Completeness Theorems for k-SUM and Geometric Friends: Deciding fragments of Linear Integer Arithmetic

Geri Gokaj[1], Marvin Künnemann[1]

[1]Karlsruhe Institute of Technology

Abstract

In the last three decades, the k-SUM hypothesis has emerged as a satisfying explanation of long-standing time barriers for a variety of algorithmic problems. Yet to this day, the literature knows of only few proven consequences of a refutation of this hypothesis. Taking a descriptive complexity viewpoint, we ask: What is the largest logically defined class of problems *captured* by the k-SUM problem?

To this end, we introduce a class $\mathsf{FOP}_{\mathbb{Z}}$ of problems corresponding to deciding sentences in Presburger arithmetic/linear integer arithmetic over finite subsets of integers. We establish two large fragments for which the k-SUM problem is complete under fine-grained reductions:

1. The k-SUM problem is complete for deciding the sentences with k existential quantifiers.
2. The 3-SUM problem is complete for all 3-quantifier sentences of $\mathsf{FOP}_{\mathbb{Z}}$ expressible using at most 3 linear inequalities.

Specifically, a faster-than-$n^{\lceil k/2 \rceil \pm o(1)}$ algorithm for k-SUM (or faster-than-$n^{2\pm o(1)}$ algorithm for 3-SUM, respectively) directly translate to polynomial speedups of a general algorithm for *all* sentences in the respective fragment.

Observing a barrier for proving completeness of 3-SUM for the entire class $\mathsf{FOP}_{\mathbb{Z}}$, we turn to the question which other – seemingly more general – problems are complete for $\mathsf{FOP}_{\mathbb{Z}}$. In this direction, we establish $\mathsf{FOP}_{\mathbb{Z}}$-completeness of the *problem pair* of Pareto Sum Verification and Hausdorff Distance under n Translations under the L_∞/L_1 norm in $\mathbb{Z}^d$. In particular, our results invite to investigate Pareto Sum Verification as a high-dimensional generalization of 3-SUM.

1 Introduction

Consider a basic question in complexity theory: How can we determine for which problems an essentially quadratic-time algorithm is best possible? If a given problem A admits an algorithm running in $n^{2+o(1)}$ time, and it is known that A cannot be solved in time $O(n^{2-\epsilon})$ for any $\epsilon > 0$, then clearly the $n^{2+o(1)}$ algorithm has *optimal* runtime, up to subpolynomial factors. This question can be asked more generally for any $k \geq 1$ and time $n^{k\pm o(1)}$. To this day, the theoretical computer science community is far from able to resolve this question unconditionally. However, a surge of results over recent years uses conditional lower bounds based on plausible hardness assumptions to shed some light on why some problems seemingly cannot be solved in time $O(n^{k-\epsilon})$ for any $\epsilon > 0$. Most notably, reductions from k-OV, k-SUM and the weighted k-clique problem have been used to establish $n^{k-o(1)}$-time conditional lower bounds, often matching known algorithms; see [60] for a detailed survey.

In this context, the 3-SUM hypothesis is arguably the first – and particularly central – hardness assumption for conditional lower bounds. Initially introduced to explain various quadratic-time barriers observed in computational geometry [44], it has since been used to show quadratic-time hardness for a wealth of problems from various fields [62, 56, 6, 49, 37, 3, 28]. Its generalization, the k-SUM[1] hypothesis, has led to further conditional lower bounds beyond the quadratic-time regime [39, 4, 1, 2, 51]. For a more comprehensive overview, we refer to [60].

The centrality of the 3-SUM hypothesis for understanding quadratic-time barriers begs an interesting question: Does 3-SUM fully capture quadratic-time solvability, in the sense that it is hard for the entire class $\mathsf{DTIME}(n^2)$? Alas, Bloch, Buss, and Goldsmith [14] give evidence that we are unlikely to prove this: If 3-SUM is hard for $\mathsf{DTIME}(n^2)$ under quasilinear reductions, then $\mathsf{P} \neq \mathsf{NP}$. Thus, to understand precisely the role of 3-SUM to understand quadratic-time computation, the more reasonable question to ask is:

What is the largest class $\mathcal{C}$ of problems such that 3-SUM is $\mathcal{C}$-hard?[2]

Finding a large class $\mathcal{C}$ for which 3-SUM is hard may be seen as giving evidence for the 3-SUM hypothesis. Furthermore, such a result may clarify the true expressive power of the 3-SUM hypothesis, much like the NP-completeness of 3-SAT highlights its central role for polynomial intractability.

1.1 Our approach

We approach our central question from a descriptive complexity perspective. This line of research has been initiated by Gao et al. [45], who establish the sparse OV problem as complete for the class of model checking first-order properties. One can interpret this result as showing that the OV problem expresses relational database queries in the sense that a truly subquadratic algorithm for OV would improve the fine-grained data complexity of such queries (see [45] for details). Related works further delineate the fine-grained hardness of model checking first-order properties and related problem classes [59, 18, 16, 11, 17, 41], see Section 1.3 for more discussion.

Towards continuing the line of research on fine-grained completeness theorems, we introduce a class of problems corresponding to deciding formulas in linear integer arithmetic over finite sets of integers. Specifically, consider the vectors

$$x_1 = (x_1[1], \dots, x_1[d_1]), \dots, x_k = (x_k[1], \dots, x_k[d_k])$$

as quantified variables, and let $t_1, \dots, t_l$ be free variables. Moreover, let

$$X := \{x_1[1], \dots, x_1[d_1], \dots x_k[1], \dots, x_k[d_k], t_1, \dots, t_l\},$$

and let ψ be a quantifier-free linear arithmetic formula over variables in X. We consider the model-checking problem for formulas ϕ in the prenex normal form

$$\phi := Q_1 x_1 \dots Q_k x_k : \psi,$$

[1]The k-SUM problem asks, given sets $A_1, \dots, A_k$ of n numbers, whether there exist $a_1 \in A_1, \dots, a_k \in A_k$ such that $\sum_{i=1}^{k} a_i = 0$. The k-SUM hypothesis states that for no $\epsilon > 0$ there exists a $O(n^{\lceil k/2 \rceil - \epsilon})$ time algorithm that solves k-SUM.

[2]Note that there are different reasonable notions of reductions to consider. Rather than the quasilinear reductions used by Bloch et al., we will consider the currently more commonly used notion of fine-grained reductions; see Section 1.2 for details on the notion of completeness that we will use.

where the quantifiers $Q_1, \dots, Q_k \in \{\exists, \forall\}$ are arbitrary. Formally, for such a ϕ, we define the model checking problem $\mathsf{FOP}_{\mathbb{Z}}(\phi)$ as follows[3]

$$\begin{aligned} &\mathsf{FOP}_{\mathbb{Z}}(\phi): \\ &\quad \textbf{Input: } \text{Finite sets } A_1 \subseteq \mathbb{Z}^{d_1}, \dots, A_k \subseteq \mathbb{Z}^{d_k} \text{ and } \hat{t_1}, \dots, \hat{t_l} \in \mathbb{Z}. \\ &\textbf{Problem: } \text{Does } Q_1 x_1 \in A_1 \dots Q_k x_k \in A_k : \psi[(t_1, \dots, t_l)\backslash(\hat{t_1}, \dots, \hat{t_l})] \text{ hold?} \end{aligned} \tag{1}$$

We let $n := \max_i\{|A_i|\}$ denote the input size and will assume throughout the paper that all input numbers (i.e., the coordinates of the vectors in $A_1, \dots, A_k$ and the values $\hat{t_1}, \dots, \hat{t_l}$) are chosen from a polynomially sized universe, i.e., $\{-U, \dots, U\}$ with $U \le n^c$ for some c. Let $\mathsf{FOP}_{\mathbb{Z}}$ be the union of all $\mathsf{FOP}_{\mathbb{Z}}(\phi)$ problems, where ϕ has at least 3 quantifiers.[4] Besides 3-SUM, a variety of interesting problems is contained in $\mathsf{FOP}_{\mathbb{Z}}$; we discuss a few notable examples below and further examples in Section B in the Appendix.

Frequently, we distinguish formulas in $\mathsf{FOP}_{\mathbb{Z}}$ using their quantifier structure; e.g., $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\forall)$ describes the class of model checking problems $\mathsf{FOP}_{\mathbb{Z}}(\phi)$ where in ϕ we have $Q_1 = Q_2 = \exists$ and $Q_3 = \forall$. Furthermore, we let $\mathsf{FOP}_{\mathbb{Z}}^k$ be the union of all $\mathsf{FOP}_{\mathbb{Z}}(\phi)$ problems, where ϕ consists of precisely k quantifiers, regardless of their quantifier structure. For a quantifier $Q \in \{\exists, \forall\}$, we write Q^k for the repetition $\underbrace{Q \dots Q}_{k \text{ times}}$. Finally, we remark that a small subset of $\mathsf{FOP}_{\mathbb{Z}}$ has already been studied by An et al. [11], for a discussion see Section 1.3.

1.2 Our Contributions

We seek to determine completeness results for the class $\mathsf{FOP}_{\mathbb{Z}}$. In particular: What are the largest fragments of this class for which 3-SUM (or more generally, k-SUM) is complete? Is there a problem that is complete for the entire class?

Intuitively, we say that a $T_A(n)$-time solvable problem A is *(fine-grained) complete* for a $T_{\mathcal{C}}(n)$-time solvable class of problems $\mathcal{C}$, if the existence of an $O(T_A(n)^{1-\epsilon})$-time algorithm for A with $\epsilon > 0$ implies that for *all* problems C in $\mathcal{C}$ there exists $\delta > 0$ such that C can be solved in time $O(T_{\mathcal{C}}(n)^{1-\delta})$. We extend this notion to completeness of a *family* of problems, since strictly speaking, any (geometric) problem over $\mathbb{Z}^d$ expressible in linear integer arithmetic corresponds to a family of formulas $\mathsf{FOP}_{\mathbb{Z}}$ (one for each $d \in \mathbb{N}$). Formally, consider a family of problems $\mathcal{P}$ with an associated time bound $T_{\mathcal{P}}(n)$ and a class of problems $\mathcal{C}$ with an associated time bound $T_{\mathcal{C}}(n)$; usually $T_{\mathcal{P}}(n), T_{\mathcal{C}}(n)$ denote the running time of the fastest known algorithm solving all problems in $\mathcal{P}$ or $\mathcal{C}$, respectively (often, we omit these time bounds, as they are clear from context).[5] We say that $\mathcal{P}$ is *(fine-grained) complete* for $\mathcal{C}$, if

1. the family $\mathcal{P}$ is a subset of the class $\mathcal{C}$, and
2. if for all problems P in $\mathcal{P}$ there exists $\epsilon > 0$ such that P can be solved in time $O(T_{\mathcal{P}}(n)^{1-\epsilon})$, then for all problems C in $\mathcal{C}$ there exists some $\delta > 0$ such that we can solve C in time $O(T_{\mathcal{C}}(n)^{1-\delta})$.

[3]Below, we use the notation $\psi[(t_1, \dots, t_l)\backslash(\hat{t_1}, \dots, \hat{t_l})]$ to denote the substitution of the variables $t_1, \dots, t_l$ by $\hat{t_1}, \dots, \hat{t_l}$ respectively.

[4]It is not too difficult to see that all formulas with 2 quantifiers can be model-checked in near-linear time; see Section C.2 in the Appendix for further details.

[5]Here, we use *family* and *class* as a purely semantic and intuitive distinction: A family consists of a small set of similar problems, and a class consists of a large and diverse variety of problems.

That is, a polynomial-factor improvement for solving the problems in $\mathcal{P}$ would lead to a polynomial-factor improvement in solving *all* problems in $\mathcal{C}$. If a singleton family $\mathcal{P} = \{P\}$ is fine-grained complete for $\mathcal{C}$, then we also say that P is fine-grained complete for $\mathcal{C}$. We work with standard hypotheses and problems encountered in fine-grained complexity; for detailed definition of these, we refer to Section A in the Appendix.

1.2.1 k-SUM is complete for the existential fragment of $\mathsf{FOP}_{\mathbb{Z}}$

Consider first the existential fragment of $\mathsf{FOP}_{\mathbb{Z}}$, i.e., formulas exhibiting only existential quantifiers. Any $\mathsf{FOP}_{\mathbb{Z}}$ formula with k existential quantifiers can be decided using a standard meet-in-the-middle approach, augmented by orthogonal range search, in time $\tilde{O}(n^{\lceil k/2 \rceil})$[6], we refer to Section C in the Appendix for details. Since k-SUM is a member of $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$, this running time is optimal up to subpolynomial factors, assuming the k-SUM Hypothesis. As our first contribution, we provide a converse reduction. Specifically, we show that a polynomially improved k-SUM algorithm would give a polynomially improved algorithm for solving the entire class. In our language, we show that k-SUM is fine-grained complete for formulas of $\mathsf{FOP}_{\mathbb{Z}}$ with k existential quantifiers.

Theorem 1.1 (k-SUM is $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$-complete)**.** *Let $k \geq 3$ and assume that k-SUM can be solved in time $T_{k\mathrm{SUM}}(n)$. For any problem P in $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$, there exists some c such that P can be solved in time $O(T_{k\mathrm{SUM}}(n) \log^c n)$.*

Thus, if there are $k \geq 3$ and $\epsilon > 0$ such that we can solve k-SUM in time $O(n^{\lceil k/2 \rceil - \epsilon})$, then we can solve all problems in $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ in time $O(n^{\lceil k/2 \rceil - \epsilon'})$ for any $0 < \epsilon' < \epsilon$. By a simple negation argument, we conclude that k-SUM is also complete for the class of problems $\mathsf{FOP}_{\mathbb{Z}}(\forall^k)$.

The above theorem generalizes and unifies previous reductions from problems expressible as $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ formulas to 3-SUM, using different proof ideas: Jafargholi and Viola [48, Lemma 4] give a simple randomized linear-time reduction from triangle detection in sparse graphs to 3-SUM, and a derandomization via certain combinatorial designs; for a proof using the completeness theorem see Example B.7 in the Appendix. Dudek, Gawrychowski, and Starikovskaya [37] study the family of 3-linear degeneracy testing (3-LDT), which constitutes a large and interesting subset of $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$: This family includes, for any $\alpha_1, \alpha_2, \alpha_3, t \in \mathbb{Z}$, the *3-partite* formula $\exists a_1 \in A_1 \exists a_2 \in A_2 \exists a_3 \in A_3 : \alpha_1 a_1 + \alpha_2 a_2 + \alpha_3 a_3 = t$ and the *1-partite* formula $\exists \alpha_1, \alpha_2, \alpha_3 \in A : \alpha_1 a_1 + \alpha_2 a_2 + \alpha_3 a_3 = t \wedge a_1 \neq a_2 \wedge a_2 \neq a_3 \wedge a_1 \neq a_3$. The authors show that each such formula is either trivial or subquadratic *equivalent* to 3-SUM. For 3-partite formulas, a reduction to 3-SUM is essentially straightforward. For 1-partite formulas, Dudek et al. [37] use color coding.[7] As further examples for reductions from $\mathsf{FOP}_{\mathbb{Z}}$ problems to k-SUM, we highlight a reduction from Vector k-SUM to k-SUM [5] as well as a reduction from $(\min, +)$-convolution to 3-SUM (see [13, 35]) based on a well-known bit-level trick due to Vassilevska Williams and Williams [62], which allows us to reduce inequalities to equalities.

Perhaps surprisingly in light of its generality and applicability, Theorem 1.1 is obtained via a very simple, deterministic reduction that combines the tricks from [5, 62]. This generality comes at the cost of polylogarithmic factors (which we do not optimize), which depend on the number of inequalities occurring in the considered formula; for the details see Section 3.

[6] We use the notation $\tilde{O}(T) := T \log^{O(1)}(T)$ to hide polylogarithmic factors.

[7] We remark that the reverse direction, i.e., 3-SUM-hardness of non-trivial formulas, is technically much more involved and can be regarded as the main technical contribution of [37].

1.2.2 Completeness for counting witnesses

We provide a certain extension of the above completeness result to the problem class of *counting* witnesses to existential $\mathsf{FOP}_{\mathbb{Z}}$ formulas[8]. Counting witnesses is an important task particularly in database applications (usually referred to as model counting). Furthermore, we will make use of witness counting to *decide* certain quantified formulas in subsequent results detailed below and found in Section 4.

Theorem 1.2. *Let $k \geq 3$ be odd. If there is $\epsilon > 0$ such that we can count the number of witnesses for k-SUM in time $O(n^{\lceil k/2 \rceil - \epsilon})$, then for all problem P in $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$, there is some $\epsilon' > 0$ such that we can count the number of witnesses for P in time $O(n^{\lceil k/2 \rceil - \epsilon'})$.*

Leveraging the recent breakthrough by [29] that 3-SUM is subquadratic equivalent to counting witnesses of 3-SUM, we obtain the corollary that *3-SUM is hard even for counting witnesses of* $\mathsf{FOP}_{\mathbb{Z}}(\exists^3)$.

Corollary 1.3. *For all problems P in $\mathsf{FOP}_{\mathbb{Z}}(\exists^3)$, there is some $\epsilon_P > 0$ such that we can count the number of witnesses for P in randomized time $O(n^{2-\epsilon_P})$ if and only if there is some $\epsilon' > 0$ such that 3-SUM can be solved in randomized time $O(n^{2-\epsilon'})$.*

1.2.3 Completeness for general quantifier structures of $\mathsf{FOP}_{\mathbb{Z}}$

In light of our first completeness result, one might wonder whether k-SUM is complete for deciding all k-quantifier formulas in $\mathsf{FOP}_{\mathbb{Z}}$, regardless of the quantifier structure of the formulas.

Note that for these general quantifier structures, a baseline algorithm with running time $\tilde{O}(n^{k-1})$ can be achieved by a combination of brute-force and orthogonal range queries; see Section C in the Appendix for details.

However, by [11, Theorem 15] there exists a $\mathsf{FOP}_{\mathbb{Z}}(\exists^{k-1}\forall)$-formula ϕ that cannot be solved in time $O(n^{k-1-\epsilon})$-time unless the 3-uniform hyperclique hypothesis is false (see the discussion in Section 1.3 and Hypothesis A.12). Thus, proving that 3-SUM is complete for all 3-quantifier formulas would establish that the 3-uniform hyperclique hypothesis implies the 3-SUM hypothesis – this would be a novel tight reduction among important problems/hypotheses in fine-grained complexity theory. For $k \geq 4$, it becomes even more intricate: the conditionally optimal running time of $n^{k-1\pm o(1)}$ for $\mathsf{FOP}_{\mathbb{Z}}(\exists^k\forall)$ formulas exceeds the conditionally optimal running time of $n^{\lceil \frac{k}{2} \rceil \pm o(1)}$ for $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ formulas.

We are nevertheless able to obtain a completeness result for general quantifier structures: Specifically, we show that if two geometric problems over $\mathbb{Z}^d$ can be solved in time $O(n^{2-\epsilon_d})$ where $\epsilon_d > 0$ for all d, then each k-quantifier formula in $\mathsf{FOP}_{\mathbb{Z}}$ can be decided in time $O(n^{k-1-\epsilon})$ for some $\epsilon > 0$. These problems are (1) a variation of the Hausdorff distance that we call *Hausdorff distance under n Translations* and (2) the Pareto Sum problem; the details are covered in Section 5. These results enable a clear picture of the class $\mathsf{FOP}_{\mathbb{Z}}$. Furthermore, they still relate to the aim of studying the power of the 3-SUM problem, by showcasing clear limits to the power of 3-SUM.

Hausdorff Distance under n Translations Among the most common translation-invariant distance measures for given point sets B and C is the Hausdorff Distance under Translation [32, 24, 26, 31, 55, 47].To define it, we denote the directed Hausdorff distance under the L_∞ metric

[8]A witness for a $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ formula $\exists a_1 \in A_1 \ldots \exists a_k \in A_k : \varphi$ with $\hat{t_1}, \ldots, \hat{t_l} \in \mathbb{Z}$ is a tuple $(a_1, \ldots, a_k) \in A_1 \times \cdots \times A_k$ that satisfies the formula $\varphi[(t_1, \ldots, t_l)\backslash(\hat{t_1}, \ldots, \hat{t_l})]$.

by $\delta_{\overrightarrow{H}}(B,C) := \max_{b\in B}\min_{c\in C}\|b-c\|_\infty$.[9] The Hausdorff distance under translation $\delta^T_{\overrightarrow{H}}(B,C)$ is defined as the minimum Hausdorff distance of B and an arbitrary translation of C, i.e.,

$$\delta^T_{\overrightarrow{H}}(B,C) \coloneqq \min_{\tau\in\mathbb{R}^d}\delta_{\overrightarrow{H}}(B, C+\{\tau\}) = \min_{\tau\in\mathbb{R}^d}\max_{b\in B}\min_{c\in C}\|b-(c+\tau)\|_\infty.$$

For $d=2$, Bringmann and Nusser [24] were able to show a $(|B||C|)^{1-o(1)}$ time lower bound based on the orthogonal vector hypothesis, and there exists a matching $\tilde{O}(|B||C|)$ upper bound by Chew et al. [33].

We shall establish that restricting the translation vector to be among a set of m candidate vectors yields a central problem in $\mathsf{FOP}_{\mathbb{Z}}$. Specifically, we define the Hausdorff distance under Translation in A, denoted as $\delta^{T(A)}_{\overrightarrow{H}}(B,C)$, by

$$\delta^{T(A)}_{\overrightarrow{H}}(B,C) \coloneqq \min_{\tau\in A}\delta_{\overrightarrow{H}}(B, C+\{\tau\}) = \min_{\tau\in A}\max_{b\in B}\min_{c\in C}\|b-(c+\tau)\|_\infty.$$

Correspondingly, we define the problem *Hausdorff distance under m Translations* as: Given $A,B,C\subseteq\mathbb{Z}^d$ with $|A|\le m$, $|B|,|C|\le n$ and a distance value $\gamma\in\mathbb{N}$, determine whether $\delta^{T(A)}_{\overrightarrow{H}}(B,C)\le\gamma$. Note that this can be rewritten as a $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$-formula; see Example B.6 in the Appendix for details.

The Hausdorff distance under m Translations occurs naturally when approximating the Hausdorff distance under translation: Specifically, common algorithms compute a set A of $|A|=f(\epsilon)$ translations such that $\delta^{T(A)}_{\overrightarrow{H}}(B,C)\le(1+\epsilon)\delta^T_{\overrightarrow{H}}(B,C)$. Generally, this problem is then solved by performing $|A|$ computations of the Hausdorff distance, which yields $\tilde{O}(|A|n)=\tilde{O}(f(\epsilon)n)$-time algorithms [58]. Improving over the $\tilde{O}(mn)$-time baseline for Hausdorff Distance under m Translations would thus lead to immediate improvements for approximating the Hausdorff Distance under Translation. Our results will establish additional consequences of fast algorithms for this problem: an $O(n^{2-\epsilon_d})$-time algorithm with $\epsilon_d>0$ for Hausdorff distance under n Translations would give an algorithmic improvement for the classes of $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$- and $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\forall)$-formulas.

Verification of Pareto Sums Our second geometric problem is a verification version of computing *Pareto sums*: Given point sets $A,B\subseteq\mathbb{Z}^d$, the Pareto sum C of A,B is defined as the Pareto front of their sumset $A+B=\{a+b\mid a\in A, b\in B\}$. Put differently, the Pareto sum of A,B is a set of points C satisfying (1) $C\subseteq A+B$, (2) for every $a\in A$ and $b\in B$, the vector $a+b$ is dominated[10] by some $c\in C$ and (3) there are no distinct $c,c'\in C$ such that c' dominates c. The task of computing Pareto sums appears in various multicriteria optimization settings [12, 57, 38, 54]; fast output-sensitive algorithms (both in theory and in practice) have recently been investigated by Hespe, Sanders, Storandt, and Truschel [46].

We consider the following problem as *Pareto Sum Verification*: Given $A,B,C\subseteq\mathbb{Z}^d$, determine whether

$$\forall a\in A\,\forall b\in B\,\exists c\in C: a+b\le c.$$

[9] Since we will exclusively consider the *directed* Hausdorff distance under Translation, we will drop "directed" throughout the paper.

[10] We consider the usual domination notion: A vector $u\in\mathbb{Z}^d$ is dominated by some vector $v\in\mathbb{Z}^d$ (written $u\le v$) if and only if in all dimensions $i\in[d]$ it holds that $u[i]\le v[i]$.

The complexity of Pareto Sum Verification[11] is tightly connected to output-sensitive algorithms for Pareto sum. Specifically, solving Pareto Sum Verification reduces to *computing* the Pareto sum C when given inputs A, B of size at most n with the promise that $|C| = \Theta(n)$; see Section 7.3 for details. The work of Hespe et al. [46] gives a practically fast $O(n^2)$-time algorithm in this case for $d = 2$; note that for $d \geq 3$, we still obtain an $\tilde{O}(n^2)$-time algorithm via our baseline algorithm, which is described in Section C.3 in the Appendix.

A problem pair that is complete for $\mathsf{FOP}_{\mathbb{Z}}$ As a pair, these two geometric problems turn out to be fine-grained complete for the class $\mathsf{FOP}_{\mathbb{Z}}$.

Theorem 1.4. *There is a function $\epsilon(d) > 0$ such that both of the following problems can be solved in time $O(n^{2-\epsilon(d)})$*

- *Pareto Sum Verification,*
- *Hausdorff distance under n Translations,*

if and only if for each problem P in $\mathsf{FOP}_{\mathbb{Z}}^k$ with $k \geq 3$ there exists an $\epsilon_P > 0$ such that P can be solved in time $O(n^{k-1-\epsilon_P})$.

The above theorem shows that a single pair of natural problems captures the fine-grained complexity of the expressive and diverse class $\mathsf{FOP}_{\mathbb{Z}}$. As an illustration just how expressive this class is, we observe the following barriers:[12]

1. If there is some $\epsilon > 0$ such that all problems in $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\forall)$ (or $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$) can be solved in time $O(n^{2-\epsilon})$, then OVH (and thus SETH) is false [11, Theorem 16].
2. If there is some $\epsilon > 0$ such that all problems in $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$ (or $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\forall)$) can be solved in time $O(n^{2-\epsilon})$, then the Hitting Set Hypothesis is false [11, Theorem 12].
3. If for all problems P in $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\forall)$ (or $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$), there exists some $\epsilon > 0$ such that we can solve P in $O(n^{2-\epsilon})$, then the 3-uniform Hyperclique Hypothesis is false [11, Theorem 15].
4. If for all problems P in $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$ ($\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\forall), \mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$, or $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\forall)$), there exists some $\epsilon > 0$ such that we can solve P in time $O(n^{2-\epsilon})$, then the 3-SUM Hypothesis is false (Theorem 1.1 with Lemma 2.4).
5. If for all problems in $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$ (or $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$), there exists some $\epsilon > 0$ such that we can solve the problem in $O(n^{2-\epsilon})$, then MaxConv lower bound[13] can be solved in time $O(n^{2-\epsilon})$ (Lemma B.8).

Theorem 1.4 raises the question whether for any constant dimension d, the Hausdorff distance under n Translations admits a subquadratic reduction to Pareto Sum Verification. A positive answer would establish Pareto Sum Verification as complete for the *entire* class $\mathsf{FOP}_{\mathbb{Z}}$. We elaborate on this in Section 8.

[11] We remark that our problem definition only checks a single of the three given conditions, specifically, condition (2). However, in Section 7.3, we will establish that the verifying *all three* conditions reduces to verifying this single condition. More specifically, for sets A, B, C of size at most n, we obtain that if we can solve Pareto Sum Verification in time $T(n)$, then we can check whether C is the Pareto sum of A, B in time $O(T(n))$.

[12] The first three statements follow from $\mathsf{FOP}_{\mathbb{Z}}$ generalizing the class *PTO* studied in [11], see Section 1.3. The remaining statements rely on the additive structure of $\mathsf{FOP}_{\mathbb{Z}}$.

[13] See Definition A.13.

1.2.4 3-SUM is complete for $\mathsf{FOP}_{\mathbb{Z}}$ formulas of low inequality dimension

Returning to our motivating question, we ask: Since it appears unlikely to prove completeness of 3-SUM for all $\mathsf{FOP}_{\mathbb{Z}}$ formulas (as this requires a tight 3-uniform hyperclique lower bound for 3-SUM), can we at least identify a large fragment of $\mathsf{FOP}_{\mathbb{Z}}$ for which 3-SUM is complete? In particular, can we extend our first result of Theorem 1.1 from existentially quantified formulas to substantially different problems in $\mathsf{FOP}_{\mathbb{Z}}$, displaying other quantifier structures?

Surprisingly, we are able to show that 3-SUM is complete for *low-dimensional* $\mathsf{FOP}_{\mathbb{Z}}$ formulas, *independent of their quantifier structure*. To formalize this, we introduce the *inequality dimension* of a $\mathsf{FOP}_{\mathbb{Z}}$ formula as the smallest number of linear inequalities required to model it. More formally, consider a $\mathsf{FOP}_{\mathbb{Z}}$ formula $\phi = Q_1 x_1 \in A_1, \ldots, Q_k x_k \in A_k : \psi$ with $A_i \subseteq \mathbb{Z}^{d_i}$. The *inequality dimension* of ϕ is the smallest number s such that there exists a Boolean function $\psi' : \{0,1\}^s \to \{0,1\}$ and (strict or non-strict) linear inequalities $L_1, \ldots, L_s$ in the variables $\{x_i[j] : i \in \{1,\ldots,k\}, j \in \{1,\ldots,d_i\}\}$ and the free variables such that $\psi(x_1,\ldots,x_k)$ is equivalent to $\psi'(L_1,\ldots,L_s)$. As an example, the 3-SUM formula $\exists a \in A \exists b \in B \exists c \in C : a+b=c$ has inequality dimension 2, as $a+b=c$ can be modelled as a conjunction of the two linear inequalities $a+b \le c$ and $a+b \ge c$, whereas no single linear inequality can model $a+b=c$.

We show that 3-SUM is fine-grained complete for model-checking $\mathsf{FOP}_{\mathbb{Z}}^3$ formulas with inequality dimension at most 3. This result is our perhaps most interesting technical contribution and intuitively combines our result that 3-SUM is hard for counting $\mathsf{FOP}_{\mathbb{Z}}$ witnesses (Corollary 1.3) with a geometric argument, specifically, that the union of n unit cubes in $\mathbb{R}^3$ can be decomposed into the union of $O(n)$ pairwise interior- and exterior-disjoint axis-parallel boxes. To this end, we extend a result from [31], which constructs pairwise interior-disjoint axis-parallel boxes, to also achieve exterior-disjointness. For more details, see the Technical Overview below and Section 6.

Theorem 1.5. *There is an algorithm deciding* 3*-SUM in randomized time* $O(n^{2-\epsilon})$ *for an* $\epsilon > 0$*, if and only if for each problem* P *in* $\mathsf{FOP}_{\mathbb{Z}}^k$ *with* $k \ge 3$ *and inequality dimension at most* 3*, there exists some* $\epsilon' > 0$ *such that we can solve* P *in randomized time* $O(n^{k-1-\epsilon'})$*.*

Note that this fragment of $\mathsf{FOP}_{\mathbb{Z}}$ contains a variety of interesting problems. A general example is given by comparisons of sets defined using the sumset arithmetic[14], which correspond to formulas of inequality dimension at most 2: E.g., checking, given sets $A, B, C \subseteq \mathbb{Z}$ and $t \in \mathbb{Z}$, whether C is an additive t-approximation of the sumset $A+B$ is equivalent to verifying the conjunction of the $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$ problem[15] $A + B \subseteq C + \{0,\ldots,t\}$ and (2) the $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$ problem[16] $C \subseteq A+B$. Likewise, this extends to λ-multiplicative approximations of sumsets. Furthermore, the problems corresponding to general sumset comparisons like $\alpha_1 A_1 + \cdots + \alpha_i A_i \subseteq \alpha_{i+1} A_{i+1} + \cdots + \alpha_k A_k + \{-\ell,\ldots,u\}$ have inequality dimension at most 2 as well.

Our results of Theorems 1.4 and 1.5 suggests to view Pareto Sum Verification as a geometric, high-dimensional generalization of 3-SUM. Furthermore, it remains an interesting problem to establish the highest d such that 3-SUM is complete for $\mathsf{FOP}_{\mathbb{Z}}$ formulas of inequality dimension at most d; for a discussion see Section 8.

Further Applications As an immediate application of our first completeness theorem, we obtain a simple proof of a $n^{4/3-o(1)}$ lower bound for the 4-SUM problem based on the 3-uniform hyperclique

[14]The sumset arithmetic uses the sumset operator $X+Y$ to denote the sumset $\{x+y \mid x \in X, y \in Y\}$ and λX to denote $\{\lambda x \mid x \in X\}$.

[15]Note that the corresponding formula is $\forall a \in A \forall b \in B \exists c \in C : (c \le a+b) \wedge (a+b \le c+t)$, which clearly has inequality dimension at most 2.

[16]Note that the corresponding formula is $\forall c \in C \exists a \in A \exists b \in B : a+b=c$, which clearly has inequality dimension at most 2.

hypothesis; see Section 7 for details. Specifically, by Theorem 1.1, it suffices to model the 3-uniform 4-hyperclique problem as a problem in $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists\exists)$. The resulting conditional lower bound is implicitly known in the literature, as it can alternatively be obtained by combining a 3-uniform hyperclique lower bound for 4-cycle given in [53] with a folklore reduction from 4-cycle to 4-SUM (see [48] for a deterministic reduction from 3-cycle to 3-SUM).

Theorem 1.6. *If there is some $\epsilon > 0$ such that 4-SUM can be solved in time $O(n^{\frac{4}{3}-\epsilon})$, then the 3-uniform hyperclique hypothesis fails.*

Similarly, we can also give a simple proof for a known lower bound for 3-SUM.

Another application of our results is to establish class-based conditional bounds. As a case in point, consider the problem of computing the Pareto sum of $A, B \subseteq \mathbb{Z}^d$: Clearly, this problem can be solved in time $\tilde{O}(n^2)$ by explicitly computing the sumset $A + B$ and computing the Pareto front using any algorithm running in near-linear time in its input, e.g. [43]. We prove the following conditional optimality results already in the case when the desired output (the Pareto sum of A, B) has size $\Theta(n)$.

Theorem 1.7 (Pareto Sum Computation Lower Bound)**.** *The following conditional lower bounds hold for output-sensitive Pareto sum computation:*

1. *If there is $\epsilon > 0$ such that we can compute the Pareto sum C of $A, B \subseteq \mathbb{Z}^2$, whenever C is of size $\Theta(n)$, in time $O(n^{2-\epsilon})$, then the 3-SUM hypothesis fails (thus, for any $\mathsf{FOP}^k_{\mathbb{Z}}$ formula ϕ of inequality dimension at most 3, there is $\epsilon' > 0$ such that ϕ can be decided in time $O(n^{k-1-\epsilon'})$).*
2. *If for all $d \geq 2$, there is $\epsilon > 0$ such that we can compute the Pareto sum C of $A, B \subseteq \mathbb{Z}^d$, whenever C is of size $\Theta(n)$, in time $O(n^{2-\epsilon})$, then there is some $\epsilon' > 0$ such that we can decide all $\mathsf{FOP}_{\mathbb{Z}}$ formulas with k quantifiers not ending in $\exists\forall\exists$ or $\forall\exists\forall$ in time $O(n^{k-1-\epsilon'})$.*

Our lower bound for 2D strengthens a quadratic-time lower bound found by Funke et al. [42] based on the $(\min, +)$-convolution hypothesis to hold already under the weaker (i.e., more believable) 3-SUM hypothesis. For higher dimensions, we furthermore strengthen the conditional lower bound via its connection to $\mathsf{FOP}_{\mathbb{Z}}$.

We conclude with remaining open questions in Section 8.

1.3 Further Related Work

To our knowledge, the first investigation of the connection between classes of model-checking problems and central problems in fine-grained complexity was given by Williams [59], who shows that the k-clique problem is complete for the class of existentially-quantified first order graph properties, among other results. As important follow-up work, Gao et al. [45] establish OV as complete problem for model-checking any first-order property.

Subsequent results include classification results for $\exists^k\forall$-quantified first-order graph properties [18], fine-grained upper and lower bounds for counting witnesses of first-order properties [36], completeness theorems for multidimensional ordering properties [11] (discussed below), completeness and classification results for optimization classes [16, 17] as well as an investigation of sparsity for monochromatic graph properties [41].

We remark that An et al. [11] study completeness results for a strict subset of $\mathsf{FOP}_{\mathbb{Z}}$ formulas: Specifically, they introduce a class $\mathsf{PTO}_{k,d}$ of k-quantifier first-order sentences over inputs $\mathbb{N}^d$ (or, without loss of generality $\{1, \ldots, n\}^d$) that may only use *comparisons* of coordinates (and constants). Note that such sentences lack additive structure, and indeed the fine-grained complexity differs decisively from $\mathsf{FOP}_{\mathbb{Z}}$: E.g., for $\mathsf{PTO}(\exists\exists\exists)$ formulas, they establish the sparse triangle detection problem as complete, establishing a conditionally tight running time of $m^{2\omega/(\omega+1)\pm o(1)}$. This is in stark contrast to $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$ formulas, for which we establish 3-SUM as complete problem, yielding a conditionally optimal running time of $n^{2\pm o(1)}$. In particular, for each 3-quantifier structure $Q_1Q_2Q_3$, a $O(n^{2-\epsilon})$-time algorithm for all $\mathsf{FOP}_{\mathbb{Z}}(Q_1Q_2Q_3)$ problems would break a corresponding hardness barrier[17].

Since any $\mathsf{PTO}_{k,d}$ formula is also a $\mathsf{FOP}_{\mathbb{Z}}$ formula with the same quantifier structure, any hardness result in [11] for $\mathsf{PTO}(Q_1,\dots,Q_k)$ carries over to $\mathsf{FOP}_{\mathbb{Z}}(Q_1,\dots,Q_k)$. On the other hand, any of our algorithmic results for $\mathsf{FOP}_{\mathbb{Z}}(Q_1,\dots,Q_k)$ transfers to its subclass $\mathsf{PTO}(Q_1,\dots,Q_k)$.

2 Technical Overview

In this section, we sketch the main ideas behind our proofs.

Completeness of k-SUM for $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ With the right ingredients, proving that k-SUM is complete for $\mathsf{FOP}_{\mathbb{Z}}$ formulas with k existential quantifiers (Theorem 1.1) is possible via a simple approach: We observe that any $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ formula ϕ can be rewritten such that we may assume that ϕ is a conjunction of m inequalities. We then use a slight generalization of a bit-level trick of [62] to reduce each inequality to an equality, incurring only $O(\log n)$ overhead per inequality (intuitively, we need to guess the most significant bit position at which the left-hand side and the right-hand side differ). Thus, we obtain $O(\log^m n)$ conjunctions of m equalities; each such conjunction can be regarded as an instance of Vector k-SUM. Using a straightforward approach for reducing Vector k-SUM to k-SUM given in [5], the reduction to k-SUM follows. We give all details in Section 3.

Counting witnesses and handling multisets While the reduction underlying Theorem 1.1 preserves the existence of solutions, it fails to preserve the number of solutions. The challenge is that when applying the bit-level trick to reduce inequalities to equalities, we need to make sure that for each witness of a $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ formula ϕ, there is a unique witness in the k-SUM instances produced by the reduction. While it is straightforward to ensure that we do not produce multiple witnesses, the subtle issue arises that distinct witnesses for ϕ may be mapped to the same witness in the k-SUM instances. It turns out that it suffices to solve a *multiset* version of #k-SUM, i.e., to count all witnesses in a k-SUM instance in which each input number may occur multiple times.

Thus, to obtain Theorem 1.2, we show a fine-grained equivalence of Multiset #k-SUM and #k-SUM, for all odd $k \geq 3$. This fine-grained equivalence, which we prove via a heavy-light approach, might be of independent interest.[18] Combining this equivalence with an inclusion-exclusion argument, we may thus lift Theorem 1.1 to a counting version for all odd $k \geq 3$.

In the reductions below, we will make crucial use of the immediate corollary of Theorem 1.2 and [29] that for each $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$ formula ϕ, there exists a subquadratic reduction from counting witnesses for ϕ to 3-SUM (Corollary 1.3).

[17] Specifically, an $O(n^{2-\epsilon})$ time algorithm for problems in $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists), \mathsf{FOP}_{\mathbb{Z}}(\forall\forall\forall), \mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$, or $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\forall)$ with $\epsilon > 0$ would refute the 3-SUM hypothesis. Furthermore, an $O(n^{2-\epsilon})$ time algorithm for problems in $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$, $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\forall)$, $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$, or $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\forall)$ with $\epsilon > 0$ would immediately yield an improvement for the MaxConv lower bound problem [35]; for details see Section B in the Appendix.

[18] We remark that it is plausible that the proof of the subquadratic equivalence of 3-SUM and #3-SUM due to Chan et al. [29] could be extended to establish subquadratic equivalence with Multiset #3-SUM as well. Note, however, that a fine-grained equivalence of #k-SUM and k-SUM is not known for any $k \geq 4$.

On general quantifier structures We perform a systematic study on the different quantifier structures for $k = 3$. Due to simple negation arguments, we only have to perform a systematic study on the classes of problems $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$, $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$, $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$, $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$.

First, we state a simple lemma establishing syntactic complete problems, for the classes above.

Lemma 2.1 (Syntactic Complete problems (Informal Version))**.** *Let $Q_1, Q_2 \in \{\exists, \forall\}$. We can reduce every formula of the class $\mathsf{FOP}_{\mathbb{Z}}(Q_1Q_2\exists)$ to the formula*

$$Q_1\tilde{a_1} \in \tilde{A}_1 Q_2\tilde{a_2} \in \tilde{A}_2 \exists\tilde{a_3} \in \tilde{A}_3 : \tilde{a_1} + \tilde{a_2} \leq \tilde{a_3}.$$

On the quantifier change $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists) \to \mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$. We rely on the subquadratic equivalence between 3-SUM and a functional version of 3-SUM called All-ints 3-SUM, which aims to determine for every $a \in A$ whether there is a solution involving a. A randomized subquadratic equivalence was given in [61], which can be turned deterministic [52].

This equivalence allows us to use the bit-level trick to turn inequalities to equalities, despite it seemingly not interacting well with the quantifier structure $\forall\exists\exists$ at first sight. This results in a proof of the following hardness result.

Lemma 2.2. *If 3-SUM can be solved in time $O(n^{2-\epsilon})$ for an $\epsilon > 0$, then all problems P of $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$ can be solved in time $O(n^{2-\epsilon_P})$ for an $\epsilon_P > 0$.*

On the quantifier change $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists) \to \mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$. As a first result for the class $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$, we are able to show equivalence to 3-SUM for a specific problem in this class, thus introducing a 3-SUM equivalent problem with a different quantifier structure in comparison to 3-SUM. Specifically, we consider the problem of verifying additive t-approximation of sumsets. We are able to precisely characterize the fine-grained complexity depending on t.

Formally, we show the following theorem.

Theorem 2.3. *Consider the Additive Sumset Approximation problem of deciding, given $A, B, C \subseteq \mathbb{Z}, t \in \mathbb{Z}$, whether*

$$A + B \subseteq C + \{0, \ldots, t\}.$$

This problem is

- *solvable in time $O(n^{2-\delta})$ with $\delta > 0$, whenever $t = O(n^{1-\epsilon})$ for any $\epsilon > 0$,*
- *not solvable in time $O(n^{2-\epsilon})$, whenever $t = \Omega(n)$ assuming the Strong convolutional 3-SUM hypothesis.*

Furthermore, subquadratic hardness holds under the standard 3-SUM Hypothesis if no restriction on t is made.

The above theorem is essentially enabling a quantifier change transforming the $\exists\exists\exists$ quantifier structure for which 3-SUM is complete into a subquadratic equivalent problem with a quantifier structure $\forall\forall\exists$. Moreover, the 3-SUM hardness is a witness to the hardness of the class $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$.

Let us remark a few interesting aspects: The algorithmic part follows from sparse convolution techniques going back to Cole and Hariharan [34], see [20] for a recent account and also [27, 22, 19]. Specifically, whenever $t = O(n^{1-\epsilon})$, it holds that $|C + \{0, \ldots, t\}| = O(n^{2-\epsilon})$ and intuitively, we can use an output-sensitive convolution algorithm to compute $A + B$ and compare it to $C + \{0, \ldots, t\}$.[19]

[19]The argument is slightly more subtle, since we need to avoid computing $A + B$ if its size exceeds $O(n^{2-\epsilon})$.

Our result indicates that an explicit construction of $C + \{0, \ldots, t\}$ is required, since once it may get as large as $\Omega(n^2)$, we obtain a $n^{2-o(1)}$-time lower bound assuming the Strong 3-SUM hypothesis.

The lower bound follows from describing the 3-SUM problem alternatively as $(A + B) \cap C \neq \emptyset$, which is equivalent to the negation of $(A + B) \subseteq \bar{C}$, where $\bar{C}$ denotes the complement of C. Thus, we aim to cover the complement of C by intervals of length t. While this appears impossible for 3-SUM, we employ the subquadratic equivalence of 3-SUM and its convolutional version due to Patrascu [56]. This problem will deliver us the necessary structure to represent this complement with the addition of few auxilliary points.

The reverse reduction from Additive Sumset Approximation to 3-SUM follows from Theorem 1.5 (as Additive Sumset approximation has inequality dimension 2).

On completeness results for $\mathsf{FOP}_{\mathbb{Z}}^k$ The above ingredients establish our completeness theorems by exhaustive search over remaining quantifiers. Specifically, by a combination of Theorem 2.3, which shows that Additive Sumset Approximation is 3-SUM hard, and a combination of Lemma 2.2 and Theorem 1.1, we get:

Lemma 2.4. *There is a function $\epsilon(d) > 0$ such that the Verification of Pareto Sum problem can be solved in time $O(n^{2-\epsilon(d)})$ if and only if all problems P in the classes*

- $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \ldots Q_{k-3}\exists\exists\exists)$,$\mathsf{FOP}_{\mathbb{Z}}(Q_1 \ldots Q_{k-3}\forall\forall\forall)$,
- $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \ldots Q_{k-3}\forall\exists\exists)$,$\mathsf{FOP}_{\mathbb{Z}}(Q_1 \ldots Q_{k-3}\exists\forall\forall)$,
- $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \ldots Q_{k-3}\forall\forall\exists)$,$\mathsf{FOP}_{\mathbb{Z}}(Q_1 \ldots Q_{k-3}\exists\exists\forall)$,

where $Q_1, \ldots Q_{k-3} \in \{\exists, \forall\}$ and $k \geq 3$, can be solved in time $O(n^{k-1-\epsilon_P})$ for an $\epsilon_P > 0$.

Similarly, for quantifier structures ending in $\exists\forall\exists$ and $\forall\exists\forall$ $\mathsf{FOP}_{\mathbb{Z}}$, we obtain the following completeness result.

Lemma 2.5. *There is a function $\epsilon(d) > 0$ such that the Hausdorff Distance under n Translations problem can be solved in time $O(n^{2-\epsilon(d)})$ if and only if all problems P in the classes*

- $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \ldots Q_{k-3}\exists\forall\exists)$, $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \ldots Q_{k-3}\forall\exists\forall)$,

where $Q_1, \ldots Q_{k-3} \in \{\exists, \forall\}$ and $k \geq 3$, can be solved in time $O(n^{k-1-\epsilon_P})$ for an $\epsilon_P > 0$.

The combination of Lemma 2.4 and Lemma 2.5, thus suffice to prove Theorem 1.4.

The 3-SUM completeness of formulas with inequality dimension at most 3 As a first idea, one could try to solve problems of different quantifier structures by just counting witnesses. Consider in the following the example $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$.

Assume we are promised that the formula $\forall a \in A \forall b \in B \exists c \in C \psi(a, b, c)$ satisfies a kind of *disjointness* property, specifically that for every $(a, b) \in A \times B$ there exists at most one $c \in C$ such that $\psi(a, b, c)$. Then satisfying the formula boils down to checking whether the number of witnesses (a, b, c) satisfiying $\psi(a, b, c)$ equals to $|A| \cdot |B|$.

To create this *disjointness* effect, we use the following geometric approach: We show that one can re-interpret the formula as $\forall a \in A \forall b \in B : a + b \in \bigcup_{c' \in C'} V(c')$, where $A, B, C' \subseteq \mathbb{Z}^3$, C' is a set of size $O(n)$ and $V(c')$ is an orthant associated to c'. Using an adapted variant of [31], we decompose this union of orthants in $\mathbb{R}^3$ (which we may equivalently view as sufficiently large congruent cubes) into a set $\mathcal{R}$ of $O(n)$ *disjoint* boxes. Thus, it remains to notice that the resulting problem, i.e., for all $a \in A, b \in B$ is there a box $R \in \mathcal{R}$ such that $a+b$ is contained in R, is a $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$ formula with the desired disjointness property, which can be handled as argued above.

For the class $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$, we perform a slightly more involved argument. The classes $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$ and $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$ reduce to 3-SUM regardless of the inequality dimension due to Theorem 1.1 and Lemma 2.2.

3 k-SUM is complete for existential $\mathsf{FOP}_{\mathbb{Z}}$ formulas

We begin with a simple completeness theorem that k-SUM is complete for the class of problems $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$. Since k-SUM is indeed a $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$-formula, it remains to show a fine-grained reduction from any $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ formula to k-SUM.

As a first step towards this theorem, we consider how to reduce a conjunction of m linear inequalities to a vector k-SUM instance.

Lemma 3.1. *Consider vectors $a_1 \in \{-U,\dots,U\}^{d_1},\dots,a_k \in \{-U,\dots,U\}^{d_k}$, integers $S_1,\dots,S_m \in \{-U,\dots,U\}$, for each $i \in \{1,\dots,m\}, j \in \{1,\dots,k\}$, vectors $c_{i,j} \in \mathbb{Z}^{d_j}$, a sufficiently large number M* [20] *and a formula*

$$\psi := \bigwedge_{i=1}^{m} \left(c_{i,j}^T a_j \geq S_i\right).$$

There exist $O(1)$ time computable functions $f_1^{\ell,\psi},\dots,f_k^{\ell,\psi}, g^{\ell,\psi,W}$ such that the following statements are equivalent

1. *The formula $\bigwedge_{i=1}^m \left(\sum_{j=1}^k c_{i,j}^T a_j \geq S_i\right)$ holds.*
2. *There are $\ell \in \{1,\dots,\lceil \log_2(M)\rceil\}^m, W \in \{1,\dots,k\}^m$ such that $f_1^{\ell,\psi}(a_1)+\dots+f_k^{\ell,\psi}(a_k) = g^{\ell,\psi,W}(S_1,\dots,S_m)$.*

Moreover, if the second item holds, there is a unique choice of such ℓ and W.

Essentially, the above lemma enables a reduction from a conjunction of inequality checks to a conjunction of equality checks. The full proof can be found in Section D in the Appendix. We can now continue with our completeness theorem.

Theorem 1.1 (k-SUM is $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$-complete)**.** *Let $k \geq 3$ and assume that k-SUM can be solved in time $T_{k\mathrm{SUM}}(n)$. For any problem P in $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$, there exists some c such that P can be solved in time $O(T_{k\mathrm{SUM}}(n)\log^c n)$.*

Proof. The high-level approach is as follows: Firstly, we massage our linear arithmetic formula into a normal form which is suitable to apply Lemma 3.1, and apply Lemma 3.1 to transform each inequality to an equality. Lastly, we get vector k-SUM instances which can be transformed into k-SUM instances as described in Lemma A.5.

Consider any fixed $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ formula ϕ. Let $A_1 \subseteq \mathbb{Z}^{d_1},\dots,A_k \subseteq \mathbb{Z}^{d_k}$ and $\hat{t}_1,\dots,\hat{t}_\ell \in \mathbb{Z}$ be an instance of $\mathsf{FOP}_{\mathbb{Z}}(\phi)$. We first substitute the free variables $t_1,\dots,t_\ell$ by $\hat{t}_1,\dots\hat{t}_\ell$, which yields the sentence $\phi[(t_1,\dots,t_l)\backslash(\hat{t_1},\dots,\hat{t_l})]$. We then transform this sentence into disjunctive normal form (DNF), more specifically, into the following form:

$$\exists a_1 \in A_1 \dots \exists a_k \in A_k : \bigvee_{h=1}^{H}\bigwedge_{i=1}^{m}\left(\sum_{j=1}^{k} c_{h,i,j}^T a_j \geq S_{h,i}\right), \tag{2}$$

[20] Let $C := \max\{|c_{i,j}[k]| : i \in \{1,\dots,m\}, j \in \{1,\dots,k\}, k \in \{1,\dots,d_j\}\}$ and $D := \max_{i=1}^k d_i$ We construct M to be sufficiently large for our purposes i.e $M := 4DUC$.

where $H, m \in \mathbb{N}$, each $S_{1,1}, \ldots, S_{H,m} \in \mathbb{Z}$, and for each $h \in \{1, \ldots, H\}, i \in \{1, \ldots, m\}, j \in \{1, \ldots, k\}, c_{h,i,j} \in \mathbb{Z}^{d_j}$. Transforming a formula into DNF is a standard routine in Linear Integer Arithmetic, see, e.g. [50]. Note that we may in particular assume that each conjunction involves only inequalities of the form $\sum_{j=1}^{k} c_{h,i,j}^T a_j \geq S_{h,i}$. To see this, note that over the integers, a strict inequality $x > y$ is equivalent to the non-strict inequality $x \geq y + 1$. Furthermore, an equality $x = y$ can be rewritten as the conjunction of two inequalities $x \leq y \wedge y \leq x$, similarly $x \neq y$ can be rewritten as $x < y \vee y < x$, and finally $x \leq y$ is equivalent to $-x \geq -y$.

Due to the commutativity of disjunction and existential quantifiers, Equation (2) is equivalent to $\bigvee_{h=1}^{H} \phi_h$, where

$$\phi_h := \exists a_1 \in A_1 \ldots \exists a_k \in A_k : \bigwedge_{i=1}^{m} \left(\sum_{j=1}^{k} c_{h,i,j}^T a_j \geq S_{h,i} \right).$$

In the remainder of the proof, we will show that any ϕ_h can be decided in time $O(T_{k\mathrm{SUM}}(n) \log^m n)$ time, from which the claim follows, as H and m are constants.

Thus, it suffices to consider an arbitrary formula of the form

$$\hat{\phi} := \exists a_1 \in A_1 \ldots \exists a_k \in A_k : \bigwedge_{i=1}^{m} \left(\sum_{j=1}^{k} c_{i,j}^T a_j \geq S_i \right). \tag{3}$$

By a simple application of Lemma 3.1 deciding Equation 3 can be reduced to deciding the following expression:

$$\begin{aligned} &\exists l \in \{1, \ldots, \lceil \log_2(M) \rceil\}^m \exists W \in \{1, \ldots, k\}^m \exists a_1 \in f_1^{l,\hat{\phi}}(A_1) \ldots \exists a_k \in f_k^{l,\hat{\phi}}(A_k) : \\ &\qquad\qquad\qquad\qquad a_1 + \cdots + a_k = g^{l,\hat{\phi},W}(S_1, \ldots, S_m). \end{aligned} \tag{4}$$

Equation 4 can be decided by $(\lceil \log_2(M) \rceil^m \cdot k^m) = O(\lceil \log_2(M) \rceil^m)$ calls to (M, m)-vector k-SUM instances. These can be reduced to k-SUM instances using Lemma A.5 [5], in time $\tilde{O}(n \log^m(M)) = \tilde{O}(n)$, where $M = \mathrm{poly}(n)$ is chosen according to the proof of Lemma 3.1. For the correctness, we remark that due to the commutativity of existential quantifiers, Equation 4 is equivalent to

$$\begin{aligned} &\exists a_1 \in A_1 \ldots \exists a_k \in A_k \exists l \in \{1, \ldots, \lceil \log_2((k+1)M) \rceil\}^m \exists W \in \{1, \ldots, k\}^m : \\ &f_1^{l,\hat{\phi}}(a_1) + \cdots + f_k^{l,\hat{\phi}}(a_k) = g^{l,\hat{\phi},W}(S_1, \ldots, S_m) \\ \overset{\text{Lm.3.1}}{\iff} &\exists a_1 \in A_1 \ldots \exists a_k \in A_k : \bigwedge_{i=1}^{m} \sum_{j=1}^{k} c_{i,j}^T a_j \geq S_i. \end{aligned}$$

□

4 On counting witnesses in $\mathsf{FOP}_{\mathbb{Z}}$

In this section, we show reductions from counting witnesses of $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ formulas to $\#k$-SUM, specifically, we prove Theorem 1.2. To do so, we adapt the proof of Theorem 1.1 given in Section 3 to a counting version. As discussed in Section 2, this requires us to work with a multiset version of $\#k$-SUM. Handling multisets is thus the main challenge addressed in this section. Formally, we say that a multiset is a set A together with a function $f : A \to \mathbb{N}$. For $a \in A$, we abbreviate $n_a := f(a)$ as the multiplicity of a. To measure multiset sizes, we still think of each a to have n_a copies in the input, i.e. the size of A is $\sum_{a \in A} n_a$.

Definition 4.1 ((U,d)-vector Multiset $\#k$-SUM)**.** *Let $X := \{-U,\dots,U\}^d$. Given k multisets $A_1,\dots,A_k \subseteq X$ and $t \in X$, we ask for the total number of k-SUM witnesses, that is*

$$\sum_{\substack{a_1+\cdots+a_k=t,\\ a_1\in A_1,\dots,a_k\in A_k}} \prod_{i=1}^{k} n_{a_i}.$$

Furthermore, define Multiset $\#k$-SUM as $(U,1)$-vector Multiset $\#k$-SUM and M-multiplicity $\#k$-SUM as Multiset $\#k$-SUM with the additional restriction that the multiplicity of each element is limited, that is for all $a \in A_1 \cup \cdots \cup A_k : n_a \leq M$ holds. Lastly, $\#k$-SUM is defined as 1-Multiplicity $\#k$-SUM and (U,d)-vector $\#k$-SUM is (U,d)-vector Multiset $\#k$-SUM where for all $a \in A_1 \cup \cdots \cup A_k : n_a = 1$ holds.

For the case of $\mathsf{FOP}^3_{\mathbb{Z}}$ we will also introduce the #All-ints version of the above problems, which just asks to determine for each $a_1 \in A_1$ the number of witnesses.

The following proof will be similar to the proof of Lemma A.5 of Abboud et al. in [5].

Lemma 4.2 ((U,d)-vector Multiset $\#k$-SUM $\leq_{\lceil k/2\rceil}$ Multiset $\#k$-SUM)**.** *If Multiset $\#k$-SUM can be solved in time $T(n)$ then (U,d)-vector Multiset $\#k$-SUM can be solved in time $O(nd\log(U) + T(n))$.*

Proof. Let $X := \{-U,\dots,U\}^d$ be our universe, $A_1,\dots,A_k \subseteq X$ and $t \in X$. Shift all vector entries to make them positive by adding U to them, accordingly shift the vector entries in t by kU. Moreover, let $B := 2kU + 1$. We define $b(x) := \sum_{j=0}^{d-1} x[j]B^j$. Now, construct multisets $A'_1,\dots,A'_k$ for $i \in \{1,\dots,k\}$ as $A'_i = \{b(a) : a \in A_i\}$, where for each $a \in A'_i$, we set the multiplicity of $n_{b(a)} := n_a$. Meaning, we adopt the multiplicities of each transformed element from its multiplicity in the original set. Moreover, set $t' := b(t)$.

Consider a solution $(a_1,\dots,a_k) \in A_1 \times \cdots \times A_k$ such that $\sum_{i=1}^{k} a_i = t$, then for each $j \in \{0,\dots,d-1\}$, it holds that $a_1[j] + \cdots + a_k[j] = t[j]$. As there is no overflow when adding, that is for each $a_1[j] + \cdots + a_k[j] < B$, we get

$$\begin{aligned} & a_1 + \cdots + a_k = t \\ \iff & \sum_{j=0}^{d-1} a_1[j]B^j + \cdots + \sum_{j=0}^{d-1} a_k[j]B^j = \sum_{j=0}^{d-1} t[j]B^j. \end{aligned}$$

The reduction creates a Multiset $\#k$-SUM instance with universe size $k \cdot B^d = \Theta(U^d)$, and runs in time $O(nd\log U)$. $\square$

Next, we give a simple approach to solve Multiset $\#k$-SUM when all multiplicities are comparably small.

Lemma 4.3 (M-multiplicity $\#k$-SUM$\leq_{\lceil k/2\rceil}$ $\#k$-SUM)**.** *If $\#k$-SUM can be solved in time $T(n)$ then M-multiplicity $\#k$-SUM can be solved in time $\tilde{O}(T(nM^{k-1}))$.*

Proof. Let $A_1,\dots,A_k \subseteq \{U,\dots,-U\}$. We create zero indexed vectors of length k^2+1. Let $B := 3M$. For clarity, the first entry will be denoted by $v[0]$, and the other entries we denote by two-dimensional coordinates $u,w \in \{1,\dots,k\}$, where $v[(u,w)] := v[k(u-1)+w]$. For each $i \in \{1,\dots,k\}$,$a \in A_i, j_1,\dots j_{i-1}, j_{i+1},\dots,j_{k-1} \in \{1,\dots,M\}$ and $j_i \in \{1,\dots,n_a\}$, we add the following vectors with $v[0] = a$ to the newly created set $\hat{A}_i$:

$$v(a, j_1, \dots, j_k)[u, w] = \begin{cases} 0 & \text{if } u = w, \\ j_u & \text{if } u = i \text{ and } u \neq w, \\ B - j_u & \text{if } w = i \text{ and } u \neq w, \\ 0 & \text{else.} \end{cases}$$

For $i \in \{1, \dots, k\}$ and a new constructed vector $v(a, j_1, \dots, j_k)$, we denote $\mathsf{ind}(v(a, j_1, \dots, j_k)) := j_i$. There are $O(n \cdot M^{k-1})$ many vectors in $\hat{A}_1, \dots, \hat{A}_k$. We set $\hat{t} := (t, B, \dots, B)$. Consider now witnesses $(\hat{a_1}, \dots, \hat{a_k}) \in \hat{A}_1 \times \dots \hat{A}_k$ such that $\hat{a_1} + \dots + \hat{a_k} = \hat{t}$. By construction, for this witness, for each pair $u, w \in \{1, \dots, k\}$, we have $\mathsf{ind}(\hat{a_u}) = \mathsf{ind}(\hat{a_w})$, and moreover, for each $i \in \{1, \dots, k\}$ the dimensions $(u-1)i + 1, \dots, (u-1)i + k$ lead to the fact that $\mathsf{ind}(\hat{a_1}) = \dots = \mathsf{ind}(\hat{a_k})$. Due to how the j_i are chosen, a witness $(a_1, \dots, a_k) \in A_1 \times \dots \times A_k$ will correspond to

$$\prod_{i=1}^{k} |\{1, \dots, M\} \cap \{1, \dots, n_{a_i}\}| = \prod_{i=1}^{k} n_{a_i}$$

many witnesses $(\hat{a_1}, \dots, \hat{a_k}) \in \hat{A}_1 \times \dots \times \hat{A}_k$. Finally, convert this $(U, k^2 + 1)$-vector #k-SUM instance into a #k-SUM problem, using Lemma A.5. $\square$

For later purposes, we will need the following version of the above lemma.

Observation 4.4. *If #All-ints 3-SUM can be solved in time $T(n)$ for an $\epsilon > 0$, then we can solve #All-ints M-multiplicity 3-SUM in time $\tilde{O}(T(nM^2))$.*

The same proof as the above can be performed, at the end it just remains to sum up for each a the number of solutions for $v(a, j_1, j_2, j_3)$.

Lemma 4.5. *For odd $k \geq 3$, if there exists an algorithm for the #k-SUM problem running in time $O(n^{\lceil k/2 \rceil - \epsilon})$ for an $\epsilon > 0$, then there exists an algorithm for the Multiset #k-SUM problem running in time $O(n^{\lceil k/2 \rceil - \epsilon'})$ for an $\epsilon' > 0$.*

Proof. We proceed with a heavy-light approach. Assume there exists an $O(n^{\lceil k/2 \rceil - \epsilon})$ algorithm for the #k-SUM problem. Set $c := (k-1)(\lceil k/2 \rceil)$. Firstly, we count the number of solutions $(a_1, \dots, a_k) \in A_1 \times \dots \times A_k$, where $n_{a_1}, \dots, n_{a_k} \leq n^{\epsilon/c}$ using the reduction described in Lemma 4.3. This takes time

$$\begin{aligned} \tilde{O}\left((n \cdot (n^{\epsilon/c})^{k-1})^{\lceil k/2 \rceil - \epsilon}\right) &= \tilde{O}\left(n^{1 + \frac{\epsilon}{\lceil k/2 \rceil}}\right)^{\lceil k/2 \rceil - \epsilon} \\ &= \tilde{O}\left(n^{\lceil k/2 \rceil - \epsilon + \epsilon - \frac{\epsilon^2}{\lceil k/2 \rceil}}\right) \\ &= O\left(n^{\lceil k/2 \rceil - \epsilon'}\right), \end{aligned}$$

where $\epsilon' > 0$. It remains to calculate the number of witnesses $(a_1, \dots, a_k)$, where for at least one $i \in \{1, \dots, k\}$, we have high-multiplicity, meaning $n_{a_i} > n^{\epsilon/c}$ holds. Consider the case that $a_1 \in A_1$ is a high-multiplicity number (the case where $a_i \in A_i$ with $i \neq 1$ is a high-multiplicity number is analogous). For each high-multiplicity number a_1 in A_1 we do the following. Solve the $(k-1)$-SUM instance with sets $A_2, \dots, A_k$ and target $t - a_1$. There are at most $n^{1-(\epsilon/c)}$ many high-multiplicity numbers in A_1, and solving the $(k-1)$-SUM instance takes time $O(n^{(k-1)/2})$. We get a total runtime of

$$\begin{aligned} n^{1-\frac{\epsilon}{c}} \cdot \tilde{O}(n^{(k-1)/2}) &= \tilde{O}(n^{1-(\epsilon/c)+(k-1)/2}) \\ &= \tilde{O}(n^{(k+1)/2-(\epsilon/c)}) \\ &= O(n^{\lceil k/2 \rceil - \epsilon''}), \end{aligned}$$

where $\epsilon'' > 0$, which concludes the proof. □

We can alter the above proof a bit to get the following result.

Observation 4.6. *If #All-ints 3-SUM can be solved in time $O(n^{2-\epsilon})$ for an $\epsilon > 0$, then we can solve #All-ints 3-SUM on multisets in time $O(n^{2-\epsilon'})$ for an $\epsilon' > 0$.*

Proof. For the proof, we basically proceed like the above. We count for each $a_1 \in A_1$ the number of low-multiplicity solutions (multiplicity at most $n^{\frac{\epsilon}{4}}$) using the Observation 4.4 in time $\tilde{O}\left(\left(n \cdot (n^{\frac{\epsilon}{4}})\right)^{2-\epsilon}\right) = \tilde{O}\left(\left(n^{1+\frac{\epsilon}{4}}\right)^{2-\epsilon}\right) = O(n^{2-\epsilon'})$, where $\epsilon' > 0$. Now, we proceed with a brute-force argument. For each $a_1 \in A_1$, which is high-multiplicity(multiplicity higher than $n^{\frac{\epsilon}{4}}$) we create a 2-SUM instance with target $t - a_1$. The witnesses can be counted by a naive algorithm in time $\tilde{O}(n)$. With a simple 2-SUM algorithm, we can count the number of high-multiplicity solutions in time $O(n^{1-\frac{\epsilon}{4}}) \cdot \tilde{O}(n) = O(n^{2-\epsilon''})$, where $\epsilon'' > 0$. □

Theorem 1.2. *Let $k \geq 3$ be odd. If there is $\epsilon > 0$ such that we can count the number of witnesses for k-SUM in time $O(n^{\lceil k/2 \rceil - \epsilon})$, then for all problem P in $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$, there is some $\epsilon' > 0$ such that we can count the number of witnesses for P in time $O(n^{\lceil k/2 \rceil - \epsilon'})$.*

Proof. Consider a general $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ formula. After substituting the free variables, we can assume the formula to be of the following form, by the same arguments used in the proof of the Theorem 1.1.

$$\exists a_1 \in A_1 \ldots \exists a_k \in A_k : \left(\bigvee_{h=1}^{H} \bigwedge_{j=1}^{m} \sum_{i=1}^{k} c_{h,j,i}^T a_i \geq S_{h,j} \right),$$

where $d_1, \ldots, d_k \in \mathbb{N}_{>0}$, $A_1 \subseteq \{-U, \ldots, U\}^{d_1}, \ldots, A_k \subseteq \{-U, \ldots, U\}^{d_k}$ and for $h \in \{1, \ldots, H\}, j \in \{1, \ldots, m\}, i \in \{1, \ldots, k\}$, we have $S_{h,j} \in \mathbb{Z}$, $c_{h,j,i} \in \{-U, \ldots, U\}^{d_i}$.

We first consider the special case consisting of the a single conjunction ($H = 1$), that is

$$\phi = \bigwedge_{j=1}^{m} \sum_{i=1}^{k} c_{j,i}^T a_i \geq S_j.$$

By $\#(\phi)$, we denote the number of witnesses that satisfy the above formula ϕ. Formally, we have

$$\#(\phi) := \# \left\{ (a_1, \ldots, a_k) \in A_1 \times \cdots \times A_k : \bigwedge_{j=1}^{m} \sum_{i=1}^{k} c_{j,i}^T a_i \geq S_j \right\}.$$

By Lemma 3.1, and Observation A.15, we get that $\#\left(\phi\right)$ can be computed by

$$\sum_{\substack{\ell \in \{1, \ldots, \lceil \log_2(U) \rceil\}, \\ b \in \{1, \ldots, k\}}} \#\{(a_1, \ldots, a_k) : f_1^{\phi,l}(a_1) + \cdots + f_k^{\phi,l}(a_k) = g^{\phi,l,W}(S_1, \ldots, S_m), a_1 \in A_1, \ldots, a_k \in A_k\}$$

where $\#\{(a_1,\ldots,a_k) \in A_1 \times \cdots \times A_k : f_1^{\phi,l}(a_1) + \cdots + f_k^{\phi,l}(a_k) = g^{\phi,l,W}(S_1,\ldots,S_m)\}$ is a (M,m)-vector Multiset $\#k$-SUM instance, where M comes from the application of Lemma 3.1. We go through the following chain of transformations to reduce this counting problem to a $\#k$-SUM instance. Firstly, the (M,m)-vector Multiset $\#k$-SUM instance will be transformed into a Multiset $\#k$-SUM instance using Lemma 4.2. Afterwards, we transform the Multiset $\#k$-SUM instances into $\#k$-SUM instances using Lemma 4.5. These yield a sub-$n^{\lceil k/2 \rceil}$ fine-grained reduction that preserve witnesses. This concludes the fine-grained reduction to $\#k$-SUM for the case of a single conjunction.

We turn back to the general formula, which we can now count the witnesses for by a simple application of the inclusion-exclusion principle. Specifically, by inclusion-exclusion, we have

$$\#\left(\bigvee_{h=1}^{H} \underbrace{\bigwedge_{j=1}^{m} \sum_{i=1}^{k} c_{h,j,i}^T a_i \geq S_{h,j}}_{\psi_h}\right) = \sum_{l=1}^{H} (-1)^{l+1} \sum_{\substack{U \subseteq \{1,\ldots,H\},\\ |U|=l}} \#\left(\bigwedge_{h\in U} \psi_h\right).$$

Notice that $\#\left(\bigwedge_{h\in U} \psi_h\right)$ adheres to the special case above and can thus be reduced to $\#k$-SUM. Finally, evaluating the above expression can be done by $O(2^H)$ many calls to the reduction to $\#k$-SUM. This concludes the proof, as in our setting $H = O(1)$ holds. $\square$

By combining the subquadratic equivalence between 3-SUM and #3-SUM (see Theorem A.10 due to Chan et al. [29] in the Appendix) and the above, we get the result:

Corollary 1.3. *For all problems P in $\mathsf{FOP}_{\mathbb{Z}}(\exists^3)$, there is some $\epsilon_P > 0$ such that we can count the number of witnesses for P in randomized time $O(n^{2-\epsilon_P})$ if and only if there is some $\epsilon' > 0$ such that 3-SUM can be solved in randomized time $O(n^{2-\epsilon'})$.*

The above proof can also be adapted for the special case $k = 3$ to count for each $a_1 \in A_1$ the number of witnesses involving a_1, by plugging in the appropriate All-ints versions. Together with the equivalence between #All-ints 3-SUM and 3-SUM of Chan et al. [29], we get

Corollary 4.7. *For all problems P in $\mathsf{FOP}_{\mathbb{Z}}(\exists^3)$, we are able to count for each $a_1 \in A_1$ the number of witnesses involving a_1 in randomized time $O(n^{2-\epsilon})$ for an $\epsilon > 0$, if 3-SUM can be solved in randomized time $O(n^{2-\epsilon'})$ for an $\epsilon' > 0$.*

5 Completeness Theorems for General Quantifier Structures

As Theorem 1.1 establishes 3-SUM as the complete problem for the class $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$, we would like to similarly explore complete problems for other quantifier structures. Let us recall our main geometric problems.

Definition 5.1 (Verification of d-dimensional Pareto Sum)**.** *Given sets $A, B, C \subseteq \mathbb{Z}^d$. Does the set C dominate $A + B$, that is does for all $a \in A, b \in B$ exist a $c \in C$, with $c \geq a + b$?*

It is easy to see that Verification of d-dimensional Pareto Sum is in $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$.

Definition 5.2 (Hausdorff Distance under n Translations)**.** *Given sets $A, B, C \subseteq \mathbb{Z}^d$ with at most n elements and a $\gamma \in \mathbb{N}$, the Hausdorff distance under n Translations problem asks whether the following holds:*

$$\delta_{\vec{H}}^{T(A)}(B,C) := \min_{\tau\in A} \delta_{\vec{H}}(B, C + \{\tau\}) = \min_{\tau\in A} \max_{b\in B} \min_{c\in C} \|b - (c+\tau)\|_\infty \leq \gamma.$$

We show the following result firstly, which allows us to assume without loss of generality a certain normal form.

Lemma 5.3. *Let $Q_1, Q_2 \in \{\exists, \forall\}$. A general $\mathsf{FOP}_{\mathbb{Z}}(Q_1Q_2\exists)$ formula, with input set $A_1 \subseteq \mathbb{Z}^{d_1}, A_2 \subseteq \mathbb{Z}^{d_2}, A_3 \subseteq \mathbb{Z}^{d_3}$, where $|A_1| = |A_2| = |A_3| = n$, can be reduced to the $\mathsf{FOP}_{\mathbb{Z}}(Q_1Q_2\exists)$ formula*

$$Q_1 a_1' \in A_1' Q_2 a_2' \in A_2' \exists a_3' \in A_3' : a_1' + a_2' \leq a_3'$$

in time $O(n)$, where $|A_1'| = |A_2'| = n$ and $|A_3'| = O(n)$.

Proof. After substituting the free variables, by the same arguments used in the proof of the Theorem 1.1, we can rewrite the general $\mathsf{FOP}_{\mathbb{Z}}(Q_1Q_2\exists)$ formula into the form:

$$Q_1 a_1 \in A_1 Q_2 a_2 \in A_2 \exists a_3 \in A_3 : \underbrace{\bigvee_{i=1}^{H} \bigwedge_{j=1}^{m} \alpha_{i,j}^T a_1 + \beta_{i,j}^T a_2 \leq \gamma_{i,j}^T a_3 + S_{i,j}}_{\varphi} .$$

We set

$$A_1' := \left\{ \begin{pmatrix} \alpha_{1,1}^T a_1, \dots, \alpha_{1,m}^T a_1, \dots, \alpha_{H,1}^T a_1, \dots, \alpha_{H,m}^T a_1 \end{pmatrix} : a_1 \in A_1 \right\},$$
$$A_2' := \left\{ \begin{pmatrix} \beta_{1,1}^T a_2, \dots, \beta_{1,m}^T a_2, \dots, \beta_{H,1}^T a_2, \dots, \beta_{H,m}^T a_2 \end{pmatrix} : a_2 \in A_2 \right\}.$$

The set A_3' will be the following set of vectors, which intuitively will allow choices for each disjunct

$$\left\{ \begin{pmatrix} M, \dots, M, \dots, M, \gamma_{i,1}^T a_3 + S_{i,1}, \dots, \gamma_{i,m}^T a_3 + S_{i,m}, M, \dots, M \end{pmatrix} : i \in \{1, \dots, H\}, a_3 \in A_3 \right\}.$$

For the newly built sets, we have $A_1' \subseteq \mathbb{Z}^{H \cdot m}, A_2' \subseteq \mathbb{Z}^{H \cdot m}, A_3' \subseteq \mathbb{Z}^{H \cdot m}$, and $|A_1'| = |A_2'| = n$, whereas $|A_3'| = H \cdot n$. For fixed $(a_1', a_2') \in A_1' \times A_2'$ it is now easy to see that there exists an $a_3' \in A_3'$ such that $a_1' + a_2' \leq a_3'$ if and only if φ holds. We can assume M to be a sufficiently large number that upper bounds any possible sum e.g $M = 2 \cdot \max\{\|a_1'\|_1 + \|a_2'\|_1 : a_1 \in A_1', a_2' \in A_2'\}$.

□

The above lemma immediately gives us complete syntactic problems for our classes. It remains to establish connections between the different quantifier structure classes, and explore natural variants of the syntactic problems in the following.

The syntactic complete problem for the class $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$ turns out to be equivalent to Hausdorff Distance under n Translations. We obtain:

Lemma 5.4 (Hausdorff Distance under n Translations is complete for $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$)**.** *There is a function $\epsilon(d) > 0$ such that Hausdorff Distance under n Translations can be solved in time $O(n^{2-\epsilon(d)})$ if and only if all problems P in $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$ can be solved in time $O(n^{2-\epsilon_P})$ for an $\epsilon_P > 0$.*

Proof. Let $\phi := \exists a \in A \forall b \in B \exists c \in C : \varphi$ a formula in $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$. By Lemma 5.3, we can rewrite ϕ equivalently into the form $\phi' := \exists a' \in A' \forall b' \in B' \exists c' \in C' : a' + b' \leq c'$. Let $M := 2 \cdot \max\{\|a'\|_1 + \|b'\|_1 + \|c'\|_1 : a' \in A', b' \in B', c' \in C'\}$, be a sufficiently large number. Construct $\hat{A} = -A' + \{(M, \dots, M)^T\}$, $\hat{B} = B' + \{(3M, \dots, 3M)^T\}$ and $\hat{C} = C' + \{(M, \dots, M)^T\}$. Thus we have

$$\exists a' \in A' \forall b' \in B' \exists c' \in C' : a' + b' \leq c' \iff \exists a' \in A' \forall b' \in B' \exists c' \in C' : \bigwedge_{i=1}^{d} a[i] + b[i] - c[i] \leq 0$$
$$\iff \exists a' \in A' \forall b' \in B' \exists c' \in C' : \bigwedge_{i=1}^{d} b[i] - (c[i] - a[i]) \leq 0.$$

Then, equivalently

$$\exists a' \in A' \forall b' \in B' \exists c' \in C' : \bigwedge_{i=1}^{d} (b[i] + 3M) - ((c[i] + M) - (a[i] - M)) \leq M \iff$$
$$\min_{\hat{a} \in \hat{A}} \max_{\hat{b} \in \hat{B}} \min_{\hat{c} \in \hat{C}} \left\| \hat{b} - (\hat{c} + \hat{a}) \right\|_{\infty} \leq M.$$

The proof of the membership of Hausdorff Distance under n Translations in $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$ can be found in Example B.6. ☐

Similarly, the Verification of Pareto Sum problem is complete for the class $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$.

Lemma 5.5 (Verification of Pareto Sum is complete for $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$)**.** *Verification of Pareto Sum can be solved in time $O(n^{2-\epsilon})$ for an $\epsilon > 0$ if and only if all problems P in $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$ can be solved in time $O(n^{2-\epsilon'})$ for an $\epsilon' > 0$.*

Proof. Let $\phi := \forall a \in A \forall b \in B \exists c \in C : \varphi$ a formula in $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$ after replacing the free variables. By Lemma 5.3, we can rewrite ϕ equivalently into the form $\phi' := \forall a' \in A' \forall b' \in B' \exists c' \in C' : a' + b' \leq c'$. Clearly, Verification of Pareto Sum is in the class $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$. ☐

5.1 $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists) \to \mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$

We continue with handling the class $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$. By simply making use of Corollary 4.7, one can easily prove that 3-SUM is hard for the class $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$. In the following, we will show a deterministic proof, as Corollary 4.7 makes use of the subquadratic equivalence between 3-SUM and #All-ints 3-SUM, which relies on randomization techniques[21].

Lemma 2.2. *If 3-SUM can be solved in time $O(n^{2-\epsilon})$ for an $\epsilon > 0$, then all problems P of $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$ can be solved in time $O(n^{2-\epsilon_P})$ for an $\epsilon_P > 0$.*

Proof. We devise Algorithm 1 to show that all problems in $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$ can be reduced to 3-SUM in subquadratic time.

Firstly notice that

$$\forall a_1 \in A_1 \exists a_2 \in A_2 \exists a_3 \in A_3 : \bigvee_{h=1}^{H} \bigwedge_{i=1}^{m} \sum_{j=1}^{3} c_{h,i,j}^{T} a_j \geq S_{h,i}$$

holds, if by the commutativity of disjunction and existential quantifier the following holds:

$$\forall a_1 \in A_1 \bigvee_{h=1}^{H} \exists a_2 \in A_2 \exists a_3 \in A_3 : \bigwedge_{i=1}^{m} \sum_{j=1}^{3} c_{h,i,j}^{T} a_j \geq S_{h,i}$$

For each $h \in \{1, \ldots, H\}$, we aim to retrieve the $a_1 \in A_1$, which satisfy

$$\exists a_2 \in A_2 \exists a_3 \in A_3 : \bigwedge_{i=1}^{m} \sum_{j=1}^{3} c_{h,i,j}^{T} a_j \geq S_{h,i}.$$

[21] The equivalence between 3-SUM and #3-SUM has been recently made deterministic after the published version of this paper by Fischer, Jin and Xu [40].

Algorithm 1 $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists) \leq_2$ 3-SUM.

Input: Sets A_1, A_2, A_3 and a linear arithmetic formula (with already substituted free variables)

$$\phi = \bigvee_{h=1}^{H} \bigwedge_{i=1}^{m} \underbrace{\sum_{j=1}^{k} c_{h,i,j}^T a_j \geq S_{h,i}}_{=:\psi} .$$

Output: Whether $\forall a_1 \in A_1, \exists a_2 \in A_2, \exists a_3 \in A_3 : \phi$ holds.

$S \leftarrow \{\}$
for $h \in \{1, \ldots, H\}$ **do**
  for $\ell = (l_1, \ldots, l_m) \in \{1, \ldots, \log_2(M)\}^k, W = (W_1, \ldots, W_m) \in \{1, 2, 3\}^m$ **do**
    $A_1' \leftarrow f_1^{\ell,\psi}(A_1), A_2' \leftarrow f_2^{\ell,\psi}(A_2), A_3' \leftarrow f_3^{\ell,\psi}(A_3), t \leftarrow g^{W,\ell,\psi}(S_1, \ldots, S_m)$ using Lemma 3.1.
    Convert the above vector 3-SUM instance into a 3-SUM instance, with sets A_1'', A_2'', A_3'' using Lemma A.5.
    Call All-ints 3-SUM on A_1'', A_2'', A_3''.
    For each transformed version of $a_1 \in A_1$, which we call $a_1'' \in A_1''$, that is part of a witness (a_1'', a_2'', a_3''), add the original $a_1 \in A_1$ to the set S.
  end for
end for
return $\begin{cases} \text{Yes} & \text{if } S = A_1, \\ \text{No} & \text{else.} \end{cases}$

By Lemma 3.1 the above holds if and only if there exist $\ell = (l_1, \ldots, l_m) \in \{1, \ldots, \log_2(M)\}^3, W = (W_1, \ldots, W_m) \in \{1, 2, 3\}^m$ such that

$$f_1^{\ell,\psi}(a_1) + f_2^{\ell,\psi}(a_2) + f_3^{\ell,\psi}(a_3) = g^{W,\ell,\psi}(S_{h,1}, \ldots, S_{h,m}).$$

Thus, it remains to find all $a_1 \in A_1$ that satisfy the above equation. Clearly these are all the $a_1' \in A_1'$, which satisfy the vector 3-SUM problem instantiated by the sets A_1', A_2', A_3' and t.

We transform the given vector 3-SUM instance, into a 3-SUM instance using Lemma A.5 in time $\tilde{O}(n)$.

By a call to All-ints 3-SUM (which is subquadratic equivalent to 3-SUM [61]), we can retrieve all solutions of this 3-SUM instance.

We remark, that the last transformation to 3-SUM in the algorithm preserve the witnesses, that is (a_1, a_2, a_3) is a witness in $A_1 \times A_2 \times A_3$ if and only if (a_1'', a_2'', a_3'') is a witness in $A_1'' \times A_2'' \times A_3''$. $\square$

In a more general light, the above proof also allows us to determine for each $a \in A_1$ whether it is involved in a solution.

5.2 $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists) \to \mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$

We explore the connection between the problem Additive Sumset Approximation, which is a member of the class $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$, and the 3-SUM problem. The following theorem will play a key role to enable the discovery of the relationship between 3-SUM and other quantifier structures.

Theorem 2.3. *Consider the Additive Sumset Approximation problem of deciding, given $A, B, C \subseteq \mathbb{Z}, t \in \mathbb{Z}$, whether*

$$A + B \subseteq C + \{0, \ldots, t\}.$$

This problem is

- *solvable in time $O(n^{2-\delta})$ with $\delta > 0$, whenever $t = O(n^{1-\epsilon})$ for any $\epsilon > 0$,*
- *not solvable in time $O(n^{2-\epsilon})$, whenever $t = \Omega(n)$ assuming the Strong convolutional 3-SUM hypothesis.*

Furthermore, subquadratic hardness holds under the standard 3-SUM Hypothesis if no restriction on t is made.

Proof. 1. For the first part consider $t = O(n^{1-\epsilon})$, for an $\epsilon > 0$. We make all numbers positive by adding a large constant M to our sets, thus let $A' := A + M$, $B' := B + M$ and $C' := C + 2M$. Let $\hat{C} := C' + [t]$. The set $\hat{C}$, can be computed naively in time $O(n \cdot n^{1-\epsilon}) = O(n^{2-\epsilon})$. To compute the sumset $A' + B'$, we perform a boolean convolution of the characteristic vectors describing A' and B' using the deterministic output-sensitive convolution algorithm of Bringmann et al. [20] (see Theorem A.4 in the Appendix). Specifically, we run the output-sensitive $\tilde{O}(t)$-time algorithm for at most the number of steps required for outputs of size $t = \#\hat{C}$. Thus, it either returns $A' + B'$ correctly, which we then compare against $\hat{C}$, or we stop it with the result that $\#(A' + B')$ exceeds $\#\hat{C}$, and the answer is trivially NO. In both cases, we use time $\tilde{O}(t) = \tilde{O}(n^{2-\epsilon})$.

2. The sumset expression $A + B \subseteq C + [t]$ is equivalent to $\forall a \in A \forall b \in B \exists c : a + b \in \{c\} + [t]$. We reduce in the way such that an original 3-SUM is a YES instance iff the produced sumset expression is a NO instance. Thus, a reduction from 3-SUM needs to somehow certify that $a + b$ cannot be a solution.

   To show our reduction, we reduce from the more structured convolutional 3-SUM problem, which is subquadratic equivalent to 3-SUM. Consider sequences A, B, C, we ask if $\exists i, j, k \in [n-1] : a[i] + b[j] = c[k] \wedge i + j = k$. We construct the following sets where $W := 100U$, where U denotes the maximal number occurring in the sets: $A' = \{a[i] + iW : i \in [n-1]\}, B' = \{b[i] + iW : i \in [n-1]\}, C' = \{c[i] + iW : i \in [n-1]\}$. Hence, we get that for $a' \in A', b' \in B'$, there are $i, j \in [n-1]$, with $i + j \in [n-1]$, such that

$$\begin{aligned} a' + b' &= a[i] + iW + b[j] + jW \\ &= a[i] + b[j] + (i+j)W \in \{(i+j)W, \ldots, (i+j)W + 2U\}. \end{aligned}$$

   Let $c' = c[i + j] + (i + j)W$. By the above property (a', b') is not a witness if and only if $a' + b' \neq c'$. Put differently, $a' + b'$ is in the "complement" set $\{(i+j)W, \ldots, (i+j)W + 2U\} \setminus \{c'\}$. Let $t := 2U$. Due to adjacent "complement" sets being at least $97U$ apart, we equivalently have (a', b') is not a witness if and only if $a' + b' \in \{c' - 1 - t, \ldots, c' - 1\} \cup \{c' + 1, \ldots, c' + t + 1\}$. We can construct the above set for a $c' \in C'$ using two help points $c'_1 := c' - 1 - t, c'_2 := c' + 1$. Let $\hat{C} := \{c'_1 : c' \in C'\} \cup \{c'_2 : c' \in C'\}$. Thus, $\hat{C} + [t]$ is the union of all the sets $I_{c'}$ for $c' \in C'$, and $A' + B' \subseteq \hat{C} + [t]$ is a NO Additive Sumset approximation instance if and only if A, B, C is a YES convolutional 3-SUM instance.

   Note that the above reduction transform any given convolutional 3-SUM instance with universe $U = \Theta(n)$ to an equivalent instance of the problem with $t = 2U = \Theta(n)$. Thus, any algorithm deciding the problem within time $O(n^{2-\epsilon})$ for an $\epsilon > 0$ refutes the strong 3-SUM

Hypothesis. Furthermore, by subquadratic equivalence of 3-SUM and convolutional 3−SUM without any restriction on t in time $O(n^{2-\delta})$ for a $\delta > 0$ would refute the 3-SUM Hypothesis. □

Notice that the above problem is a member of the class $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$. Formally, we can rewrite the Additive Sumset Approximation problem as

$$\forall a \in A \forall b \in B \exists c \in C : c \leq a + b \leq c + t.$$

The reduction from Additive Sumset Approximation to 3-SUM seems nontrivial on the first sight. In Section 6 we will explore a tool, which will give us this reduction.

5.3 Completeness results for the class $\mathsf{FOP}_{\mathbb{Z}}^k$

We turn to combining the above insights to establish (a pair of) complete problems for the class $\mathsf{FOP}_{\mathbb{Z}}$.

Lemma 2.4. *There is a function $\epsilon(d) > 0$ such that the Verification of Pareto Sum problem can be solved in time $O(n^{2-\epsilon(d)})$ if and only if all problems P in the classes*

- $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\exists\exists\exists)$,$\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\forall\forall\forall)$,
- $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\forall\exists\exists)$,$\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\exists\forall\forall)$,
- $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\forall\forall\exists)$,$\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\exists\exists\forall)$,

where $Q_1, \dots Q_{k-3} \in \{\exists, \forall\}$ and $k \geq 3$, can be solved in time $O(n^{k-1-\epsilon_P})$ for an $\epsilon_P > 0$.

Proof. We firstly bruteforce the first $k-3$ quantifiers. It then remains to solve a formula ϕ in $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$, $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$ or $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$. If the formula ϕ is in $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$ then Lemma 5.5 concludes the proof. If the formula ϕ is in $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$ or in $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$, we make use of Theorem 1.1 or Lemma 2.2 to reduce it to a 3-SUM instance. By Theorem 2.3, we can reduce this 3-SUM instance to an instance of Additive Sumset Approximation. Finally, we conclude by the remark that Additive Sumset Approximation is in $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$ and conclude with an application of Lemma 5.5. □

Lemma 2.5. *There is a function $\epsilon(d) > 0$ such that the Hausdorff Distance under n Translations problem can be solved in time $O(n^{2-\epsilon(d)})$ if and only if all problems P in the classes*

- $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\exists\forall\exists)$, $\mathsf{FOP}_{\mathbb{Z}}(Q_1 \dots Q_{k-3}\forall\exists\forall)$,

where $Q_1, \dots Q_{k-3} \in \{\exists, \forall\}$ and $k \geq 3$, can be solved in time $O(n^{k-1-\epsilon_P})$ for an $\epsilon_P > 0$.

Proof. We firstly bruteforce the first $k-3$ quantifiers. By a potential negation argument, it remains to solve a formula ϕ in $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\forall)$. We conclude with an application of Lemma 5.4. □

We can finally turn our attention to a completeness Theorem for the whole class $\mathsf{FOP}_{\mathbb{Z}}^k$.

Theorem 1.4. *There is a function $\epsilon(d) > 0$ such that both of the following problems can be solved in time $O(n^{2-\epsilon(d)})$*

- *Pareto Sum Verification,*
- *Hausdorff distance under n Translations,*

if and only if for each problem P in $\mathsf{FOP}_{\mathbb{Z}}^k$ with $k \geq 3$ there exists an $\epsilon_P > 0$ such that P can be solved in time $O(n^{k-1-\epsilon_P})$.

Proof. The proof is simply an application of Lemma 2.5 or Lemma 2.4, depending on the quantifier structure of the problem P. $\square$

Essentially, these two problems capture the complexity of the class $\mathsf{FOP}_{\mathbb{Z}}^3$ and can be seen as the most important problems in $\mathsf{FOP}_{\mathbb{Z}}^k$.

6 The 3-SUM problem is complete for $\mathsf{FOP}_{\mathbb{Z}}$ formulas with Inequality Dimension at most 3

In this section, we show that 3-SUM captures an interesting subclass of $\mathsf{FOP}_{\mathbb{Z}}$ formulas with arbitrary quantifier structure, namely the formulas of sufficiently small *inequality dimension.* Let us recall the notion of inequality dimension.

Definition 6.1 (Inequality Dimension of a Formula)**.** *Let $\phi = Q_1 x_1 \in A_1, \ldots, Q_k x_k \in A_k : \psi$ be a $\mathsf{FOP}_{\mathbb{Z}}$ formula with $A_i \subseteq \mathbb{Z}^{d_i}$.*

The inequality dimension *of ϕ is the smallest number s such that there exists a Boolean function $\psi' : \{0,1\}^s \to \{0,1\}$ and (strict or non-strict) linear inequalities $L_1, \ldots, L_s$ in the variables $\{x_i[j] : i \in \{1,\ldots,k\}, j \in \{1,\ldots,d_i\}\}$ and the free variables such that $\psi(x_1,\ldots,x_k)$ is equivalent to $\psi'(L_1,\ldots,L_s)$.*

In the following, we look at the class of problems $\mathsf{FOP}_{\mathbb{Z}}^k$ with the restriction of inequality dimension at most 3. We use the following naming convention for boxes.

Definition 6.2. *A d-box in $\mathbb{R}^d$ is the cartesian product of d proper intervals $s_1 \times \cdots \times s_d$, where s_i is an open, closed or half-open interval. We call a cartesian product of only closed intervals a closed box and a cartesian product of only open intervals an open box.*

Given a set of n closed boxes in a set R of $2d$ dimensions, and d-dimensional points $a \in A, b \in B$, we can check in $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists\exists)$ whether $a+b$ lies in one of the boxes as follows:

$$\exists a \in A \exists b \in B \exists r \in R : \bigwedge_{i=1}^{d} r[i] \leq a[i] + b[i] \wedge a[i] + b[i] \leq r[d+i].$$

In fact, we are not limited to closed boxes, if a box is open or half open in a dimension, one can adjust the inequalities in this dimension appropriately.

In order to prove our main theorem in this section, we need to partition the union of n unit cubes in $\mathbb{R}^3$ into pairwise interior- and exterior-disjoint boxes. While Chew et al. [31] studied such a decomposition of unit cubes with the requirement of only interior-disjoint boxes, we need an extension of their result to guarantee disjoint exteriors.

Lemma 6.3 (Disjoint decomposition of the union of cubes in $\mathbb{R}^3$)**.** *Let $\mathcal{C}$ be a set of n axis-aligned congruent cubes in $\mathbb{R}^3$. The union of these cubes, can be decomposed into $O(n)$ boxes whose interiors and exteriors are disjoint in time $O(n \log^2 n)$.*

For a proof, see Section E in the Appendix.

Theorem 6.4. *There is an algorithm deciding 3-SUM in randomized time $O(n^{2-\epsilon})$ for an $\epsilon > 0$ if and only if for each problem P in the classes $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$ and $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$ of inequality dimension at most 3 there exists an $\epsilon' > 0$ such that we can solve P in randomized time $O(n^{2-\epsilon'})$.*

Proof. For the first direction due to Theorem 2.3, we can reduce 3-SUM to an instance of Additive Sumset Approximation,

$$\forall a \in A \forall b \in B \exists c \in C : c \leq a + b \wedge a + b \leq c + t,$$

which has inequality dimension 2. Let us continue with the other direction. Let $\phi := Q_1 a \in A \forall b \in B \exists c \in C : \varphi$, where $Q_1 \in \{\exists, \forall\}$ and φ is a quantifier free linear arithmetic formula with inequality dimension 3. Let $L_1 := \alpha_1^T a + \beta_1^T b \leq \gamma_1^T c + S_1$, $L_2 := \alpha_2^T a + \beta_2^T b \leq \gamma_2^T c + S_2$ and $L_3 := \alpha_3^T a + \beta_3^T b \leq \gamma_3^T c + S_3$ after replacing the free variables. Assume that the formula φ is given in DNF, thus each co-clause has at most 3 atoms, chosen from L_1, L_2, L_3 and their negations. Let

$$A' := \left\{ \begin{pmatrix} \alpha_1^T a \\ \alpha_2^T a \\ \alpha_3^T a \end{pmatrix} : a \in A \right\}, B' := \left\{ \begin{pmatrix} \beta_1^T b \\ \beta_2^T b \\ \beta_3^T b \end{pmatrix} : b \in B \right\}, C' := \left\{ \begin{pmatrix} \gamma_1^T c + S_1 \\ \gamma_2^T c + S_2 \\ \gamma_3^T c + S_3 \end{pmatrix} : c \in C \right\}$$

Thus each co-clause consists of conjunctions of a subset of the following set

$$\{a'[0] + b'[0] \leq c'[0], a'[0] + b'[0] \geq c'[0] + 1, a'[1] + b'[1] \leq c'[1],$$
$$a'[1] + b'[1] \geq c'[1] + 1, a'[2] + b'[2] \leq c'[2], a'[2] + b'[2] \geq c'[2] + 1\}.$$

Let the co-clauses of φ be $V_1, \ldots, V_h$. Thus, we aim to decide a formula of the form:

$$Q_1 a' \in A' \forall b' \in B' \exists c' \in C' : \bigvee_{i=1}^{h} V_i \tag{5}$$

For each co-clause V_i, $i \in \{1, \ldots, h\}$ it holds that V_i is of the form

$$\bigwedge_{k \in V_i^K} L_k \wedge \bigwedge_{j \in V_i^J} \neg L_j,$$

for some $V_i^J, V_i^K \subseteq \{1, 2, 3\}$ and $V_i^J \cap V_i^K = \emptyset$.

Let us consider for each fixed $c' \in C'$ the following possibly empty orthant in $\mathbb{R}^3$.

$$\mathcal{S}(V_i, c') := \{x \in \mathbb{R}^3 : \bigwedge_{k \in V_i^K} x[k] \leq c'[k] \wedge \bigwedge_{j \in V_i^J} x[j] \geq c'[j] + 1\}.$$

By construction, it is immediate that for a fixed c' and $(a', b') \in A' \times B'$ that (a', b', c') fulfill the co-clause V_i if and only if $a' + b' \in \mathcal{S}(V_i, c')$. Thus, equivalently to Equation (5), we ask

$$Q_1 a' \in A' \forall b' \in B' \exists c' \in C' : \bigvee_{i=1}^{h} \left(a' + b' \in S(V_i, c')\right).$$

Having a closer look, $\bigvee_{i=1}^{h} (a' + b' \in S(V_i, c'))$ is true if and only if $a' + b'$ lies in one of the orthants $S(V_i, c')$.

We argue that we may represent the orthant $\mathcal{S}(V_i, c')$ as an appropriately chosen cube in $\mathbb{R}^3$. To this end, let $M := 2 \cdot \max\{\|a\|_1 + \|b\|_1 + \|c\|_1 : a' \in A', b' \in B', c' \in C'\}$ be a sufficiently large number. We can interpret $\mathcal{S}(V_i, c')$ as a cube of the type $\mathcal{C}_{i,c'} = [m_0, m_0'] \times [m_1, m_1'] \times [m_2, m_2']$, where for $u \in \{0, 1, 2\}$, we define:

$$m_u := \begin{cases} -M & u \notin V_i^K, u \notin V_i^J, \\ -2M + c[u] & u \in V_i^K, \\ c[u] + 1 & u \in V_i^J, \end{cases} \qquad m_u' := \begin{cases} M & u \notin V_i^K, u \notin V_i^J, \\ c[u] & u \in V_i^K, \\ 2M + c[u] + 1 & u \in V_i^J. \end{cases}$$

The cubes are axis-aligned and have side length $2M$. Due to the large size of the cube we get for fixed $c' \in C'$ that $a' + b' \in \mathcal{S}(V_i, c')$ if and only if $a' + b'$ lies inside the cube $\mathcal{C}_{i,c'}$.

By Lemma 6.3, we can decompose the collection of cubes $\mathcal{C}_{i,c'}$ for $i \in \{1, \ldots, H\}, c' \in C'$ into $l = O(n)$ disjoint boxes $\mathcal{R} := \{R_1, \ldots, R_l\}$ in time $O(n \log^2 n)$. Let us now go through a case distinction based on the first quantifier.

- If $Q_1 = \forall$, equivalent to ϕ we ask

  $$\forall a' \in A' \forall b' \in B' \exists i \in \{1, \ldots, l\} : a' + b' \text{ lies in } R_i.$$

  By replacing each $i \in \{1, \ldots, l\}$ by a 6-tuple denoting the dimensions of the box R_i, we can reduce counting the number of (a', b', R_i) with $a' + b' \in R_i$ to 3-SUM using Corollary 1.3. Due to the disjointness of the boxes R_i, we know that no (a', b') can be in different boxes $R_i, R_{i'}$ with $i \neq i'$.

  Thus, we can decide our original question by checking whether the number of such witnesses equals $|A'| \cdot |B'|$, concluding the fine-grained reduction to 3-SUM.

- Assume now that $Q_1 = \exists$. Thus, equivalently to ϕ, we ask.

  $$\exists a' \in A' \forall b' \in B' \exists i \in \{1, \ldots, l\} : a' + b' \text{ lies in } R_i.$$

  We can now make use of Corollary 4.7. Count for each $a' \in A'$ the number of *witnesses* (a', b', R_i) with $a' + b' \in R'$. We claim that it remains to check whether there is some a' that is involved in $|B'|$ witnesses. To see this, note that due to the disjointness of the R_i's, for any $a' \in A'$ we have that the number of (b', R_i) with $a' + b' \in R_i$ is equal to the number of b' such that there exists R_i with $a' + b' \in R_i$. Again, the desired reduction to 3-SUM follows. $\square$

We remark that, by [15], we know that the complexity of the union of orthants in $\mathbb{R}^d$ has worst case complexity $O(n^{\lfloor d/2 \rfloor})$. Thus, the above proof does not seem directly generalizable for higher inequality dimensions.

Notice that the decomposition into disjoint boxes, was performed to get a disjointness property. Problems with inherent, disjointness such as problems with an inbuilt convolution constraint can be directly reduced to 3-SUM regardless of the inequality dimension.

Theorem 1.5. *There is an algorithm deciding* 3*-SUM in randomized time* $O(n^{2-\epsilon})$ *for an* $\epsilon > 0$*, if and only if for each problem* P *in* $\mathsf{FOP}_{\mathbb{Z}}^k$ *with* $k \geq 3$ *and inequality dimension at most* 3*, there exists some* $\epsilon' > 0$ *such that we can solve* P *in randomized time* $O(n^{k-1-\epsilon'})$*.*

Proof. We apply a bruteforce search for the first $k - 3$ quantifiers. Thus it remains to solve a $\mathsf{FOP}_{\mathbb{Z}}^3$ formula. By combining Theorem 6.4, Theorem 1.1 and Lemma 2.2 if 3-SUM admits a $O(n^{2-\epsilon})$ time algorithm for an $\epsilon > 0$, then so does every possible $\mathsf{FOP}_{\mathbb{Z}}^k$ formula with inequality dimension 3. Concluding, this gives us an $O(n^{k-1-\epsilon'})$ time algorithm. $\square$

The above theorem gives us immediate reductions to 3-SUM for many seemingly unrelated problem of different quantifier structure and semantic.

For instance, as a direct application of the above theorem we can conclude the equivalence of the Additive Sumset Approximation problem to 3-SUM, together with Theorem 2.3.

Lemma 6.5 (Additive Sumset Approximation $\leq_2$ 3-SUM)**.** *If the* 3*-SUM problem can be solved in (randomized) time* $O(n^{2-\epsilon})$ *for an* $\epsilon > 0$ *then Additive Sumset Approximation problem can be solved in randomized time* $O(n^{2-\epsilon'})$ *for an* $\epsilon' > 0$*.*

Proof. Notice that Additive Sumset Approximation can be rewritten in inequality dimension 2 as $\forall a \in A \forall b \in B \exists c \in C : c \leq a + b \wedge a + b \leq c + t$. Thus, we conclude by an application of Theorem 1.5. $\square$

7 Applications

In this section, we introduce applications of our results. In general, by usage of Theorem 1.1, we get surprisingly simple reductions to the k-SUM problem. While the first two applications can be shown simply by an ad-hoc argument, we make use of our completeness theorems to showcase elegant and simple proofs. Let us see as a first application the following lower bound, which corresponds to a known implicit result in fine-grained complexity theory.

7.1 A lower bound for 4-SUM

As a perhaps surprisingly simple application of Theorem 1.1, we obtain a deterministic proof of the implicit conditional lower bound for 4-SUM from the 3-uniform hyperclique problem.

Theorem 1.6. *If there is some $\epsilon > 0$ such that 4-SUM can be solved in time $O(n^{\frac{4}{3}-\epsilon})$, then the 3-uniform hyperclique hypothesis fails.*

Proof. We firstly model the 3-uniform 4 hyperclique problem as a problem in the class $\mathsf{FOP}_{\mathbb{Z}}(\exists^4)$. Assume the graph to be 4-partite, that is $V = V_1 \dot\cup V_2 \dot\cup V_3 \dot\cup V_4$.

- Let $E_1 \subseteq \mathbb{N}^3$ be the set of edges connecting V_1, V_2, V_3 with edges of the form (v_{1a}, v_{1b}, v_{1c}).
- Let $E_2 \subseteq \mathbb{N}^3$ be the set of edges connecting V_1, V_2, V_4 with edges of the form (v_{2a}, v_{2b}, v_{2d}).
- Let $E_3 \subseteq \mathbb{N}^3$ be the set of edges connecting V_1, V_3, V_4 with edges of the form (v_{3a}, v_{3c}, v_{3d}).
- Let $E_4 \subseteq \mathbb{N}^3$ be the set of edges connecting V_2, V_3, V_4 with edges of the form (v_{4b}, v_{4c}, v_{4d}).

Now the formula will just be

$$\begin{aligned} \exists e_1 \in E_1 \exists e_2 \in E_2 \exists e_3 \in E_3 \exists e_4 \in E_4 : & e_1[1] = e_2[1] \wedge e_2[1] = e_3[1] \\ & \wedge e_1[2] = e_2[2] \wedge e_2[2] = e_4[1] \\ & \wedge e_1[3] = e_3[2] \wedge e_3[2] = e_4[2] \\ & \wedge e_2[3] = e_3[3] \wedge e_3[3] = e_4[3]. \end{aligned}$$

For correctness, we remark that any witness $e_1 \in E_1, \dots, e_4 \in E_4$ for the above formula consistently chooses the same vertex $v_i \in V_i$ for each i. Thus, $\{v_1, \dots, v_4\}$ forms a 4-hyperclique, as all 4 connecting edges are present, as witnessed by $e_1, \dots, e_4$. Conversely, any 4-hyperclique $\{v_1, \dots, v_4\}$ with $v_i \in V_i$ yields the witness $(v_1, v_2, v_3) \in E_1, (v_1, v_2, v_4) \in E_2, (v_1, v_3, v_4) \in E_3$, and $(v_2, v_3, v_4) \in E_4$ in the above formula.

The sets E_1, E_2, E_3, E_4 will be of size $O(n^3)$, where n denotes the number of vertices. Assuming there was an algorithm of runtime $O(n^{4/3-\epsilon})$ for the 4-SUM problem, then by Theorem 1.1, we would have an algorithm for the class $\mathsf{FOP}_{\mathbb{Z}}(\exists^4)$ in time $\tilde{O}(n^{4/3-\epsilon})$. Finally, through the reduction above, we get an algorithm in runtime $\tilde{O}(n^{4-\epsilon/3})$ for deciding the 3-uniform 4 hyperclique problem, refuting the hypothesis. $\square$

7.2 A lower bound for 3-SUM

In similar spirit to Example B.7 in the Appendix, we can also give a direct lower bound for 3-SUM from the Exact Triangle problem, which seeks to find a triangle in an undirected graph, whose edge weights sum up to a certain value t. Formally, for an undirected graph $G = (V, E)$ with a cost function $c : E \to \mathbb{Z}$ on the edges and a value $t \in \mathbb{Z}$, we ask if there exist distinct vertices $v_1, v_2, v_3 \in V$ such that $c(v_1, v_2) + c(v_2, v_3) + c(v_3, v_1) = t$ holds.[60]. While the corresponding lower bound is known in the literature, by a reduction from BMM [48, 61], we still can give a simple direct proof.

Lemma 7.1. *Let $\epsilon > 0$, there is no algorithm solving 3-SUM in time $O(n^{1.5-\epsilon})$ if there is no algorithm solving the Exact Triangle problem in time $O(n^{3-\epsilon'})$ for an $\epsilon' > 0$.*

Proof. We show how to find an exact triangle in a directed graph. Let E be the set of edges in a directed graph. Each edge has a unique id and consider α, ω to be functions that denote the start and endpoint of an edge respectively, furthermore let $c(e)$ denote the cost of an edge. Thus $E' = \{(e_{id}, \alpha(e), \omega(e), c(e)) : e \in E\}$. To detect an exact triangle we can simply ask

$$\begin{aligned}\exists e_1 \in E \exists e_2 \in E \exists e_3 \in E : &e_1[0] \neq e_2[0] \wedge e_2[0] \neq e_3[0] \wedge e_1[0] \neq e_3[0] \wedge \\ &e_1[2] = e_2[1] \wedge e_2[2] = e_3[1] \wedge e_3[2] = e_1[1] \wedge \\ &e_1[3] + e_2[3] + e_3[3] = t.\end{aligned}$$

The set E is of at most quadratic size, thus an $O(n^{1.5-\epsilon})$ time algorithm with an $\epsilon > 0$ would imply an $O(n^{3-\epsilon'})$ time algorithm for exact triangle for some $\epsilon' > 0$. □

7.3 A lower bound on the computation of Pareto Sums

In the following, we explore how the 3-SUM hardness of Verification of Pareto Sum translates to a hardness result for the problem of computing Pareto Sums. Let us firstly justify the naming of the Verification of Pareto Sum problem, by showing it to be subquadratic equivalent to the more natural extended version of Verification of Pareto Sum. Throughout this section, we consider dimensions $d \geq 2$.

Definition 7.2 (Verification of Pareto Sum (Extended version))**.** *Given sets $A, B, C \subseteq \mathbb{Z}^d$. Do the following properties hold simultaneously:*

- *(Inclusion): $C \subseteq A + B$,*
- *(Dominance): C dominates $A + B$. More formally, for every $a \in A, b \in B$ there exists $c \in C$ with $c \geq a + b$.*
- *(Minimality): There are no $c, c' \in C$ with $c \neq c'$ and $c \leq c'$.*

We make use of the following lemma and its construction for the results in this section.

Lemma 7.3. *Given sets $A, B, C \subseteq \mathbb{Z}^d$ of size at most n, one can construct sets $\tilde{A}, \tilde{B}, \tilde{C} \subseteq \mathbb{Z}^d$ of size $\Theta(n)$ in time $\tilde{O}(n)$ such that (1) $\tilde{A}, \tilde{B}, \tilde{C}$ always satisfy the minimality and inclusion condition and (2) $\tilde{A}, \tilde{B}, \tilde{C}$ fulfill the dominance condition if and only if A, B, C fulfill the dominance condition.*

Proof. Let M be a sufficiently large number, that is $M := 2 \cdot \max\{\|a\|_1 + \|b\|_1 + \|c\|_1 : a \in A, b \in B, c \in C\}$. For ease of notation, we assume the dimension to be even, if the dimension is odd the same construction where we omit the last dimension will work. We construct from the sets A, B, C the following sets:

$$\tilde{A} := A \cup \left\{ c + \begin{pmatrix} 2M \\ -2M \\ \vdots \\ 2M \\ -2M \end{pmatrix} : c \in C \right\} \cup \left\{ \begin{pmatrix} 6M \\ -3M \\ \vdots \\ 6M \\ -3M \end{pmatrix} \right\},$$

$$\tilde{B} := B \cup \left\{ \begin{pmatrix} -2M \\ +2M \\ \vdots \\ -2M \\ +2M \end{pmatrix} \right\}.$$

The following are possible members of the pareto sum $\tilde{A} + \tilde{B}$:

1. $a + b$, where $a \in A, b \in B$,
2. $a + (-2M, +2M, \dots, -2M, +2M)^T$, where $a \in A$,
3. $c + (2M, -2M, \dots, +2M, -2M)^T + b$, where $c \in C, b \in B$,
4. $c + (2M, -2M, \dots, +2M, -2M)^T + (-2M, +2M, \dots, -2M, +2M)^T = c$, where $c \in C$,
5. $b + (6M, -3M, \dots, 6M, -3M)^T$, where $b \in B$,
6. $(6M, -3M, \dots, 6M, -3M)^T + (-2M, +2M, \dots, -2M, +2M)^T = (4M, -M, \dots, 4M, -M)^T$.

Notice that, the elements of type 3 are dominated by the element of type 6. The elements 2, 4, 5, 6 are incomparable.

We let C' be the union of elements of type 2, 4, 5, 6. and let $\tilde{C}$ be the minimal elements of C', which can be found in time $\tilde{O}(n)$ by [43]. Furthermore, the inclusion property holds trivially as $\tilde{C}$ is chosen from elements of the sumset $\tilde{A} + \tilde{B}$.

We remark at this point that, minimizing does not affect the dominance condition or the inclusion condition.

For the correctness assume A, B, C fulfill the dominance condition, then all elements of type 1 are dominated by some elements of type 4. Thus $\tilde{A}, \tilde{B}$ and $\tilde{C}$ fufill the dominance condition.

On the contrary assume that for some $a \in A, b \in B$ that $a + b$ is truly dominates all $c \in C$, then an element of type 1 would need to be in $\tilde{C}$ for the dominance condition to hold. ☐

Lemma 7.4. *There is an $O(n^{2-\epsilon})$ time algorithm for an $\epsilon > 0$ for Verification of Pareto Sum (Extended Version) if and only if there is an $O(n^{2-\epsilon'})$ time algorithm for an $\epsilon' > 0$ for Verification of Pareto Sum.*

Proof. For the first direction, let us assume we are given an instance of Verification of Pareto Sum (Extended Version). Thus, we ask whether C fulfills the following 3 criteria.

1. $\forall a \in A \forall b \in B \exists c \in C : a + b \leq c$,
2. $\forall c \in C \exists a \in A \exists b \in B : a + b = c$,

3. $\neg \exists c \in C \exists c' \in C' : c \neq c' \wedge c \leq c'$.

Requirement 1 can be immediately checked by a call to Verification of Pareto Sum. Requirement 2 is in $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$, thus can be reduced to 3-SUM by Lemma 2.2. This 3-SUM instance can be reduced to an instance of Additive Sumset approximation by Theorem 2.3, which is in $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$. By the completeness of Verification of Pareto Sum for the class $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$, we can conclude by a call to Verification of Pareto Sum. Requirement 3 is known to be decidable in time $\tilde{O}(n)$ by orthogonal range tree methods [43].

For the other direction, assume we are given an instance of Verification of Pareto Sum, we simply conclude by an application of Lemma 7.3. $\square$

Thus, for subquadratic reductions, we can restrict ourselves to the Verification of Pareto Sum problem, which essentially only checks the dominance condition.

Let us now consider the natural problem of computing the Pareto Sum.

Definition 7.5 (Pareto Sum)**.** *Given sets $A, B \subseteq \mathbb{Z}^d$, compute a set $C \subseteq \mathbb{Z}$, such that A, B, C satisfy the Inclusion, Dominance and Minimality condition.*

In the following, we argue why the lower bounds to Verification of Pareto Sum translate to lower bounds to computation of the Pareto Sum. Formally, we prove:

Lemma 7.6. *If there is an output sensitive algorithm to compute the Pareto Sum of sets $A, B \subseteq \mathbb{Z}^d$ in time $O(n^{2-\epsilon})$ for an $\epsilon > 0$, then one can also decide Verification of Pareto Sum of sets A, B, C in time $O(n^{2-\epsilon'})$ for an $\epsilon' > 0$.*

Proof. We make use of Lemma 7.3. Construct the sets $\tilde{A}, \tilde{B}$ as in Lemma 7.3.

We then compute the Pareto Sum of $\tilde{A}, \tilde{B}$ by using an output sensitive algorithm, where we stop the output after $|A| + |C| + |B| + 2$ many steps. Let C' be defined as in Lemma 7.3. If $|C'| > |A| + |C| + |B| + 1$, we conclude that the Verification of Pareto Sum instance is a NO instance.

Now we argue as in Lemma 7.3, the elements in the set C' are only of the type $2, 4, 5, 6$ if and only if A, B, C satisfy the dominance condition, which is equivalent to saying A, B, C are a YES instance to Verification of Pareto Sum. $\square$

We conclude the section with our resulting hardness results for computing Pareto Sums.

Theorem 1.7 (Pareto Sum Computation Lower Bound)**.** *The following conditional lower bounds hold for output-sensitive Pareto sum computation:*

1. *If there is $\epsilon > 0$ such that we can compute the Pareto sum C of $A, B \subseteq \mathbb{Z}^2$, whenever C is of size $\Theta(n)$, in time $O(n^{2-\epsilon})$, then the 3-SUM hypothesis fails (thus, for any $\mathsf{FOP}_{\mathbb{Z}}^k$ formula ϕ of inequality dimension at most 3, there is $\epsilon' > 0$ such that ϕ can be decided in time $O(n^{k-1-\epsilon'})$).*

2. *If for all $d \geq 2$, there is $\epsilon > 0$ such that we can compute the Pareto sum C of $A, B \subseteq \mathbb{Z}^d$, whenever C is of size $\Theta(n)$, in time $O(n^{2-\epsilon})$, then there is some $\epsilon' > 0$ such that we can decide all $\mathsf{FOP}_{\mathbb{Z}}$ formulas with k quantifiers not ending in $\exists\forall\exists$ or $\forall\exists\forall$ in time $O(n^{k-1-\epsilon'})$.*

Proof.

1. Combine the 3-SUM hardness of Verification of Pareto Sum (Lemma 2.4) together with Lemma 7.6.

2. Consider a formula ϕ in $\mathsf{FOP}_{\mathbb{Z}}^k$ not ending in $\exists\forall\exists$ or $\forall\exists\forall$. We brute-force over the first $k-3$ quantifiers. Thus, it remains to solve a formula in $\mathsf{FOP}_{\mathbb{Z}}^3$(with a different quantifier structure than $\exists\forall\exists$ or $\forall\exists\forall$), which can be reduced to Verification of Pareto Sum. Finally, conclude by an application of Lemma 7.6.

$\square$

8 Future Work

While we exhibit a pair of problems that is complete for the class $\mathsf{FOP}_{\mathbb{Z}}$, one could still ask whether there is a subquadratic reduction from Hausdorff distance under n Translations to Verification of Pareto Sum. As a result there would be a single complete problem (or rather the canonical multidimensional family of a single geometric problem) for $\mathsf{FOP}_{\mathbb{Z}}$.

Is Verification of Pareto Sum complete for the class $\mathsf{FOP}_{\mathbb{Z}}$*?*

Interestingly, previous completeness theorems [45] were able to establish a problem of quantifier structure $\forall\forall\exists$ (the Orthogonal Vectors problem) as complete by making use of a technique in [61] that was originally used to show subcubic equivalence between All-Pairs Negative Triangle and Negative Triangle. However, a major problem we encounter is that while the third quantifier in the Orthogonal Vectors problem ranges over a sparse (intuitively: subpolynomially sized) domain (i.e., the dimensions of the vectors), the third quantifier in Pareto Sum Verification ranges over a linearly sized domain (i.e., the set C).

Finally, we ask if our 3-SUM completeness result for arbitrary quantifier structures can be improved upon.

Can we establish a $d > 3$ *such that* 3*-SUM is complete for* $\mathsf{FOP}_{\mathbb{Z}}$ *formulas of inequality dimension at most* d*?*

Acknowledgements

The authors thank the ITCS reviewers for the constructive feedback as well as Karl Bringmann and Nick Fischer for helpful discussions.

References

[1] Amir Abboud, Arturs Backurs, Karl Bringmann, and Marvin Künnemann. Fine-grained complexity of analyzing compressed data: Quantifying improvements over decompress-and-solve. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 192–203. IEEE Computer Society, 2017. `doi:10.1109/FOCS.2017.26`.

[2] Amir Abboud, Arturs Backurs, Karl Bringmann, and Marvin Künnemann. Impossibility results for grammar-compressed linear algebra. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL: `https://proceedings.neurips.cc/paper/2020/hash/645e6bfdd05d1a69c5e47b20f0a91d46-Abstract.html`.

[3] Amir Abboud, Karl Bringmann, Seri Khoury, and Or Zamir. Hardness of approximation in p via short cycle removal: cycle detection, distance oracles, and beyond. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1487–1500. ACM, 2022. `doi:10.1145/3519935.3520066`.

[4] Amir Abboud and Kevin Lewi. Exact weight subgraphs and the k-sum conjecture. In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, volume 7965 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2013. doi:10.1007/978-3-642-39206-1_1.

[5] Amir Abboud, Kevin Lewi, and Ryan Williams. Losing weight by gaining edges. In Andreas S. Schulz and Dorothea Wagner, editors, *Algorithms - ESA 2014 - 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings*, volume 8737 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2014. doi:10.1007/978-3-662-44777-2_1.

[6] Amir Abboud and Virginia Vassilevska Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 434–443. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.53.

[7] Pankaj K. Agarwal and Alex Steiger. An output-sensitive algorithm for computing the union of cubes and fat boxes in 3d. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 10:1–10:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. URL: https://doi.org/10.4230/LIPIcs.ICALP.2021.10, doi:10.4230/LIPICS.ICALP.2021.10.

[8] Josh Alman, Timothy M. Chan, and R. Ryan Williams. Polynomial representations of threshold functions and algorithmic applications. In Irit Dinur, editor, *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. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.57.

[9] Stephen Alstrup, Gerth Stølting Brodal, and Theis Rauhe. New data structures for orthogonal range searching. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 198–207. IEEE Computer Society, 2000. doi:10.1109/SFCS.2000.892088.

[10] Amihood Amir, Timothy M. Chan, Moshe Lewenstein, and Noa Lewenstein. On hardness of jumbled indexing. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, volume 8572 of *Lecture Notes in Computer Science*, pages 114–125. Springer, 2014. doi:10.1007/978-3-662-43948-7_10.

[11] Haozhe An, Mohit Gurumukhani, Russell Impagliazzo, Michael Jaber, Marvin Künnemann, and Maria Paula Parga Nina. The fine-grained complexity of multi-dimensional ordering properties. *Algorithmica*, 84(11):3156–3191, 2022. URL: https://doi.org/10.1007/s00453-022-01014-x, doi:10.1007/S00453-022-01014-X.

[12] Christian Artigues, Marie-José Huguet, Fallou Gueye, Frédéric Schettini, and Laurent Dezou. State-based accelerations and bidirectional search for bi-objective multi-modal shortest paths. *Transportation Research Part C: Emerging Technologies*, 27:233–259, 2013.

[13] Arturs Backurs, Piotr Indyk, and Ludwig Schmidt. Better approximations for tree sparsity in nearly-linear time. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2215–2229. SIAM, 2017. `doi:10.1137/1.9781611974782.145`.

[14] Stephen A Bloch, Jonathan F Buss, and Judy Goldsmith. How hard are n 2-hard problems? *ACM SIGACT News*, 25(2):83–85, 1994.

[15] Jean-Daniel Boissonnat, Micha Sharir, Boaz Tagansky, and Mariette Yvinec. Voronoi diagrams in higher dimensions under certain polyhedral distance functions. *Discret. Comput. Geom.*, 19(4):485–519, 1998. `doi:10.1007/PL00009366`.

[16] Karl Bringmann, Alejandro Cassis, Nick Fischer, and Marvin Künnemann. Fine-grained completeness for optimization in P. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPIcs*, pages 9:1–9:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. URL: `https://doi.org/10.4230/LIPIcs.APPROX/RANDOM.2021.9`, `doi:10.4230/LIPICS.APPROX/RANDOM.2021.9`.

[17] Karl Bringmann, Alejandro Cassis, Nick Fischer, and Marvin Künnemann. A structural investigation of the approximability of polynomial-time problems. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 30:1–30:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. URL: `https://doi.org/10.4230/LIPIcs.ICALP.2022.30`, `doi:10.4230/LIPICS.ICALP.2022.30`.

[18] Karl Bringmann, Nick Fischer, and Marvin Künnemann. A fine-grained analogue of schaefer's theorem in P: dichotomy of exists^k-forall-quantified first-order graph properties. In Amir Shpilka, editor, *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*, volume 137 of *LIPIcs*, pages 31:1–31:27. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. URL: `https://doi.org/10.4230/LIPIcs.CCC.2019.31`, `doi:10.4230/LIPICS.CCC.2019.31`.

[19] Karl Bringmann, Nick Fischer, and Vasileios Nakos. Sparse nonnegative convolution is equivalent to dense nonnegative convolution. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1711–1724. ACM, 2021. `doi:10.1145/3406325.3451090`.

[20] Karl Bringmann, Nick Fischer, and Vasileios Nakos. Deterministic and las vegas algorithms for sparse nonnegative convolution. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 3069–3090. SIAM, 2022. `doi:10.1137/1.9781611977073.119`.

[21] Karl Bringmann, Ahmed Ghazy, and Marvin Künnemann. Exploring the approximability landscape of 3sum. In Timothy M. Chan, Johannes Fischer, John Iacono, and Grzegorz Herman, editors, *32nd Annual European Symposium on Algorithms, ESA 2024, September 2-4, 2024, Royal Holloway, London, United Kingdom*, volume 308 of *LIPIcs*, pages 34:1–34:15.

Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. URL: `https://doi.org/10.4230/LIPIcs.ESA.2024.34`, `doi:10.4230/LIPICS.ESA.2024.34`.

[22] Karl Bringmann and Vasileios Nakos. Fast n-fold boolean convolution via additive combinatorics. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 41:1–41:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. URL: `https://doi.org/10.4230/LIPIcs.ICALP.2021.41`, `doi:10.4230/LIPICS.ICALP.2021.41`.

[23] Karl Bringmann and Vasileios Nakos. A fine-grained perspective on approximating subset sum and partition. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 1797–1815. SIAM, 2021. `doi:10.1137/1.9781611976465.108`.

[24] Karl Bringmann and André Nusser. Translating hausdorff is hard: Fine-grained lower bounds for hausdorff distance under translation. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational Geometry, SoCG 2021, June 7-11, 2021, Buffalo, NY, USA (Virtual Conference)*, volume 189 of *LIPIcs*, pages 18:1–18:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. URL: `https://doi.org/10.4230/LIPIcs.SoCG.2021.18`, `doi:10.4230/LIPICS.SOCG.2021.18`.

[25] 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 Madhu Sudan, editor, *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 261–270. ACM, 2016. `doi:10.1145/2840728.2840746`.

[26] Timothy M. Chan. Minimum l_∞ hausdorff distance of point sets under translation: Generalizing klee's measure problem. In Erin W. Chambers and Joachim Gudmundsson, editors, *39th International Symposium on Computational Geometry, SoCG 2023, June 12-15, 2023, Dallas, Texas, USA*, volume 258 of *LIPIcs*, pages 24:1–24:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. URL: `https://doi.org/10.4230/LIPIcs.SoCG.2023.24`, `doi:10.4230/LIPICS.SOCG.2023.24`.

[27] Timothy M. Chan and Moshe Lewenstein. Clustered integer 3sum via additive combinatorics. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 31–40. ACM, 2015. `doi:10.1145/2746539.2746568`.

[28] Timothy M. Chan, Virginia Vassilevska Williams, and Yinzhan Xu. Hardness for triangle problems under even more believable hypotheses: reductions from real apsp, real 3sum, and OV. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1501–1514. ACM, 2022. `doi:10.1145/3519935.3520032`.

[29] Timothy M. Chan, Virginia Vassilevska Williams, and Yinzhan Xu. Fredman's trick meets dominance product: Fine-grained complexity of unweighted apsp, 3sum counting, and more. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 419–432. ACM, 2023. `doi:10.1145/3564246.3585237`.

[30] Ruiwen Chen and Rahul Santhanam. Improved algorithms for sparse MAX-SAT and max-k-csp. In Marijn Heule and Sean A. Weaver, editors, *Theory and Applications of Satisfiability Testing - SAT 2015 - 18th International Conference, Austin, TX, USA, September 24-27, 2015, Proceedings*, volume 9340 of *Lecture Notes in Computer Science*, pages 33–45. Springer, 2015. doi:10.1007/978-3-319-24318-4_4.

[31] L. Paul Chew, Dorit Dor, Alon Efrat, and Klara Kedem. Geometric pattern matching in d -dimensional space. *Discret. Comput. Geom.*, 21(2):257–274, 1999. doi:10.1007/PL00009420.

[32] L Paul Chew and Klara Kedem. Improvements on geometric pattern matching problems. In *Algorithm Theory—SWAT'92: Third Scandinavian Workshop on Algorithm Theory Helsinki, Finland, July 8–10, 1992 Proceedings 3*, pages 318–325. Springer, 1992.

[33] L. Paul Chew and Klara Kedem. Improvements on geometric pattern matching problems. In Otto Nurmi and Esko Ukkonen, editors, *Algorithm Theory - SWAT '92, Third Scandinavian Workshop on Algorithm Theory, Helsinki, Finland, July 8-10, 1992, Proceedings*, volume 621 of *Lecture Notes in Computer Science*, pages 318–325. Springer, 1992. doi:10.1007/3-540-55706-7_28.

[34] Richard Cole and Ramesh Hariharan. Verifying candidate matches in sparse and wildcard matching. In John H. Reif, editor, *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 592–601. ACM, 2002. doi:10.1145/509907.509992.

[35] Marek Cygan, Marcin Mucha, Karol Wegrzycki, and Michal Wlodarczyk. On problems equivalent to (min, +)-convolution. *ACM Trans. Algorithms*, 15(1):14:1–14:25, 2019. doi:10.1145/3293465.

[36] Holger Dell, Marc Roth, and Philip Wellnitz. Counting answers to existential questions. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 113:1–113:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. URL: https://doi.org/10.4230/LIPIcs.ICALP.2019.113, doi:10.4230/LIPICS.ICALP.2019.113.

[37] Bartlomiej Dudek, Pawel Gawrychowski, and Tatiana Starikovskaya. All non-trivial variants of 3-ldt are equivalent. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 974–981. ACM, 2020. doi:10.1145/3357713.3384275.

[38] Matthias Ehrgott and Xavier Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR-spektrum*, 22:425–460, 2000.

[39] Jeff Erickson. New lower bounds for convex hull problems in odd dimensions. *SIAM J. Comput.*, 28(4):1198–1214, 1999. doi:10.1137/S0097539797315410.

[40] Nick Fischer, Ce Jin, and Yinzhan Xu. New applications of 3sum-counting in fine-grained complexity and pattern matching, 2024. To appear in SODA 2025. URL: https://arxiv.org/abs/2410.20764, arXiv:2410.20764.

[41] Nick Fischer, Marvin Künnemann, and Mirza Redzic. The effect of sparsity on k-dominating set and related first-order graph properties. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 4704–4727. SIAM, 2024. `doi:10.1137/1.9781611977912.168`.

[42] Daniel Funke, Demian Hespe, Peter Sanders, Sabine Storandt, and Carina Truschel. Pareto sums of pareto sets: Lower bounds and algorithms, 2024. URL: `https://arxiv.org/abs/2409.10232`, `arXiv:2409.10232`.

[43] Harold N. Gabow, Jon Louis Bentley, and Robert Endre Tarjan. Scaling and related techniques for geometry problems. In Richard A. DeMillo, editor, *Proceedings of the 16th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1984, Washington, DC, USA*, pages 135–143. ACM, 1984. `doi:10.1145/800057.808675`.

[44] Anka Gajentaan and Mark H. Overmars. On a class of o(n2) problems in computational geometry. *Comput. Geom.*, 5:165–185, 1995. `doi:10.1016/0925-7721(95)00022-2`.

[45] Jiawei Gao, Russell Impagliazzo, Antonina Kolokolova, and Ryan Williams. Completeness for first-order properties on sparse structures with algorithmic applications. *ACM Trans. Algorithms*, 15(2):23:1–23:35, 2019. `doi:10.1145/3196275`.

[46] Demian Hespe, Peter Sanders, Sabine Storandt, and Carina Truschel. Pareto sums of pareto sets. In Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman, editors, *31st Annual European Symposium on Algorithms, ESA 2023, September 4-6, 2023, Amsterdam, The Netherlands*, volume 274 of *LIPIcs*, pages 60:1–60:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. URL: `https://doi.org/10.4230/LIPIcs.ESA.2023.60`, `doi:10.4230/LIPICS.ESA.2023.60`.

[47] Daniel P. Huttenlocher and Klara Kedem. Computing the minimum hausdorff distance for point sets under translation. In Raimund Seidel, editor, *Proceedings of the Sixth Annual Symposium on Computational Geometry, Berkeley, CA, USA, June 6-8, 1990*, pages 340–349. ACM, 1990. `doi:10.1145/98524.98599`.

[48] Zahra Jafargholi and Emanuele Viola. 3sum, 3xor, triangles. *Algorithmica*, 74(1):326–343, 2016. URL: `https://doi.org/10.1007/s00453-014-9946-9`, `doi:10.1007/S00453-014-9946-9`.

[49] Tsvi Kopelowitz, Seth Pettie, and Ely Porat. Higher lower bounds from the 3sum conjecture. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1272–1287. SIAM, 2016. URL: `https://doi.org/10.1137/1.9781611974331.ch89`, `doi:10.1137/1.9781611974331.CH89`.

[50] Daniel Kroening and Ofer Strichman. *Decision procedures*. Springer, 2016.

[51] Marvin Künnemann. A tight (non-combinatorial) conditional lower bound for klee's measure problem in 3d. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 555–566. IEEE, 2022. `doi:10.1109/FOCS54457.2022.00059`.

[52] Andrea Lincoln, Virginia Vassilevska Williams, Joshua R. Wang, and R. Ryan Williams. Deterministic time-space trade-offs for k-sum. In Ioannis Chatzigiannakis, Michael Mitzenmacher,

Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 58:1–58:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. URL: https://doi.org/10.4230/LIPIcs.ICALP.2016.58, doi:10.4230/LIPICS.ICALP.2016.58.

[53] Andrea Lincoln, Virginia Vassilevska Williams, and R. Ryan Williams. Tight hardness for shortest cycles and paths in sparse graphs. In Artur Czumaj, editor, *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. SIAM, 2018. doi:10.1137/1.9781611975031.80.

[54] Thibaut Lust and Daniel Tuyttens. Variable and large neighborhood search to solve the multiobjective set covering problem. *Journal of Heuristics*, 20:165–188, 2014.

[55] André Nusser. *Fine-grained complexity and algorithm engineering of geometric similarity measures*. PhD thesis, Saarland University, Saarbrücken, Germany, 2021. URL: https://publikationen.sulb.uni-saarland.de/handle/20.500.11880/33904.

[56] Mihai Puatracscu. Towards polynomial lower bounds for dynamic problems. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 603–610. ACM, 2010. doi:10.1145/1806689.1806772.

[57] Britta Schulze, Kathrin Klamroth, and Michael Stiglmayr. Multi-objective unconstrained combinatorial optimization: a polynomial bound on the number of extreme supported solutions. *Journal of Global Optimization*, 74(3):495–522, 2019.

[58] Carola Wenk. *Shape matching in higher dimensions*. PhD thesis, Free University of Berlin, Dahlem, Germany, 2003. URL: http://www.diss.fu-berlin.de/2003/151/index.html.

[59] Ryan Williams. Faster decision of first-order graph properties. In Thomas A. Henzinger and Dale Miller, editors, *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 80:1–80:6. ACM, 2014. doi:10.1145/2603088.2603121.

[60] Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the international congress of mathematicians: Rio de janeiro 2018*, pages 3447–3487. World Scientific, 2018.

[61] 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. IEEE Computer Society, 2010. doi:10.1109/FOCS.2010.67.

[62] Virginia Vassilevska Williams and Ryan Williams. Finding, minimizing, and counting weighted subgraphs. *SIAM J. Comput.*, 42(3):831–854, 2013. doi:10.1137/09076619X.

A Preliminaries

Let us first state problems and hypotheses we work with, starting with the k-SUM problem.

Definition A.1 (The k-SUM problem). *Given k finite sets of at most n integers $A_1, A_2, \ldots, A_k \subseteq \{-n^k, \ldots, 0, \ldots, n^k\}$, and an integer t, determine if $\exists a_1 \in A_1 \exists a_2 \in A_2 \ldots \exists a_k \in A_k : \sum_{i=1}^{k} a_i = t$.*

There is a simple meet-in-the-middle approach to solve the above problem in time $O(n^{\lceil k/2 \rceil + o(1)})$. It is widely believed that this is optimal, as stated in the following hypothesis.

Hypothesis A.2 (The k-SUM Hypothesis). *Let $k \geq 3$. There is no algorithm solving the k-SUM problem in time $O(n^{\lceil \frac{k}{2} \rceil - \epsilon})$ for any $\epsilon > 0$.*

For discussion of its plausibility, we refer to the survey [60]. We work in the standard word RAM model with words of $O(\log(n))$ bits. As we aim to relate runtime between quadratic time problems, we need a different notion of reduction than those used to prove NP completeness.

Fine-grained reductions were first introduced for subcubic runtimes in [61]. A general definition can be found in [60]. In this paper, the following definition of fine-grained reduction, also used in [16] suffices. A more detailed description on fine-grained reductions can be found in [25].

Definition A.3 (Fine-grained reductions). *Consider problems P_1, P_2 with presumed time complexities T_1, T_2, respectively. A fine-grained reduction from P_1 to P_2 is an algorithm $\mathcal{A}$ with oracle access to P_2. Whenever $\mathcal{A}$ uses an $O(T_2(n)^{1-\epsilon})$ algorithm for oracle calls to P_2 (for an $\epsilon > 0$), there exists an $\epsilon' > 0$, such that $\mathcal{A}$ runs in time $O(T_1(n)^{1-\epsilon'})$. We write this as $(P_1, T_1) \leq (P_2, T_2)$.*

We introduce for problems P, Q the notation $P \leq_c Q$ if and only if $(P, n^c) \leq (Q, n^c)$ holds. Furthermore, P and Q are fine-grained equivalent, denoted by $P \equiv_c Q$, if and only if $(P, n^c) \leq (Q, n^c)$ and $(Q, n^c) \leq (P, n^c)$ holds. We call P_1, P_2 subquadratic equivalent if $P_1 \equiv_2 P_2$.

A convolution of two vectors $x := (x_0, \ldots, x_{n-1})$ $y := (y_0, \ldots, y_{n-1})$ is defined as $z = (z_0, \ldots, z_{2n-2})$, where

$$z[k] := \sum_{i,j \in [n-1], i+j=k} x_i \cdot y_j.$$

It is a well known fact that one can compute the sumset $A+B$ by computing the convolution of the characteristic vectors of A and B. In particular, we also make use of the theorem from Bringmann et al. [20] on sparse convolutions.

Theorem A.4. *There is a deterministic algorithm to compute the convolution of two nonnegative vectors $A, B \in \mathbb{N}^n$ in time $O(t\,polylog(n\Delta))$, where t denotes the number of non-zero entries in the output convolution vector and Δ denotes the maximum entry size of the vectors A, B.*

The convolutional $3-$SUM problem asks for three sequences A, B, C of size n, whether $\exists 0 \leq i, j, k \leq n-1 : a[i] + b[j] = c[k]$. It is well known by a reduction from Pătrașcu [56] that there is no $O(n^{2-\epsilon})$ for an $\epsilon > 0$ algorithm for the Convolutional 3-SUM problem if and only if the 3-SUM hypothesis holds. The strong convolutional 3-SUM hypothesis asks this question over a linearly sized universe [10], that is $A, B, C \subseteq \{-n, \ldots, n\}$, and has recently been shown to be equivalent to the strong 3-SUM hypothesis (3-SUM over the universe $\{-n^2, \ldots, n^2\}$) [21].

Let us fix the notation throughout the paper. We denote the sumset $U + V := \{u + v : u \in U, v \in V\}$. The i-th entry of a vector v is denoted by $v[i]$. For natural numbers n, we denote by $n[i]$ the i-th bit of n, where $n[0]$ is the least significant bit. Furthermore, we say the i-th bit of a natural number is set iff $n[i] = 1$. For vectors u, v we write $u \leq v$ if and only if for all dimensions i it holds that $u[i] \leq v[i]$. If $u \leq v$, we say u is dominated by v.

For a unary function $f : U \to M$ and a set $A \subseteq U$ denote $f(A) := \{f(a) : a \in A\}$. We abbreviate by $[t] := \{0, \ldots, t\}$. The notation $\tilde{O}(T) := T \log^{O(1)} T$ is used to hide poly-logarithmic factors. We denote the cardinality of a set A by $\#A$ or $|A|$. Linear Integer arithmetic refers to the first order logic over the domain $\mathbb{Z}$ with vocabulary: equality ($=$), inequality ($<, >, \geq, \leq$), and addition ($+$). We use $\binom{V}{k}$ to denote all k-element subsets of V.

The (M, d)-vector k-SUM problem is defined as follows [5]. For given k sets $A_1, \ldots, A_k$ of size at most n where each $A_i \subseteq \{-M, \ldots, M\}^d$ and a target $t \in \{-M, \ldots, M\}^d$, do there exist $a_1 \in A_1, \ldots, a_k \in A_k : a_1 + \cdots + a_k = t$? Through the standard technique of interpreting vectors as integers we get:

Lemma A.5. *The (M, d)-vector k-SUM problem can be reduced to the k-SUM problem with universe size $\{0, \ldots, (kM + 1)^d\}$ in time $O(n \log M)$.*

For a proof of the above see the proof of Abboud et al. [5] or our multiset adaptation of the proof in Lemma 4.2. A functional version of the 3-SUM problem we will require is the following:

Definition A.6 (All-ints 3-SUM)**.** *Given sets A, B, C of at most n integers $A, B, C \subseteq \{-n^k, \ldots, n^k\}$ for each $a \in A$ determine, whether there exist $b \in B$ and $c \in C$ such that $a + b + c = t$.*

Lemma A.7 (All-ints 3-SUM $\equiv_2$ 3-SUM [61])**.** *There exists a $O(n^{2-\epsilon})$ time algorithm for the All-ints 3-SUM problem for an $\epsilon > 0$ if and only if there exists a $O(n^{2-\epsilon'})$ time algorithm for 3-SUM for an $\epsilon' > 0$.*

For a proof see results from Williams et al. [61]. It is known that the reduction can be made deterministic for instance using the 3-SUM self reduction from Lincoln et al. [52] combined with the technique introduced by Williams et al. [61]. We continue with a 3-SUM version, which aims to count witnesses.

Definition A.8 (#3-SUM)**.** *Given sets A, B, C of at most n integers $A, B, C \subseteq \{-n^k, \ldots, n^k\}$. The #3-SUM problem asks for the number of triplets $(a, b, c) \in A \times B \times C$ such that $a + b + c = 0$.* [22]

Definition A.9 (All-ints #3-SUM)**.** *Given sets A, B, C of at most n integers $A, B, C \subseteq \{-n^k, \ldots, n^k\}$. The All-ints #3-SUM problem, asks to determine for each $a \in A$ the number of $(b, c) \in B \times C$ such that $a + b + c = 0$.*

In our paper, we make use of the following recent powerful result from Chan et al. [29]

Theorem A.10 ([29])**.** *The following problems are all subquadratic equivalent under randomized fine-grained reductions:*

- *#All-ints 3-SUM,*
- *3-SUM,*
- *#3-SUM.*

Definition A.11 (3-Uniform k-hyperclique problem)**.** *Given a k-partite 3-uniform hypergraph $G = (V, E)$, that is the vertices are a disjoint union of sets $V_1, V_2, \ldots, V_k$ of size n each, and E is a set of edges of the form v_a, v_b, v_c where a, b, c are distinct and $v_a \in V_a, v_b \in V_b, v_c \in V_c$. The problem asks if there exists a $k-$Clique in G, that is vertices $v_1 \in V_1, \ldots, v_k \in V_k$, such that for all*

$$a, b, c \in \binom{\{1, \ldots, k\}}{3}$$

there exists an edge $\{v_a, v_b, v_c\}$.

[22] This version of 3-SUM is equivalent to the version where $a + b + c = t$ for an integer t is asked. In particular by setting $C' := C - \{t\}$, we get a reduction that preserves all solutions.

There is a naive algorithm deciding the above problem in $O(n^k)$. It is strongly believed that this runtime is optimal. One reason is that matrix multiplication techniques that speed up clique detections in graphs (rather than hypergraphs) seem to fail, see for instance [53]. Furthermore, a faster algorithm would lead to an exponential improvement over current $2^{n-o(n)}$ algorithms for MAX 3-SAT, see [30, 8]. Thus, we work with the following hypothesis.

Hypothesis A.12 (3-Uniform Hyperclique Hypothesis)**.** *There is no $O(n^{k-\epsilon})$ algorithm solving the 3-Uniform k-hyperclique problem for $k \geq 4$ and an $\epsilon > 0$.*

In [35], Cygan et al. studied the MaxConv lower bound problem, but were unable to give a nontrivial upperbound. They managed to only show a reduction from this problem to the L_p necklace alignment problem. The problem is defined as follows:

Definition A.13 (MaxConv lower bound)**.** *Given integer arrays A, B, C of length n. Determine whether $C[k] \leq \max_{i+j=k}(A[i] + B[j])$ holds.*

A key in our proofs will be a slightly generalized version of a lemma, whose aim it is to reduce inequality checking to a logarithmic amount of equality checks [62].

Lemma A.14 (Bit-trick)**.** *For any non-negative integers $x_1, x_2, \ldots, x_k, z \in \{0, \ldots, U\}$, we have the following equivalence:*

$$x_1 + \cdots + x_k > z \iff \text{There are } \ell \in \{1, \ldots, [\lceil \log_2(U) \rceil]\}, b \in \{1, 2, \ldots, k\} : \\ pre_\ell(x_1) + \cdots + pre_\ell(x_k) = pre_\ell(z) + b,$$

where $pre_\ell(x)$ denotes the number remaining when taking the first l bits of x, where the most significant bit is considered the first bit. Formally, for $z \in \{0, \ldots, U\}$ and $\ell \in \{0, \ldots, \lceil \log_2(U) \rceil\}$ $pre_\ell(z) := \lfloor \frac{z}{2^{B-\ell}} \rfloor$, where B denotes the number of bits in z.

Proof. If $pre_\ell(x_1) + pre_\ell(x_2) + \cdots + pre_\ell(x_k) = pre_\ell(z) + b$, for B-bit integers holds, we have

$$\begin{aligned}
&\sum_{i=B-\ell}^{B-1} 2^i x_1[i] + \cdots + \sum_{i=B-l}^{B-1} 2^i x_k[i] = \sum_{i=B-\ell}^{B-1} 2^i z[i] + 2^{B-l} b \\
\implies &\sum_{i=B-\ell}^{B-1} 2^i (x_1[i] + x_2[i] + \cdots + x_k[i]) = \sum_{i=B-\ell}^{B-1} 2^i z[i] + 2^{B-l} b \\
\implies &\sum_{i=1}^{k} x_i = z + \sum_{i=0}^{B-\ell-1} (x_1[i] + x_2[i] + \cdots + x_k[i] - z[i]) + 2^{B-\ell} b \\
\implies &\sum_{i=1}^{k} x_i > z.
\end{aligned}$$

Assume now that $x_1 + x_2 + \cdots + x_k > z$ holds, let

$$\hat{\ell} := \min\{i : 1 \leq i \leq B \wedge pre_i(x_1) + pre_i(x_2) + \cdots + pre_i(x_k) > pre_i(z)\}.$$

If $b \leq k$, the statement holds, for the sake of a contradiction assume

$$pre_{\hat{\ell}}(x_1) + pre_{\hat{\ell}}(x_2) + \cdots + pre_{\hat{\ell}}(x_k) \geq pre_{\hat{\ell}}(z) + k + 1.$$

We know that the following holds for any B-bit integer x and any $1 \leq l \leq B$

$$2pre_{\ell-1}(x) \leq pre_{\ell}(x) \leq 2pre_{\ell-1}(x) + 1.$$

Thus, we have

$$\begin{aligned} 2\left(pre_{\hat{\ell}-1}(x_1) + \cdots + pre_{\hat{\ell}-1}(x_k)\right) + k &\geq pre_{\hat{\ell}}(x_1) + \cdots + pre_{\hat{\ell}}(x_k) \\ &\geq pre_{\hat{\ell}}(z) + k + 1 \\ &\geq 2pre_{\hat{\ell}-1}(z) + k + 1. \end{aligned}$$

Concluding

$$pre_{\hat{\ell}-1}(x_1) + pre_{\hat{\ell}-1}(x_2) + \cdots + pre_{\hat{\ell}-1}(x_k) \geq pre_{\hat{\ell}-1}(z) + 1/2,$$

and thereby also

$$pre_{\hat{\ell}-1}(x_1) + pre_{\hat{\ell}-1}(x_2) + \cdots + pre_{\hat{\ell}-1}(x_k) > pre_{\hat{\ell}-1}(z),$$

contradicting the minimality of $\hat{\ell}$. □

We conclude this section with an observation.

Observation A.15. *The choices of ℓ and b in Lemma A.14 are not unique, but there is a unique $\ell \in \{1, \ldots, \lceil \log_2(U) \rceil\}$ and $b \in \{1, \ldots, k\}$ such that the following holds*

$$\sum_{i=1}^{k} x_i > z \iff \sum_{i=1}^{k} pre_{\ell}(x_i) = pre_{l}(z) + b \wedge \sum_{i=1}^{k} pre_{\ell-1}(x_i) = pre_{\ell-1}(z).$$

B Examples of problems in $\mathsf{FOP}_{\mathbb{Z}}$

In the following, we give a plethora of examples of problems in $\mathsf{FOP}_{\mathbb{Z}}$.

Example B.1 (3-Average free set)**.** *Consider a finite set of integers A. Is there no arithmetic progression of length 3? We can use the negation of the following sentence to obtain an answer.*

$$\exists a_1 \in A \exists a_2 \in A \exists a_3 \in A : a_1 < a_2 < a_3 \wedge a_1 + a_3 = 2a_2.$$

We use the condition $a_1 < a_2 < a_3$ as to avoid $a_1 = a_2 = a_3$. The 3-average free set problem is known to be subquadratic equivalent to 3-SUM [37], which itself is another example.

Example B.2 (3-SUM and Conv-3SUM)**.** *Consider a finite set of integers A. Are there three numbers in A summing up to 0? I.e.,*

$$\exists a_1 \in A \exists a_2 \in A \exists a_3 \in A : a_1 + a_2 + a_3 = 0.$$

We can also express its subquadratic equivalent formulation convolutional 3-SUM[23] *as a two-dimensional formula: Representing a sequence $X[0 \ldots n-1]$ as $X' = \{(i, X[i]) \mid 0 \leq i < n\}$, we ask whether given sequences $A[0 \ldots n-1], B[0 \ldots n-1], C[0 \ldots n-1]$ satisfy:*

$$\exists (i, a[i]) \in A' \; \exists (j, b[j]) \in B' \; \exists (k, c[k]) \in C' : a[i] + b[j] = c[k] \wedge i + j = k$$

[23] For further discussion, see Section A.

More generally, we can view these problems as natural database queries with numerical data, e.g.:

Example B.3 (Orthogonal Range and Database queries)**.** *Consider a database $\mathcal{D}$, where each entity consists of a tuple $(id, age, income)$. The following query is part of the class $\mathsf{FOP}_{\mathbb{Z}}(\exists^3)$.*
Do there exist three different people in $\mathcal{D}$, whose average age is below 30, whose income is in the range $[10000, 20000]$, and whose incomes accumulate to more than 50000?

All the above examples use existentially quantified variables. This existential fragment will be of particular interest for us.

Other quantifier structures also give rise to natural algorithmic problems.

Example B.4 (Universal 3-SUM)**.** *Given sets $A, B, C \subseteq \mathbb{Z}$. Does for all $a \in A$ exist $b \in B$ and $c \in C$ such that $c = a + b$. Clearly the problem is in $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$. Alternatively, one can also view the problem as a sumset expression, namely $C \subseteq A + B$.*

We note that Universal 3-SUM seems to be a weaker version of All-ints 3-SUM and that in general the class $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$ seems to be the most likely to admit a subquadratic algorithm, as it has the weakest known lower bound barrier.

Example B.5 (Verification of Δ-approximations of sumsets)**.** *Bringmann and Nakos introduce a notion of additive approximation of sumsets in [23]. We consider the problem of verifying whether a set additively approximates a sumset. Specifically, let $A + B := \{a + b : a \in A, b \in B\}$ denote the sumset of A and B. We say that a set C is an additive Δ-approximation*[24] *of $A + B$ whenever*

$$\forall a \in A \forall b \in B \exists c \in C : c \leq a + b \leq c + \Delta \iff A + B \subseteq C + \{0, \ldots, \Delta\}.$$

Note that we could also require each number in C to be a sum itself by taking the conjunction with the Universal 3-SUM formula:

$$\forall c \in C \exists a \in A \exists b \in B : a + b = c \iff C \subseteq A + B.$$

The above example makes use of the free variables in the definition of the problem $\mathsf{FOP}_{\mathbb{Z}}(\phi)$, in this case the additive approximation constant Δ is a free variable and can be instantiated by an input natural number. Interestingly, the conjunction of both conditions $A + B \subseteq C + \{0, \ldots, \Delta\}$ and $C \subseteq A + B$ can be reduced to 3-SUM, as $A + B \subseteq C + \{0, \ldots, \Delta\}$ is in $\mathsf{FOP}_{\mathbb{Z}}(\forall\forall\exists)$ and has inequality dimension 2, and $C \subseteq A + B$ is in $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$.

Example B.6 (Hausdorff Distance under n Translations)**.** *We recall the definition of the Hausdorff Distance under n Translations problem over d-dimensional sets A, B, C, and a value $\gamma \in \mathbb{N}$, where we ask whether the following holds*

$$\delta_{\vec{H}}^{T(A)}(B, C) = \min_{\tau \in A} \max_{b \in B} \min_{c \in C} \|b - (c + \tau)\|_\infty \leq \gamma.$$

Clearly, $\|b - (c + \tau)\|_\infty \leq \gamma$ if and only if for all dimensions $i \in \{1, \ldots d\}$, we have $b[i] - (c[i] + \tau[i]) \leq \gamma$ and $(c[i] + \tau[i]) - b[i] \leq \gamma$. Thus, it remains to check

$$\exists \tau \in A \forall b \in B \exists c \in C : \bigwedge_{i=1}^{d} b[i] - (c[i] + \tau[i]) \leq \gamma \wedge (c[i] + \tau[i]) - b[i] \leq \gamma.$$

[24] We remark that we use here a simplified notion of additive approximation that is closely related to the ones used by Bringmann et al. [23].

Example B.7 (Triangle Detection)**.** *Let E be the set of edges in a directed graph. Each edge has an id and consider α, ω to be functions that denote the start and endpoint of an edge respectively. Thus $E' = \{(e_{id}, \alpha(e), \omega(e)) : e \in E\}$. To detect a triangle we can simply ask*

$$\begin{aligned}\exists e_1 \in E \exists e_2 \in E \exists e_3 \in E : & e_1[0] \neq e_2[0] \wedge e_2[0] \neq e_3[0] \wedge e_1[0] \neq e_3[0] \wedge \\ & e_1[2] = e_2[1] \wedge e_2[2] = e_3[1] \wedge e_3[2] = e_1[1].\end{aligned}$$

Thus a $\tilde{O}(n^{1+\epsilon})$ time algorithm for 3-SUM implies a $\tilde{O}(m^{1+\epsilon})$ time algorithm for triangle detection.

Let us now take a look at the MaxConv lower bound problem (see Definition A.13).

Lemma B.8. *The MaxConv lower bound problem is a member of the class $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$, and a member of the class $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$.*

Proof. 1. Let

$$\begin{aligned}A' &= \{(A[i], i) : i \in [n-1]\} \\ B' &= \{(B[j], j) : j \in [n-1]\} \cup \{(-M, -j) : j \in \{1, \ldots, n-1\}\} \\ C' &= \{(C[k], k) : k \in [n-1]\},\end{aligned}$$

where $M = 3 \cdot \max(A \cup B \cup C)$. We ask

$$\exists c' \in C' \forall a' \in A' \exists b' \in B' : (i + j = k \wedge C'[k] > A'[i] + B'[j]).$$

Thus, we can formulate MaxConv lower bound in $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$.

We give a short proof to this equivalence in the following. We have that $C[k]$ is a non witness to MaxConv lower bound if and only if $\forall i \in [n-1] : C[k] > A[i] + B[k-i]$. Now, let us make a case distinction on i. Consider the case $i \leq k$, then only $j = k - i \in [n-1]$ fulfills $i + j = k$, and due to the fact that $c[k]$ is a non witness $C[k] > A[i] + B[j]$. For the case $i > k$, we have j negative, which trivially fulfills $C[k] > A[i] + B[j]$.

For the other direction notice that there exists a $k \in [n-1]$ which fulfills for all indices $i + j = k$ that $C[k] > A[i] + B[j]$.

2. For the following constructed sets:

$$\begin{aligned}A' &= \{(A[i], i) : i \in [n-1]\} \\ B' &= \{(B[j], j) : j \in [n-1]\} \\ C' &= \{(C[k], k) : k \in [n-1]\} \cup \{(-M, k) : k \in \{n, ..., 2n\}\},\end{aligned}$$

where $M = 3 \cdot \max A \cup B \cup C$. It can be easily seen that the following formula in $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$ models the MaxConv lower bound problem,

$$\forall c' \in C' \exists a \in A' \exists b' \in B' : i + j = k \wedge C[k] \leq A[i] + B[j].$$

$\square$

The above lemma presents a witness to the hardness of the classes $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists\exists)$ and $\mathsf{FOP}_{\mathbb{Z}}(\exists\forall\exists)$.

C Baseline Algorithms

Lemma C.1. *Every problem in the class* $\mathsf{FOP}_{\mathbb{Z}}(\exists^k)$ *can be decided in time* $\tilde{O}(n^{\lceil k/2 \rceil})$*.*

Proof. To this end, after substituting the free variables $\hat{t_1}, \ldots \hat{t_l}$ in the quantifier free part of ϕ, without loss of generality, by the same arguments used in the proof of Theorem 1.1, we can transform every such linear arithmetic formula $\phi[(t_1, \ldots, t_l)\backslash(\hat{t_1}, \ldots, \hat{t_l})]$ into the following form:

$$\exists a_1 \in A_1 \ldots \exists a_k \in A_k : \bigvee_{h=1}^{H} \bigwedge_{i=1}^{m} \sum_{j=1}^{k} c_{h,i,j}^T a_j \geq S_{h,i}.$$

Due to the commutativity of disjunction and existential quantifiers, we can rewrite the above in the form:

$$\bigvee_{h=1}^{H} \exists a_1 \in A_1 \ldots \exists a_k \in A_k : \bigwedge_{i=1}^{m} \sum_{j=1}^{k} c_{h,i,j}^T a_j \geq S_{h,i}.$$

Now it suffices to have a look at H instances of the following problem, whose results are combined disjunctively

$$\exists a_1 \in A_1 \ldots \exists a_k \in A_k : \bigwedge_{i=1}^{m} \sum_{j=1}^{k} c_{i,j}^T a_j \geq S_i,$$

where $S_i \in \mathbb{Z}, c_{i,j} \in \mathbb{Z}^{d_j}$. Consider summing up the tuples by their corresponding coefficients, that is we consider the sets

$$A_j^{'} := \left\{ \begin{pmatrix} c_{1,j}^T a \\ c_{2,j}^T a \\ \vdots \\ c_{m,j}^T a \end{pmatrix} : a \in A_j \right\},$$

and define $S := (S_1, S_2, \ldots, S_m)^T$. We are now left with the following problem

$$\exists a_1' \in A_1' \ldots \exists a_k' \in A_k' : \sum_{j=1}^{\lceil k/2 \rceil} a_j' \geq \sum_{j=\lceil k/2 \rceil+1}^{k} a_j' + S$$

Precompute the left-hand side sums and the right-hand side sums in $O(n^{\lceil k/2 \rceil})$. Store all possible sums in the left-hand side in a range tree, see [9]. Iterating over the precomputed sums in the right-hand side and querying them in the range tree gives us an algorithm in time $\tilde{O}(n^{\lceil k/2 \rceil})$. $\square$

Lemma C.2. *Every formula* ϕ *in one of the classes*

$$\mathsf{FOP}_{\mathbb{Z}}(\exists\exists), \mathsf{FOP}_{\mathbb{Z}}(\forall\forall), \mathsf{FOP}_{\mathbb{Z}}(\exists\forall), \mathsf{FOP}_{\mathbb{Z}}(\forall\exists),$$

can be decided in time $\tilde{O}(n)$*.*

Proof. If ϕ is in the class $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists)$, we can conclude by the same algorithm as in Lemma C.1. If ϕ is in the class $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists)$, after substituting the free variables, by the same arguments used in the proof of Theorem 1.1, we can transform ϕ to an equivalent formula of the type:

$$\forall a_1 \in A_1 \exists a_2 \in A_2 : \bigvee_{h=1}^{H} \bigwedge_{i=1}^{m} c_{h,i,1}^T a_1 + c_{h,i,2}^T a_2 \geq S_{h,i},$$

which is equivalent to

$$\forall a_1 \in A_1 \bigvee_{h=1}^{H} \exists a_2 \in A_2 : \bigwedge_{i=1}^{m} c_{h,i,1}^T a_1 + c_{h,i,2}^T a_2 \geq S_{h,i}.$$

Consider

$$A_2^{(h)} := \left\{ \begin{pmatrix} c_{h,1,2}^T a \\ c_{h,2,2}^T a \\ \vdots \\ c_{h,m,2}^T a \end{pmatrix} : a \in A_2 \right\},$$

$S^{(h)} := (S_{h,1}, \ldots, S_{h,m})^T$, and finally $a_1^{(h)} := (c_{h,1,1}^T a_1, c_{h,2,1}^T a_1, c_{h,m,1}^T a_1)$. Then, it remains to solve:

$$\forall a_1 \in A_1 \bigvee_{h=1}^{H} \exists a_2 \in A_2^{(h)} : a_2 \geq S^{(h)} - a_1^{(h)}.$$

Thus, we insert in H different orthogonal range trees, the elements of $A_2^{(h)}$.

By iterating over all the elements in A_1, and checking whether for any $h \in \{1, \ldots, H\}$, the vector $S^{(h)} - a_1^{(h)}$ is dominated by some $a_2 \in A_2^{(h)}$ via a range tree query, we can conclude.

Finally, by simply negating the formula and pushing the negation inwards, we can always get a formula in $\mathsf{FOP}_{\mathbb{Z}}(\exists\exists)$ or $\mathsf{FOP}_{\mathbb{Z}}(\forall\exists)$. $\square$

Lemma C.3. *Every problem in the class $\mathsf{FOP}_{\mathbb{Z}}^k$ can be decided in time $\tilde{O}(n^{k-1})$.*

Proof. Let ϕ be in $\mathsf{FOP}_{\mathbb{Z}}$. After substituting the free variables, we brute force the first $k-2$ quantifiers. It remains to solve a formula φ with 2 quantifiers. We can conclude by a simple application of Lemma C.2 $\square$

D Proof of Lemma 3.1

Proof. We define the transformations for $f_j^{\ell,\psi}$ with $j \in \{1, \ldots, k\}$ and the transformation $g^{\ell,W,\psi}$ as the following vectors with $2m$ dimensions:

$$\underbrace{\begin{pmatrix} pre_{\ell[1]}(M + c_{1,j}^T a_j) \\ \vdots \\ pre_{\ell[m]}(M + c_{m,j}^T a_j) \\ pre_{\ell[1]-1}(M + c_{1,j}^T a_j) \\ \vdots \\ pre_{\ell[m]-1}(M + c_{m,j}^T a_j) \end{pmatrix}}_{=:f_j^{\ell,\psi}(a_j)} \quad \underbrace{\begin{pmatrix} pre_{\ell[1]}\left(S_1 - 1 + kM\right) + W[1] \\ \vdots \\ pre_{\ell[m]}(S_m - 1 + kM) + W[m] \\ pre_{\ell[1]-1}\left(S_1 - 1 + kM\right) \\ \vdots \\ pre_{\ell[m]-1}(S_m - 1 + kM) \end{pmatrix}}_{=:g^{\ell,W,\psi}(S_1,\ldots,S_m)}.$$

Notice that pre_ℓ is applied on non-negative integers due to the choice of M. For all $\ell \in$

$\{1, \ldots, \lceil \log_2(M) \rceil\}^m, W \in \{1, \ldots, k\}^m$, we have that

$$
\begin{aligned}
& f_1^{\ell,\psi}(a_1) + \cdots + f_k^{\ell,\psi}(a_k) = g^{\ell,\psi,W}(S_1, \ldots, S_m) \\
\iff & \underbrace{\bigwedge_{i=1}^{m} \sum_{j=1}^{k} pre_{\ell[i]}(M + c_{i,j}^T a_j) = pre_{\ell[i]}\left(S_i - 1 + kM\right) + W[i]}_{=:\varphi_1(\ell,W)} \\
\wedge & \underbrace{\bigwedge_{i=1}^{m} \sum_{j=1}^{k} pre_{\ell[i]-1}(M + c_{i,j}^T a_j) = pre_{\ell[i]-1}\left(S_i - 1 + kM\right)}_{=:\varphi_2(\ell,W)}.
\end{aligned}
$$

By Lemma A.14 (also Observation A.15), there exist unique $\ell' \in \{1, \ldots, \lceil \log_2(M) \rceil\}^m, W' \in \{1, \ldots, k\}^m$ such that

$$
\varphi_1(\ell', W') \wedge \varphi_2(\ell', W') \iff \bigwedge_{i=1}^{m} \sum_{j=1}^{k} M + c_{i,j}^T a_j > S_i - 1 + kM,
$$

which is equivalent to the desired conjunction of inequalities. The function of the dimensions $m+1, \ldots, 2m$, and in particular $\varphi_2(\ell', W')$ are to ensure the uniqueness for the choices of b and ℓ in Lemma A.14. ☐

E Proof of Disjoint boxes Lemma

Lemma E.1. *We can decompose a rectilinear shape of complexity n in $O(n)$ interior- and exterior-disjoint rectangles, some of which may be degenerate, in time $O(n \log n)$.*

Proof. At the beginning, we aim to decompose the rectilinear shape into interior-disjoint rectangles. For this purpose, we perform a sweep-line algorithm. The sweep-line goes through the x-direction. Firstly, we maintain a collection of components given by a segment in y direction, which will intuitively denote the opening of components(or rectangles) which are not overlapping. Each vertex event, will now either create a new component, will enlarge an existing component, will shrink an existing component, or will make it disjoint. By shooting a ray up and down from this event point vertex we can create a new rectangle, together with the current existing component. The vertices intersected by this ray, will form the starting points for the new component. For details see the figure below. As each vertex will create at most one new rectangle, we conclude that the number of rectangles is in $O(n)$.

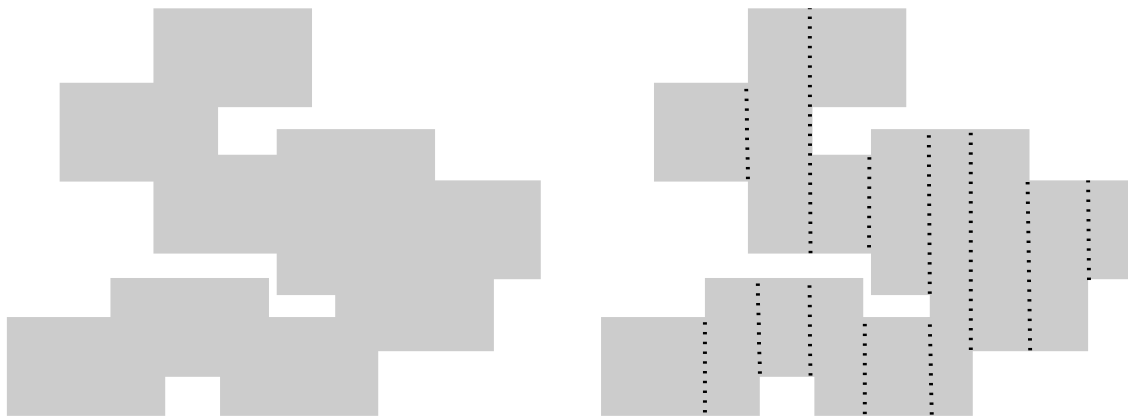

Figure 1: To the left we can find the rectilinear shape. To the right we find the decomposition into interior disjoint rectangles.

The rectangles formed by the above procedure are kept as open 2-boxes. We now turn to the face segments of these rectangles. If a vertex event point splits a segment, we decompose the segment at this point. Clearly at most $O(n)$ segments (1-boxes) can be created in this way. The 1-boxes we will keep as open, and the vertices as closed 0-boxes (or points). $\square$

Lemma E.2. *The union of n unit cubes can be decomposed into $O(n)$ interior- and exterior-disjoint axis-aligned boxes, some of which may be degenerate, in time $O(n \log^2 n)$.*

Proof. We adapt the algorithm of [31] to also create exterior-disjoint boxes. Slice the three-dimensional space by planes parallel to the z axis for planes $z = 1, z = 2, z = 3 \dots$ (without loss of generality, we can assume the cubes to be starting at $z = 1$). Consider the slab of cubes generated by cutting the cubes by the plane $z = i$ and $z = i+1$, where we denote by n_i the number of cubes cut by the plane $z = i$. The complexity of the slab is in $O(n_i + n_{i+1})$ by [15].

We repeat the following for each slab.

Let E be the portion of cubes that lie within the slab bounded by $z = i$ and $z = i + 1$, and let the silhouette of E be the projection on both $z = i$ and $z = i + 1$ of all vertical lines whose intersection with E is one unit long – we refer to this vertical length as *height*.

We firstly construct a decomposition of the silhouette of E into interior- and exterior-disjoint boxes. To achieve this we perform a decomposition as given in Lemma E.1 of the projection of the silhouette onto the plane $z = i$. After achieving this decomposition, we create 3-boxes out of the rectangles, with height 1, which we keep open. Out of the segments, we create open 2-boxes with height 1, and from the vertices we create open 1-boxes of height 1.

Let S be the projection of the silhouette of E onto the plane $z = i$, and S' be the projection of the silhouette of E onto the plane $z = i + 1$. Moreover, let E' be the intersection of E with the plane $z = i$, and let F' denote the points on E' whose vertical line length (i.e., height) changes. Computing F' (with the associated heights) costs time $O((n_i + n_{i+1}) \log^2(n_i + n_{i+1}))$ by computing the union of the cubes in the slab by [7].

Consider F' without S, where S is the union of the formed rectangle decomposition of the silhouette of E. We perform the Lemma E.1 decomposition onto this rectilinear shape, with the restriction that the rays shot up and down do not cross S. Create the open 3-boxes just like in the step for the silhouette, with the difference being that we use the height of each produced rectangle (which by construction is uniform over the rectangle). We ignore the segments and vertices which lie adjacent to S (they have height 1 and are already taken care of). For the remaining segments and vertices, we create open 2-boxes and closed 1-boxes respectively, where we take the height of the larger adjacent box. Finally, in contrast to the silhouette we also create open 2-boxes for the faces, which lie on top of the created open 3-boxes (or rather the face lying on the height of the 3-box).

Proceed analogously for E'' which is the intersection of E onto the plane $z = i + 1$ and F'', the points on E'' whose vertical line length changes.

Finally, it remains to treat the rectilinear shapes E' and E'' separately, as none of the boxes include the faces on the plane $z = i$ and $z = i + 1$. For this again, we perform the Lemma E.1 decomposition for E' and E'' respectively, where we keep the rectangles and segments, as open 2 and 1-boxes respectively, and the vertices as closed 0-boxes which lie on the respective plane. $\square$