0$, under either the 3SUM hypothesis or the APSP hypothesis. Combining their approach with Theorem 1.5, we also obtain improved lower bounds.

Theorem 1.11 (Dynamic Approximate Shortest Path). Assuming the 3SUM hypothesis, for any constant integer $k \geq 3$ and $\varepsilon, \delta > 0$, no algorithm can support insertion and deletion of edges and support querying $(k-\delta)$-approximate distance between two vertices in $O(m^{\frac{1}{2\varepsilon^2}-\varepsilon})$ time per update and query, after an $O(n^3)$ time pre-processing, in $n$-vertex $m$-edge undirected unweighted graphs, where $m = \Theta(n^{1+\frac{1}{2\varepsilon^2}})$.

All-Nodes Shortest Cycles. We also explore the conditional lower bound of the All-Nodes Shortest Cycles problem, which was not considered by [4]. This problem, we are given a graph and are required to compute the length of the shortest cycle through every vertex. This problem was first considered by Yuster [67], who gave an $O(n^3/k)$ time algorithm for weighted undirected graph. It was later improved by Agarwal and Ramachandran [7] to $O(n^3)$ time. Sankowski and Węgrzycki [59] showed the same $O(n^3)$ time bound for unweighted directed graphs.

The study of the All-Nodes Shortest Cycles problem in the approximate setting was initiated by Dalirrooyfard, Jin, Vassilevska Williams and Wein [32], who gave various algorithms and conditional lower bounds for approximate All-Nodes Shortest Cycles. In particular, they showed an $O(mn^k/k)$ time algorithm for $(k+\varepsilon)$-approximate All-Nodes Shortest Cycles in undirected unweighted graphs, for arbitrary $\varepsilon > 0$.

Using Theorem 1.5, we show the following conditional lower bound, suggesting that $m^{1+\frac{1}{2\varepsilon^2}}$ is likely the correct running time for $k$-approximate All-Nodes Shortest Cycles.

Theorem 1.12 (All-Nodes Shortest Cycles). Fix any integer $k \geq 4$. Assuming the 3SUM hypothesis, for any $\varepsilon, \delta > 0$, there is no $O(m^{1+1/k-\varepsilon})$ time algorithm that can solve the All-Nodes Shortest Cycles problem within $(k/3 - \delta)$ approximation factors on $n$-vertex $m$-edge graphs with $m = \Theta(n^{1+\frac{1}{2\varepsilon^2}})$.

Triangle Detection. As mentioned, [4] showed a conditional lower bound for Triangle Detection on a 4-cycle free graph, assuming Triangle Detection on $n$-vertex graphs with maximum degree at most $\sqrt{n}$ requires $n^{2-o(1)}$ time.

Even though All-Edges Sparse Triangle on $n$-vertex graphs with maximum degree at most $\sqrt{n}$ requires $n^2-o(1)$ time under either the 3SUM hypothesis or the APSP hypothesis [54, 65], the same is not known for Triangle Detection. In fact, it is an open problem to base the hardness of Triangle Detection on some central hypotheses in fine-grained complexity, explicitly asked by [4]. Towards resolving this open problem, [4] proposes a new hypothesis, which they call the Strong Zero-Triangle conjecture. It states that detecting a zero-weight triangle in an edge-weighted tripartite graph with vertex parts of sizes $A, B, C$ and with integer weights in $\{-W, \ldots, W\}$ requires $\min\{W(AB + BC + CA), ABC\}^{1-o(1)}$ time. Under this hypothesis, they showed that Triangle Detection on $n$-vertex graphs with maximum degree at most $\sqrt{n}$ requires $n^2-o(1)$ time.

As a side result, we make another progress towards this open problem. Our hardness result is based on a more well-known hypothesis called the Strong 3SUM hypothesis, which was first proposed by Amir, Chan, Lewenstein and Lewenstein [12], and later
used by, e.g., [1, 2, 22, 46, 51]. We remark that there is no known direct relations between the Strong Zero-Triangle conjecture and the Strong 3SUM hypothesis. As far as we know, neither, either, or both of them could be true.

**Hypothesis 1.13** (Strong 3SUM hypothesis). In the Word-RAM model with $O(\log n)$-bit words, 3SUM on size-$n$ set of integers from $\{-n^2, n^2\}$ cannot be solved in $O(n^{1+\varepsilon})$ time, for any positive constant $\varepsilon > 0$.

We show the following:

**Theorem 1.14.** Under the Strong 3SUM hypothesis, Triangle Detection on $n$-node graphs with maximum degree $O(n^{1/4})$ requires $n^{1.5-o(1)}$ time.

Our lower bound is arguably lower than that in [4]: in terms of $m$, their lower bound is $m^{5/3-o(1)}$, while ours is $m^{6/5-o(1)}$. Nevertheless, basing the hardness of Triangle Detection on a more popular hypothesis gives more confidence that it requires super-linear time.

Combining the techniques of [4] with Theorem 1.14, one can obtain an $m^{1+O(1)}$-time lower bound for 4-cycle detection in $m$-edge graphs, assuming the Strong 3SUM hypothesis. Of course, the exponent here can only be lower than what [4] obtained from their Triangle Detection hypothesis.

### 1.2 Technical Overview

In this section, we will describe the high-level ideas of our reductions from 3SUM to 3SUM on Sidon sets, and subsequently to All-Edges Sparse Triangle on quasirandom graphs.

The additive energy of a set $A \subseteq \mathbb{Z}$, which we call $E(A)$, is defined as the number of tuples $(a, b, c, d) \in A$ such that $a+b = c+d$. The first component of our reduction is an efficient algorithm for 3SUM on sets with very large $|A|/K$ for some small $K$) additive energy. For sets $A$ with $E(A) < n^3/K$ (moderate energy), we use self-reduction of 3SUM to reduce the instance to a number of smaller instances, so that the total additive energy of the smaller instances is small. This means that there are very few tuples $(a, b, c, d)$ from the same instance such that $a+b = c+d$ and $(a, b) \neq (c, d)$, so it suffices to remove these numbers from the instances that contain them. In the following, we describe each of these steps in more details.

**Efficient algorithm for sets with high additive energy.** First, suppose we need to solve 3SUM on a size-$n$ set $A$ where $E(A) \geq n^3/K$ for some small $K$. By the Balog-Szemerédi-Gowers Theorem, such a set $A$ contains a large subset $A' \subseteq A$ of size $|A'| \geq K^{-O(1)}n$, and with small doubling, $|A'| = K^O(1)|A'|$. Furthermore, such a subset can be found in subquadratic time, by an adaptation of an algorithmic version of the Balog-Szemerédi-Gowers Theorem given by Chan and Lewenstein [26]. If we are able to solve the tripartite version of 3SUM on sets $A', A, A$, then we can remove $A'$ from $A$ afterwards. We repeat this procedure until either the size of $A$ becomes truly sublinear, when we can use brute-force to solve 3SUM on $A$, or the energy of $A$ becomes small, when we can apply the reduction for the moderate energy case. Either way, as the size of $A'$ is large, we only need to repeat $K^{O(1)}$ times.

Therefore, it suffices to give an efficient algorithm for solving tripartite 3SUM on size-$n$ sets $A, B, C$, where $A$ has small doubling, i.e., $|A+A|$ is small. We will show an algorithm that runs in $\tilde{O}(\sqrt{n}|A+A|)$ time, which is truly subquadratic when $|A+A|$ is sufficiently small. The algorithm roughly works as follows. First, we show an efficient algorithm that partitions $B$ into subsets $B_1, B_2, \ldots, B_m$, so that each $B_i$ is a subset of a shift of $A$, i.e., there exists $s_j$ such that $B_j \subseteq A + s_j$. Furthermore, our algorithm also finds $C_j$, which is a superset of $C \setminus (A+B_j)$, so that $\sum_j |C_j|$ is bounded by $\tilde{O}(|A+A|)$ (we cannot afford to simply set $C_j$ to be $C \cap (A+B_j)$, as finding $C \cap (A+B_j)$ is not simpler than solving 3SUM on $A, B_j, C$). Clearly, for some $i$ and $b \in B_i$, the only possible numbers in $C$ that can form a 3SUM solution with $b$ are from $C_i$. If the size of $C_i$ is smaller than some parameter $t$, we enumerate all pairs $b \in B_i$ and $c \in C_i$, and check whether they are in a 3SUM solution in $O(1)$ time; over all such $C_i$, it takes $\tilde{O}(nt)$ time. Otherwise, we use sparse convolution [20] to compute $A + B_i$ and test whether $C_i \cap (A+B_i)$ is empty in $\tilde{O}(|C_i| + |A + B_i|)$ time. Note that sparse convolution runs in $\tilde{O}(|A + B_i|) \leq \tilde{O}(|A + (A + s_j)|) = \tilde{O}(|A + A|)$ time, so over all such $C_i$, it takes $\tilde{O}(|\sum_j |C_j| + \sum_j |C_j| - |A + A|) = \tilde{O}(|A + A| + \frac{|A+A|}{n})$. Setting $t = \frac{|A+A|}{n}$ gives the desired $\tilde{O}(\sqrt{n}|A+A|)$ time.

To give some intuition why it is possible to find $B_1, \ldots, B_m$ and $C_1, \ldots, C_m$, we describe the following (inefficient) algorithm that is analogous to our efficient algorithm. Suppose for each $z \in \mathbb{Z}$, we add it to a set $S$ with probability $\frac{\log n}{z}$. For every $b \in B$, there exists $s \in S$ such that $b + A + s$ if and only if $S \cap (b - A) \neq \emptyset$, which happens with high probability. Thus, for every $b$, we can arbitrarily assign it one of the $s \in S$ such that $b + A + s$. Grouping $b \in B$ assigned with the same $s$ to the same group forms the partition $B_1, \ldots, B_m$. Then let $C_i = C \setminus (s_i + A + A)$. It is a superset of $C \setminus (A+B_j)$ since $B_j \subseteq s_i + A$. Also, each $c \in C_i$ is in $C_j$ if and only if $s_i \in (A+A+c)$. As each number in $(A+A+c)$ is added to $S$ with probability $O(\frac{\log n}{z^2})$, the expected number of such $s_i$ is $\tilde{O}(\frac{|A+A|}{n})$, i.e., $c$ appears in $\tilde{O}(\frac{|A+A|}{n})$ many $C_i$ in expectation. Summing over all $c \in C$ gives the desired $\sum_i |C_i| \leq \tilde{O}(|A + A|)$ bound.

**Self-reduction for sets with moderate additive energy.** It is well-known that the 3SUM problem has an efficient self-reduction [16] through almost-linear hash functions, such as modulo a random prime, or Dietzfelbinger’s hash function (see e.g., [25, 36, 37]). Say the input range of a 3SUM instance is $[-U, U]$ and consider an almost-linear hash family $H$ mapping from $[-U, U]$ to $[m]$. The almost-linear property states that, for any $H \in H$ and every $a, b \in [-U, U]$, $H(a) + H(b) + H(-a - b)$ can only have $U^{O(1)}$ possible values. Let us first review the high level ideas of the self-reduction of 3SUM. The self-reduction first samples $H \sim H$, and creates a bucket $G_x$ for every $x \in [m]$ that contains every number $a \in A$ where $H(a) = x$. Then we enumerate triples of buckets, and solve a tripartite 3SUM on numbers from these three buckets. By the almost-linear property, we only need to enumerate $m^{U^{O(1)}}$ triples of buckets. Suppose each bucket has $O(m)$ numbers, we get $m^{U^{O(1)}}$ small instances of 3SUM on sets of sizes $O(m)$.

Suppose the input set $A$ has moderate additive energy $E(A) < n^3/K$, i.e., there are only $O(n^3/K)$ tuples $(a, b, c, d) \in A$ with $a + b = c + d$ and $(a, b) \neq (c, d)$. For simplicity, let us focus on the tuples with distinct $a, b, c, d$. Suppose that we can bound the probability of $a, b, c, d$ being in the same bucket by $p$. By the almost-linear property, this bucket appears in $ml^{O(1)}$ small instances. Over all
small instances, the expected number of tuples \((a, b, c, d)\) in an instance with \(a + b = c + d\) and \(\{a, b\} \neq \{c, d\}\) contributed this way is roughly \(n^4 \cdot p \cdot m^{|U|+1}\). Such tuples can appear in a small instance via other possibilities, such as the case where \(a, c\) are in a bucket while \(b, d\) are in another bucket, and these two buckets belong to some small instance. However, we ignore such cases in this overview for simplicity. For each such tuple, we remove all its 4 numbers from the small instance they belong to, but we need to pay \(O(n/m)\) time for each number in order to check 3SUM solutions involving them in a brute-force way. Therefore, the overall running time becomes \(n^4 \cdot p \cdot m^{|U|+1}\) and the small instances are on Sidon sets (after we convert the tripartite instance to one-set version in a standard way). In order for this running time to be truly subquadratic, we need \(p\) to be close to \(1/n^2\). This is possible when \(m\) is close to \(n\) and the hash family has almost 3-wise independence guarantees.

Unfortunately, given the almost-linearity requirement, it seems difficult to achieve full 3-wise independence: for three integers \(x, y, z\) with \(x + y + z = 0\), the hash value of \(z\) is almost determined (up to \(|U|+1\) possibilities) by the hash values of \(x\) and \(y\). We mitigate this by only requiring 3-universality on triples \((x, y, z)\) with certain properties. More specifically, our hash functions satisfy that \(\Pr[H(x) = H(y) = H(z)]\) is roughly \(\frac{1}{m}\) on integers \(x, y, z\) such that there does not exist integers \(a, b, y\) with small absolute values such that \(a + b + y = 0\), \(a, b, y\) are not all zeros and \(ax + by + yz = 0\). Furthermore, we borrow the proof idea from [39] that uses Behrend’s set [18] to split each bucket to multiple sub buckets so that triples of integers with such relations do no appear in the same sub bucket.

Reduction to pseudorandom graphs. Next, we show hardness of All-Edges Sparse Triangle on pseudorandom graphs by reducing from 3SUM on Sidon sets. The reduction follows a previous line of reduction from 3SUM to All-Edges Sparse Triangle via 3SUM Convolution and Exact Triangle [54, 64, 65]. As we will show in our reductions, the number of tuples \((a, b, c, d)\) with \(a + b = c + d\) in the 3SUM instance relates to the number of 4-cycles in the All-Edges Sparse Triangle instance, so starting from a 3SUM instance on Sidon set helps reducing the number of 4-cycles in the All-Edges Sparse Triangle instance. Along the way of the reduction, we also achieve a hardness result for Exact Triangle on graphs with certain properties (see full version for details).

Comparison with [4]. The short cycle removal technique of [4] can be seen as removing short cycles directly in the input graph of a Triangle Detection or All-Edges Sparse Triangle instance, which incurs some overhead in time complexity. In comparison, our approach removes 4-cycles in a more indirect way: we trace the hardness of All-Edges Sparse Triangle back to 3SUM, and remove tuples \((a, b, c, d)\) with \(a + b = c + d\) and \(\{a, b\} \neq \{c, d\}\) (or, “arithmetic 4-cycles”) in the 3SUM instance, which translates to removing 4-cycles in the All-Edges Sparse Triangle instance. The benefit of our approach is that we can exploit the additive structure of 3SUM and apply various tools from additive combinatorics and additive algorithms, so that our reduction removes cycles more efficiently. However, one advantage of their results is that their lower bounds hold under APSP hypothesis as well as 3SUM hypothesis.

1.3 Independent Works
Concurrently and independently, Abboud, Bringmann, and Fischer [3] proved similar fine-grained lower bounds for the 4-Cycle Enumeration problem (or, 4-Cycle Listing) (Theorem 1.7), and for the Approximate Distance Oracle problem (Theorem 1.10). Additionally, they obtained fine-grained lower bounds for Approximate Distance Oracle with stretch \(2 \leq \alpha < 3\) [3, Theorem 1.3]. Similar to our proof, [3] also uses an energy reduction framework, with some technical parts implemented differently. One noticeable difference is that they reduce the additive energy of 3SUM instance \(A\) to \(O(|A|^{2+\delta})\) (for arbitrary constant \(\delta > 0\), while we perform more steps to reduce it all the way down to \(2|A|^4 - |A|\) (i.e., \(A\) is a Sidon set). The former bound already suffices for the application to 4-Cycle Listing and Approximate Distance Oracles in [3], whereas the latter bound allows us to show further results such as the 3SUM-hardness of 4-LDT (Theorem 1.3).

Concurrently and independently, Abboud, Khouby, Leibowitz, and Safer [5] also gave an algorithm for listing all 4-cycles in an \(n\)-node \(m\)-edge undirected graph in \(\tilde{O}(\min\{n^2, m^{6/3}\}) + t\) time.

1.4 Further Related Works
Chan and Lewenstein [26] designed subquadratic-time algorithms for a certain clustered version of 3SUM. One of their key components is an algorithmic version of the Balog-Szemerédi-Gowers Theorem in a very generalized scenario that requires one to keep track of the uncovered edges in a dense bipartite graph, which needs quadratic time in total (see [26, Theorem 2.3]). Our proof borrows one of their subroutines, but does not require the full generality of their algorithm. This then allows us to use Ruzsa triangle inequality to avoid spending quadratic time, which is crucial to our proof.

Many problems studied in fine-grained complexity or parameterized complexity have a monochromatic version and a colorful version. The monochromatic version cannot be harder than the colorful version, by a simple reduction using color coding [10]. For several important problems, such as OV and 3SUM, the reverse direction is also true, proved by simple gadget reductions. However, for some other problems, the reduction is extremely nontrivial. Examples include Euclidean Closest Pairs [6, 9, 47, 66], Bidirectional Graph Diameter [33–35, 48], 4-Cycle Detection [4, 49], 3-LDTs [39]. Our work provides another example of this phenomenon: 4-partite version of 4-LDTs can be easily shown to be 4SUM hard, but for 1-partite version of some 4-LDTs it takes a lot of effort to just prove 3SUM-hardness. On the other hand, for some problems, the monochromatic property can be nontrivially used in designing algorithms that run faster than the colorful version, e.g., Element Distinctness in the low-space setting ([15, 17, 29, 50]).

1.5 Paper Organization
We give necessary definitions and backgrounds in Section 2. Next, we reduce 3SUM to 3SUM on sets with moderate additive energy in Section 3 and then further reduce to 3SUM on Sidon sets in Section 4.

In the full version of this paper, this result is used to show the 3SUM-hardness of All-Edges Sparse Triangle on quasirandom

\(^\text{To get subquadratic overall time, they need to apply this lemma on compressed instances.}\)
graphs, which is then applied to show conditional lower bounds of 4-Cycle Enumeration, Distance Oracles, Dynamic Shortest Paths and All-Nodes Shortest Cycles. In the full version, we also show algorithms that match the above-mentioned conditional lower bounds of 4-Cycle Enumeration. In the full version, we show a lower bound on Triangle Detection under the Strong 3SUM hypothesis.

Finally, we conclude with several open problems in Section 5.

2 PRELIMINARIES

Denote \([n] = \{1, 2, \ldots, n\}\).

We use the convention that \((a \mod p) \in \{0, 1, \ldots, p - 1\} \subseteq \mathbb{Z}\) regardless of the sign of \(a \in \mathbb{Z}\). For a set \(A\), denote \(A \mod p = \{a \mod p : a \in A\}\).

2.1 Problem Definitions

The 3SUM problem is defined as follows.

**Definition 2.1 (3SUM).** Given an integer set \(A \subseteq \mathbb{Z} \cap [-U, U]\) of size \(|A| = n\), decide if there exist \(a, b, c \in A\) such that \(a + b + c = 0\).

**Hypothesis 2.2 (3SUM hypothesis).** In the Word-RAM model with \(O(\log n)\)-bit words, 3SUM with input range \(U = n^3\) cannot be solved in \(O(n^{2-\varepsilon})\) time, for any positive constant \(\varepsilon > 0\).

There are several variants of the 3SUM problem studied in the literature. To name a few, one could require integers \(a, b, c\) to be distinct, or could ask for \(a + b + c = t\) for a given target \(t\) not necessarily zero. Another variant is the 3-partite version (sometimes called colorful 3SUM), where the input contains three sets \(A, B, C\) instead of a single set, and \(a \in A, b \in B, c \in C\) is required. It is a standard exercise to show the equivalence of these variants of 3SUM (see also [39]).

Sidon sets are well-studied objects in additive combinatorics. We consider the computational problem of deciding whether a set is a Sidon set.

**Definition 2.3 (Sidon Set Verification).** Set \(A \subseteq \mathbb{Z}\) is called a Sidon set if \(A\) contains no solutions to \(a + b = c + d\) except when \(\{a, b\} = \{c, d\}\).

In the Sidon Set Verification problem, we are given an integer set \(A \subseteq \mathbb{Z} \cap [-U, U]\) of size \(|A| = n\), and need to decide whether \(A\) is a Sidon set.

The 3SUM problem and the Sidon Set Verification problem are special cases of the more general (homogeneous) k-Variate Linear Degeneracy Testing (k-LDT) problem [39]: for a fixed homogeneous linear equation \(\sum_{i=1}^{k} \beta_i a_i = 0\) with non-zero integer coefficients \(\beta_i\), given an input integer set \(A\), find a good solution \((a_1, \ldots, a_k) \in A^k\) that satisfies the equation. We consider two variants for the definition of good solutions:

1. The solution contains distinct \(a_i\). This is the definition used in [39].
2. The solution is nontrivial, as defined below.

**Definition 2.4 (Nontrivial solutions to a linear equation).** A solution \((a_1, \ldots, a_k)\) to the equation \(\sum_{i=1}^{k} \beta_i a_i = 0\) is called trivial, if for every \(a \in \{a_1, \ldots, a_k\}\) it holds that \(\sum_{a_i = a} \beta_i = 0\). All other solutions are called nontrivial.

The distinct \(a_i\) definition is more restrictive than the nontrivial definition. Note that nontrivial solutions can only exist when \(\sum_{a_i = a} \beta_i = 0\).

For example:

- In the AVERAGE problem (equation \(a_1 - 2a_2 + a_3 = 0\)), the trivial solutions are \((a_1, a_2, a_3) = (a, a, a)\) for all \(a \in A\). In this case, the definition coincides with the distinct \(a_i\) definition.
- In the Sidon Set Verification problem (equation \(a_1 + a_2 - a_3 = a_4 = 0\)), the trivial solutions are \((a_1, a_2, a_3, a_4) = (a, b, a, b)\) or \((a, b, b, a)\) for all \(a, b \in A\). For \(w \neq 0\), \((a, a+2w, a+w, a+w)\) is a nontrivial solution, but does not have distinct \(a_i\).

In this paper, we only consider (homogeneous) 3-LDTs and 4-LDTs. Moreover, we only study the hardness of (homogeneous) 4-LDTs.
with zero coefficient sum $\sum_{i=1}^{d} \beta_i = 0$, because $k$-LDTs with non-homogeneous equations or nonzero coefficient sum are known to be either trivial or $k$SUM-hard [39].

### 2.2 Additive Combinatorics

Denote $a + B := \{a + b : b \in B\}$, and $a \cdot B := \{ab : b \in B\}$.

**Definition 2.5** (Sumset). For sets $A, B \subseteq \mathbb{Z}$, define their sumset as $A + B := \{a + b : a \in A, b \in B\}$.

For finite $A \subseteq \mathbb{Z}$, $|A + A|/|A|$ is called the doubling constant of $A$.

We use the sparse convolution algorithm to compute sumsets [13, 19–21, 26, 30, 44, 52, 56].

**Theorem 2.6** (Sparse convolution, [20]). Given two integer sets $A, B \subseteq \{U\}$, there is a deterministic algorithm that computes their sumset $A + B$ with output-sensitive time complexity $O(|A + B| \cdot \text{polylog } U)$.

**Definition 2.7** (Additive energy). Let $A \subseteq \mathbb{Z}$ be a finite set. The additive energy of $A$ is defined as

$$E(A) := \left|\{(a, b, c, d) \in A \times A \times A \times A : a + b = c + d\}\right|.$$

Let $r_A(x) := \left|\{(a, b) \in A \times A : a + b = x\}\right|$. Then

$$E(A) = \sum_{x} r_A(x)^2.$$

It holds that

$$2|A|^2 - |A| \leq E(A) \leq |A|^3,$$

where the lower bound comes from the trivial solutions with $(a, b) = (c, d)$. This lower bound is achieved if and only if $A$ is a Sidon set.

We use the following standard tools in additive combinatorics (see e.g., [58, 61, 69]).

**Lemma 2.8** (Ruzsa sum triangle inequality). For finite integer sets $A, B, C$,

$$|A + B| |C| \leq |A + C| |B + C|.$$

**Lemma 2.9**. For a finite set $A \subseteq \mathbb{Z}$ and non-zero coefficients $\beta_1, \beta_2, \beta_3, \beta_4 \in \mathbb{Z} \setminus \{0\}$,

$$\left|\{(a_1, a_2, a_3, a_4) \in A^4 : \beta_1 a_1 + \beta_2 a_2 + \beta_3 a_3 + \beta_4 a_4 = 0\}\right| \leq E(A).$$

The proof of Lemma 2.9 is deferred to the full version.

### 3 REDUCTION TO MODERATE-ENERGY 3SUM

In general, a 3SUM instance $A$ of size $|A| = n$ can have additive energy up to $n^2$ asymptotically. In this section, we provide a reduction from an arbitrary 3SUM instance $A$ to another 3SUM instance $A'$ with moderate additive energy, $E(\hat{A}) < O(|A|^{1-\varepsilon})$, for some positive constant $\varepsilon > 0$. The reduction is formally summarized in the following theorem.

**Theorem 3.1** (Reduction to moderate-energy 3SUM). There exist universal constants $\delta > 0$ and $\varepsilon > 0$ such that the following holds. Given an integer set $A \subseteq \mathbb{Z} \cap [-U, U]$ of size $|A| = n$, there is a randomized algorithm in $O(n^{2-\delta} \cdot \text{polylog } U)$ time that, with probability 1, either

(a) finds a 3SUM solution $a, b, c \in A, a + b + c = 0$, or

(b) returns a subset $\hat{A} \subseteq A$, such that $A$ has a 3SUM solution if and only if $\hat{A}$ has one.

Moreover, the probability that Case (b) occurs and $E(\hat{A}) > |\hat{A}|^{3-\varepsilon}$ is at most $1/3$.

The proof of Theorem 3.1 has two ingredients: an algorithmic version of the celebrated Balog-Szemerédi-Gowers Theorem [14, 45], and a subquadratic-time algorithm for 3-partite 3SUM when one of the input sets has small doubling.

The BSG theorem states that any set $A \subseteq \mathbb{Z}$ with high additive energy $E(A) \geq |A|^3/K$ must have a subset $A' \subseteq A$ that has large size $|A'| \geq K^{-O(1)}|A|$, and small doubling, $|A' + A'| \leq K^{O(1)}|A'|$. The following lemma gives a subquadratic-time randomized algorithm for finding such subset $A'$.

**Lemma 3.2** (BSG lemma). There exist universal constants $\alpha > 0, \beta > 0$ such that the following holds. Given an integer set $A \subseteq \mathbb{Z} \cap [-U, U]$ of size $|A| = n$, and a parameter $1 \leq K \leq n$, there is a randomized algorithm in $O(n^{\alpha} \cdot \text{polylog } U)$ time that, with probability 1, either

(i) returns a subset $A' \subseteq A$ such that $|A'| \geq n/K^2$ and $|A' + A'| \leq n^{\alpha}$, or

(ii) outputs “failure”.

Moreover, if $E(A) > n^3/K$, then the failure probability is at most $1/n^2$.

The second ingredient we need is a specialized algorithm for the 3-partite 3SUM problem, where we are given three integer sets $A, B, C \subseteq \mathbb{Z} \cap [-U, U]$, and need to find $a \in A, b \in B, c \in C$ such that $a + b + c = 0$. This algorithm has sub-quadratic running time provided $|A + A|$ is small.

**Lemma 3.3** (3SUM with small doubling). Given input sets $A, B, C \subseteq \mathbb{Z} \cap [-U, U]$ with max$\{|A|, |B|, |C|\} \leq n$, the 3-partite 3SUM problem can be solved by a Las Vegas randomized algorithm with time complexity

$$O\left(\frac{n^3}{|A|} \cdot \text{polylog } U\right).$$

Now we prove Theorem 3.1 using Lemma 3.2 and Lemma 3.3.

**Proof of Theorem 3.1.** Let $A \subseteq \mathbb{Z} \cap [-U, U]$ be the input 3SUM instance of size $|A| = n$. Let $\varepsilon > 0$ be a small constant to be determined.

The reduction is described in Algorithm 1. It maintains a subset $\hat{A} \subseteq A$ initialized to $A$, and repeatedly uses the BSG lemma (Lemma 3.2) to peel off a large subset $A' \subseteq \hat{A}$ with small doubling, and then uses Lemma 3.3 to find 3SUM solutions involving $A'$, i.e., $(a, b, c) \in A' \times A \times \hat{A}$ with $a + b + c = 0$. The algorithm is terminated whenever a 3SUM solution is found (Case (a) of Theorem 3.1). We return this subset $\hat{A}$ (Case (b) of Theorem 3.1) once the BSG lemma reports failure (Line 7). If $|\hat{A}|$ becomes smaller than $n^{1-\varepsilon}$ (Line 8), we can afford to solve 3SUM on $|\hat{A}|$ by brute-force in $O(|\hat{A}|^2) \leq O(n^{2-2\varepsilon})$ time, and return the found 3SUM solution, or return an empty set $\hat{A} := \emptyset$ if no solution is found.

Algorithm 1 maintains the invariant that $A$ has a 3SUM solution if and only if $\hat{A}$ has one. Indeed, if 3-partite 3SUM on $A', \hat{A}$ has no solution, then any 3SUM solution of $\hat{A}$ must be contained in

\[\text{[412]}\]
Algorithm 1: Reduction to moderate-energy 3SUM

1. Initialize $\hat{A} ← A$
2. while $|\hat{A}| ≥ n^{1-\varepsilon}$ do
   3. Apply BSG lemma (Lemma 3.2) to $\hat{A}$ with $K := |\hat{A}|^r$
   4. if BSG lemma successfully returned a subset $A' \subseteq \hat{A}$ then
      5. Solve 3-partite 3SUM on $A', \hat{A}, A$ using Lemma 3.3, and terminate if a solution is found
      6. $\hat{A} ← \hat{A} \setminus A'$
   7. else return $\hat{A}$
8. Solve 3SUM on $\hat{A}$ by brute-force in $O(|\hat{A}|^2)$ time

$\hat{A} \setminus A'$, so we can remove $A'$. This shows that with probability 1 either Case (a) or Case (b) in the theorem statement holds.

Now we analyze the time complexity of Algorithm 1. Starting from $\hat{A} = A$, each iteration of the while loop removes from $\hat{A}$ a subset $A'$ of size $|A'| ≥ |\hat{A}|/K^\beta = |\hat{A}|^{1-\varepsilon} ≥ n^{1-\varepsilon},$ so the total number of iterations is at most $|A|/n^{1-\varepsilon}(1-\varepsilon\beta) = n^{\varepsilon(\beta+1)}$, where the coefficients $\beta$ and $\varepsilon$ are nontrivial $4$-term relations.

In each iteration, ignoring poly-logarithmic factors, Lemma 3.2 has time complexity $|\hat{A}| \cdot K^\alpha = |\hat{A}| \cdot |\hat{A}|^{\alpha\varepsilon} ≤ n^{\varepsilon\alpha}$, and Lemma 3.3 has time complexity $|\hat{A}| \cdot |A'|^{\alpha\varepsilon} ≤ |\hat{A}|^{1.5 \alpha\varepsilon+\beta/2} ≤ n^{1.5\varepsilon(\alpha\varepsilon+\beta/2)}$.

Summing over all iterations, the total time complexity of Algorithm 1 is at most $n^{\varepsilon(\beta+1-\varepsilon\beta)} \cdot O(n^{1.5\varepsilon(\alpha\varepsilon+\beta/2)} \log U) ≤ O(n^{1.6\varepsilon(\alpha\varepsilon+\beta/2)}),$ where we set $\varepsilon = 0.1/(\alpha + 1.5\beta + 1)$.

In each iteration, the probability that $E(\hat{A}) > |\hat{A}|/K = |\hat{A}|^{1-\varepsilon}$ yet the BSG lemma outputs failure is at most $1/|\hat{A}|^2 ≤ 1/n^{2-\varepsilon}$. By a union bound over at most $n^{\varepsilon(\beta+1-\varepsilon\beta)}$ iterations, with at most $1/n^{2-\varepsilon}$ probability we eventually return $\hat{A}$ with large additive energy $E(\hat{A}) > |\hat{A}|^{1-\varepsilon}$.

A minor issue is that the time bound in (4) is in expectation rather than worst-case, since the 3-partite 3SUM algorithm in Lemma 3.3 is Las Vegas randomized. To fix this, we terminate Algorithm 1 (and default to return $\hat{A} ← A$) after executing longer than 10 times the expected time bound, which additionally incurs $1/10$ failure probability by Markov’s inequality.

4 REDUCTION TO 3SUM ON SIDON SETS

In this section, we further reduce a moderate-energy 3SUM instance to a 3SUM instance on Sidon sets. In fact, the produced instance avoids not only Sidon 4-tuples, but all small-coefficient 4-term linear relations as well. To state our formal result, we make the following technical definition, which is also crucially in our proof.

Definition 4.1 ($k$-term $\ell$-relation). We say $k$ integers $a_1, \ldots, a_k$ have an $\ell$-relation, if there exist integer coefficients $\beta_1, \ldots, \beta_k \in [-\ell, \ell]$ that have sum $\sum_{i=1}^k \beta_i = 0$ and are not all zero, such that $\sum_{i=1}^k \beta_i a_i = 0$. Moreover, we say $\sum_{i=1}^k \beta_i a_i = 0$ is a nontrivial $\ell$-relation, if $(a_1, \ldots, a_k)$ is a nontrivial solution to the equation $\sum_{i=1}^k \beta_i a_i = 0$ (see Definition 2.4).

We only consider 3-term and 4-term relations. For example, a nontrivial 3-term arithmetic progression $(a, b, c)$ form a nontrivial 3-term 2-relations $a - 2b + c = 0$ (where the coefficients $1, (-2), 1$ have zero sum and maximum magnitude 2), and a Sidon 4-tuple $(a, b, c, d)$ (where $(a, b) \neq (c, d)$) form a nontrivial 4-term 1-relation $a + b - c - d = 0$. Here are more examples: integers 101, 103, 109 have a nontrivial 3-term 4-relation $3 \cdot 101 - 4 \cdot 103 + 1 \cdot 109 = 0$, but do not have any 3-term 3-relations. Integers 999, 101, 103, 109 have a nontrivial 4-term 4-relation $0 \cdot 999 + 3 \cdot 101 - 4 \cdot 103 + 1 \cdot 109 = 0$. Integers 101, 103, 109 have a nontrivial 4-term 3-relation $3 \cdot 101 - 2 \cdot 103 - 2 \cdot 103 + 2 \cdot 109 = 0$, and also a trivial 4-term 1-relation $0 \cdot 101 + 1 \cdot 103 - 1 \cdot 103 + 0 \cdot 109 = 0$.

We prove the following theorem.

Theorem 4.2 (Generalized version of Theorem 1.1). For any constants $\ell ≥ 1$ and $\varepsilon > 0$, solving 3SUM on size-$n$ sets of integers bounded by $[-n^{3+\delta}, n^{3+\delta}]$ avoiding nontrivial 4-term $\ell$-relations requires $n^{\varepsilon+\alpha(1)}$ time, assuming the 3SUM hypothesis.

In particular, solving 3SUM on Sidon sets is 3SUM-hard, proving Theorem 1.1. Note that if a set avoids nontrivial 4-term $\ell$-relations, it does not contain four distinct numbers that have an $\ell$-relation either. Thus, avoiding nontrivial 4-term $\ell$-relations is a stronger condition and Theorem 4.2 also holds if we replace "nontrivial 4-term $\ell$-relations" with "4-term $\ell$-relations involving 4 distinct numbers".

In comparison, the main technique of [39] can establish a special case of Theorem 4.2, the 3SUM-hardness of 3SUM on sets avoiding nontrivial 3-term $\ell$-relations. Although a weaker asker of our result is that our reduction is Las Vegas randomized, while their reduction is deterministic.

Using Theorem 4.2, it is not difficult to show the 3SUM-hardness of detecting solutions to any nontrivial 4-LDT (Theorem 1.3). The proof uses standard techniques in fine-grained complexity similar to those applied in 3SUM hardness proofs (e.g., [39]), but it involves some technical details, and is therefore deferred to the full version.

In the following, we prove Theorem 4.2, by applying a careful self-reduction on the moderate-energy 3SUM instance obtained from Theorem 3.1.

It is well-known that 3SUM has an efficient self-reduction [16] through almost-linear hash functions, such as modulo a random prime, or Dietzfelbinger’s hash function (see e.g., [25, 36, 37]). We will use the same self-reduction with a few modifications.

In the following, for integer parameters $m ≤ U$, we always consider hash families $\mathcal{H}$ consisting of hash functions of the form $H : \mathcal{Z} \cap [-U, U] → [m]$.

We always assume a hash function $H ∈ \mathcal{H}$ can be described by a seed of length $U^{\alpha(1)}$, and evaluating $H(x)$ can be done in $U^{\alpha(1)}$ time given $x$ and the description of $H$. First we define the almost-linearity property of a hash family.
Definition 4.3 (Almost linearity). For an integer set $\Delta$, we say a hash family $\mathcal{H}$ is $\Delta$-almost-linear, if for all hash functions $H \in \mathcal{H}$ and all $x, y \in \mathbb{Z} \cap [-U, U]$, 
\[ H(x) + H(y) + H(-x - y) \in \Delta. \]
The set $\Delta$ should be computable in $\text{poly}(|\Delta| \log U)$ time. Sometimes we also say $\mathcal{H}$ is $|\Delta|$-almost-linear.

The standard 3SUM self-reduction proceeds as follows: sample $H \in \mathcal{H}$, and place input integer $x$ in the bucket numbered $H(x)$. By almost-linearity, it suffices to solve $3$-partition 3SUM on the three buckets numbered $i, j, -i - j - d$ respectively, over all $i \in [m], j \in [m], d \in \Delta$. There are $m^2|\Delta|$ small instances, and we need to set $|\Delta| = \Omega(1)$ for time-efficiency.\(^7\)

Similar to previous works, in order to bound the size of the instances generated by the self-reduction, we require the hash family $\mathcal{H}$ to be almost 2-universal: for $x \neq y$, $\Pr_{H \in \mathcal{H}}[H(x) = H(y)] \approx 1/m$. However, in our scenario of removing distinct numbers $a, b, c$ with $a + b = c + d$, we need stronger independence guarantees in order to bound the probability that $a, b, c, d$ all receive the same hash value. Unfortunately, given the almost-linearity requirement, it seems difficult to achieve 3-wise independence: for three integers $x, y, z$ with $z = x + y$, the hash value of $z$ is almost determined (up to $|\Delta|$ possibilities) by the hash values of $x$ and $y$. Nevertheless, we can achieve the desired independence guarantee for three integers that avoid 3-relations (for some small $\ell$). We formally state the properties of our hash family $\mathcal{H}$ in the following lemma, whose proof can be found in the full version.

Lemma 4.4 (Hash family). Let $\ell = \lceil \exp((\log U)^{1/3}) \rceil$. Given an integer $m \in [1, U]$, there is a hash family $\mathcal{H} \subseteq (\{H : \mathbb{Z} \cap [-U, U] \to [m]\})$ such that:
- $\mathcal{H}$ is $\Omega(1)$-almost-linear.
- For every $x, y \in \mathbb{Z} \cap [-U, U]$, $x \neq y$, we have
  \[ \Pr_{H \in \mathcal{H}}[H(x) = H(y)] \leq \frac{\Omega(1)}{m}. \]
- For every $x, y, z \in \mathbb{Z} \cap [-U, U]$ that do not have any 3-term $t$-relations, we have
  \[ \Pr_{H \in \mathcal{H}}[H(x) = H(y) = H(z)] \leq \frac{\Omega(1)}{m^2}. \]

To deal with integers that do have 3-term $t$-relations, we borrow the proof idea from [39] that uses Behrend’s set [18] to forbid these integers occurring simultaneously. The following adaptation of Behrend’s construction will be proved in the full version.

Lemma 4.5 (Behrend’s construction). Let $\ell = \lceil \exp((\log U)^{1/3}) \rceil$. Given set $A = \{a_1, \ldots, a_n\} \subseteq \mathbb{Z} \cap [-U, U]$, there exists a deterministic $O(n \log U)$-time algorithm that partitions $A$ into $b = \exp(O((\log U)^{2/3}))$ disjoint subsets $B_1, B_2, \ldots, B_b$, such that $B_1$ avoids nontrivial 3-term $t$-relations, for all $1 \leq i \leq b$.

In our reduction, we start with a 3SUM instance $A \subseteq \mathbb{Z} \cap [-U, U]$ (where $U = n^3$) of size $|A| \leq n$ and moderate additive energy $E(A) \leq |A|^{1/2}$ for some constant $\varepsilon > 0$. Such an instance $A$ is generated (with $2/3$ success probability) by Theorem 3.1 from an arbitrary $n$-size 3SUM instance with input range $[-n^3, n^3]$ (see Hypothesis 2.2).

In the following we assume $U = n^3$. Let $\ell = \lceil \exp((\log U)^{1/3}) \rceil$ be the same parameter from Lemma 4.5 and Lemma 4.4.

The first step is to perform a self-reduction on $A$, which generates many small 3SUM instances that have few nontrivial 4-term $t$-relations.

Definition 4.6 (Self-reduction). Given $A \subseteq \mathbb{Z} \cap [-U, U]$ (where $U = n^3$) of size $|A| \leq n$, and a small constant parameter $\epsilon \in (0, \epsilon)$, generate smaller 3SUM instances as follows. Let $m = \lceil n^{1/\epsilon} \rceil$.

1. Sample $H : \mathbb{Z} \cap [-U, U] \to [m]$ from the hash family $\mathcal{H}$ in Lemma 4.4, and use $H$ to partition $A$ into groups $G_i := \{a \in A : H(a) = i\}$ for $i \in [m]$.

2. For each group $G_i$ of too large size $|G_i| > n^{1+\epsilon}/m$, remove all its elements (i.e., redefine $G_i = \emptyset$). Use brute-force to check for 3SUM solutions in $A$ involving these removed elements.

3. Use Lemma 4.5 to partition $A$ into $b = \exp(O((\log U)^{2/3}))$ Behrend sets $B_1, B_2, \ldots, B_b$. For each $(x', y', z') \in [b]^3$, $(x, y, z) \in [m]^3$ with $x+y+z \in \Delta$, generate the following 3SUM instance:
   \[ (q_x U + (G_x \cap B_x')) \cup (q_y U + (G_y \cap B_y')) \cup (q_z U + (G_z \cap B_z')), \]
   where $q_x := 10\ell^2, q_y := 100\ell^2, q_z := -q_x - q_y$.

We need the following property on the shifting coefficients $q_x, q_y$ and $q_z$ defined in Definition 4.6.

Claim 4.7. For all integers $\beta_x, \beta_y, \beta_z \in \mathbb{Z} \cap [-\ell, \ell]$, we have $|\beta_x q_x + \beta_y q_y + \beta_z q_z| > 5\ell$ unless $\beta_x = \beta_y = \beta_z$.

Proof. Suppose $|\beta_x q_x + \beta_y q_y + \beta_z q_z| \leq 5\ell$, or equivalently,
   \[ |(\beta_x - \beta_z)q_x + (\beta_y - \beta_z)q_y| \leq 5\ell. \]
   If $\beta_x - \beta_z = 0$ and $\beta_y - \beta_z = 0$, then $\beta_x = \beta_y = \beta_z$. If $\beta_y - \beta_z = 0$ and $\beta_x - \beta_z \neq 0$, then $LHS = |\beta_x - \beta_z| \cdot q_x \geq 10\ell > RHS$, a contradiction. Finally, if $\beta_x - \beta_z \neq 0$ and $\beta_y - \beta_z \neq 0$, then $LHS \geq |\beta_y - \beta_z| \cdot q_y - |\beta_x - \beta_z| \cdot q_x > 2q_x \cdot |\beta_x - \beta_z| > 80\ell^2 > RHS$, a contradiction. $\square$

It is easy to see that the self-reduction from Definition 4.6 preserves the 3SUM solutions of $A$. Indeed, a 3SUM solution that involves any integers from $A \setminus (G_1 \cup \cdots \cup G_m)$ must be found in step (1). Among the remaining integers, a 3SUM solution $a_1 + a_2 + a_3 = 0$ with $a_1 \in G_x, a_2 \in G_y, a_3 \in G_z$ must satisfy $x + y + z \in \Delta$, so the shifted version of this solution, $(q_x U + a_1) + (q_y U + a_2) + (q_z U + a_3)$, must be included in one of the 3SUM instances (Eq. 7) generated in step (2).

Observation 4.8. Step (1) in Definition 4.6 runs in $n^{2-\gamma+o(1)}$ time in expectation.

Proof. For each integer $a \in A$, by 2-universality of $H$ (Eq. 5), the expected size of $G_H(a)$ is at most $1 + (n - 1) \cdot \frac{\Omega(1)}{m}$. Note
that we remove a only if $|G_{H(a)}| > \frac{n!^4}{m}$, which happens with probability at most $\frac{1}{n!^2-o(1)}$ by Markov’s inequality, so the total number of removed elements is at most $n^{2-o(1)}$ in expectation.

Sort A at the beginning. For each removed integer $a \in A$, it takes an $O(n)$-time scan to check for 3SUM solutions involving $a$. Hence, the expected total time to check removed elements is $n^{2-o(1)}$. □

The instances generated by the self-reduction (Definition 4.6) may still contain a few nontrivial 4-term $t$-relations. The next step is to remove the elements involved in such relations, so that the remaining elements in each instance are completely free of nontrivial 4-term $t$-relations. To do this, we first need to analyze the expected total number of nontrivial 4-term $t$-relations across all the generated instances. To better understand the following technical parts, readers are encouraged to think of the representative case $\beta_1 = 3 = 1, \beta_2 = \beta_4 = -1$, i.e., Sidon 4-tuples.

Lemma 4.9 (Types of nontrivial 4-term $t$-relations). Denote the 3SUM instance defined in Eq. (7) by $P_X \cup P_Y \cup P_Z$ for short. Then, every nontrivial 4-term $t$-relation $\sum_{i=1}^{4} \beta_i a_i' = 0$ on integers $a_i' \in P_X \cup P_Y \cup P_Z$ must have one of the following two types (up to permuting indices $\{1, 2, 3, 4\}$ and/or $\{X, Y, Z\}$):

1. $a'_1, a'_2, a'_3, a'_4 \in P_X$, and all $\beta_i$ are non-zero.

We say this relation is induced by the nontrivial $t$-relation $\sum_{i=1}^{4} \beta_i a_i = 0$ in $A$, where $a_i = a'_i - qxU \in G_x \cap B_x$ ($i \in \{4\}$).

2. $a'_1, a'_2, a'_3 \in P_Y$, $a'_4 \in P_Y$, $\beta_1 + \beta_2 = \beta_3 = \beta_4 = 0$, and all $\beta_i$ are non-zero.

We say this relation is induced by the $t$-relation $\sum_{i=1}^{4} \beta_i a_i = 0$ in $A$, where $a_1 = a'_1 - qxU, a_2 = a'_2 - qxU \in G_x \cap B_x$, and $a_3 = a'_3 - qxU, a_4 = a'_4 - qxU \in G_y \cap B_y$.

The proof of Lemma 4.9 is deferred to the full version. Lemma 4.9 shows that the nontrivial 4-term $t$-relations $\sum_{i=1}^{4} \beta_i a_i' = 0$ in the generated instances (Eq. (7)) are always induced by 4-term $t$-relations $\sum_{i=1}^{4} \beta_i a_i = 0$ (where $\beta_i$ are non-zero) from the original input set $A$. For each of them, we can bound the expected number of nontrivial relations it induces in the generated instances, and by linearity of expectation this allows us to bound the total number of such relations in the generated instances. This is the key property of our reduction.

Lemma 4.10. The expected total number of nontrivial 4-term $t$-relations in all instances (Eq. (7)) generated by the self-reduction (Definition 4.6) is at most $E(A) \cdot n^{o(1)}/m$.

Proof. For each 4-term $t$-relation $\sum_{i=1}^{4} \beta_i a_i = 0$ where $a_i \in A, \beta_i \neq 0$ for all $i \in \{4\}$, we separately analyze the expected number of nontrivial 4-term $t$-relations $\sum_{i=1}^{4} \beta_i a_i' = 0$ it induces for each of the two types defined in Lemma 4.9.

• Type 1: $a'_1, a'_2, a'_3, a'_4 \in P_X$, and $a_1 = a'_1 - qxU$. Note that $\sum_{i=1}^{4} \beta_i a_i = 0$ is also a nontrivial relation.

This can happen only if $a_1, a_2, a_3, a_4 \in G_x \cap B_x$, for some $x \in [m]$ and $x' \in [b]$. We show that $a_1, a_2, a_3, a_4$ contain at least 3 distinct integers. Otherwise, $\{a_1, a_2, a_3, a_4\} \subseteq \{u, v\}$ for some $u \neq v$, and combining $\sum_{i=1}^{4} \beta_i = 0$ and $u \sum_{i=1}^{4} \beta_i a_i + v \sum_{i=1}^{4} \beta_i a_i = 0$ would imply $\sum_{i=1}^{4} \beta_i a_i = 0$, contradicting the fact that $\sum_{i=1}^{4} \beta_i a_i = 0$ is a nontrivial relation.

Without loss of generality assume $a_1, a_2, a_3$ are distinct. Since $a_1, a_2, a_3 \in B_x$, they do not have any 3-term $t$-relation. Then, by the 3-universal property (Eq. (6)) of the hash family, we have

$$
\Pr_{H \in H} \left[ a_1, a_2, a_3, a_4 \in G_x \text{ for some } x \in [m] \right] 
\leq \Pr_{H \in H} \left[ H(a_1) = H(a_2) = H(a_3) \right] 
\leq \frac{n^{o(1)}}{m^2}.
$$

If $a_1, a_2, a_3, a_4 \in G_x \cap B_x$ happens for some $x \in [m]$ and $x' \in [b]$, then it may induce a nontrivial 4-term $t$-relation in every instance that involve $G_x \cap B_x$. Such instances (indexed by $(x, y, z, x', y', z') \in [m]^3 \times [b]^3$ in Definition 4.6) should satisfy $y + z = \Delta - x$, so there are only $|\Delta| \cdot b^2$ such instances. So the expected total number of nontrivial 4-term $t$-relations induced by $\sum_{i=1}^{4} \beta_i a_i = 0$ is at most $\frac{n^{o(1)}}{m} \cdot |\Delta| \cdot b^2$ in expectation.

• Type 2: $a'_1, a'_2, a'_3 \in P_Y$, $a'_4 \in P_Y$, $\beta_1 + \beta_2 = \beta_3 = \beta_4 = 0$, and all $\beta_i$ are non-zero.

This can happen only if $a_1, a_2, a_3 \in G_x \cap B_x$, $a_4 \in G_y \cap B_y$ for some $x, y \in [m]$ and $x', y' \in [b]$. Observe that $a_1 \neq a_2$, since otherwise we must have $a_3 = a_4$ as well, which would imply $a'_1 = a'_2$ and $a'_3 = a'_4$, contradicting the assumption that $\sum_{i=1}^{4} \beta_i a_i = 0$ is a nontrivial relation.

Then, by the 2-universal property (Eq. (5)) of the hash family (Lemma 4.4), we have

$$
\Pr_{H \in H} \left[ a_1, a_2 \in G_x \text{ for some } x \in [m] \right] 
\leq \Pr_{H \in H} \left[ H(a_1) = H(a_2) \right] 
\leq \frac{n^{o(1)}}{m}.
$$

If $a_1, a_2 \in G_x \cap B_x, a_3, a_4 \in G_y \cap B_y$ happen for some $x, y \in [m]$ and $x', y' \in [b]$, then it may induce a nontrivial 4-term $t$-relation in every instance that involve both $G_x \cap B_x$ and $G_y \cap B_y$. There are only $|\Delta| \cdot b$ such instances, so the expected total number of nontrivial 4-term $t$-relations induced by $\sum_{i=1}^{4} \beta_i a_i = 0$ is at most $n^{o(1)}/(|\Delta| \cdot b)$ in expectation.

There are at most $O(t^3)$ equations $\sum_{i=1}^{4} \beta_i a_i = 0$ with integer coefficients $\beta_i \in [-t, t] \setminus \{0\}$, and $\sum_{i=1}^{4} \beta_i = 0$, and each of them has at most $E(A)$ solutions in $A$ by Lemma 2.9, so there are at most $O(t^3 E(A))$ such 4-term $t$-relations in $A$. Summing over all of them (and accounting for all possible ways of permuting $\{X, Y, Z\}$ and/or $\{1, 2, 3, 4\}$ in the types), by linearity of expectation, the expected total number of induced nontrivial 4-term $t$-relations over all generated instances is $O(t^3 E(A)) \cdot n^{o(1)}/m \leq E(A) \cdot n^{o(1)}/m$. □

Now, we describe how to efficiently report all the nontrivial 4-term $t$-relations in the generated instances.

Lemma 4.11. We can report all the nontrivial 4-term $t$-relations in all instances (Eq. (7)) generated by Definition 4.6 in time linear in their number, plus $n^{2+2\gamma+o(1)}$ additional time.

Proof. We first do the following pre-processing step. For every $x \in [m], x' \in [b]$, and every $\beta_1, \beta_2 \in \mathbb{Z} \cap [-t, t] \setminus \{0\}$, compute the set of tuples

$$
D_{x, x'}^{\beta_1, \beta_2} = \{(\beta_1 a_1 + \beta_2 a_2, a_1, a_2) : a_1, a_2 \in G_x \cap B_{x'}, a_1 \neq a_2\}. \tag{8}
$$
Then for every \( t \in \mathbb{Z} \) and \( \beta \in \mathbb{Z} \cap [\ell, \ell] \setminus \{0\} \), compute a bucket \( L_t^{\beta} \) that contains all tuples \( (t, a_1, a_2) \) appearing in any set \( D_{x,x'}^{\beta} \). Technically, every tuple in the bucket \( L_t \) also records which set \( D_{x,x'}^{\beta} \) it comes from. This pre-processing step can be implemented in time

\[
\tilde{O}(t^{2} \cdot \sum_{x \in [m], x' \in [b]} |G_x \cap B_{x'}|^2) \leq \tilde{O}(t^{2}mb \cdot (n^{1+y}/m)^2) \leq n^{2+2y+o(1)}/m.
\]

To report all the nontrivial 4-term \( \ell \)-relations \( \sum_{i=1}^{4} \beta_i a_i = 0 \) in the generated instances, again we separately consider type 1 and type 2 as defined in Lemma 4.9.

- Type 1: induced by nontrivial \( \ell \)-relation \( \sum_{i=1}^{4} \beta_i a_i = 0 \), where \( a_i \in G_x \cap B_{x'} \) (\( t \in \{4\} \)). For every \( x \in [m], x' \in [b] \), to find all nontrivial 4-term \( \ell \)-relations in \( G_x \cap B_{x'} \) simply enumerate \( \beta_1, \beta_2, \beta_3, \beta_4 \) and compare \( D_{x,x'}^{\beta_1} \), \( D_{x,x'}^{\beta_2} \), \( D_{x,x'}^{\beta_3} \), \( D_{x,x'}^{\beta_4} \) to find common sums \( \beta_1 a_1 + \beta_2 a_2 = -\beta_3 a_3 - \beta_4 a_4 \). Then we immediately find the induced relations \( \sum_{i=1}^{4} \beta_i a_i = 0 \) in all instances that involve \( G_x \cap B_{x'} \).

- Type 2: induced by \( \ell \)-relation \( \sum_{i=1}^{4} \beta_i a_i = 0 \), where \( \beta_1 + \beta_2 = \beta_3 + \beta_4 = 0 \) and \( a_1, a_2 \in G_x \cap B_{x'}, a_3, a_4 \in G_y \cap B_{y'} \). For every \( t \in \mathbb{Z} \) and \( \beta_1, \beta_2 \in \mathbb{Z} \cap [-t, t] \setminus \{0\} \), enumerate every pair of \( (t, a_1, a_2) \in \mathbb{Z} \cap [-t, t] \setminus \{0\} \), and compare \( D_{x,x'}^{\beta_1} \), \( D_{x,x'}^{\beta_2} \) to find common sums \( \beta_1 a_1 + \beta_2 a_2 = -\beta_3 a_3 - \beta_4 a_4 \). Then it induces a nontrivial \( \ell \)-relation \( \sum_{i=1}^{4} \beta_i a_i = 0 \) in every instance involving both \( G_x \cap B_{x'} \) and \( G_y \cap B_{y'} \), where \( a_1', a_2' \in \mathbb{Q} U + (G_x \cap B_{x'}), a_3', a_4' \in \mathbb{Q} U + (G_y \cap B_{y'}) \).

In both cases, after the pre-processing is finished, reporting the \( \ell \)-relations does not incur any extra overhead in time complexity.

After finding all the nontrivial 4-term \( \ell \)-relations in the generated instances, the final step is to remove these involved integers, so that the remaining integers in each instance completely avoid all nontrivial 4-term \( \ell \)-relations. The reduction is summarized as follows.

**Theorem 4.12.** Suppose for some constant \( \delta > 0 \) there exists an \( O(n^{1-\delta}) \)-time algorithm \( \mathcal{A} \) that solves 3SUM on input set \( A_0 \subset \mathbb{Z} \cap [-n^{\delta}, n^{\delta}] \) (where \( C = 12000 \)) of size \( |A_0| \leq n_0 \) that avoids nontrivial 4-term \( \ell \)-relations.

Then, there is an algorithm that solves 3SUM on size-\( n \) input set \( A \subset \mathbb{Z} \cap [-n^\epsilon, n^\epsilon] \) in \( O(n^{2-\delta}) \) time for some constant \( \delta' > 0 \) depending on \( \delta \). Moreover, this reduction is Las Vegas randomized.

Proof. Given input set \( A \subset \mathbb{Z} \cap [-n^\epsilon, n^\epsilon] \) of size \( |A| = n \), first run the sub-quadratic time reduction in Theorem 3.1 to obtain an equivalent input set \( \tilde{A} \subset A \), which has moderate additive energy \( E(\tilde{A}) \leq |\tilde{A}|^{1/2} \) with at least \( 2/3 \) probability, for some constant \( c > 0 \).

Set \( \gamma = \delta \epsilon / 200 \). Then, apply the self-reduction from Definition 4.6 on \( \tilde{A} \), and obtain \( \delta' \cdot m^2 \cdot |\tilde{A}| \leq m^2 \cdot n^{1+o(1)} \) small 3SUM instances.
We would like to thank Virginia Vassilevska Williams for many helpful discussions and suggestions. We also thank Ryan Williams for helpful discussions.

REFERENCES

[1] Amir Abboud, Arturs Backurs, Karl Bringmann, and Marvin Künnemann. 2017. Fine-Grained Complexity of Analyzing Compressed Data: Quantifying Improvements over Decompress-and-Solve. In Proceedings of the 58th Annual Symposium on Foundations of Computer Science (FOCS). 192–203. https://doi.org/10.1109/FOCS.2017.26

[2] Amir Abboud, Arturs Backurs, Karl Bringmann, and Marvin Künnemann. 2020. Impossibility results for grammar-compressed linear algebra. In Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS). 8810–8820.

The construction is as follows: uniformly independently sample m integers from \([N]\), which produce \(s = \Theta m^2 \cdot N\) Sidon 4-tuples in expectation. Remove one integer from each Sidon 4-tuple, and the remaining integers form a Sidon set of expected size \(m - s \geq m/2\), by setting \(m = c \cdot N^{1/3}\) for a small enough constant \(c\).

[3] Amir Abboud, Karl Bringmann, and Nick Fischer. 2022. Stronger 3-SUM Lower Bounds for Approximate Distance Oracles via Additively Combinatorics. CoRR abs/2211.07098 (2022). arXiv:2211.07098v1. To appear in STOC 2023.

[4] Amir Abboud, Karl Bringmann, Seri Khoury, and O. Ryan. 2022. Hardness of approximation in \(p\) via short cycle removal: cycle detection, distance oracles, and beyond. In Proceedings of the 56th Annual ACM SIGACT Symposium on Theory of Computing (STOC). 1487–1500. https://doi.org/10.1145/3519935.3520066

[5] Amir Abboud, Seri Khoury, Orite Leibowitz, and Ron Safir. 2022. Listing 4-Cycles. CoRR abs/2211.10022 (2022). arXiv:2211.10022v1

[6] Amir Abboud, Aviad Rubinstein, and R. Ryan Williams. 2017. Distributed PCT 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. IEEE Computer Society, 25–36. https://doi.org/10.1109/FOCS.2017.12

[7] Udit Agarwal and Vijaya Ramachandran. 2018. Fine-grained complexity for sparse graphs: In Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC). 259–262. https://doi.org/10.1145/3188745.3188888

[8] Josh Alman and Virginia Vassilevska Williams. 2021. A refined laser method and faster matrix multiplication. In Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA). 522–539. https://doi.org/10.1137/1.9781611976465.32

[9] Josh Alman and Ryan Williams. 2015. Probabilistic Polynomials and Hamming Nearest Neighbors. In 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015. IEEE Computer Society, 136–150. https://doi.org/10.1109/FOCS.2015.18

[10] Noga Alon, Raphael Yuster, and Uri Zwick. 1995. Color-coding. J. ACM 42, 4 (1995). 844–856. https://doi.org/10.1145/210332.210337

[11] Noga Alon, Raphael Yuster, and Uri Zwick. 1997. Finding and Counting Given Length Cycles. Algorithmica 17, 3 (1997). 209–223. https://doi.org/10.1007/BF02521859

[12] Amirhosed Amir, Timothy M. Chan, Moshe Lewenstein, and Noa Lewenstein. 2014. On Hardness of Jumbled Indexing. In Proceedings of the 41st International Colloquium on Automata, Languages and Programming (ICALP) 114–125. https://doi.org/10.1007/978-3-662-43948-7_10

[13] Andrew Arnold and Daniel S. Roche. 2015. Output-Sensitive Algorithms for Subset and Sparse Polynomial Multiplication. In Proceedings of the 2015 ACM on International Symposium on Symbolic and Algebraic Computation (ISSAC). 29–36. https://doi.org/10.1145/2755996.2756653

[14] Antal Balog and Endre Szemerédi. 1994. A statistical theorem of set addition. Combinatorica 14, 3 (1994), 263–268. https://doi.org/10.1007/BF01219794

[15] Níkhlí Bansal, Shashwat Garg, Jesper Nederlof, and Níkhlí Vyá. 2018. Faster Space-Efficient Algorithms for Subset Sum, k-Sum, and Related Problems. SIAM J. Comput. 47, 5 (2018), 1755–1777. https://doi.org/10.1137/17M1158203

[16] Ilya Baran, Erik D. Demaine, and Mihai Pătraşcu. 2008. Subquadratic Algorithms for SSSUM. Algorithmica 50, 4 (2008), 584–596. https://doi.org/10.1007/s00453-006-9036-3

[17] Paul Beame, Raphael Clifford, and Widad Machmouchi. 2013. Element Distinctness, Frequency Moments, and Sliding Windows. In Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS). 290–299. https://doi.org/10.1109/FOCS.2013.39

[18] F. A. Behrend. 1946. On sets of integers which contain no three terms in arithmetic progression. Proc. Nat. Acad. Sci. U.S.A. 32 (1946), 331–332. https://doi.org/10.1073/pnas.32.12.331

[19] Karl Bringmann, Nick Fischer, and Vasileios Nakos. 2021. Sparse nonnegative approximation in discrete algorithms. In Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC). 1711–1724. https://doi.org/10.1145/3406255.3451090

[20] Karl Bringmann, Nick Fischer, and Vasileios Nakos. 2022. Deterministic and Las Vegas Algorithms for Sparse Nonconvex Optimization. In Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA). 3069–3090. https://doi.org/10.1137/1.9781611972073.119

[21] Karl Bringmann and Vasileios Nakos. 2021. Fast n-Fold Boolean Convolution via Additively Combinatorics. In Proceedings of the 48th International Colloquium on Automata, Languages, and Programming (ICALP). 411–417. https://doi.org/10.4230/LIPIcs.ICALP.2021.41

[22] Karl Bringmann and Philip Wellnitz. 2021. On Near-Linear-Time Algorithms for Dense Subset Sum. In Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA). 1777–1796. https://doi.org/10.1137/9781611976465.107

[23] Noraf Carmeli and Markus Kröll. 2020. Enumeration complexity of conjunctive queries with functional dependencies. Theor. Comput. Syst. 64, 5 (2020), 828–860. https://doi.org/10.1007/s00224-019-09937-9

[24] Noraf Carmeli and Markus Kröll. 2021. On the enumeration complexity of unions of conjunctive queries. ACM Trans. Database Syst. 46, 2 (2021), 1–41. https://doi.org/10.1145/3450263

[25] Timothy M. Chan and Qingyong He. 2020. Reducing 3SUM to Convolution-3SUM. In Proceedings of the 3rd Symposium on Simplicity in Algorithms (SODA). 1–7. https://doi.org/10.1137/1.9781611976014.1

[26] Timothy M. Chan and Moshe Lewenstein. 2015. Clustered Integer 3SUM via Additive Combinatorics. In Proceedings of the 47th Annual ACM on Symposium on Theory of Computing (STOC). 31–40. https://doi.org/10.1145/2746539.2746568

417
[27] Shiri Chechik. 2014. Approximate distance oracles with constant query time. In Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC). 545–553. https://doi.org/10.1145/2591796.2592503

[28] Shiri Chechik. 2018. Near-optimal approximate decremental all pairs shortest paths. In Proceedings of the 2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS). 170–181. https://doi.org/10.1109/FOCS.2018.00025

[29] Lijie Chen, Ce Jin, R. Ryan Williams, and Hongxun Wu. 2022. Truly Low-Space Element Distinctness and Subset Sum via Pseudorandom Hash Functions. In Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA). 1661–1678. https://doi.org/10.1137/1616179703.67

[30] Richard Cole and Ramesh Hariharan. 2002. Verifying candidate matches in sparse and wildcard matching. In Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC). 592–601. https://doi.org/10.1109/SFCS.2002.1059992

[31] Sören Dahlgaard, Mathias Bæk Tejs Knudsen, and Morten Støckel. 2017. Finding even cycles faster via capped k-walks. In Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC). 112–120. https://doi.org/10.1109/3055399.3055459

[32] Mina Draief, Ce Jin, Virginia Vassilevska Williams, and Nicole Wein. 2022. Approximation Algorithms and Hardness for m-Pairs Shortest Paths and All-Nodes Shortest Cycles. CoRR abs/2204.03076 (2022). https://doi.org/10.48550/arXiv.2204.03076. To appear in FOCS 2022.

[33] Mina Draief, Ray Li, and Virginia Vassilevska Williams. 2021. Hardness of Approximate Diameter. Now for Undirected Graphs. In 62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, October 17-20, 2021. IEEE, 1021–1032. https://doi.org/10.1109/FOCS52979.2021.00102

[34] Mina Draief, Virginia Vassilevska Williams, Nikhil Vyas, and Nicole Wein. 2019. Tight Approximation Algorithms for Bichromatic Graph Diameter and Related Problems. In 46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece (LIPIcs, Vol. 132) Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 47:1–47:15. https://doi.org/10.4230/LIPIcs.ICALP.2019.47

[35] Mina Draief and Nicole Wein. 2021. Tight conditional lower bounds for approximating diameter in directed graphs. In STOC ’21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021. ACM, 1697–1710. https://doi.org/10.1145/3406325.3451130

[36] Martin Dietzfelbinger. 1996. Universal Hashing and k-Wise Independent Random Variables via Integer Arithmetic without Primes. In Proceedings of the 13th Annual Conference on Computational Complexity (Comput Complexity’96). 165–177. https://doi.org/10.1137/1.9781611975482.5

[37] Martin Dietzfelbinger. 1996. Universal Hashing via Integer Arithmetic Without Primes. In Proceedings of the 13th Annual Conference on Computational Complexity (Comput Complexity’96). 165–177. https://doi.org/10.1137/1.9781611975482.5

[38] Martin Dietzfelbinger. 1996. Universal Hashing and k-Wise Independent Random Variables via Integer Arithmetic without Primes. In Proceedings of the 13th Annual Conference on Computational Complexity (Comput Complexity’96). 165–177. https://doi.org/10.1137/1.9781611975482.5

[39] Martin Dietzfelbinger. 2018. Universal Hashing via Integer Arithmetic without Primes. In Proceedings of the 52nd Annual ACM Symposium on Theory of Computing (STOC). 654–663. https://doi.org/10.1145/3188956.3188968. To appear in SODA 2023.

[40] Martin Mucha, Karol Węgrzycki, and Michal Włodarczyk. 2019. A subquadratic approximation scheme for partition. In Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). 76–88. https://doi.org/10.1137/1.9781611975482.5

[41] Vasileios Nakos. 2020. Nearly Optimal Sparse Polynomial Multiplication. IEEE Trans. Inf. Theory 66, 11 (2020), 7231–7236. https://doi.org/10.1109/TIT.2020.2993985

[42] Fernando Florenzano, Cristian Riveros, Martín Ugarte, Stijn Vansummeren, and Chloe Ching-Yun Hsu and Chris Umans. 2017. On Multidimensional and Monotone k-SUM. In Proceedings of the 42nd International Symposium on Mathematical Foundations of Computer Science (MFCS). https://doi.org/10.48550/ARXIV.1704.03076. 257–279.

[43] Sebastian Forster, Gramoz Goranci, and Monika Henzinger. 2021. Dynamic hardness for shortest cycles and paths in sparse graphs. In Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA) 1256–1252. https://doi.org/10.1137/1.9781611975482.5

[44] Andrea Lincoln, Virginia Vassilevska Williams, and Ryan Williams. 2018. Tight hardness for shortest cycles and paths in sparse graphs. In Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA) 1256–1252. https://doi.org/10.1137/1.9781611975482.5

[45] Xin Lyu and Weihao Zhu. 2022. Time-Space Tradeoffs for Element Distinctness and Subset Sum via Pseudorandomness. https://doi.org/10.48550/ARXIV.2202.05734. To appear in SODA 2023.