A Framework for ETH-Tight Algorithms and Lower Bounds in Geometric Intersection Graphs*†

Mark de Berg
Eindhoven University of Technology
Eindhoven, The Netherlands
M.T.d.Berg@tue.nl

Hans L. Bodlaender
Utrecht University and
Eindhoven University of Technology
Utrecht, The Netherlands
H.L.Bodlaender@uu.nl

Sándor Kisfaludi-Bak
Eindhoven University of Technology
Eindhoven, The Netherlands
S.Kisfaludi.Bak@tue.nl

Dániel Marx
Hungarian Academy of Sciences
(MTA SZTAKI)
Budapest, Hungary
DMarx@cs.bme.hu

Tom C. van der Zanden
Utrecht University
Utrecht, The Netherlands
T.C.vanderZanden@uu.nl

ABSTRACT

We give an algorithmic and lower-bound framework that facilitates the construction of subexponential algorithms and matching conditional complexity bounds. It can be applied to a wide range of geometric intersection graphs (intersections of similarly sized fat objects), yielding algorithms with running time $2^{O(n^{1-1/d})}$ for any fixed dimension $d \geq 2$ for many well known graph problems, including Independent Set, r-Dominating Set for constant r, and Steiner Tree. For most problems, we get improved running times compared to prior work; in some cases, we give the first known subexponential algorithm in geometric intersection graphs. Additionally, most of the obtained algorithms work on the graph itself, i.e., do not require any geometric information. Our algorithmic framework is based on a weighted separator theorem and various treewidth techniques.

The lower bound framework is based on a constructive embedding of graphs into d-dimensional grids, and it allows us to derive matching $2^{\Omega(n^{1-1/d})}$ lower bounds under the Exponential Time Hypothesis even in the much more restricted class of d-dimensional induced grid graphs.

CCS CONCEPTS

• **Theory of computation → Computational geometry**; *Parameterized complexity and exact algorithms*;

KEYWORDS

Geometric intersection graphs, Subexponential algorithms, Geometric separator, Treewidth, Graph minors

ACM Reference Format:
Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C. van der Zanden. 2018. A Framework for ETH-Tight Algorithms and Lower Bounds in Geometric Intersection Graphs. In *Proceedings of 50th Annual ACM SIGACT Symposium on the Theory of Computing (STOC'18).* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3188745.3188854

1 INTRODUCTION

Many hard graph problems that seem to require $2^{\Omega(n)}$ time on general graphs, where n is the number of vertices, can be solved in subexponential time on planar graphs. In particular, many of these problems can be solved in $2^{O(\sqrt{n})}$ time on planar graphs. Examples of problems for which this so-called *square-root phenomenon* [21] holds include Independent Set, Vertex Cover, Hamiltonian Cycle. The great speed-ups that the square-root phenomenon offers lead to the question: are there other graph classes that also exhibit this phenomenon, and is there an overarching framework to obtain algorithms with subexponential running time for these graph classes? The planar separator theorem [19, 20] and treewidth-based algorithms [8] offer a partial answer to this question. They give a general framework to obtain subexponential algorithms on planar graphs or, more generally, on H-minor free graphs. It builds heavily on the fact that H-minor free graphs have treewidth $O(\sqrt{n})$ and, hence, admit a separator of size $(\sqrt{n})$. A similar line of work is emerging in the area of geometric intersection graphs, with running times of the form $n^{O(n^{1-1/d})}$, or in one case $2^{O(n^{1-1/d})}$ in the d-dimensional case [23, 25]. The main goal of our paper is to establish a framework for a wide class of geometric intersection graphs that is similar to the framework known for planar graphs, while guaranteeing the running time $2^{O(n^{1-1/d})}$.

The *intersection graph* $G[F]$ of a set F of objects in $\mathbb{R}^d$ is the graph whose vertex set is F and in which two vertices are connected when the corresponding objects intersect. *(Unit-)disk graphs*, where F consists of (unit) disks in the plane are a widely studied class of intersection graphs. Disk graphs form a natural generalization of

*This work was supported by the NETWORKS project, funded by the Netherlands Organization for Scientific Research NWO under project no. 024.002.003 and by the ERC Consolidator Grant SYSTEMATICGRAPH (No. 725978) of the European Research Council.
†The full version is available at [9] (http://arxiv.org/abs/1803.10633).

STOC'18, June 25–29, 2018, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5559-9/18/06…$15.00
https://doi.org/10.1145/3188745.3188854

planar graphs, since any planar graph can be realized as the intersection graph of a set of disks in the plane. In this paper we consider intersection graphs of a set F of *fat objects*, where an object $o \subseteq \mathbb{R}^d$ is α-*fat*, for some $0 < \alpha \leqslant 1$ if there are balls B_{in} and B_{out} in $\mathbb{R}^d$ such that $B_{\text{in}} \subseteq o \subseteq B_{\text{out}}$ and $\text{radius}(B_{\text{in}})/\text{radius}(B_{\text{out}}) \geqslant \alpha$. For example, disks are 1-fat and squares are $(1/\sqrt{2})$-fat. From now on we assume that α is an absolute constant, and often simply speak of fat objects. Note that we do not require the objects in F to be convex, or even connected. Thus our definition is very general. In particular, it does not imply that F has near-linear union complexity, as is the case for so-called locally-fat objects [2]. In most of our results we furthermore assume that the objects in F are *similarly sized*, meaning that the ratio of their diameters is bounded by a fixed constant.

Several important graph problems have been investigated for (unit-)disk graphs or other types of intersection graphs [1, 4, 11, 12, 23]. However, an overarching framework that helps designing subexponential algorithms has remained elusive. A major hurdle to obtain such a framework is that even unit-square graphs can already have arbitrarily large cliques and so they do not necessarily have small separators or small treewidth. One may hope that intersection graphs have low cliquewidth or rankwidth—this has proven to be useful for various dense graph classes [7, 24]—but unfortunately this is not the case even when considering only unit interval graphs [14]. One way to circumvent this hurdle is to restrict the attention to intersection graphs of disks of *bounded ply* [3, 15]. This prevents large cliques, but the restriction to bounded-ply graphs severely limits the inputs that can be handled. A major goal of our work is thus to give a framework that can even be applied when the ply is unbounded.

Our first contribution: an algorithmic framework for geometric intersection graphs of fat objects. As mentioned, many subexponential results for planar graphs rely on planar separators. Our first contribution is a generalization of this result to intersection graphs of (arbitrarily-sized) fat objects in $\mathbb{R}^d$. Since these graphs can have large cliques we cannot bound the number of vertices in the separator. Instead, we build a separator consisting of cliques. We then define a weight function γ on these cliques—in our applications it suffices to define the weight of a clique C as $\gamma(|C|) := \log(|C| + 1)$. We define the weight of a separator as the sum of the weights of its constituent cliques C_i, which is useful since for many problems a separator can intersect the solution vertex set in $2^{O(\sum_i \gamma(|C_i|))}$ many ways. Formally, the theorem can be stated this way:

Theorem 1.1. *Let F be a set of n α-fat objects in $\mathbb{R}^d$ and let γ be a weight function such that $\gamma(t) = O(t^{1-1/d-\varepsilon})$, for constants $d \geqslant 2$, $\alpha > 0$, and $\varepsilon > 0$. Then the intersection graph $G[F]$ has a $(6^d/(6^d+1))$-balanced separator and a clique partition $\mathcal{C}(F_{\text{sep}})$ of F_{sep} with weight $O(n^{1-1/d})$. Such a separator and a clique partition $\mathcal{C}(F_{\text{sep}})$ can be computed in $O(n^{d+2})$ time if the objects have constant complexity.*

A direct application of our separator theorem is a $2^{O(n^{1-1/d})}$ algorithm for Independent Set. For general fat objects, only the 2-dimensional case was known to have such an algorithm [22].

Our separator theorem can be seen as a generalization of the work of Fu [13] who considers a weighting scheme similar to ours. However, Fu's result is significantly less general as it only applies to unit balls and his proof is arguably more complicated. Our result can also be seen as a generalization of the separator theorem of Har-Peled and Quanrud [15] which gives a small separator for constant ply—indeed, our proof borrows some ideas from theirs.

Finally, the technique employed by Fomin et al. [11] in two dimensions has also similar qualities; in particular, the idea of using cliques as a basis for a separator can also be found there, and leads to subexponential parameterized algorithms, even for some problems that we do not tackle here.

After proving the weighted separator theorem for arbitrarily-sized fat objects, we switch to similarly-sized objects. Here the idea is as follows: We find a suitable clique-decomposition $\mathcal{P}$ of the intersection graph $G[F]$, contract each clique to a single vertex, and then work with the contracted graph $G_{\mathcal{P}}$ where the node corresponding to a clique C gets weight $\gamma(|C|)$. We then prove that the graph $G_{\mathcal{P}}$ has constant degree and, using our separator theorem, we prove that $G_{\mathcal{P}}$ has weighted treewidth $O(n^{1-1/d})$. Moreover, we can compute a tree decomposition of this weight in $2^{O(n^{1-1/d})}$ time. Thus we obtain a framework that gives $2^{O(n^{1-1/d})}$-time algorithms for intersection graphs of similarly-sized fat objects for many problems for which treewidth-based algorithms are known. Our framework recovers and often slightly improves the best known results for several problems,[1] including Independent Set, Hamiltonian Cycle and Feedback Vertex Set. Our framework also gives the first subexponential algorithms in geometric intersection graphs for, among other problems, r-Dominating Set for constant r, Steiner Tree and Connected Dominating Set. See the full version for more details. (The full version also contains the proofs of theorems and lemmas whose proofs have been omitted from this extended abstract.)

Furthermore, we show that our approach can be combined with the *rank-based approach* [5], a technique to speed up algorithms for connectivity problems. Table 1 summarizes the results we obtain by applying our framework; in each case we have matching upper and lower bounds on the time complexity of $2^{\Theta(n^{1-1/d})}$ (where the lower bounds are conditional on the Exponential Time Hypothesis).

A desirable property of algorithms for geometric graphs is that they are *robust*, meaning that they can work directly on the graph without knowledge of the underlying geometry. Most of the known algorithms are in fact non-robust, which could be a problem in applications, since finding a geometric representation of a given geometric intersection graph is NP-hard [6] (and many recognition problems for geometric graphs are ER-complete [17]). One of the advantages of our framework is that it yields robust algorithms for many problems. To this end we need to generalize our scheme slightly: We no longer work with a clique partition to define the contracted graph $G_{\mathcal{P}}$, but with a partition whose classes are the union of constantly many cliques. We show that such a partition can be found efficiently without knowing the set F defining the given intersection graph. Thus we obtain robust algorithms for many of the problems mentioned above, in contrast to known results which almost all need the underlying set F as input.

[1] Note that most of the earlier results are in the parameterized setting, but we do not consider parameterized algorithms here.

Table 1: Summary of our results. In each case we list the most inclusive class where our framework leads to algorithms with $2^{O(n^{1-1/d})}$ running time, and the most restrictive class for which we have a matching lower bound. We also list whether the algorithm is robust.

Problem	Algorithm class	Robust	Lower bound class
Independent Set	Fat	no	Unit Ball, $d \geq 2$
Independent Set	Sim. sized fat	yes	Unit Ball, $d \geq 2$
r-Dominating Set, $r = \text{const}$	Sim. sized fat	yes	Induced Grid, $d \geq 2$
Steiner Tree	Sim. sized fat	yes	Induced Grid, $d \geq 2$
Feedback Vertex Set	Sim. sized fat	yes	Induced Grid, $d \geq 2$
Conn. Vertex Cover	Sim. sized fat	yes	Unit Ball, $d \geq 2$ **or** Induced Grid, $d \geq 3$
Conn. Dominating Set	Sim. sized fat	yes	Induced Grid, $d \geq 2$
Conn. Feedback Vertex Set	Sim. sized fat	yes	Unit Ball, $d \geq 2$ **or** Induced Grid, $d \geq 3$
Hamiltonian Cycle/Path	Sim. sized fat	no	Induced Grid, $d \geq 2$

Our second contribution: a framework for lower bounds under ETH. The $2^{O(n^{1-1/d})}$-time algorithms that we obtain for many problems immediately lead to the question: is it possible to obtain even faster algorithms? For many problems on planar graphs, and for certain problems on ball graphs the answer is no, assuming the Exponential Time Hypothesis (ETH) [16]. However, these lower bound results in higher dimensions are scarce, and often very problem-specific. Our second contribution is a framework to obtain tight ETH-based lower bounds for problems on d-dimensional grid graphs (which are a subset of intersection graphs of similarly-sized fat objects). The obtained lower bounds match the upper bounds of the algorithmic framework. Our lower bound technique is based on a constructive embedding of graphs into d-dimensional grids, for $d \geqslant 3$, thus avoiding the invocation of deep results from Robertson and Seymour's graph minor theory. This *Cube Wiring Theorem* implies that for any constant $d \geq 3$, any connected graph on m edges is the minor of the d-dimensional grid hypercube of side length $O(m^{\frac{1}{d-1}})$ (see Theorem 3.8). For $d = 2$, we give a lower bound for a customized version of the 3-SAT problem. Now, these results make it possible to design simple reductions for our problems using just three custom gadgets per problem; the gadgets model variables, clauses, and connections between variables and clauses, respectively. By invoking Cube Wiring or our custom satisfiability problem, the wires connecting the clause and variable gadgets can be routed in a very tight space. Giving these three gadgets immediately yields the tight lower bound in d-dimensional grid graphs (under ETH) for all $d \geq 2$. Naturally, the same conditional lower bounds are implied in all containing graph classes, such as unit ball graphs, unit cube graphs and also in intersection graphs of similarly sized fat objects. Similar lower bounds are known for various problems in the parameterized complexity literature[4, 23]. The embedding in [23] in particular has a denser target graph than a grid hypercube, where the "edge length" of the cube contains an extra logarithmic factor compared to ours (see Theorem 2.17 in [23]) and thereby gives slightly weaker lower bounds.

2 THE ALGORITHMIC FRAMEWORK

2.1 Separators for Arbitrarily-Sized Fat Objects

Let F be a set of n α-fat objects in $\mathbb{R}^d$ for some constant $\alpha > 0$, and let $G[F] = (F, E)$ be the intersection graph induced by F. We say that a subset $F_{\text{sep}} \subseteq F$ is a *β-balanced separator* for $G[F]$ if $F \setminus F_{\text{sep}}$ can be partitioned into two subsets F_1 and F_2 with no edges between them and with $\max(|F_1|, |F_2|) \leqslant \beta n$. For a given decomposition $C(F_{\text{sep}})$ of F_{sep} into cliques and a given weight function γ we define the *weight* of F_{sep}, denoted by $\text{weight}(F_{\text{sep}})$, as $\text{weight}(F_{\text{sep}}) := \sum_{C \in C(F_{\text{sep}})} \gamma(|C|)$. Next we prove that $G[F]$ admits a balanced separator of weight $O(n^{1-1/d})$ for any cost function $\gamma(t) = O(t^{1-1/d-\varepsilon})$ with $\varepsilon > 0$. Our approach borrows ideas from Har-Peled and Quanrud [15], who show the existence of small separators for low-density sets of objects, although our arguments are significantly more involved.

Step 1: Finding candidate separators. Let H_0 be a minimum-size hypercube containing at least $n/(6^d + 1)$ objects from F, and assume without loss of generality that H_0 is the unit hypercube centered at the origin. Let $H_1, \ldots, H_m$ be a collection of $m := n^{1/d}$ hypercubes, all centered at the origin, where H_i has edge length $1 + \frac{2i}{m}$. Note that the largest hypercube, H_m, has edge length 3, and that the distance between consecutive hypercubes H_i and H_{i+1} is $1/n^{1/d}$.

Each hypercube H_i induces a partition of F into three subsets: a subset $F_{\text{in}}(H_i)$ containing all objects that lie completely in the interior of H_i, a subset $F_{\partial}(H_i)$ containing all objects that intersect the boundary ∂H_i of H_i, and a subset $F_{\text{out}}(H_i)$ containing all objects that lie completely in the exterior of H_i. Obviously an object from $F_{\text{in}}(H_i)$ cannot intersect an object from $F_{\text{out}}(H_i)$, and so $F_{\partial}(H_i)$ defines a separator in a natural way. It will be convenient to add some more objects to these separators, as follows. We call an object *large* when its diameter is at least $1/4$, and *small* otherwise. We will add all large objects that intersect H_m to our separators. Thus our candidate separators are the sets $F_{\text{sep}}(H_i) := F_{\partial}(H_i) \cup F_{\text{large}}$, where F_{large} is the set of all large objects intersecting H_m. We show that our candidate separators are balanced:

Lemma 2.1. *For any $0 \leqslant i \leqslant m$ we have*

$$\max\left(|F_{\text{in}}(H_i) \setminus F_{\text{large}}|, |F_{\text{out}}(H_i) \setminus F_{\text{large}}|\right) < \frac{6^d}{6^d + 1} n.$$

Proof. Consider a hypercube H_i. Because H_0 contains at least $n/(6^d+1)$ objects from F, we immediately obtain

$$\left|F \cap (F_{\text{out}}(H_i) \setminus F_{\text{large}})\right| \leqslant |F \cap F_{\text{out}}(H_0)| \leqslant |F \setminus F_{\text{in}}(H_0)| < \left(1 - \frac{1}{6^d+1}\right) n = \frac{6^d}{6^d+1} n.$$

To bound $\left|F_{\text{in}}(H_i) \setminus F_{\text{large}}\right|$, consider a subdivision of H_i into 6^d sub-hypercubes of edge length $\frac{1}{6}(1+\frac{2i}{m}) \leqslant 1/2$. We claim that any sub-hypercube H_{sub} intersects fewer than $n/(6^d+1)$ small objects from F. To see this, recall that small objects have diameter less than 1/4. Hence, all small objects intersecting H_{sub} are fully contained in a hypercube of edge length less than 1. Since H_0 is a smallest hypercube containing at least $n/(6^d+1)$ objects from F, H_{sub} must thus contain fewer than $n/(6^d+1)$ objects from F, as claimed. Each object in $F_{\text{in}}(H_i)$ intersects at least one of the 6^d sub-hypercubes, so we can conclude that $\left|F_{\text{in}}(H_i) \setminus F_{\text{large}}\right| < \left(6^d/(6^d+1)\right)n$. □

Step 2: Defining the cliques and finding a low-weight separator. Define $F^* := F \setminus (F_{\text{in}}(H_0) \cup F_{\text{out}}(H_m) \cup F_{\text{large}})$. Note that $F_\partial(H_i) \subseteq F^*$ for all i. We partition F^* into *size classes* F^*_s, based on the diameter of the objects. More precisely, for integers s with $1 \leqslant s \leqslant s_{\max}$, where $s_{\max} := \lceil (1-1/d)\log n \rceil - 2$, we define

$$F^*_s := \left\{ o \in F^* : \frac{2^{s-1}}{n^{1/d}} \leqslant \text{diam}(o) < \frac{2^s}{n^{1/d}} \right\}.$$

We furthermore define F^*_0 to be the subset of objects $o \in F^*$ with $\text{diam}(o) < 1/n^{1/d}$. Note that $2^{s_{\max}}/n^{1/d} \geqslant 1/4$, which means that every object in F^* is in exactly one size class.

Each size class can be decomposed into cliques, as follows. Fix a size class F^*_s, with $1 \leqslant s \leqslant s_{\max}$. Since the objects in F are α-fat for a fixed constant $\alpha > 0$, each $o \in F^*_s$ contains a ball of radius $\alpha \cdot (\text{diam}(o)/2) = \Omega(\frac{2^s}{n^{1/d}})$. Moreover, each object $o \in F^*_s$ lies fully or partially inside the outer hypercube H_m, which has edge length 3. This implies we can stab all objects in F^*_s using a set P_s of $O((\frac{n^{1/d}}{2^s})^d)$ points. Thus there exists a decomposition $\mathcal{C}(F^*_s)$ of F^*_s consisting of $O(\frac{n}{2^{sd}})$ cliques. In a similar way we can argue that there exists a decomposition $\mathcal{C}(F_{\text{large}})$ of F_{large} into $O(1)$ cliques. For F^*_0 the argument does not work since objects in F^*_0 can be arbitrarily small. Hence, we create a singleton clique for each object in F^*_0. Together with the decompositions of the size classes F^*_s and of F_{large} we thus obtain a decomposition $\mathcal{C}(F^*)$ of F^* into cliques.

A decomposition of $F_{\text{sep}}(H_i)$ into cliques is induced by $\mathcal{C}(F^*)$, which we denote by $\mathcal{C}(F_{\text{sep}}(H_i))$. Thus, for a given weight function γ, the weight of $F_{\text{sep}}(H_i)$ is $\sum_{C \in \mathcal{C}(F_{\text{sep}}(H_i))} \gamma(|C|)$. Our goal is now to show that at least one of the separators $F_{\text{sep}}(H_i)$ has weight $O(n^{1-1/d})$, when $\gamma(t) = O(t^{1-1/d-\varepsilon})$ for some $\varepsilon > 0$. To this end we will bound the total weight of all separators $F_{\text{sep}}(H_i)$ by $O(n)$. Using that the number of separators is $n^{1/d}$ we then obtain the desired result.

Lemma 2.2. *If $\gamma(t) = O(t^{1-1/d-\varepsilon})$ for some $\varepsilon > 0$ then $\sum_{i=1}^m \text{weight}(F_{\text{sep}}(H_i)) = O(n)$.*

Proof. First consider the cliques in $\mathcal{C}(F^*_0)$, which are singletons. Since objects in F^*_0 have diameter less than $1/n^{1/d}$, which is the distance between consecutive hypercube H_i and H_{i+1}, each such object is in at most one set $F_\partial(H_i)$. Hence, its contribution to the total weight $\sum_{i=1}^m \text{weight}(F_{\text{sep}}(H_i))$ is $\gamma(1) = O(1)$. Together, the cliques in $\mathcal{C}(F^*_0)$ thus contribute $O(n)$ to the total weight.

Next, consider $\mathcal{C}(F_{\text{large}})$. It consists of $O(1)$ cliques. In the worst case each clique appears in all sets $F_\partial(H_i)$. Hence, their total contribution to $\sum_{i=1}^m \text{weight}(F_{\text{sep}}(H_i))$ is bounded by $O(1) \cdot \gamma(n) \cdot n^{1/d} = O(n)$.

Now consider a set $\mathcal{C}(F^*_s)$ with $1 \leqslant s \leqslant s_{\max}$. A clique $C \in \mathcal{C}(F^*_s)$ consists of objects of diameter at most $2^s/n^{1/d}$ that are stabbed by a common point. Since the distance between consecutive hypercubes H_i and H_{i+1} is $1/n^{1/d}$, this implies that C contributes to the weight of $O(2^s)$ separators $F_{\text{sep}}(H_i)$. The contribution to the weight of a single separator is at most $\gamma(|C|)$. (It can be less than $\gamma(|C|)$ because not all objects in C need to intersect ∂H_i.) Hence, the total weight contributed by all cliques, which equals the total weight of all separators, is

$$\sum_{s=1}^{s_{\max}} \sum_{C \in \mathcal{C}(F^*_s)} (\text{weight contributed by } C) \leqslant \sum_{s=1}^{s_{\max}} \sum_{C \in \mathcal{C}(F^*_s)} 2^s \gamma(|C|) = \sum_{s=1}^{s_{\max}} \left(2^s \sum_{C \in \mathcal{C}(F^*_s)} \gamma(|C|) \right).$$

Next we wish to bound $\sum_{C \in \mathcal{C}(F^*_s)} \gamma(|C|)$. Define $n_s := |F^*_s|$ and observe that $\sum_{s=1}^{s_{\max}} n_s \leqslant n$. Recall that $\mathcal{C}(F^*_s)$ consists of $O(n/2^{sd})$ cliques, that is, of at most $cn/2^{sd}$ cliques for some constant c. To make the formulas below more readable we assume $c = 1$ (so we can omit c), but it is easily checked that this does not influence the final result asymptotically. Similarly, we will be using $\gamma(t) = t^{1-1/d-\varepsilon}$ instead of $\gamma(t) = O(t^{1-1/d-\varepsilon})$. Because γ is positive and concave, the sum $\sum_{C \in \mathcal{C}(F^*_s)} \gamma(|C|)$ is maximized when the number of cliques is maximal, namely $\min(n_s, n/2^{sd})$, and when the objects are distributed as evenly as possible over the cliques. Hence,

$$\sum_{C \in \mathcal{C}(F^*_s)} \gamma(|C|) \leqslant \begin{cases} n_s & \text{if } n_s \leqslant n/2^{sd} \\ (n/2^{sd}) \cdot \gamma\left(\frac{n_s}{n/2^{sd}}\right) & \text{otherwise} \end{cases}$$

We now split the set $\{1, \ldots, s_{\max}\}$ into two index sets S_1 and S_2, where S_i contains all indices s such that $n_s \leqslant n/2^{sd}$, and S_2 contains all remaining indices. Thus

$$\sum_{s=1}^{s_{\max}} \left(2^s \sum_{C \in \mathcal{C}(F^*_s)} \gamma(|C|) \right) = \sum_{s \in S_1} \left(2^s \sum_{C \in \mathcal{C}(F^*_s)} \gamma(|C|) \right) + \sum_{s \in S_2} \left(2^s \sum_{C \in \mathcal{C}(F^*_s)} \gamma(|C|) \right) \tag{1}$$

The first term in (1) can be bounded by

$$\sum_{s \in S_1} \left(2^s \sum_{C \in \mathcal{C}(F^*_s)} \gamma(|C|) \right) \leqslant \sum_{s \in S_1} 2^s n_s \leqslant \sum_{s \in S_1} 2^s (n/2^{sd}) = n \sum_{s \in S_1} 1/2^{s(d-1)} = O(n),$$

where the last step uses that $d \geqslant 2$. For the second term we get

$$\begin{aligned}\sum_{s\in S_2}\left(2^s \sum_{C\in C(F_s^*)} \gamma(|C|)\right) &\leqslant \sum_{s\in S_2}\left(2^s(n/2^{sd})\cdot\gamma\left(\frac{n_s}{n/2^{sd}}\right)\right)\\ &\leqslant \sum_{s\in S_2}\left(\frac{n}{2^{s(d-1)}}\cdot\left(\frac{n_s 2^{sd}}{n}\right)^{1-1/d-\varepsilon}\right)\\ &\leqslant n\sum_{s\in S_2}\left(\frac{n_s}{n}\right)^{1-1/d-\varepsilon}\frac{1}{2^{sd\varepsilon}}\\ &\leqslant n\sum_{s\in S_2}\left(\frac{1}{2^{d\varepsilon}}\right)^s\\ &= O(n).\end{aligned}$$

□

The two lemmas above imply the existence of the separator of Theorem 1.1 with the desired balance and weight. Computing the separator in polynomial time can be done in a more or less brute-force manner; this is explained in the full version.

Corollary 2.3. *Let F be a set of n fat objects in $\mathbb{R}^d$, where d is a constant. Then* Independent Set *on the intersection graph $G[F]$ can be solved in $2^{O(n^{1-1/d})}$ time.*

Proof. Let $\gamma(t) := \log(t+1)$, and compute a separator F_{sep} for $G[F]$ using Theorem 1.1. For each subset $S_{\text{sep}} \subseteq F_{\text{sep}}$ of independent (that is, pairwise non-adjacent) vertices we find the largest independent set S of G such that $S \supseteq S_{\text{sep}}$, by removing the closed neighborhood of S_{sep} from G and recursing on the remaining connected components. Finally, we report the largest of all these independent sets. Because a clique $C \in C(F_{\text{sep}})$ can contribute at most one vertex to S_{sep}, we have that the number of candidate sets S_{sep} is at most

$$\prod_{C\in C(F_{\text{sep}})}(|C|+1) = 2^{\sum_{C\in C(F_{\text{sep}})}\log(|C|+1)} = 2^{O(n^{1-1/d})}.$$

Since all components on which we recurse have at most $(6^d/(6^d+1))n$ vertices, the running time $T(n)$ satisfies

$$T(n) = 2^{O(n^{1-1/d})}T((6^d/(6^d+1))n) + \text{poly}(n),$$

which solves to $T(n) = 2^{O(n^{1-1/d})}$. □

2.2 An Algorithmic Framework for Similarly-Sized Fat Objects

We restrict our attention to *similarly-sized* fat objects. More precisely, we consider intersection graphs of sets F of objects such that, for each $o \in F$, there are balls B_{in} and B_{out} in $\mathbb{R}^d$ such that $B_{\text{in}} \subseteq F \subseteq B_{\text{out}}$, and $\text{radius}(B_{\text{in}}) = \alpha$ and $\text{radius}(B_{\text{out}}) = 1$ for some fatness constant $\alpha > 0$. The restriction to similarly-sized objects makes it possible to construct a clique cover of F with the following property: if we consider the intersection graph $G[F]$ where the cliques are contracted to single vertices, then the contracted graph has constant degree. Moreover, the contracted graph admits a tree decomposition whose weighted treewidth is $O(n^{1-1/d})$. This tool allows us to solve many problems on intersection graphs of similarly-sized fat objects.

Our tree-decomposition construction uses the separator theorem from the previous subsection. That theorem also states that we can compute the separator for $G[F]$ in polynomial time, provided we are given F. However, finding the separator if we are only given the graph and not the underlying set F is not easy. Note that deciding whether a graph is a unit-disk graph is already ER-complete [17]. Nevertheless, we show that for similarly-sized fat objects we can find certain tree decompositions with the desired properties, purely based on the graph $G[F]$.

κ-partitions, $\mathcal{P}$-contractions, and separators. Let $G = (V, E)$ be the intersection graph of an (unknown) set F of similarly-sized fat objects, as defined above. The separators in the previous section use cliques as basic components. We need to generalize this slightly, by allowing connected unions of a constant number of cliques as basic components. Thus we define a *κ-partition* of G as a partition $\mathcal{P} = (V_1, \ldots, V_k)$ of V such that every partition class V_i induces a connected subgraph that is the union of at most κ cliques. Note that a 1-partition corresponds to a clique cover of G.

Given a κ-partition $\mathcal{P}$ of G we define the *$\mathcal{P}$-contraction* of G, denoted by $G_{\mathcal{P}}$, to be the graph obtained by contracting all partition classes V_i to single vertices and removing loops and parallel edges. In many applications it is essential that the $\mathcal{P}$-contraction we work with has maximum degree bounded by a constant. From now on, when we speak of the degree of a κ-partition $\mathcal{P}$ we refer to the degree of the corresponding $\mathcal{P}$-contraction.

The following theorem and its proof are very similar to Theorem 1.1, but it applies only for similarly-sized objects because of the degree bound on $G_{\mathcal{P}}$. The other main difference is that the separator is defined on the $\mathcal{P}$-contraction of a given κ-partition, instead of on the intersection graph G itself.

Theorem 2.4. *Let $G = (V, E)$ be the intersection graph of a set of n similarly-sized fat objects in $\mathbb{R}^d$, and let γ be a weight function such that $\gamma(t) = O(t^{1-1/d-\varepsilon})$, for constants $d \geqslant 2$ and $\varepsilon > 0$. Suppose we are given a κ-partition $\mathcal{P}$ of G such that $G_{\mathcal{P}}$ has maximum degree at most Δ, where κ and Δ are constants. Then there exists a $(6^d/(6^d+1))$-balanced separator for $G_{\mathcal{P}}$ of weight $O(n^{1-1/d})$.*

The following lemma shows that a partition $\mathcal{P}$ as needed in Theorem 2.4 can be computed even in the absence of geometric information.

Lemma 2.5. *Let $G = (V, E)$ be the intersection graph of an (unknown) set of n similarly-sized fat objects in $\mathbb{R}^d$ for some constant $d \geqslant 2$. There there exist constants κ and Δ such that a κ-partition $\mathcal{P}$ for which $G_{\mathcal{P}}$ has maximum degree Δ can be computed in polynomial time.*

Proof. Let $S \subseteq V$ be a maximal independent set in G (that is, it is inclusion-wise maximal). We assign each vertex $v \in V \setminus S$ to an arbitrary vertex $s \in S$ that is a neighbor of v; such a vertex s always exists since S is maximal. For each vertex $s \in S$ define $V_s := \{s\} \cup \{v \in V \setminus S : v \text{ is assigned to } s\}$. We prove that the partition $\mathcal{P} := \{V_s : s \in S\}$, which can be computed in polynomial time, has the desired properties.

Let o_v denote the (unknown) object corresponding to a vertex $v \in V$, and for a partition class V_s define $U(V_s) := \bigcup_{v\in V_s} o_v$. We call $U(V_s)$ a *union-object*. Let $\mathcal{U}_S := \{U(V_s) : s \in S\}$. Because

the objects defining G are similarly-sized and fat, there are balls $B_{\text{in}}(o_v)$ of radius $\alpha = \Omega(1)$ and $B_{\text{out}}(o_v)$ of radius 1 such that $B_{\text{in}}(o_v) \subseteq o_v \subseteq B_{\text{out}}(o_v)$.

Now observe that each union-object $U(V_s)$ is contained in a ball of radius 3. Hence, we can stab all balls $B_{\text{in}}(o_v)$, $v \in V_s$ using $O(1)$ points, which implies that $\mathcal{P}$ is a κ-partition for some $\kappa = O(1)$.

To prove that the maximum degree of $G_{\mathcal{P}}$ is $O(1)$, we note that any two balls $B_{\text{in}}(s)$, $B_{\text{in}}(s')$ with $s, s' \in S$ are disjoint (because S is an independent set in G). Since all union-objects $U(s')$ that intersect $U(s)$ are contained in a ball of radius 9, an easy packing argument now shows that $U(s)$ intersects $O(1)$ union-objects $U(s)$. Hence, the node in $G_{\mathcal{P}}$ corresponding to V_s has degree $O(1)$. □

Weighted tree decompositions for $\mathcal{P}$-contractions. Recall that a *tree decomposition* of a graph $G = (V, E)$ is a pair (T, σ) where T is a tree and σ is a mapping from the vertices of T to subsets of V called *bags*, with the following properties. Let $\text{Bags}(T, \sigma) := \{\sigma(u) : u \in V(T)\}$ be the set of bags associated to the vertices of T. Then we have: (1) For any vertex $u \in V$ there is at least one bag in $\text{Bags}(T, \sigma)$ containing it. (2) For any edge $(u, v) \in E$ there is at least one bag in $\text{Bags}(T, \sigma)$ containing both u and v. (3) For any vertex $u \in V$ the collection of bags in $\text{Bags}(T, \sigma)$ containing u forms a subtree of T.

The *width* of a tree decomposition is the size of its largest bag minus 1, and the *treewidth* of a graph G equals the minimum width of a tree decomposition of G. We will need the notion of *weighted treewidth* [26]. Here each vertex has a weight, and the *weighted width* of a tree decomposition is the maximum over the bags of the sum of the weights of the vertices in the bag (note: without the -1). The *weighted treewidth* of a graph is the minimum weighted width over its tree decompositions.

Now let $\mathcal{P} = (V_1, \ldots, V_k)$ be a κ-partition of a given graph G which is the intersection graph of similarly-sized fat objects, and let γ be a given weight function on partition classes. We apply the concept of weighted treewidth to $G_{\mathcal{P}}$, where we assign each vertex V_i of $G_{\mathcal{P}}$ a weight $\gamma(|V_i|)$. Because we have a separator for $G_{\mathcal{P}}$ of low weight by Theorem 2.4, we can prove a bound on the weighted treewidth of $G_{\mathcal{P}}$ using standard techniques.

Lemma 2.6. *Let $\mathcal{P}$ be a κ-partition of a family of similarly-sized fat objects such that $G_{\mathcal{P}}$ has maximum degree at most Δ, where κ and Δ are constants. Then the weighted treewidth of $G_{\mathcal{P}}$ is $O(n^{1-1/d})$ for any weight function γ with $\gamma(t) = O(t^{1-1/d-\varepsilon})$.*

By combining Lemmas 2.5 and 2.6 we can obtain a κ-partition such that $G_{\mathcal{P}}$ has constant degree, and such that the weighted treewidth of $G_{\mathcal{P}}$ is as desired. In the full version we show how to use existing algorithms for computing tree decompositions of approximately optimal width to obtain the following main theorem.

Theorem 2.7. *Let $G = (V, E)$ be the intersection graph of an (unknown) set of n similarly-sized α-fat objects in $\mathbb{R}^d$, and let γ be a weight function such that $1 \leqslant \gamma(t) = O(t^{1-1/d-\varepsilon})$, for constants $d \geqslant 2$, $\alpha > 0$, and $\varepsilon > 0$. Then there exist constants κ and Δ such that there is a κ-partition $\mathcal{P}$ with the following properties: (i) $G_{\mathcal{P}}$ has maximum degree at most Δ, and (ii) $G_{\mathcal{P}}$ has weighted treewidth $O(n^{1-1/d})$. Moreover, such a partition $\mathcal{P}$ and a corresponding tree decomposition of weight $O(n^{1-1/d})$ can be computed in $2^{O(n^{1-1/d})}$ time.*

2.3 Basic Algorithmic Applications

In this section, we give examples of how κ-partitions and weighted tree decompositions can be used to obtain subexponential-time algorithms for classical problems on geometric intersection graphs.

Given a κ-partition $\mathcal{P}$ and a weighted tree decomposition of $G_{\mathcal{P}}$ of width τ, we note that there exists a nice tree decomposition of G (i.e., a "traditional", non-partitioned tree decomposition) with the property that each bag is a subset of the union of a number of partition classes, such that the total weight of those classes is at most τ. This can be seen by creating a nice version of the weighted tree decomposition of $G_{\mathcal{P}}$, and then replacing every introduce/forget bag (that introduces/forgets a class of the partition) by a series of introduce/forget bags (that introduce/forget the individual vertices). We call such a decomposition a *traditional tree decomposition*. Using such a decomposition, it becomes easy to give algorithms for problems for which we already have dynamic-programming algorithms operating on nice tree decompositions. We can re-use the algorithms for the leaf, introduce, join and forget cases, and either show that the number of partial solutions remains bounded (by exploiting the properties of the underlying κ-partition) or show that we can discard some irrelevant partial solutions.

We present several applications for our framework, resulting in $2^{O(n^{1-1/d})}$ algorithms for various problems. In addition to the Independent Set algorithm for fat objects based on our separator, we also give a robust algorithm for similarly sized fat objects. This adds robustness compared to the state of the art [23]. In the rest of the applications, our algorithms work on intersection graphs of d-dimensional similarly sized fat objects; this is usually a larger graph class than what has been studied. We have non-robust algorithms for Hamiltonian Path and Hamiltonian Cycle; this is a simple generalization from the algorithm for unit disks that has been known before [11, 18]. For Feedback Vertex Set, we give a robust algorithm with the same running time improvement, over a non-robust algorithm that works in 2-dimensional unit disk graphs [11]. For r-Dominating Set, we give a robust algorithm for $d \geq 2$, which is the first subexponential algorithm in dimension $d \geq 3$, and the first robust subexponential for $d = 2$ [22]. (The algorithm in [22] is for Dominating Set in unit disk graphs.) Finally, we give robust algorithms for Steiner Tree, r-Dominating Set, Connected Vertex Cover, Connected Feedback Vertex Set and Connected Dominating Set, which are – to our knowledge – also the first subexponential algorithms in geometric intersection graphs for these problems.

In the following, we let t refer to a node of the tree decomposition T, let X_t denote the set of vertices in the bag associated with t, and let $G[t]$ denote the subgraph of G induced by the vertices appearing in bags in the subtree of T rooted at t. We fix our weight function to be $\gamma(k) = \log(k + 1)$.

Theorem 2.8. *Let $\gamma(k) = \log(k+1)$. If a κ-partition and a weighted tree decomposition of width at most τ is given,* Independent Set *and* Vertex Cover *can be solved in time $2^{\kappa\tau} n^{O(1)}$.*

Proof. A well-known algorithm (see, e.g., [8]) for solving Independent Set on graphs of bounded treewidth, computes, for each bag t and subset $S \subseteq X_t$, the maximum size $c[t, S]$ of an independent subset $\hat{S} \subset G[t]$ such that $\hat{S} \cap X_t = S$.

An independent set never contains more than one vertex of a clique. Therefore, since X_t is a subset of the union of partition classes $V_i, i \in \sigma(b)$, and from each partition class we can select at most κ vertices (one vertex from each clique), the number of subsets $\hat{S}$ that need to be considered is at most $\prod_{i\in\sigma(b)}(|V_i|+1)^\kappa = \exp\left(\sum_{i\in\sigma(b)} \kappa \log(|V_i|+1)\right) = 2^{\kappa\tau}$.

Applying the standard algorithm for Independent Set on a traditional tree decomposition, using the fact that only solutions that select at most one vertex from each clique get a non-zero value, we obtain the claimed algorithm. Minimum vertex cover is simply the complement of maximum independent set. □

Corollary 2.9. *Let d be a fixed constant. Then* Independent Set *and* Vertex Cover *can be solved in $2^{O(n^{1-1/d})}$ time on intersection graphs of similarly-sized fat objects in $\mathbb{R}^d$, even if the geometric representation is not given.*

In the remainder of this section, because we need additional assumptions that are derived from the properties of intersection graphs, we state our results in terms of algorithms operating directly on intersection graphs. However, note that underlying each of these results is an algorithm operating on a weighted tree decomposition of the contracted graph.

To obtain the algorithm for Independent Set, we exploited the fact that we can select at most one vertex from each clique, and that thus, we can select at most κ vertices from each partition class. For Dominating Set, our bound for the treewidth is however not enough. Instead, we need the following, stronger result, which states that the weight of a bag in the decomposition can still be bounded by $O(n^{1-1/d})$, even if we take the weight to be the total weight of the classes in the bag *and* that of their distance-r neighbors:

Theorem 2.10. *Let G be an intersection graph of n similarly-sized fat objects in $\mathbb{R}^d$, and let $r \geqslant 1$ be a constant. For any weight function γ, there exists a constant $\kappa = O(1)$ such that G has a κ-partition $\mathcal{P}$ and a corresponding $G_{\mathcal{P}}$ of maximum degree at most Δ, where $G_{\mathcal{P}}$ has a weighted tree decomposition with the additional property that for any bag b, the total weight of the partition classes $\{V_i \in \mathcal{P} \mid$ (some vertex in) V_i is within distance r of some $V_j \in \sigma(b)\}$ is $O(n^{1-1/d})$.*

Proof. As per Theorem 2.7, there exist constants $\kappa, \Delta = O(1)$ such that G has a κ-partition in which each class of the partition is adjacent to at most Δ other classes.

We now create a new geometric intersection graph G', which is made by copying each vertex (and its corresponding object) at most κ^r times. We create the following κ^r-partition $\mathcal{P}^r$: for each class V_i of the original partition, create a class that contains a copy of the vertices from V_i and copies of the vertices from the classes within distance at most r from V_i. This graph G^r has at most $\kappa^r n = O(n)$ vertices, and it is an intersection graph of similarly-sized objects; furthermore, the set $\mathcal{P}^r$ has low union ply. Therefore, we can find a weighted tree decomposition of $G^r_{\mathcal{P}^r}$ of width $O(n^{1-1/d})$ by Lemma 2.6.

This decomposition can also be used as a decomposition for the original κ-partition, by replacing each partition class with the corresponding original partition class. □

Theorem 2.11. *Let r and d be fixed constants. Then r-*Dominating Set *can be solved in $2^{O(n^{1-1/d})}$ time on intersection graphs of similarly-sized fat objects in $\mathbb{R}^d$.*

Proof. We first present the argument for Dominating Set. It is easy to see that from each partition class, we need to select at most $\kappa^2(\Delta+1)$ vertices: each partition class can be partitioned into at most κ cliques, and each of these cliques is adjacent to at most $\kappa(\Delta+1)$ other cliques. If we select at least $\kappa(\Delta+1)+1$ vertices from a clique, we can instead select only one vertex from the clique, and select at least one vertex from each neighboring clique.

We once again proceed by dynamic programming on a traditional tree decomposition (see e.g. [8] for an algorithm solving Dominating Set using tree decompositions). However, rather than needing just two states per vertex (in the solution or not), we need three: a vertex can be either in the solution, not in the solution and not dominated, or not in the solution and dominated. After processing each bag, we discard partial solutions that select more than $\kappa^2(\Delta+1)$ vertices from any class of the partition. Note that all vertices of each partition class are introduced before any are forgotten, so we can guarantee we do indeed never select more than $\kappa^2(\Delta+1)$ vertices from each partition class.

The way vertices outside the solution are dominated or not is completely determined by the vertices that are in the solution and are neighbours of the vertices in the bag. While the partial solution does not track this explicitly for vertices that are forgotten, by using the fact that we need to select at most $\kappa\Delta$ vertices from each class of the partition, and the fact that Theorem 2.10 bounds the total weight of the neighbourhood of the partition classes in a bag, we see that there are at most $\Pi_i(|V_i|+1)^{\kappa^2(\Delta+1)} = \exp(\kappa^2(\Delta+1)\sum_i \log(|V_i|+1)) = 2^{O(n^{1-1/d})}$, where the product (resp., sum) is taken over all partition classes V_i that appear in the current bag or are a neighbors of such a class.

For the generalization where $r > 1$, the argument that we need to select at most $\kappa(\Delta+1)$ vertices from each clique still holds: moving a vertex from a clique with more than $\kappa(\Delta+1)$ vertices selected to an adjacent clique only decreases the distance to any vertices it helps cover. The dynamic programming algorithm needs, in a partial solution, to track at what distance from a vertex in the solution each vertex is. This, once again, is completely determined by the solution in partition classes at distance at most r; the number of such cases we can bound using Theorem 2.10. □

2.4 Rank-Based Approach

To illustrate how our algorithmic framework can be combined with the rank-based approach, we now give an algorithm for Steiner Tree. We consider the following variant of Steiner Tree:

Steiner Tree
Input: A graph $G = (V, E)$, a set of terminal vertices $K \subseteq V$ and integer s.
Question: Decide if there is a vertex set $X \subseteq V$ of size at most s, such that $K \subseteq X$, and X induces a connected subgraph of G.

We only consider the unweighted variant of Steiner Tree, as the weighted Steiner Tree problem is NP-complete, even on a clique (so we should not expect Theorem 2.12 to hold for the weighted case).

Theorem 2.12. *Let $d \in \mathbb{Z}_+$ be a constant. Then* Steiner Tree *can be solved in $2^{O(n^{1-1/d})}$ time on intersection graphs of d-dimensional similarly-sized fat objects.*

Proof. The algorithm works by dynamic programming on a traditional tree decomposition. The leaf, introduce, join and forget cases can be handled as they are in the conventional algorithm for Steiner Tree on tree decompositions, see e.g. [5]. However, after processing each bag, we can reduce the number of partial solutions that need to be considered by exploiting the properties of the underlying κ-partition.

To this end, we first need a bound on the number of vertices that can be selected from each class of the κ-partition $\mathcal{P}$.

Lemma 2.13. *Let C be a clique in a κ-sized clique cover of a partition class $V_i \in \mathcal{P}$. Then any optimal solution X contains at most $\kappa(\Delta+1)$ vertices from C that are not also in K. Furthermore, any optimal solution thus contains at most $\kappa^2(\Delta+1)$ vertices (that are not also in K) from each partition class.*

Proof. To every vertex $v \in (C \cap X) \setminus K$ we greedily assign a *private neighbor* $u \in X \setminus C$ such that u is adjacent to v and u is not adjacent to any other previously assigned private neighbor. If this process terminates before all vertices in $(C \cap X) \setminus K$ have been assigned a private neighbor, then the remaining vertices are redundant and can be removed from the solution.

We now note that since the neighborhood of C can be covered by at most $\kappa(\Delta+1)$ cliques, this gives us an upper bound on the number of private neighbors that can be assigned and thus bounds the number of vertices that can be selected from any partition class. □

The algorithm for Steiner Tree presented in [5] is for the weighted case, but we can ignore the weights by setting them to 1. A partial solution is then represented by a subset $\hat{S} \subseteq X_t$ (representing the intersection of the partial solution with the vertices in the bag), together with an equivalence relation on $\hat{S}$ (which indicates which vertices are in the same connected component of the partial solution).

Since we select at most $\kappa^2(\Delta+1)$ vertices from each partition class, we can discard partial solutions that select more than this number of vertices from any partition class. Then the number of subsets S considered is at most

$$\prod_{i \in \sigma(b)} (|V_i|+1)^{\kappa^2(\Delta+1)} = \exp\left(\kappa^2(\Delta+1) \cdot \sum_{i \in \sigma(b)} \log(|V_i|+1)\right) \leqslant \exp\left(\kappa^2(\Delta+1)\tau\right).$$

For any such subset $\hat{S}$, the number of possible equivalence relations is $2^{\Theta(|\hat{S}| \log |\hat{S}|)}$. However, the rank-based approach [5] provides an algorithm called "*reduce*" that, given a set of equivalence relations[2] on $\hat{S}$, outputs a representative set of equivalence relations of size at most $2^{|\hat{S}|}$. Thus, by running the reduce algorithm after processing each bag, we can keep the number of equivalence relations considered single exponential.

Since $|\hat{S}|$ is also $O(\kappa^2(\Delta+1)\tau)$ (we select at most $\kappa^2(\Delta+1)$ vertices from each partition class and each bag contains at most τ partition classes), for any subset $\hat{S}$, the rank-based approach guarantees that we need to consider at most $2^{O(\kappa^2(\Delta+1)\tau)}$ representative equivalence classes of $\hat{S}$ (for each set $\hat{S}$). □

[2] What we refer to as "equivalence relation", [5] refers to as "partition".

3 THE LOWER-BOUND FRAMEWORK

The goal of this section is to provide a general framework to exclude algorithms with running time $2^{o(n^{1-1/d})}$ in intersection graphs. To get the strongest results, we show our lower bounds where possible for a more restricted graph class, namely subgraphs of d-dimensional induced grid graphs. Induced grid graphs are intersection graphs of unit balls, so they are a subclass of intersection graphs of similarly sized fat objects. We need to use a different approach for $d = 2$ than for $d > 2$; this is because of the topological restrictions introduced by planarity. Luckily, the difference between $d = 2$ and $d > 2$ is only in the need of two different "embedding theorems"; when applying the framework to specific problems, the same gadgetry works both for $d = 2$ and for $d > 2$. In particular, in $\mathbb{R}^2$, constructing crossover gadgets is not necessary with our framework. To apply our framework, we need a graph problem $\mathcal{P}$ on grid graphs in $\mathbb{R}^d$, $d \geqslant 2$. Suppose that $\mathcal{P}$ admits a reduction from 3-SAT using constant size variable and clause gadgets and a wire gadget, whose size is a constant multiple of its length. Then the framework implies that $\mathcal{P}$ has no $2^{o(n^{1-1/d})}$ time algorithm in d-dimensional grid graphs for all $d \geq 2$, unless ETH fails. We remark that such gadgets can often be obtained by looking at classical NP-hardness proofs in the literature, and introducing minor tweaks if necessary.

3.1 Lower Bounds in Two Dimensions

To prove lower bounds in two dimensional grids, we introduce an intermediate problem.

We denote by $G^2(n_1, n_2)$ the two dimensional grid graph with vertex set $[n_1] \times [n_2]$. We say that a graph H is *embeddable* in $G^2(n_1, n_2)$ if it is a topological minor of $G^2(n_1, n_2)$, i.e., if H has a subdivision that is a subgraph of $G^2(n_1, n_2)$. Finally, for a given 3-CNF formula ϕ, its incidence graph G_ϕ is the bipartite graph on its variables and clauses, where a variable vertex and a clause vertex are connected by an edge if the variable appears in the clause.

A CNF formula ϕ with clause size at most 3 and where each variable occurs at most 3 times is called a $(3,3)$-CNF formula. Note that in such formulas the number of clauses and variables is within constant factor of each other. The $(3,3)$-SAT problem asks to decide the satisfiability of a $(3,3)$-CNF formula.

Proposition 3.1. *There is no $2^{o(n)}$ algorithm for $(3,3)$-SAT unless ETH fails.*

Our intermediate problem, Grid Embedded SAT, asks to determine the satisfiability of a $(3,3)$-CNF formula whose incidence graph is embedded in a $n \times n$ grid:

> Grid Embedded SAT
> **Input:** A $(3,3)$-CNF formula ϕ together with an embedding of its incidence graph G_ϕ in $G_2(n,n)$.
> **Question:** Is there a satisfying assignment?

Theorem 3.2. *Grid Embedded SAT has no $2^{o(n)}$ algorithm, unless ETH fails.*

3.2 Lower Bounds in Higher Dimensions – Cube Wiring

For an integer n, let $[n] = \{1, \ldots, n\}$. For a vector $\mathbf{n} := (n_1, \ldots, n_d)$ in $\mathbb{Z}_+^d$, let $\mathrm{Box}_d(\mathbf{n}) = [n_1] \times \cdots \times [n_d]$. Let $G^d(\mathbf{n})$ be the graph whose vertex set $V(G)$ is $\mathrm{Box}_d(\mathbf{n})$, and where $\mathbf{x}, \mathbf{y} \in V(G)$ are connected if and only if they are at distance 1 in $\mathbb{R}^d$. The integer points of $\mathbb{R}^d$ can be divided into parallel *layers*. The layer at "height" $h \in \mathbb{Z}$ is defined as $\ell(h) = \{\mathbf{x} \in \mathbb{Z}^d \mid x_d = h\}$. Let $\mathrm{Raise}^h : \mathbb{R}^{d-1} \to \mathbb{R}^d$ be the function that maps $\mathbb{R}^{d-1}$ into $\ell(h)$ as follows: $\mathrm{Raise}^h(x_1, \ldots, x_{d-1}) = (x_1, \ldots, x_{d-1}, h)$.

In what follows, $\mathbf{n}$ denotes a $(d-1)$-dimensional vector. Let P, Q be equal-size subsets of $\mathrm{Box}(\mathbf{n})$. Let M be a perfect matching of the graph $G_{P\times Q} := (P \cup Q, P \times Q)$.

We say that *M can be wired in $G^d(c\mathbf{n}, h)$* where c and h are positive integers, if there are vertex-disjoint paths $G^d(c\mathbf{n}, h)$ that connect $\mathrm{Raise}^1(\mathbf{p})$ to $\mathrm{Raise}^h(\mathbf{q})$ for all $(\mathbf{p}, \mathbf{q}) \in M$. Note that $G^d(c\mathbf{n}, h)$ consists of h layers, each of which is a copy of $\mathrm{Box}(c\mathbf{n})$ at a different height.

We will refer to the embedding in $\mathbb{R}^d$ of the path representing a pair $(\mathbf{p}, \mathbf{q})$ as a *wire*, and we define the *length* of a wire as the number of edges on the path. Note that the length of a wire is equal to its Euclidean length, since the edges connect adjacent points of the integer grid.

Theorem 3.3. ***(Cube Wiring Theorem)*** *Let $d \geq 3$, $\mathbf{n} \in \mathbb{Z}_+^{d-1}$, and let P and Q be two equal-size subsets of $\mathrm{Box}_{d-1}(\mathbf{n})$. Let M be a perfect matching in $G_{P\times Q} = (P \cup Q, P \times Q)$. Then M can be wired in $G^d(36\mathbf{n}, h)$, where $h = O(\sum_{i=1}^{d-1} n_i)$, and the length of each wire is $O(d \sum_{i=1}^{d-1} n_i)$.*

To simplify the description, we assume that all coordinates of $\mathbf{n}$ are powers of two, and work inside $\mathrm{Box}_{d-1}(18\mathbf{n})$. Rounding coordinates up to the next power of two gives the stated result inside $\mathrm{Box}_{d-1}(36\mathbf{n})$. Note that $\mathrm{Box}_{d-1}(\mathbf{n})$ is a "corner" of the larger $\mathrm{Box}_{d-1}(18\mathbf{n})$, so the point sets P and Q above are embedded into two such corners within the first and last layer of the grid graph.

Overview of the Proof. We obtain a wiring from P to Q by a divide-and-conquer approach. Let $n_{max} := \max_{i\in[d-1]} n_i$, and without loss of generality, assume that $n_1 = n_{max}$. We split $\mathrm{Box}_{d-1}(\mathbf{n})$ in all layers into two equal-sized sub-boxes, using a hyperplane orthogonal to the x_1-axis. Thus the points $\mathbf{z} \in P \cup Q$ with $\mathbf{z}_1 \leqslant n_{max}/2$ end up in one halfspace, while the points $\mathbf{z}$ with $\mathbf{z}_1 > n_{max}/2$ end up in the other halfspace. We then perform the crucial step, a *rough reordering*, which wires all points from P to points in an intermediate layer ℓ so they end up in the correct halfspace with respect to their target locations in Q. That is, if a point $\mathbf{p}$ and its matching point $\mathbf{q}$ were on different sides in the above split, then we wire $\mathbf{p}$ to a point $\mathbf{p}'$ in ℓ which lies in the same side as $\mathbf{q}$ (Figure 1). Next, we perform a *global movement*, which offsets all the points in the halfspace $\mathbf{x}_1 > n_{max}/2$ by $(8 + 1/2)n_{max}$ in the first coordinate, that is, the points are wired to the halfspace $\mathbf{x}_1 \geqslant 9n_{max}$. The rough reordering and the global movement can be performed in $O(n_{max})$ layers. Recall that we are working inside a $18n_1 \times 18n_2 \times \ldots \times 18n_{d-1} \times c\sum_{i=1}^{d-1} n_i$ grid. The wiring problem in the halfspaces can recursively be solved in their own separate halfspaces, and the size of grid required for this is $18n_1/2 \times 18n_2 \times \ldots \times 18n_{d-1} \times c\sum_{i=1}^{d-1} n_i$. Thus, in the original, twice larger grid, we can recursively solve the wiring problem for both halfspaces *in parallel*. After the recursive steps are finished, we have the points arranged as they should be in Q but spread out in $\mathrm{Box}_{d-1}(18\mathbf{n})$, so we *compress* it back to their true targets in $\mathrm{Box}_{d-1}(\mathbf{n})$. We will provide a more rigorous analysis later, but for now, note that after at most d rough reorderings (taking $O(dn_{max})$ layers), n_{max} will have halved. Since the number of layers required for each halving of n_{max} decreases by half each iteration, we see that the wiring can be accomplished in $O(dn_{max})$ layers.

To perform a rough reordering, we first separate the points of P into three groups: those that are already in the correct halfspace, those that need to move from the halfspace $\mathbf{x}_1 \leqslant n_{max}/2$ to the halfspace $\mathbf{x}_1 > n_{max}/2$ (and we say that the wires corresponding to those points need to be *pushed*) and those that need to move in the opposite direction (whose wires must be *pulled*). To avoid conflicts between these movements, we do them in different subgrids. An (a, b)-subgrid consists of those points whose coordinates are equal to a modulo b, together with the points of which at most one coordinate differs from a modulo b. Note that the former points make up the "vertices" of the (a, b) subgrid, and these are connected by paths of length b, the "edges" of the subgrid. We perform pushing in the $(1, 3)$-subgrid, pulling in the $(2, 3)$-subgrid, and the points that do not need to move stay in the $(0, 3)$-subgrid. Notice that we use $d \geq 3$ here; for $d \leqslant 2$, these subgrids are not disjoint.

In what follows, we introduce some further concepts needed for the proof, together with the required lemmas. The proof of these lemmas can be found in the full version.

Rearrangement lemmas. We begin by defining a discrete version of *compression* and *magnification*:

$$\mathrm{comp}_k, \mathrm{mag}_k^r : \mathbb{Z}^{d-1} \to \mathbb{Z}^{d-1},$$
$$\mathrm{comp}_k(x_1, \ldots, x_{d-1}) = \left(\left\lfloor \frac{x_1 - 1}{k} \right\rfloor, \ldots, \left\lfloor \frac{x_{d-1} - 1}{k} \right\rfloor\right),$$
$$\mathrm{mag}_k^r(x_1, \ldots, x_{d-1}) = (kx_1 + r, \ldots, kx_{d-1} + r).$$

We often use the set version of some functions, so for example if P is a point set, then let $\mathrm{comp}_k(P) = \{\mathrm{comp}_k(\mathbf{p}) \mid \mathbf{p} \in P\}$.

We can subdivide $\mathbb{Z}^{d-1}$ into small hypercubes of side length t, the vertices of which we call *t-cells*. More precisely, points $\mathbf{p}, \mathbf{p}' \in \mathbb{Z}^{d-1}$ belong to the same t-cell if and only if $\mathrm{comp}_t(\mathbf{p}) = \mathrm{comp}_t(\mathbf{p}')$.

Definition 3.4. Let k be a positive integer, and consider a point set $P \subseteq \mathbb{Z}^{d-1}$. The set P is *k-spaced* if there is an integer $0 \leqslant r < k$ such that for any $x = (x_1, \ldots, x_{d-1}) \in P$ we have $x_i \equiv r \mod k$ for all $i = 1, \ldots, d-1$. A point set $P \subseteq \mathbb{Z}^d$ is *quasi-k-spaced* if it has at most one point in each k-cell.

Lemma 3.5. *It is possible to make local and global movements in the following sense.*

(1) (Local movement) *Let P and Q be two quasi-k-spaced subsets of $\mathrm{Box}_{d-1}(k\mathbf{n})$, which have points in the same k-cells, i.e., $\mathrm{comp}_k(P) = \mathrm{comp}_k(Q)$. Then $M = \{(\mathbf{p}, \mathbf{q}) \mid \mathrm{comp}_k(\mathbf{p}) = \mathrm{comp}_k(\mathbf{q})\}$ can be wired in $G^d(k\mathbf{n}, 3)$, while keeping each wire within its k-cell of origin in all layers, and the length of each wire is $O(kd)$.*

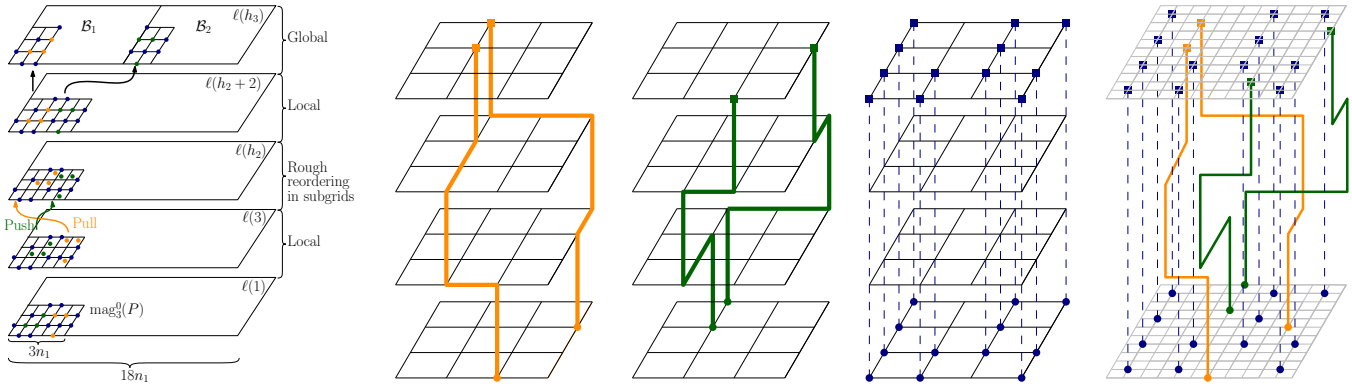

Figure 1: Left: One step in the divide and conquer approach. Right (four pictures): Schematic pictures of rough reordering in disjoint subgrids, according to first coordinates. The orange wires are pulled, the green ones are pushed. The blue wires do not need reordering. The three subgrids are weaved together in a finer grid (rightmost picture).

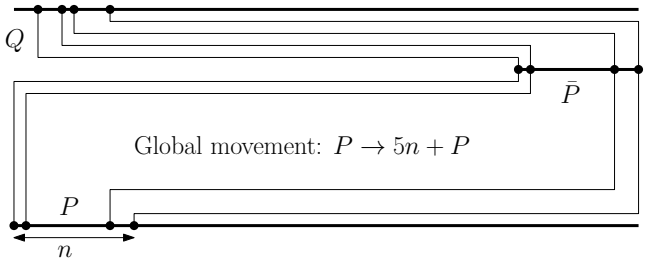

Figure 2: Pushing/pulling lemma in 2 dimensions, using two global movements.

(2) (Global movement) *Let $P \subseteq \mathrm{Box}_{d-1}(\mathbf{n})$ and let Q be a translate of P along the first coordinate, of the form $Q = \{\mathbf{p}+\mathbf{x} \mid p \in P\}$ for some fixed vector $\mathbf{x} = (kn_1, 0, \ldots, 0)$ $(k \in \mathbb{Z})$. The translation defines a matching $M = \{(\mathbf{p}, \mathbf{p}+\mathbf{x}) \mid \mathbf{p} \in P\}$, which can be wired in $G^d((k+1)n_1, n_2, \ldots, n_{d-1}, n_1+2)$, and the length of each wire is $O(kn_1)$.*

Let $\Sigma(\mathbf{n}) \stackrel{\text{def}}{=} \sum_{i=1}^{d-1} n_i$, and let $\pi_{\setminus i}$ be the projection that removes the i-th coordinate: $\pi_{\setminus i}(x_1, \ldots, x_d) = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)$.

Lemma 3.6. ***(Compression/Expansion)*** *Let $P \subset \mathrm{Box}_{d-1}(k\mathbf{n})$ be a k-spaced set and let $M = \{(\mathbf{p}, \mathbf{q}) \mid \mathbf{p} \in P, \mathbf{q} = \mathrm{comp}_k(\mathbf{p})\}$ — so M is the natural matching between P and $Q = \mathrm{comp}_k(P)$. Then M can be wired in $G^d(k\mathbf{n}, 2d-2+\Sigma(\mathbf{n}))$, where each wire has length $O(k\Sigma(\mathbf{n}))$.*

In a point set $P \subseteq \mathbb{Z}^{d-1}$, we denote the lexicographic ordering by $<_{d-1}$. The *lexicographic matching* between two equal size point sets of $\mathbb{Z}^{d-1}$ is the matching $\{(\mathbf{p}^i, \mathbf{q}^i) \mid i = 1, \ldots, |P|\}$, where $\mathbf{p}^i$ and $\mathbf{q}^i$ are the i-th points in the lexicographic order in P and Q respectively. The main lemma in the proof of Theorem 3.3 is the following:

Lemma 3.7. ***(Pushing/Pulling)*** *Let $d \geq 2$, $\mathbf{n} \in \mathbb{Z}_+^{d-1}$ and let P and Q be equal-size subsets of $\mathrm{Box}_{d-1}(\mathbf{n})$, where $n_1 \geqslant n_2 \geqslant \ldots \geqslant n_{d-1}$. Then the lexicographic matching between P and Q can be wired in $G^d(6\mathbf{n}, 3n_1+2)$. Moreover, the length of each wire is $O(\Sigma(\mathbf{n}))$.*

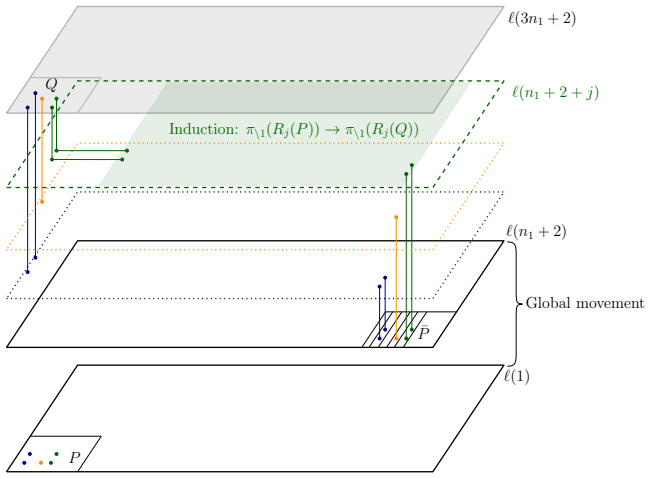

Figure 3: Pushing/pulling lemma: we use the induction hypothesis for each partition class in a separate layer.

Proof. Let the points of P and Q be $\mathbf{p}^1 <_{d-1} \mathbf{p}^2 <_{d-1} \cdots <_{d-1} \mathbf{p}_k$ and $\mathbf{q}^1 <_{d-1} \mathbf{q}^2 <_{d-1} \cdots <_{d-1} \mathbf{q}^k$. The lexicographic matching is $M = \{(p^i, q^i) \mid i \in \{1, \ldots, k\}\}$.

We use induction on the dimension d. For $d = 2$, the sets P and Q are equal size subsets of $[n]$. The wiring for $d = 2$ starts by using a global movement (Lemma 3.5) from P to its translate $5n + P \stackrel{\text{def}}{=} \{5n + p \mid p \in P\}$ — this requires $n + 2$ layers. The wires we need to continue are $\bar{P} \stackrel{\text{def}}{=} \mathrm{Raise}^{n_1+2}(5n + P)$. Next, we continue wire i from point $\bar{p}^i$ by raising its height by i units (along the x_2-coordinate), then we add a horizontal segment so that the first coordinate becomes equal to q^i (we decrease the first coordinate by $(5n + p^i - q^i)$). We finish by raising the height by $k + 1 - i$ steps. It is easy to see that these wires do not intersect. This requires $k + 2 \leqslant n_1 + 2$ layers, so overall the $d = 2$ case can be wired in $2n + 4$ layers. Each wire that we defined has length at most cn for some constant c.

For the inductive step, consider $P, Q \subseteq \mathrm{Box}_{d-1}(\mathbf{n})$. Let I_P be the set of indices in the lexicographic ordering of P that separate the ordering according to the value of the first coordinate, i.e., $i \in I_P$ if and only if $(\mathbf{p}^i)_1 < (\mathbf{p}^{i+1})_1$. We define the analogous set I_Q for the lexicographic order of Q. Let $I = I_P \cup I_Q \cup \{0, |P|\}$. Let $\mathcal{R}$ be the partition of P according to I, so

$$\mathcal{R} = \big\{ \{\mathbf{p}_a, \mathbf{p}_{a+1}, \ldots, \mathbf{p}_b\} \mid a \leqslant b, (a-1) \in I, b \in I, \{a, a+1, \ldots, b-1\} \cap I = \emptyset \big\}.$$

Note that $\mathcal{R}$ has size at most $|\mathcal{R}| \leqslant 2n_1 - 1$. We enumerate the partition classes in the lexicographic order: $\mathcal{R} = \{R_1, R_2, \ldots, R_{|\mathcal{R}|}\}$. The analogous partition $\mathcal{R}' = \{R'_1, R'_2, \ldots, R'_{|\mathcal{R}|}\}$ can be defined on Q. Notice that the lexicographic matching between P and Q is the union of the lexicographic matchings between R_j and $R'j$ for $j = 1, \ldots, |R|$. The crucial property of each partition class $R_j \in \mathcal{R}$ is that for any $\mathbf{p}^k, \mathbf{p}^l \in R_j$ and their pairs $\mathbf{q}^k, \mathbf{q}^l \in R'_j$ we have $(\mathbf{p}^k)_1 = (\mathbf{p}^l)_1$ and $(\mathbf{q}^k)_1 = (\mathbf{q}^l)_1$.

Now we are ready to define the wiring. We start with a global movement (Lemma 3.5), just as we did in 2 dimensions: we move P to $(5n_1, 0, \ldots, 0) + P$ using height $n_1 + 2$. Then we continue the wires from $\bar{P} \stackrel{\text{def}}{=} \mathrm{Raise}^{n_1+2}\big((5n_1, 0, \ldots, 0) + P\big)$, see Figure 3. For each point $\bar{p} \in \bar{P}$ whose wire belongs to the class R_j, we raise the wire j layers (into $\ell(n_1 + 2 + j)$). This introduces at most $|\mathcal{R}| \leqslant 2n_1 - 1$ new layers, and together with a top layer it gives us all our $n_1 + 2 + 2n_1 - 1 + 1 = 3n_1 + 2$ layers.

We apply the inductive step for $\pi_{\setminus 1}(R_j)$ and $\pi_{\setminus 1}(R'_j)$. This gives us a wiring in $d-1$ dimensions between these sets corresponding to the lexicographic matching between R_j and R'_j, which is a subset of the lexicographic matching between P and Q. We can embed this wiring into $\ell(n_1 + 2 + j)$ using the function $\varphi^j : \mathbb{R}^{d-1} \to \mathbb{R}^d$ that is defined as

$$\varphi^j((x_1, \ldots, x_{d-1})) = (5n_1 + \mathbf{p}_1 - x_{d-1},\ x_1, \ldots, x_{d-2},\ n_1 + 2 + j)$$

where $\mathbf{p}_1$ is the first coordinate of an arbitrary $\mathbf{p} \in R_j$, so that the "height" of the inductive step is mapped to decreasing the first coordinate within ℓ^j. (Note that by the definition of $\mathcal{R}$, we have $\mathbf{p}_1 = \mathbf{p}'_1$ for any $\mathbf{p}, \mathbf{p}' \in R_j$, thus φ^j is well-defined.) The induction implies that the wiring fits within $[6n_2] \times \cdots \times [6n_{d-1}] \times [3n_2 + 2]$. Therefore, the embedded wires do not enter $\mathrm{Raise}^{n_1+2+j}(\mathrm{Box}_{d-1}(\mathbf{n}))$, since the above embedding ends with a first coordinate which is at least $5n_1 - (3n_2 + 2) \geq n_1$. We further extend each of these wires by decreasing the first coordinate, until the wire corresponding to $\mathbf{p} \in R_j$ decreases to $\mathbf{q}_1$, where $\mathbf{q}$ is the pair of $\mathbf{p}$ in the lexicographic matching. Finally, we finish the wiring by raising all of the wires corresponding to R_j for each $j \in 1, \ldots, |\mathcal{R}|$ (extending them parallel to the d-th coordinate axis) by length $|\mathcal{R}| + 1 - j$. This completes the wiring. The length used per wire is $c \sum_{i=2}^{d-1} n_i + cn_1 = c\Sigma(\mathbf{n})$. It is routine to check that these wires are vertex disjoint. □

Our task is to wire from the bottom layer $\ell(1)$, where the point set P is embedded, to the the top layer $\ell(h_{top})$ that contains $\mathrm{Raise}^{h_{top}}(Q)$.

A *wire point* of a wire at height h is the vertex of the wire inside layer $\ell(h)$. (If there are multiple such points, let it denote the one that is the furthest away from the starting point of the wire w.) We denote by Wires the set of wires corresponding to M in the construction; furthermore, for any set of wires $T \subset \mathrm{Wires}$ let $T(h)$ be the set of wire points at height h for the wires in T. For any wire w and corresponding matching edge $(\mathbf{p}, \mathbf{q}) \in M$, denote by $\mathrm{orig}(w) = \mathbf{p}$ and $\mathrm{dest}(w) = \mathbf{q}$ the origin and destination of the wire.

Proof of Theorem 3.3. By adding dummy edges to the matching, we may assume that $P = Q = \mathrm{Box}_{d-1}(\mathbf{n})$. Without loss of generality, assume that $n_1 \geqslant n_2 \geqslant \ldots \geqslant n_{d-1}$. We assume that all coordinates of $\mathbf{n}$ are powers of two, and work inside $\mathrm{Box}_{d-1}(18\mathbf{n})$. Rounding coordinates up to the next power of two gives the stated result inside $\mathrm{Box}_{d-1}(36\mathbf{n})$.

We show that there are constants c_1, c_2 such that M can be wired in $c_1\Sigma(\mathbf{n})$ layers and $c_2 d\Sigma(\mathbf{n})$ length per wire, but starting from $\mathrm{mag}_3^0(P)$ instead of P and arriving to $\mathrm{mag}_{18}^0(Q)$ instead of Q. This is sufficient because using our compression technique described in Lemma 3.6, we can wire initially from P to $\mathrm{mag}_3^0(P)$ and in the end from $\mathrm{mag}_{18}^0(Q)$ to Q in $O(\Sigma(\mathbf{n}))$ extra layers and $O(\Sigma(\mathbf{n}))$ extra length per wire.

We use induction on $\Sigma(\mathbf{n})$. In the base case, we have $\Sigma(\mathbf{n}) = d - 1$ (i.e., $n_i = 1$ for all i), and therefore $\mathrm{mag}_3^0(P)$ can be wired in 3 layers to $\mathrm{mag}_{18}^0(Q)$ with a local movement (Lemma 3.5) since P and Q are both singletons in the 18-cell $[18]^{d-1}$.

For the inductive step, start the wiring at layer $\ell(1)$ with the 3-spaced point set $\mathrm{mag}_3^0(P) \subseteq \mathrm{Box}_{d-1}(3\mathbf{n})$. (See Figure 1.)

A wire w must be *pushed* if $(\mathrm{orig}(w))_i \leqslant n_i/2$ and $(\mathrm{dest}(w))_i > n_i/2$. Let $\mathrm{Push} \subset \mathrm{Wires}$ be the set of wires that need to be pushed. Conversely, there is a set $\mathrm{Pull} \subset \mathrm{Wires}$, the wires that need to be *pulled*, where $(\mathrm{orig}(w_{\mathbf{p}}))_i > n_i/2$ and $(\mathrm{dest}(w_{\mathbf{p}}))_i \leqslant n/2$. Due to our assumption that $P = Q = \mathrm{Box}_{d-1}(\mathbf{n})$, we have $|\mathrm{Push}| = |\mathrm{Pull}|$, therefore the pushed and pulled wires need to change places. Let $\mathrm{Stay} = \mathrm{Wires} \setminus (\mathrm{Push} \cup \mathrm{Pull})$ be the rest of the wires. Therefore, the starting points of the wires are at height one, at the points $\mathrm{Raise}^1(P) = \mathrm{Wires}(1) = \mathrm{Push}(1) \cup \mathrm{Pull}(1) \cup \mathrm{Stay}(1)$.

Using local movements with respect to 3-cells (Lemma 3.5), we connect the points of $\pi_{\setminus d}(\mathrm{Push}(1))$ into relevant points of the $(1,3)$-subgrid of $\mathrm{Box}_{d-1}(3\mathbf{n})$, and the points of $\pi_{\setminus d}(\mathrm{Pull}(1))$ to the relevant points of the $(2,3)$-subgrid of $\mathrm{Box}_{d-1}(3\mathbf{n})$. These local movements end in layer $\ell(3)$; by raising the Stay wires vertically into $\ell(3)$, we have that the point set $\mathrm{Stay}(3)$ is in the $(0,3)$ subgrid of $\mathbb{Z}^d$. We raise $\mathrm{Push}(3)$ by one layer, and $\mathrm{Pull}(3)$ by two layers; as a result the points $\mathrm{Push}(4)$ and $\mathrm{Pull}(5)$ are in the $(1,3)$ and $(2,3)$ subgrids of $\mathbb{Z}^d$ respectively. Note that for a while, we ensure the disjointness of Push, Pull and Stay by keeping them in these subgrids, which are disjoint (i.e., even the subgrid "edges" are vertex disjoint) for $d \geq 3$.

Next, we apply pushing (Lemma 3.7) in the $(1,3)$-subgrid to $\mathrm{Push}(4)$. More precisely, we regard the $(1,3)$-subgrid as a grid graph $G^d\big(6\mathbf{n}, (c/3)\Sigma(n)\big)$ (and disregard the edge subdivisions). We can apply Lemma 3.7 in this graph, to wire from the point set $\mathrm{comp}_3\big(\pi_{\setminus d}(\mathrm{Push}(4))\big)$ to $\mathrm{comp}_3\big(\pi_{\setminus d}(\mathrm{Pull}(5))\big)$ along the lexicographic matching. This wiring requires at most $(3n_1 + 2)$ layers in the $(1,3)$-subgrid. In the original graph, that becomes $3 \cdot (3n_1 + 2)$ layers, therefore the wiring ends at height $h_1 = O(n_1)$. We apply the same lemma to $\mathrm{Pull}(5)$ in the $(2,3)$-subgrid, to wire from $\mathrm{comp}_3\big(\pi_{\setminus d}(\mathrm{Pull}(5))\big)$ to $\mathrm{comp}_3\big(\pi_{\setminus d}(\mathrm{Push}(4))\big)$; this also requires height $O(n_1)$ in the original graph, and ends at height $h'_1 = O(n_1)$.

Let $h_2 = \max(h_1, h_1') = O(n_1)$. By raising the height (increasing the last coordinate) of the wire sets Push, Pull and Stay until they reach height h_2, we get to a quasi-3-spaced point set $\pi_{\setminus d}\big(\mathsf{Push}(h_2) \cup \mathsf{Pull}(h_2) \cup \mathsf{Stay}(h_2)\big) = \pi_{\setminus d}\big(\mathsf{Wires}(h_2)\big) \subseteq \mathrm{Box}_{d-1}(18\mathbf{n})$. We apply a local movement (Lemma 3.5) to make our wire points 3-spaced at height $h_2 + 2$. Finally, we apply a global movement (Lemma 3.5) on the wires in the higher half, and move them into the second half of $\mathrm{Box}_{d-1}(18\mathbf{n})$ along the first coordinate[3], that is,

$$X = \left\{\mathbf{x} \in \pi_{\setminus d}\left(\mathsf{Wires}(h_2 + 2)\right) \mid (\mathrm{comp}_3(\mathbf{x}))_1 > n_1/2\right\}$$
$$\text{is wired to } \left\{\mathbf{x} + \left(15\frac{n_1}{2}, 0, \ldots, 0\right) \mid \mathbf{x} \in X\right\}.$$

This wiring ends at a layer $h_3 = O(n_1) = c_1 n_{max}$ for some constant c_1. The length requirement per wire is $O(d)$ for the local movements, $O(dn_1)$ for the rough reordering and $O(n_1)$ for the global movement, so overall $O(dn_1) = c_2 d n_{max}$ length is used per wire so far for some constant c_2.

Let $\mathcal{B}_1 = [18\frac{n_1}{2}] \times [18n_2] \times \cdots \times [18n_{d-1}]$ and let $\mathcal{B}_2 = \mathrm{Box}_{d-1}(18\mathbf{n}) \setminus \mathcal{B}_1$. Moreover, let $W_1 = \{w \in \mathsf{Wires} \mid (\mathrm{dest}(w))_1 \leqslant \frac{n_1}{2}\}$ and $W_2 = \{w \in \mathsf{Wires} \mid (\mathrm{dest}(w))_1 > \frac{n_1}{2}\}$. Note that due to the rough reordering, the wires that are in the $\mathcal{B}_1$ box in the layer $\ell(h_2)$ are precisely W_1, while those in $\mathcal{B}_2$ are precisely W_2. By induction, there is a wiring from $\pi_{\setminus d}(W_1(h_3))$ to $\mathrm{mag}_{18}^0(Q) \cap \mathcal{B}_1$, and also from $\pi_{\setminus d}(W_2(h_3))$ to $\mathrm{mag}_{18}^0(Q) \cap \mathcal{B}_2$ that realize the matching M restricted to these parts respectively, requiring $c_1 \max(n_1/2, n_2, \ldots, n_{d-1})$ height and $c_2 d \max(n_1/2, n_2, \ldots, n_{d-1})$ length per wire. We can embed these two wirings next to each other starting from layer $\ell(h_3)$. Consider the number of layers used throughout. The value of n_{max} takes all values from the multiset $\{n_i/2^j \mid i \in [d-1],\ j = 0, 1, \ldots, \log n_i\}$ exactly once. The number of layers used is therefore

$$\sum_{i=1}^{d-1} \sum_{j=0}^{\log n_i} c_1 \frac{n_i}{2^j} < \sum_{i=1}^{d-1} 2c_1 n_i = O(\Sigma(\mathbf{n})),$$

and the length required per wire is

$$\sum_{i=1}^{d-1} \sum_{j=0}^{\log n_i} c_2 d \frac{n_i}{2^j} < \sum_{i=1}^{d-1} 2c_2 d n_i = O(d\Sigma(\mathbf{n})). \qquad \square$$

The following theorem is an easy corollary of Cube Wiring.

Theorem 3.8. *For all constants $d \geq 3$, any graph with m edges and no isolated vertices is the minor of the d-dimensional grid hypercube of side length $O(m^{\frac{1}{d-1}})$.*

Proof. Let G be an arbitrary graph with m edges. We dilate all vertices v of G into a path P_v of length $\deg_G(v)$, i.e., replace v with P_v, where each vertex of P_v is adjacent to a single neighbor of v. We also subdivide each original edge $e = uv$ of G with two new vertices, w_{eu} (adjacent to u) and w_{ev} (adjacent to v); let G' be the graph that we end up with after these modifications. Let $P = \bigcup_{v \in V} V(P_v)$ and let $Q = \{w_{ev} \mid e \in E, v \in e\}$; both sets have size $2m$. It is easy to see that both $G'[P]$ and $G'[Q]$ are subgraphs of $G_{d-1}((cm^{1/(d-1)}, \ldots, cm^{1/(d-1)})$ for some constant c; let us fix such an embedding. By the cube wiring theorem, there are vertex disjoint paths connecting the embedded vertices of P to the embedded vertices of Q in $O(dm^{1/(d-1)}) = O(m^{1/(d-1)})$ layers, along the perfect matching $E(G') \cap (P \times Q)$. This wiring together with the embeddings is a subgraph of the d-dimensional hypercube of side length $O(m^{1/(d-1)})$ from which we can get to G by applying edge contractions. $\square$

[3]Notice that we shift by $7.5n_1$ instead of $8.5n_1$, as stated in the simplified overview earlier. The reason is that we are working with a 3-spaced set X here.

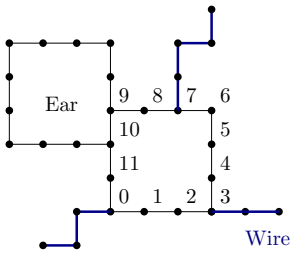

Figure 4: Variable gadget for Dominating Set. In blue it is shown where the wire gadgets are attached for a variable that occurs twice as a negative literal (corresponding to wires attached to vertices 0 and 3) and once as a positive literal (wire attached to vertex 7).

3.3 Applying the Lower-Bound Framework

Theorem 3.9. *Let $d \geqslant 2$ be a fixed constant. Then there is no $2^{o(n^{1-1/d})}$ algorithm for Dominating Set in induced grid graphs of dimension d, unless ETH fails.*

Proof. We do a reduction from Grid Embedded SAT. Let ϕ be the input formula with incidence graph G_ϕ. Our variable gadget is a cycle of length 12 with an "ear" of the same size, as depicted in Figure 4; we number the vertices of the cycle from 0 to 11. The wire gadgets are simple paths of length $3k + 1$ (for some $k \in \mathbb{Z}_+$), and the clause gadget is a single vertex. A wire that corresponds to a positive literal starts at a variable cycle vertex with index $\equiv 1 \pmod 3$, and ends at the corresponding clause vertex. For negative literals, we start at a vertex of index $\equiv 0 \pmod 3$ instead. From each variable cycle, we must select at least four vertices into our dominating set, and at least three more vertices from the ear are necessary. From the inner vertices of a wire of length $3k + 1$, we have at least k vertices in the dominating set.

Therefore, the dominating-set instance corresponding to a formula on n variables with a drawing of total wire count w and total wire length ℓ has dominating set size at least $7n + \frac{\ell - w}{3}$. It is routine to check that this is attainable if and only if the formula is satisfied. See [10] for a similar, but more detailed argument.

Two-dimensional grid graphs. Given a grid embedded drawing $\mathcal{D}$ of G_ϕ, we need to create a grid graph which incorporates the above gadgets. This can be done by changing the grid underlying $\mathcal{D}$ to a ten times denser grid; this way, we can add the variable gadgets without overlap or unwanted induced edges, and we also have space to adjust the wire length where necessary using local modifications. This transformation can be done in polynomial time, and the result is an induced grid graph drawn in an $O(n) \times O(n)$ grid. Therefore, Dominating Set has no $2^{o(\sqrt{n})}$ algorithm in induced grid graphs unless ETH fails.

Higher dimensional grid graphs.

We start with a (3, 3)-SAT formula ϕ. We place the above variable gadgets in a $d-1$-dimensional hypercube of side length $O(n^{\frac{1}{d-1}})$. The clause gadgets along with the last inner vertices of each wire are placed in the opposing face of the d-dimensional hypercube. Applying the Cube Wiring Theorem to the first and last inner vertices of the wires that have been placed in the opposing faces, we can place each wire inside the hypercube, by increasing the side length by a constant factor (depending on d). The construction fits in a hypercube of side length $O(n^{\frac{1}{d-1}})$, and the number of vertices in this induced grid graph is $O(n^{\frac{d}{d-1}})$. Thus, a $2^{o(|V|^{1-1/d})}$ algorithm for Dominating Set would translate into a $2^{o((n^{\frac{d}{d-1}})^{1-1/d})} = 2^{o(n)}$ algorithm for (3, 3)-SAT, contradicting ETH (this is demonstrated in the full version). □

4 CONCLUSION

We have presented an algorithmic and lower bound framework for obtaining $2^{\Theta(n^{1-1/d})}$ algorithms and matching conditional lower bounds for several problems in geometric intersection graphs. We find the following questions intriguing:

- Is it possible to obtain clique decompositions without geometric information? Alternatively, how hard is it to color the complement of a small diameter geometric intersection graph of fat objects?
- Many of our applications require the low degree property (i.e., the fact that $G_{\mathcal{P}}$ has bounded degree). Is the low degree property really essential for these applications? Would having low average degree be sufficient?
- Is it possible to modify the framework to work without the similar size assumption?

Finally, it would be interesting to explore the potential consequences of this framework for parameterized and approximation algorithms.

REFERENCES

[1] Jochen Alber and Jirí Fiala. 2004. Geometric separation and exact solutions for the parameterized independent set problem on disk graphs. *Journal of Algorithms* 52, 2 (2004), 134–151. https://doi.org/10.1016/j.jalgor.2003.10.001

[2] Boris Aronov, Mark de Berg, Esther Ezra, and Micha Sharir. 2014. Improved Bounds for the Union of Locally Fat Objects in the Plane. *SIAM J. Comput.* 43, 2 (2014), 543–572. https://doi.org/10.1137/120891241

[3] Julien Baste and Dimitrios M. Thilikos. 2018. Contraction-Bidimensionality of Geometric Intersection Graphs. In *12th International Symposium on Parameterized and Exact Computation (IPEC 2017) (Leibniz International Proceedings in Informatics (LIPIcs))*, Daniel Lokshtanov and Naomi Nishimura (Eds.), Vol. 89. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 5:1–5:13. https://doi.org/10.4230/LIPIcs.IPEC.2017.5

[4] Csaba Biró, Édouard Bonnet, Dániel Marx, Tillmann Miltzow, and Paweł Rzążewski. 2017. Fine-Grained Complexity of Coloring Unit Disks and Balls. In *Proceedings of the 33rd International Symposium on Computational Geometry, SoCG 2017 (LIPCS)*, Vol. 77. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 18:1–18:16. https://doi.org/10.4230/LIPIcs.SoCG.2017.18

[5] Hans L Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. 2015. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Information and Computation* 243 (2015), 86–111.

[6] Heinz Breu and David G. Kirkpatrick. 1998. Unit disk graph recognition is NP-hard. *Comput. Geom.* 9, 1-2 (1998), 3–24. https://doi.org/10.1016/S0925-7721(97)00014-X

[7] Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. 2000. Linear Time Solvable Optimization Problems on Graphs of Bounded Clique-Width. *Theory of Computing Systems* 33, 2 (2000), 125–150. https://doi.org/10.1007/s002249910009

[8] Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. 2015. *Parameterized Algorithms*. Springer.

[9] Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C. van der Zanden. 2018. Framework for ETH-tight Algorithms and Lower Bounds in Geometric Intersection Graphs. *CoRR* abs/1803.10633 (2018). arXiv:1803.10633 http://arxiv.org/abs/1803.10633

[10] Mark de Berg, Sándor Kisfaludi-Bak, and Gerhard Woeginger. 2018. The Dominating Set Problem in Geometric Intersection Graphs. In *12th International Symposium on Parameterized and Exact Computation (IPEC 2017) (Leibniz International Proceedings in Informatics (LIPIcs))*, Daniel Lokshtanov and Naomi Nishimura (Eds.), Vol. 89. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 14:1–14:12. https://doi.org/10.4230/LIPIcs.IPEC.2017.14

[11] Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. 2017. Finding, Hitting and Packing Cycles in Subexponential Time on Unit Disk Graphs. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017 (LIPICS)*, Vol. 80. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 65:1–65:15. https://doi.org/10.4230/LIPIcs.ICALP.2017.65

[12] Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. 2012. Bidimensionality and geometric graphs. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012*. SIAM, 1563–1575. http://portal.acm.org/citation.cfm?id=2095240&CFID=63838676&CFTOKEN=79617016

[13] Bin Fu. 2011. Theory and application of width bounded geometric separators. *J. Comput. System Sci.* 77, 2 (2011), 379 – 392. https://doi.org/10.1016/j.jcss.2010.05.003

[14] Martin Charles Golumbic and Udi Rotics. 2000. On the Clique-Width of Some Perfect Graph Classes. *International Journal of Foundations of Computer Science* 11, 3 (2000), 423–443. https://doi.org/10.1142/S0129054100000260

[15] Sariel Har-Peled and Kent Quanrud. 2015. Approximation algorithms for polynomial-expansion and low-density graphs. In *Proceedings of the 23rd Annual European Symposium on Algorithms, ESA 2015 (Lecture Notes in Computer Science)*, Vol. 9294. Springer, 717–728. https://doi.org/10.1007/978-3-662-48350-3_60

[16] Russell Impagliazzo and Ramamohan Paturi. 2001. On the Complexity of k-SAT. *J. Comput. System Sci.* 62, 2 (2001), 367–375. https://doi.org/10.1006/jcss.2000.1727

[17] Ross J. Kang and Tobias Müller. 2012. Sphere and Dot Product Representations of Graphs. *Discrete & Computational Geometry* 47, 3 (2012), 548–568. https://doi.org/10.1007/s00454-012-9394-8

[18] Sándor Kisfaludi-Bak and Tom C van der Zanden. 2017. On the Exact Complexity of Hamiltonian Cycle and q-Colouring in Disk Graphs. In *Proceedings 10th International Conference on Algorithms and Complexity, CIAC 2017 (Lecture Notes in Computer Science)*, Vol. 10236. Springer, 369–380.

[19] Richard J. Lipton and Robert Endre Tarjan. 1979. A separator theorem for planar graphs. *SIAM J. Appl. Math.* 36, 2 (1979), 177–189.

[20] Richard J. Lipton and Robert Endre Tarjan. 1980. Applications of a planar separator theorem. *SIAM J. Comput.* 9, 3 (1980), 615–627.

[21] Dániel Marx. 2013. The Square Root Phenomenon in Planar Graphs. In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part II*. 28. https://doi.org/10.1007/978-3-642-39212-2_4

[22] Dániel Marx and Michal Pilipczuk. 2015. Optimal Parameterized Algorithms for Planar Facility Location Problems Using Voronoi Diagrams. In *Proceedings of the 23rd Annual European Symposium on Algorithms, ESA 2015 (Lecture Notes in Computer Science)*, Vol. 9294. Springer, 865–877. https://doi.org/10.1007/978-3-662-48350-3_72

[23] Dániel Marx and Anastasios Sidiropoulos. 2014. The limited blessing of low dimensionality: when $1-1/d$ is the best possible exponent for d-dimensional geometric problems. In *Proceedings of the 30th Annual Symposium on Computational Geometry, SOCG 2014*. ACM, 67–76. https://doi.org/10.1145/2582112.2582124

[24] Sang-il Oum. 2017. Rank-width: Algorithmic and structural results. *Discrete Applied Mathematics* 231 (2017), 15–24. https://doi.org/10.1016/j.dam.2016.08.006

[25] Warren D. Smith and Nicholas C. Wormald. 1998. Geometric Separator Theorems & Applications. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science, FOCS 1998*. IEEE Computer Society, 232–243. https://doi.org/10.1109/SFCS.1998.743449

[26] Frank van den Eijkhof, Hans L. Bodlaender, and M.C.A. Koster. 2007. Safe Reduction Rules for Weighted Treewidth. *Algorithmica* 47, 2 (2007), 139–158.