Evaluating Restricted First-Order Counting Properties on Nowhere Dense Classes and Beyond

TU Wien

Daniel Mock

□

RWTH Aachen University

Peter Rossmanith

□

RWTH Aachen University

- Abstract

It is known that first-order logic with some counting extensions can be efficiently evaluated on graph classes with bounded expansion, where depth-r minors have constant density. More precisely, the formulas are $\exists x_1 \dots x_k \# y \varphi(x_1, \dots, x_k, y) > N$, where φ is an FO-formula. If φ is quantifier-free, we can extend this result to nowhere dense graph classes with an almost linear FPT run time. Lifting this result further to slightly more general graph classes, namely almost nowhere dense classes, where the size of depth-r clique minors is subpolynomial, is impossible unless FPT = W[1]. On the other hand, in almost nowhere dense classes we can approximate such counting formulas with a small additive error. Note those counting formulas are contained in FOC($\{>\}$) but not FOC₁(\mathbf{P}).

In particular, it follows that partial covering problems, such as partial dominating set, have fixed parameter algorithms on nowhere dense graph classes with almost linear running time.

2012 ACM Subject Classification Theory of computation \rightarrow Parameterized complexity and exact algorithms; Theory of computation \rightarrow Logic; Theory of computation \rightarrow Computational complexity and cryptography; Mathematics of computing \rightarrow Graph theory

Keywords and phrases nowhere dense, sparsity, counting logic, dominating set, FPT

Related Version Full version:

Funding Daniel Mock: Supported by the German Science Foundation DFG, grant no. DFG-927/15-2 Peter Rossmanith: Supported by the German Science Foundation DFG, grant no. DFG-927/15-2

1 Introduction

First-order logic can be used to express algorithmic problems. FO-model checking on certain classes of structures is therefore a meta-algorithm, which solves many problems at the same time. For example, the three classical problems that started the research on parameterized complexity are all FO-expressible: Vertex Cover, Independent Set, and Dominating Set [9, 10]. Dominating Set with the natural parameter—the size of the minimal dominating set—is W[2]-complete on general graphs, but fixed parameter tractable (fpt) on many special graph classes. The study of sparsity, initiated by Nešetřil and Ossona de Mendez, has led to the concept of bounded expansion and nowhere dense graph classes [30]. They generalize many well-known notions of sparsity, such as bounded degree, planarity, bounded genus, bounded treewidth, (topological) minor-closed, etc. and have led to quite general algorithmic results [33, 16, 6, 15]. Most notably, Grohe, Kreutzer, and Siebertz showed that FO-model checking is fpt on nowhere dense graph classes [22]. This shows, e.g., that dominating set is fpt on nowhere dense graphs, a result that was already known: Dawar and Kreutzer were able to find a specific algorithm several years earlier [7] that solves generalizations of the dominating set problem. All of them are FO-expressible, which shows how strong meta-algorithms are.

Partial dominating set, also called t-dominating set, is another generalization of dominating set: The input is a graph G and two numbers k and t. The question is, whether G contains k vertices that dominate at least t vertices. The parameter is k, as in the classical dominating set problem. (If you choose t as the parameter—which also makes sense—the problem becomes fixed-parameter tractable even on general graphs [26].) The length of an FO-formula expressing the existence of a partial dominating depends on t, which is not bounded by any function of k and therefore all the results on first-order model checking do not help when we parameterize by k only. Golovach and Villanger showed that partial dominating set remains hard on degenerate graphs [18], while Amini, Fomin, and Saurabh have shown that partial dominating set is fixed-parameter tractable in minor-closed graph classes, which generalized earlier positive results [1]. Very recently, this was improved to graph classes with bounded-expansion, while simultaneously using only linear fpt time instead of polynomial fpt time, i.e., the running time is now only f(k)n [11].

This result was achieved by another meta-theorem for the counting logic $FOC(\{>\})$ on classes of bounded expansion. $FOC(\{>\})$ is a fragment of the logic $FOC(\mathbf{P})$, introduced by Kuske and Schweikardt in order to generalize first-order logic to counting problems [28]. $FOC(\mathbf{P})$ is a very expressive counting logic and allows counting quantifiers $\#\bar{y}\varphi(\bar{x},\bar{y})$, which count for how many \bar{y} the FOC(P)-formula $\varphi(\bar{x},\bar{y})$ is true. Moreover, arithmetic operations are allowed as well as all predicates in P, which might contain comparisons, equivalence modula a number, etc. Kuske and Schweikardt showed that the FOC(P)-model checking problem is fixed parameter tracktable on graphs of bounded degree and hard on trees of bounded depth. The fragment $FOC(\{>\})$ is more restrictive and allows only counting quantifiers of single variables and no arithmetic operations. The only predicate is comparison against an arbitrary number, but not between counting terms. While $FOC(\{>\})$ -model checking is still hard on trees of bounded depth, there is an "approximation scheme" for $FOC(\{>\})$ on classes of bounded expansion [11]: An algorithm gives either the right answer or says "mayby," but only if the formula is both almost satisfied and not satisfied. For a fragment of $FOC(\{>\})$, which captures in particular the partial dominating set problem, we can compute even an exact answer to the model checking problem in linear fpt time [11]. That fragment consists of formulas of the form

$$\exists x_1 \dots \exists x_k \# y \, \varphi(y, x_1, \dots, x_k) > N, \tag{1}$$

where φ is a first-order formula and N an arbitrary number. The semantics of the *counting quantifier* $\#y \varphi(y, v_1, \ldots, v_k)$ is the number of vertices u in G such that G satisfies $\varphi(u, v_1, \ldots, v_k)$. As an example, the existence of partial dominating set can be expressed as

$$\exists x_1 \dots \exists x_k \# y \bigvee_{i=1}^k E(y, x_i) \lor y = x_i > t,$$
(2)

where k is the number of the dominating, and t the number of dominated vertices. The length of the formula only depends on k. This implies that partial dominating set can be solved in linear fpt time on classes of bounded expansion.

There is another fragment of FOC(\mathbf{P}), which should not be confused with FOC($\{>\}$). In FOC₁(\mathbf{P}), introduced by Grohe and Schweikardt [23], the counting terms may contain at most one free variable. They show that FOC₁(\mathbf{P}) is fixed-parameter tractable on nowhere dense graph classes [23]. Note that formula 2 is in FOC($\{>\}$) but not in FOC₁(\mathbf{P}) as the counting term relies on k free variables. Hence, FOC($\{>\}$) and FOC₁(\mathbf{P}) are orthogonal in there expressiveness.

Graph class	FO-MC	$\mathrm{FOC}_1(\mathbf{P})$	FOC({>})	PDS like
bounded expansion	fpt [15]	fpt [23]	hard [11] $(1+\varepsilon)$ -approx fpt [11]	fpt [11]
nowhere dense	fpt [22]	fpt [23]	hard, approx open	\mathbf{fpt}^c
almost nowhere dense	\mathbf{hard}^a	\mathbf{hard}^a	\mathbf{hard}^a	$\mathbf{hard}^a \ \mathbf{approx} \pm \delta \ \mathbf{fpt}^b$
general graphs	hard	hard	hard	hard

^a Corollary 48, ^b Corollary 2, ^c Theorem 1

Table 1 Results of this paper (in boldface) and some related known results. *Hard* means at least W[1]-hard. *PDS like* indicates problems similar to the partial dominating set problems: All problems that can be expressed by a $FOC(\{>\})$ formula of the form (1). The mentioned approximation results are quite different. Numbers are approximated either with a relative or an absolute error.

There has been some research about low degree graphs. A graph class has low degree if every (sufficiently large) graph has degree at most n^{ε} for every $\varepsilon > 0$. Examples are classes with bounded degree or classes with degree bounded by a polylogarithmic function. These graph classes are incomparable to nowhere dense classes. Especially, classes of low degree are not closed under subgraphs. On those classes, Grohe has shown that first-order model-checking can be solved in almost linear time [19]. Recently, Durand, Schweikardt, and Segoufin have generalized the result to query counting with constant delay and almost linear preprocessing time [13]. Vigny explores dynamic query evaluation on graph classes with low degree [35].

Almost nowhere dense is a property which subsumes both low degree and nowhere dense classes. Whereas a nowhere dense class \mathcal{C} can be characterized that for every r graphs do not contain up to r times subdivided cliques of arbitrary sizes, for an almost nowhere dense class arbitrary sizes are allowed, but their growth must be bounded by subpolynomial function in the size of the graph.

1.1 Our Results

In this work, we consider a fragment of $FOC(\{>\})$, which we will call *PDS-like formulas*, namely formulas of the form

$$\exists x_1 \dots \exists x_k \# y \, \varphi(y, x_1, \dots, x_k) > N$$

for a quantifier-free FO-formula φ and an (arbitrarily big) number $N \in \mathbf{Z}$. This logic is strong enough to express the partial dominating set problem as formula (2) is contained in the fragment described above. Remember that this fragment and $FOC_1(\mathbf{P})$ are orthogonal. Table 1 contains an overview of most of the results in this paper.

In formulas that start with existential quantifier it is natural to ask for a witness, if we can indeed fulfill the formula. For example, in the partial dominating set problem we are usually interested in actually finding the dominating set rather than verify than one exists. Often, this is not an issue as problems are self-reducible. Using self-reducibility to find a witness incurs a runtime penalty. The next theorem shows that solving the model checking problem, and finding a witness, for formulas in the form of 1 is possible.

▶ Theorem 1. Let C be a nowhere dense graph class. For every $\varepsilon > 0$, every graph $G \in \mathcal{C}$ and every quantifier-free first-order formula $\varphi(y\bar{x})$ we can compute a vertex tuple \bar{u}^* that maximizes $\llbracket \# y \varphi(y\bar{u}^*) \rrbracket^G$ in time $O(n^{1+\varepsilon})$.

As an immediate corollary, we get that the model-checking problem for PDS-like formulas and thus, also the partial dominating set problem are solvable in almost linear fpt time on nowhere dense graph classes, where the parameter is the length of the formulas or the solution size k respectively. Moreover, our meta-algorithm does not only work for partial dominating set, but for variants such as partial total or partial connected dominating set as well.

Note that Theorem 1 does not follow from the fact that model-checking for $FOC_1(\mathbf{P})$ or that query-counting for FO-logic is fixed-parameter tractable [23] as we do not count the number of solutions to a query, but the number of witnesses to some solution. Also, PDS-like formulas form a fragment orthogonal to $FOC_1(\mathbf{P})$. Moreover, we were not able to prove Theorem 1 by using the result from [23] as a subroutine: formulas inside a counting quantifier are allowed to have at most one free variable and this weakens self-reducibility or similar techniques drastically.

The above theorem cannot be extended to the more general case of almost nowhere dense graph classes. It turns out that even for non-counting formulas this is not possible, as the (classical) dominating set problem becomes W[1]-hard on some almost nowhere dense graph classes. This lower bound implies as a special case that plain FO-model checking is intractable on some almost nowhere dense graph classes. As far as we are aware this does not follow directly from previously known results.

However, we can go beyond nowhere dense classes if we do not insist on an exact solution: The model-checking problem for PDS-like formulas can be approximated with an *additive* subpolynomial error in almost linear fpt time on almost nowhere dense classes of graphs. To be more precise, we get the following, slightly more general result.

▶ Corollary 2. Let C be an almost nowhere dense class of graphs. For every $\varepsilon > 0$, every graph $G \in C$ and every quantifier-free first-order formula $\varphi(y\bar{x})$, we can compute in time $O(n^{1+\varepsilon})$ a vertex tuple $\bar{u} \in V(G)^{|\bar{x}|}$ with

$$|\max_{\bar{u}} \left[\!\left[\#y\,\varphi(y\bar{u})\right]\!\right]^G - \left[\!\left[\#y\,\varphi(y\bar{u}^*)\right]\!\right]^G | \leq n^{\varepsilon}.$$

Talking about characterizations of almost nowhere dense graph classes, we provide a plethora of different characterizations, similar to the ones for bounded expansion and nowhere denseness. We show that a class is almost nowhere dense classes if and only if measures like r-shallow (topological) minor, forbidden r-subdivisions and (weak) r-coloring numbers are bounded by $f(r, \varepsilon)n^{\varepsilon}$.

We also examine almost nowhere dense classes from an algorithmic point of view: Whereas it is "natural" to consider monotonicity as closure property for nowhere dense graph classes, it is similarly natural to consider closure under edge deletion for almost nowhere dense graph classes. Consider a graph class $\mathcal C$ which is closed under deleting edges. Then we show that the problem of finding an r times subdivided k-clique is fpt for every fixed r on $\mathcal C$ if and only if $\mathcal C$ is almost nowhere dense. In particular, for every graph class that is not almost nowhere dense, but closed under deletion of edges, there exists a number r such that finding r-subdivided k-cliques cannot be solved in fpt time under some complexity theoretic assumption, and, therefore, the FO model checking problem for formulas of the form $\exists \bar x \varphi(\bar x)$ where $\varphi(\bar x)$ is quantifier free and has predicates for adjacency and distance-r adjacency, cannot be solved either. The situation for distance-r independent set is different: Like finding an r-times

subdivided clique it is fpt on almost nowhere dense graph classes, but there exists a graph class which is not almost nowhere dense and is closed under edge deletion where the problem is fpt.

1.2 Techniques

For Theorem 1, we use a novel dynamic programming technique on game trees of Splitter games. Splitter games were introduced by Grohe, Kreutzer, and Siebertz [22] to solve the first-order model-checking problem on nowhere dense classes. Together with their new concept of sparse neighborhood covers they achieved small recursion trees of constant depth.

Splitter games can be understood as a localized variation of the cops and robbers game for bounded treedepth (not to be confused with locally bounded treedepth). In contrast to [22] we apply a dynamic programming approach, similar to the ones used on bounded tree-depth decompositions. In contrast to bounded treedepth, a graph decomposes into neighborhoods of small radius instead of connected components when removing vertices according to Splitter's winning strategy. A challenge is that the resulting neighborhoods—in contrast to connected components—are not disjoint and lead to double counting for counting problems (an issue that does not occur in FO-model checking). To avoid double counting we introduce so-called cover systems specifically for the subgraph "induced" by the solution. The existence of such cover systems shows that there is a disjoint selection of small neighborhoods that cover all the vertices relevant to our counting problem. By solving a certain variation of the independent set problem, we can find such a selection and can safely combine the results of local parts of the graph as in dynamic programs for bounded tree-depth.

To achieve our second result Corollary 2, we adapt the techniques of the proof for solving the corresponding exact counting problem on classes of bounded expansion [11]: We replace $\#y\,\varphi(y\bar x)$ by a sum of gradually simpler counting terms until they are simple enough to be easily evaluated. During this process we use transitive fraternal augmentations and a functional representation to encode necessary information into the graph, which is needed during the above simplification of counting terms. Along the way some difficult to handle literals appear in only a few number of terms. Ignoring them leads to the imprecision of our approximation. As the number of functional symbols in (almost) nowhere dense graph classes is not bounded by a constant as it is the case in classes of bounded expansion, the techniques from [11] have to adapted and extended. The main problem why their proof cannot be used directly is that the replacement of formulas leads to formulas of constant size in the case of bounded expansion, but to a non-constant size in our case. Here we use some new tricks and observe, that even though the transformed formulas can be of subpolynomial length, they can basically be replaced by many short formulas.

2 Preliminaries

2.1 Graphs.

We obtain results for labeled graphs. A labeled graph is a tuple $G = (V, E, P_1, \dots, P_m)$, where V is the vertex set, E is the edge set and $P_1, \dots, P_m \subseteq V$ the labels of G. The order |G| of G equals |V|. We define the size |G| of G as $|V| + |E| + |P_1| + \dots + |P_m|$. Unless otherwise noted, our graphs are undirected. For a directed graph G, the indegree of a node V equals the number of vertices U such that there is an arc U in G. The maximal indegree of all nodes in G is denoted by A



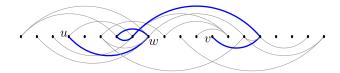


Figure 1 u is weakly 5-reachable from v by the highlighted path, but w is not weakly reachable from v.

While our results all work for labeled graphs, we will sometimes ignore labels in long chains of transformations between structures in order to keep the proof uncluttered. The presence of labels, however, is never a real problem.

2.2 Sparse Graph Classes

A graph G' is an r-subdivision of a graph G if G' can be obtained from G by replacing all edges by vertex disjoint paths with exactly r inner vertices. Similarly, G' is an $\leq r$ -subdivision of a graph G if G' is obtained from G by replacing all edges by vertex disjoint paths with at most r inner vertices. Here, the number of subdivisions may differ for each edge. In G', the vertices of G are called principal vertices and the remaining ones are called subdivision vertices. A graph G is a topological depth-r minor of a graph H if an $\leq r$ -subdivision of G is isomorphic to a subgraph of H.

- ▶ **Definition 3** (Bounded expansion). A graph class C has bounded expansion if for all $r \in \mathbf{N}$ there exists $t \in \mathbf{N}$ such that for all $G \in \mathcal{C}$, and all topological depth-r minors H of G, $||H||/|H| \leq t$.
- **Definition 4** (Nowhere dense). A graph class C is nowhere dense if for all $r \in \mathbb{N}$ there exists a $t \in \mathbb{N}$ such that no $G \in \mathcal{C}$ contains K_t as a topological depth-r minor. If a graph class is not nowhere dense it is called somewhere dense.

2.3 Weak coloring numbers

A central concept in this paper are generalized coloring numbers, especially the weak coloring numbers introduced by Kierstead and Yang [25]. An ordering π of a graph G is a linear ordering of its vertex set and the set of all such orderings is denoted by $\Pi(G)$.

▶ **Definition 5** (Kierstead and Yang [25]). A vertex $u \in V$ is weakly r-reachable from a vertex $v \in V$ with respect to $\pi \in \Pi(G)$ if $u \leq_{\pi} v$ and there exists a path P from u to v of length at most r such that $u \leq_{\pi} w$ for each $w \in V(P)$. The set of weakly r-reachable vertices from v with respect to π is denoted by WReach_r[G, π, v]. Note that v is always included in this set. We write $\operatorname{wdist}_{G,\pi}(u,v) \leq d$ if $u \in \operatorname{WReach}_r[G,\pi,v]$ or $v \in \operatorname{WReach}_r[G,\pi,u]$.

The weak r-coloring number of a graph G (and an ordering π) is defined as

$$\operatorname{wcol}_r(G, \pi) := \max_{v \in V(G)} |\operatorname{WReach}_r[G, \pi, v]|$$

 $\operatorname{wcol}_r(G) := \min_{\pi \in \Pi(G(V))} \operatorname{wcol}_r(G, \pi).$

The weak 1-coloring number of a graph is one more than its degeneracy, which is the smallest number d such that every subgraph $H \subseteq G$ has a vertex of degree at most d in H. The weak coloring number can be seen as a localized version of tree-depth, as

$$\operatorname{wcol}_1(G) \le \operatorname{wcol}_2(G) \le \cdots \le \operatorname{wcol}_{\infty}(G) = \operatorname{td}(G)$$
 [30].

Figure 1 contains an example of weak r-reachability. Weak coloring numbers can be used to characterize nowhere dense graph classes:

▶ Proposition 6 ([36, 31]). A graph class C is nowhere dense if and only if there exists a function f such that for every $r \in \mathbb{N}$, every $\varepsilon > 0$, every graph $G \in C$ satisfies $\operatorname{wcol}_r(H) \leq f(r,\varepsilon)|H|^{\varepsilon}$ for every $H \subseteq G$.

When weak coloring numbers are used within an algorithm, it is often essential to find an ordering of the vertices of a graph with a small weak coloring number. The situation is similar to efficient algorithms on tree decompositions: First a tree decomposition has to be found. Even though computing $\operatorname{wcol}_r(G)$ is NP-hard for $r \geq 3$ in general [20], it is possible to compute in parameterized quasi-linear time orderings which are approximately optimal:

▶ Proposition 7 (Grohe, Kreutzer, Siebertz [22, Cor. 5.8]). Let \mathcal{C} be a nowhere dense graph class. There is a function f such that for all $r \in \mathbb{N}$, $\varepsilon > 0$ and $G \in \mathcal{C}$ with $|G| \geq f(r, \varepsilon)$, an ordering π of G with $|\text{WReach}_r[G, \pi, v]| \leq |G|^{\varepsilon}$ for all $v \in V(G)$ can be found in time $f(r, \varepsilon) \cdot |G|^{1+\varepsilon}$.

As we are dealing with somewhere dense graph classes in Section 5, we cannot use Proposition 7 to construct orderings with small generalized coloring numbers. Revisiting the proof of Proposition 7, we notice that even without the assumption of nowhere denseness, one can find in linear time orderings which approximate the weak coloring numbers:

▶ **Lemma 8.** There is a computable function f and an algorithm that computes for a graph G a vertex ordering π such that $\operatorname{wcol}_r(G,\pi) \leq \operatorname{wcol}_{f(r)}(G)^{f(r)}$ for every $r \in \mathbb{N}$. The running time of the construction is $\operatorname{wcol}_{f(r)}(G)^{f(r)}|G|$.

Proof. We use Theorem 4.6.4 of [34] stating: "For every integer r > 0 there is a polynomial $q_r(x)$ such that for every graph G one can compute in time $q_r(\nabla_{2^{r+1}}(G))$ an orientation \vec{G} of G and a transitive fraternal augmentation $\vec{G}_1 \subseteq \ldots \subseteq \vec{G}_r$ with $\vec{G}_1 = \vec{G}$ such that $\Delta^-(\vec{G}_r) \leq q_r(\nabla_{2^{r+1}}(G))$." The proof is not contained in [34], but follows easily from Corollary 5.3 in [29]. A close look at the proof reveals that the running time is indeed $q_r(\nabla_{2^{r+1}}(G))n$, where q_r is a polynomial that is computable given r.

Using this result yields a digraph \vec{G}_r such that $\Delta^-(\vec{G}_r) \leq q_r(\nabla_{2^{r+1}}(G))$. Grohe, Kreutzer, and Siebertz show that if \vec{H} is an r-transitive fraternal augmention of a graph G with $\Delta^-(\vec{H}) \leq d$, then $\operatorname{wcol}_r(G) \leq 2(d+1)^2$ [22, Lemma 6.7]. Moreover, in the proof of this lemma it is shown that an order can be constructed in linear time that witnesses this bound on the weak coloring number.

Hence, we compute a linear order π on the vertices of \vec{G}_r such that

$$\operatorname{wcol}_{r}(G,\pi) \leq 2(q_{r}(\nabla_{2^{r+1}}(G)) + 1)^{2}.$$
(3)

The grad $\nabla_{(r-1)/2}(G)$ is bounded by the weak coloring number via $\nabla_{(r-1)/2}(G) + 1 \le \operatorname{wcol}_r(G)$ [30, Lemma 7.11]. Combining this bound with the bound in (3) yields $\operatorname{wcol}_r(G, \pi) \le 2(q_r(\operatorname{wcol}_{2^{r+2}+1}(G)+1))^2 \le \operatorname{wcol}_{2^{r+2}+1}(G)^{f(r)}$ for some f(r).

Altogether we have constructed an ordering π in time linear in $||\vec{G}_r|| \leq \Delta^-(\vec{G}_r)|G| \leq \operatorname{wcol}_{f(r)}(G)^{f(r)}|G|$ such that $\operatorname{wcol}_r(G,\pi) \leq \operatorname{wcol}_{f(r)}(G)^{f(r)}$.

2.4 Splitter game

We will use a game-based characterization of nowhere denseness introduced by Grohe, Kreutzer and Siebertz [22]. Given a graph G, a radius r and a number of rounds ℓ , the

- (ℓ,r) -Splitter game on G is an alternating game between two players called Splitter and Connector. The game starts on $G_0 = G$. In the ith round, the Connector chooses a vertex v_i from G_i . Then the Splitter chooses a vertex s_i from the radius-r neighborhood of v_i in G_i . The game continues on $G_{i+1} = G_i[v_i] - s_i$. Splitter wins if after ℓ rounds the graph is empty. Grohe, Kreutzer and Siebertz showed that nowhere dense graph classes can be characterized by Splitter games:
- ▶ Proposition 9. [22] Let C be a nowhere dense class of graphs. Then, for every r > 0, there is $\ell > 0$, such that for every $G \in \mathcal{C}$, Splitter has a strategy to win the (ℓ, r) -splitter game on

Note that a winning move of Splitter in a current play can be computed in almost linear time [22, Remark 4.7].

2.5 Sparse neighborhood covers

Even though the splitter game ends after a bounded number of rounds ℓ for nowhere dense classes, the game tree, i.e. the tree spanned by all possible plays of Splitter and Connector, can still be large, e.g. in the dimensions of n^{ℓ} . To make the game trees small and useful for algorthmic use, Grohe, Kreutzer and Siebertz introduced sparse neighborhood covers [22]. These covers group "similar" neighborhoods into a small number cluster of bounded radius. These clusters can be used instead of the neighborhoods, reducing the size of the game tree to $O(n^{1+\varepsilon})$.

- ▶ **Definition 10.** [22] For a radius $r \in \mathbb{N}$, an r-neighborhood cover \mathcal{X} of a graph G is a set of connected subgraphs of G called clusters, such that for every vertex $v \in V(G)$ there is some $X \in \mathcal{X}$ with $N_r[v] \subseteq V(X)$. The degree of v in \mathcal{X} is the number of clusters that contain v and the radius of X is the maximal radius of a cover in X. A class C admits sparse neighborhood covers if there exists $c \in \mathbb{N}$ and for all $r \in \mathbb{N}$ and all $\varepsilon > 0$ a number $d = d(r, \varepsilon)$ such that every graph $G \in \mathcal{C}$ admits an r-neighborhood cover of radius at most c and degree at most $d|G|^{\varepsilon}$.
- ▶ Proposition 11. [22] Every nowhere dense class C of graphs admits a sparse neighborhood cover. For a graph $G \in \mathcal{C}$ and $r \in \mathbf{N}$ such an r-neighborhood cover can be computed in time $f(r,\varepsilon)n^{1+\varepsilon}$ for every $\varepsilon > 0$.

Indeed, the existence of such covers is another characterization of nowhere dense classes.

▶ **Definition 12.** For a graph G with a vertex order π , $r \in \mathbb{N}$ and a vertex $v \in V(G)$, we define $X_r[G, \pi, v]$ as $\{u \in V(G) \mid v \in \operatorname{WReach}_r[G, \pi, u]\}$. We let $\mathcal{X}_r = \{X_{2r}[G, \pi, v] \mid v \in V(G)\}$.

From the proof of Proposition 11 it follows, that the set family \mathcal{X}_r is such a sparse neighborhood cover.

2.6 Low treedepth colorings

A crucial algorithmic tool in the study of bounded expansion and nowhere dense graph classes are low treedepth colorings, also known as r-centered colorings.

▶ **Definition 13.** An r-treedepth coloring of a graph G is a coloring of vertices of G such that any $r' \leq r$ color classes induce a subgraph with treedepth at most r'.

The following statement by Zhu [36] is modified such that it is constructive and holds also for a given vertex ordering π . It follows from the original proof.

▶ Proposition 14 ([36, Proof of Thm. 2.6]). If π is a vertex ordering of a graph G with $\operatorname{wcol}_{2^{r-2}}(G,\pi) \leq m$, an r-treedepth coloring can be computed with at most m colors in time O(mn).

Graph classes of bounded expansion can be characterized by low tree depth colorings, i.e., each graph has an r-tree depth coloring with at most f(r) many colors.

2.7 Logic

We are mainly interested in a small fragment of first-order counting logic, namely formulas of the form $\#y\,\varphi(y\bar x)>N$ where φ is a quantifier-free first-order formula with free variables $y\bar x$ and N is a natural number.

The length of a formula φ is denoted by $|\varphi|$ and equals its number of symbols, where the length of N counts as one. All signatures are finite and the cardinality $|\sigma|$ of a signature σ equals the number of its symbols. We often interpret conjunctive clauses $\omega \in FO$ as a set of literals and write $l \in \omega$ to indicate that l is a literal of ω .

We denote the universe of a structure G by V(G). We interpret a labeled graph $G = (V, E, P_1, \ldots, P_m)$ as a logical structure with a universe V, binary relation E and unary relations P_1, \ldots, P_m .

The notation \bar{x} stands for a non-empty tuple $x_1 \dots x_{|\bar{x}|}$. We write $\varphi(\bar{x})$ to indicate that a formula φ has free variables \bar{x} . Let G be a structure, $\bar{u} \in V(G)^{|\bar{x}|}$ be a tuple of elements from the universe of G, and β be the assignment with $\beta(x_i) = u_i$ for $i \in \{1, \dots, |\bar{x}|\}$. For simplicity, we write $G \models \varphi(\bar{u})$ and $[\![\varphi(\bar{u})]\!]^G$ instead of $(G, \beta) \models \varphi(\bar{x})$ and $[\![\varphi(\bar{x})]\!]^{(G, \beta)}$.

The logic FO is defined in the usual way for functional structures. The functional depth of a formula is the maximum level of nested function applications, e.g., the formula f(g(x)) = y has functional depth 2. We define $FO[d, \sigma]$ to be all first-order formulas with functional depth d and functional signature σ .

We will both use functional and relational structures, but we will restrict ourselves to functions of arity one and relations of arity one and two. A structure G with signature σ has multiplicity m if for every distinct pair $u, v \in V(G)$, the number of function symbols $f \in \sigma$ with $u = f_G(v)$ or $v = f_G(u)$ and relation symbols $R \in \sigma$ such that $R^G(u, v)$ is at most m.

3 Exact Evaluation on Nowhere Dense Classes

In this section we consider the model-checking problem for formulas $\exists x_1 \dots x_k \# y \varphi(y\bar{x}) > N$ on nowhere dense graph classes for quantifier-free first-order formulas φ . We show that this problem can be solved in almost linear fpt time by solving its optimization variant $\max_{\bar{u} \in V(G)^{\bar{x}}} \# y \llbracket \varphi(y\bar{u}) \rrbracket$.

3.1 Replace Formulas with Clauses

We start with a simplification of the input formula. The quantifier-free formula φ is transformed into a set of weighted positive clauses, i.e. formulas which are conjunctions of positive edge relations with an integer weight assigned to them. The advantage of positive clauses is that each vertex u satisfying $\omega(u\bar{v})$ is adjacent to a vertex in \bar{u} , making the problem very local.

▶ Lemma 15. Consider a quantifier-free FO-formula $\varphi(y\bar{x})$ with signature σ . In time $f(|\varphi|)$ one can construct a set Ω with the following properties:

- 1. The set Ω contains pairs of the form $(\mu, \omega(y\bar{x}))$ where $\mu \in \mathbf{Z}$ and $\omega(y\bar{x})$ is a conjunctive clause containing only positive literals,
- **2.** $|\Omega| \leq 4^{|\varphi|}$,
- **3.** $|\omega| \leq |\varphi|$ for each $(\mu, \omega) \in \Omega$,
- **4.** $|\mu| \leq 4^{|\varphi|}$ for every $(\mu, \omega) \in \Omega$,
- 5. for every graph G and every $\bar{u} \in V(G)^{|\bar{x}|}$, $\llbracket \#y \, \varphi(y\bar{u}) \rrbracket^G = \sum_{(\mu,\omega) \in \Omega} \mu \llbracket \#y \, \omega(y\bar{u}) \rrbracket^G.$

Proof. Let L be the set of literals in φ . We construct a formula φ' , equivalent to φ , in disjunctive normal form (disjunction of conjunctions). We can assume φ' to be complete in the sense that every atom in L occurs in every clause of φ' (either positively or negatively). Thus, every clause of φ' contains $|L| \leq |\varphi|$ literals. For every conjunctive clause ω of φ' we add the tuple $(1,\omega)$ into a set Ω . Since by completeness the clauses of φ' are mutually exclusive,

$$\llbracket \# y \,\varphi(y\bar{u}) \rrbracket^G = \sum_{(\mu,\omega)\in\Omega} \mu \llbracket \# y \,\omega(y\bar{u}) \rrbracket^G. \tag{4}$$

Fix a tuple $(\mu, \omega) \in \Omega$. Unless ω contains only positive literals, we can write it as $\omega' \wedge \neg l$, where l is a positive literal. By first ignoring l and then subtracting what we counted too much we get

$$\mu \llbracket \#y \,\omega(y\bar{x}) \rrbracket^{\vec{G}} = \mu \llbracket \#y \,\omega'(y\bar{x}) \wedge l \rrbracket^{G} - \mu \llbracket \#y \,\omega'(y\bar{x}) \rrbracket^{G}. \tag{5}$$

We remove (μ, ω) from Ω and add two new entries with conjunctive clauses as in (5) such that Ω still satisfies (4). Both newly introduced formulas contain one negative literal less. It can happen that we want to add some (μ, ω') to Ω when Ω already contains (μ', ω') . In that case we replace the latter by $(\mu + \mu', \omega')$.

We perform this procedure on Ω until no longer possible. The length of each clause in Ω is still at most $|\varphi|$. The size of Ω is at most $4^{|\varphi|}$ as the complete DNF formula φ' has at most $2^{|\varphi|}$ clauses of length at most $|\varphi|$, and applying the previously described inclusion-exclusion steps exhaustively to one clause results in at most $2^{|\varphi|}$ new clauses.

As the bound for $|\Omega|$ follows from counting the resulting clauses (without deduplicating possible duplicates), the same bound of 4^{φ} also follows for the weights.

3.2 Radius-r Decomposition Tree

In the following, we will introduce a new kind of decomposition, which heavily relies on the ideas from [22]. We call it the radius-r decomposition tree. For illustration, consider a tree-depth decomposition of a graph G. It has the property that after the removal of the root v in the decomposition, for each connected component C of G-v there exists a child of v in the decomposition that contains C. In the radius-r decomposition tree, not every connected component is represented by a child but every radius-r neighborhood of G-vinstead. Another difference is that these neighborhoods are not necessarily disjoint. We will use this radius-r decomposition tree as the structure on which a dynamic program will solve $\max_{\bar{u}} \llbracket \# y \varphi(y\bar{u}) \rrbracket^G$.

▶ **Definition 16.** Let G be a graph. Let $r, \ell \in \mathbb{N}$ be such that splitter has a winning strategy for the ℓ -round radius-2r splitter game on G. Let π be an ordering of G.

A radius-r decomposition tree $T_r(G,\pi,\ell)$ is a pair (T,β) where T is a tree of depth ℓ and $\beta \colon V(T) \to V(G)$. We construct it recursively. If G is empty, $T_r(G, \pi, \ell)$ is the empty tree.

Let $s \in V(G)$ be the first move of the winning strategy of splitter for the $(\ell, 2r)$ -splitter game on G. The root is a node t with $\beta(t) = s$. For every $v \in V(G)$ we append the decomposition tree $T_r(G[X_v], \pi, \ell - 1)$ where $X_v = X_{2r}[G - s, \pi, v]$.

Note that the case $\ell = 0$ while the graph is not empty, cannot happen due to the Splitter having a winning strategy.

- ▶ Corollary 17. Let G be a graph, π a vertex ordering of G, $r, \ell \in \mathbb{N}$ and $T = T_r(G, \pi, \ell)$ a radius-r decomposition tree. Let $t \in V(T)$ be a node and T_t be the subtree of T starting at t. Then for every $u \in W := \beta(V(T_t)) \setminus \{\beta(t)\}$ there exists a child t' of t such that $N_r^{G[W]}[u] \subseteq \beta(T_{t'})$.
- As $\mathcal{X}_r = \{X_{2r}[G, \pi, v] \mid v \in V(G)\}$ is by Proposition 11 a radius-r cover, the fact follows immediately.
- ▶ Lemma 18. Let G be a graph, π a vertex ordering of G and $r, \ell \in \mathbb{N}$. Then, the radius-r decomposition tree $T = T_r(G, \pi, \ell)$ (Definition 16) has size $|T| \leq \operatorname{wcol}_{2r}(G, \pi)^{\ell}n$ and depth ℓ . The construction time is linear in |T|.
- **Proof.** By construction, the depth of the tree is determined by the depth of the splitter game, which is ℓ .

Consider the root path P_t of some node $t \in V(T)$. Then $\beta(P_t) \subseteq \operatorname{WReach}_{2r}[G, \pi, \beta(t)]$. As the length of P_t is at most ℓ , $\beta(t)$ appears at most $\operatorname{WReach}_{2r}[G, \pi, \beta(t)]^{\ell} \leq \operatorname{wcol}_{2r}(G, \pi)^{\ell}$ times (as a β -label of nodes) in T. Thus, $|T| \leq \operatorname{wcol}_{2r}(G, \pi)^{\ell} n$.

▶ Corollary 19. Let C be a nowhere dense graph class. For every $r \in \mathbb{N}$ the r-decomposition tree has constant depth, almost linear size and can be computed in almost linear time.

3.3 Cover Systems

Given a subgraph H in G with a vertex ordering π of G. A cover system of H in G is a family Z of clusters $Z_i = X_r[G, \pi, v] \in Z$ for some $r \in \mathbb{N}$ such that every connected component C of H is contained in some Z_i . A cover system is non-overlapping if all distinct clusters have an empty intersection.

- ▶ Lemma 20. For every graph G with a vertex ordering π , every $D \subseteq V(G)$ of size k, there exists a cover system of G[N[D]] in G of size at most k where each cluster has the same radius $r \leq 2^k$.
- **Proof.** We start with the clusters $X_2[G, \pi, \min_{\pi} N[d]]$ for every $d \in D$. Call this collection \mathcal{Z} . Note that \mathcal{Z} is already a valid cover system of G[N[D]] in G. If two distinct clusters $X_r[G, \pi, z]$ and $X_r[G, \pi, z']$ from \mathcal{Z} intersect, we replace both with a new cluster $X_{2r}[G, \pi, \min_{\pi} \{z, z'\}]$ in \mathcal{Z} . Every vertex or edge covered by the two old clusters stays covered in the new one. Also, if two clusters $X_r[G, \pi, z]$ and $X_{r'}[G, \pi, z']$ are of a different radius, say, r' < r, we replace $X_{r'}[G, \pi, z']$ with $X_r[G, \pi, z']$ to match the radii of all the clusters.

We repeat this until no intersecting clusters remain. As the number of clusters decreases with every step, the radius is at most 2^k at the end.

For Theorem 1, one needs to find clusters from \mathcal{X}_r which are disjoint and maximize the sum of weights of clusters. This is captured by the following definition. We can solve this problem in almost linear time on nowhere dense graph classes, by noticing that the intersection graphs of the sparse neighborhood covers \mathcal{X}_r are almost nowhere dense. Then, one can use treedepth colorings and LinEMSOL.

Definition 21 (Disjoint Cluster Maximization). Given a graph, a set system \mathcal{X}_r as defined in Definition 12, labelled by a function $\Lambda: \mathcal{X}_r \to 2^{\Lambda}$ of size k. Each combination of a cluster $X \in \mathcal{X}_r$ and label $\lambda \in \Lambda(X)$ is weighted by a function w.

Problem: Find pairwise disjoint clusters $X_1, \ldots, X_k \in \mathcal{X}_r$ such that for each label $\lambda_i \in \Lambda$ the cluster X_i is labeled λ_i and X_1, \ldots, X_k maximize $\sum_{i=1}^k w(X_i, \lambda_i)$ for such cluster sets. Parameter: r, k

Lemma 22. Let \mathcal{C} be a nowhere dense class of graphs and $r \in \mathbb{N}$. Then there exists an almost nowhere dense graph class \mathcal{I} such that for every graph $G \in \mathcal{C}$, the intersection graph I of \mathcal{X}_r (defined in Definition 12) is contained in \mathcal{I} .

Proof. Assume \prec witnesses a good order in G. We build a new order \prec_I for I. $X_r[v]$ is a shorthand for $X_r[G, \prec, v]$. We say $X_r[v] \prec_I X_r[u]$ if $v \prec u$. Then

$$\begin{split} \operatorname{WReach}_s[I, \prec_I, X_r[u]] &= \{Y \in \mathcal{X} \mid \operatorname{path} P = Y_0 \dots Y_s \text{ of length at most } s \text{ in } I, \\ Y_0 &= X_r[u], Y_s = Y = \min_{\prec I} P \} \\ &\subseteq \{X_r[v] \in \mathcal{X} \mid v \in \operatorname{WReach}_{2rs}[G, \prec, u] \} \end{split}$$

Note that $v = \min_{\prec} X_r[v]$. Hence, the last equation follows. As WReach_{2rs} $[G, \prec, u] \leq n^{\varepsilon}$, so is WReach_s[$I, \prec_I, X_r[u]$]. Thus, \mathcal{I} is almost nowhere dense.

▶ Remark 23. Note that this result cannot be improved to a nowhere dense class of intersection graphs for \mathcal{X}_r . However, maybe there exists another sparse neighborhood cover whose intersection graph is nowhere dense.

Example: Consider the class of graphs with an independent set of size n with a star of size of log n. For the weak color ordering, order the apex to the right (this is not optimal but the weak coloring number of this ordering is $\log n$). The resulting intersection graph contains then a clique of size n. Hence, the which is somewhere dense.

▶ Lemma 24. We can solve the Disjoint Cluster Maximization problem in almost linear FPT time on nowhere dense class of graphs.

Proof. As G is from a nowhere dense graph class, we can apply Lemma 22, yielding a graph H from an almost nowhere dense graph class. The labels and weights from G are also added to H.

Two clusters $X, Y \in \mathcal{X}_r$ are disjoint in G if and only if X and Y are not adjacent in H. Hence, the original problem on G is equivalent to finding an independent set S of size in H, where $\sum_{i} w(S_i, \lambda_i)$ is maximized.

Since H is from an almost nowhere dense graph class, by Proposition 14 there exists a k-treedepth coloring of H using n^{ε} many colors. As the optimal independent set S with the constraints from above has size k, it has to be contained in the subgraph of H induced by some selection of at most k colors. Thus, for each selection of k colors, we consider the graph H' induced by those which has treedepth at most k. As this independent set variation can be expressed as MSO-formula and the objective function is linear, we can use LinEMSOL on H' to solve this problem optimally. The solution for H is then the maximum over the solutions of all H's.

Applying Lemma 22 takes almost linear time. There are $\binom{n^{\varepsilon}}{k} \leq n^{\varepsilon'}$ many color combinations and each iteration of LinEMSOL takes linear FPT time.

Let Ω be the set of weighted positive conjunctive clauses $(\mu, \omega(y\bar{x}))$, $\bar{z} \subseteq \bar{x}$ and $\bar{u} \in V(G)^{|\bar{x}|}$. With $\Omega|_{\bar{z}}$ we denote a subset of Ω with weighted clauses $(\mu, \omega(y\bar{x}))$ where every variable

occurring in ω is from \bar{z} . We define $\Omega|_{\bar{z}}[Z,\bar{u}]$ as $\sum_{v\in Z}\sum_{(\mu,\omega)\in\Omega|_{\bar{z}}}\mu[\![\omega(v\bar{u})]\!]^G$. Note that $\Omega|_{\bar{z}}[Z,\bar{u}]$ depends only on the assignment of \bar{z} and does not need the full assignment \bar{u} of \bar{x} . To illustrate the following lemma, consider a positive conjunctive clause $\omega(y\bar{x}\bar{z})$, sets $P,W\subseteq V(G)$ and $\bar{u}\in P^{\bar{x}}, \bar{w}\in W^{\bar{z}}$. To count the fulfilling vertices $v\in W$ of ω , i.e. $\Omega[W,\bar{u}]$, we want to reduce this task to counting on cover systems of $N[\bar{w}]$. However, as not all fulfilling vertices in W are adjacent to \bar{w} , we need to be more careful.

▶ Lemma 25. Let G be a graph, Ω a set of weighted positive conjunctive clauses $(\mu, \omega(y\bar{x}\bar{z}))$, $P, W \subseteq V(G)$ disjoint, $\bar{u} \in P^{\bar{x}}, \bar{w} \in W^{\bar{z}}$ such that $N[\bar{w}] \subseteq P \cup W$. For every cover system \mathcal{Z} of $G[N[\bar{w}]]$ in G[W] it holds that

$$\Omega[W,\bar{u}\bar{w}] = \Omega|_{y\bar{x}}[W,\bar{u}\bar{w}] + \sum_{Z\in\mathcal{Z}} (\Omega|_{y\bar{x}\bar{z}_Z}[Z,\bar{u}\bar{w}] - \Omega|_{y\bar{x}}[Z,\bar{u}])$$

where \bar{z}_Z are the variables z_i from \bar{z} which are assigned to a vertex in Z.

Proof. For $u \in W$, $G \models \omega(u\bar{u}\bar{w})$ only if u is adjacent to some vertex from $\bar{u}\bar{w}$, as ω is a positive conjunctive clause. Hence, $\Omega[W \setminus N[\bar{w}], \bar{u}\bar{w}] = \Omega|_{y\bar{x}}[W \setminus N[\bar{w}], \bar{u}\bar{w}]$. This also holds for $N \setminus \bigcup \mathcal{Z}$ instead. By the same observation $\Omega[Z, \bar{u}\bar{w}] = \Omega|_{y\bar{x}\bar{z}_Z}$ for $Z \in \mathcal{Z}$. We get the equality as a result of the observation above and subtracting $\Omega|_{y\bar{x}}[Z, \bar{u}\bar{w}]$ to prevent counting vertices in Z twice.

Let us consider how a solution \bar{u} for $\#y\,\varphi(y\bar{x})$ interacts with a radius-r decomposition of the input graph G where r is chosen appropriately big, e.g. 2^k resulting from Lemma 20. First, we transform φ into a set of positive clauses Ω using Lemma 15, making the application of Lemma 25 possible.

Consider some node t in T_r . When applying Lemma 25 with P as the vertices of the root path of t and W as T_t , we see that the resulting cover system \mathcal{Z} corresponds to a selection of children of t in T_r , as both use the sets X_r from Definition 12. Now imagine that we know $\Omega_{y\bar{x}\bar{z}z}[Z,\bar{u}]$ for every $Z \in \mathcal{Z}$. Note that this number only depends on the assignment of $\bar{x}z_z$ and not the vertices assigned outside P and Z. With Lemma 25 we can combine these numbers into $\Omega[W,\bar{u}]$ without needing to know the actual assignments of \bar{z}_Z in the cover system anymore! Note that $\Omega_{y\bar{x}}[Z]$ is easily computable while only knowing \bar{u} and not \bar{w} .

Thus, we can compute $[\![\# y \varphi(y\bar{u})]\!]$ bottom-up using the radius-r decomposition while only considering the vertices assigned in \bar{u} which are contained in the root path of the considered vertex.

3.4 Dynamic Program

To determine $\max_{\bar{u}} \# y \varphi(y\bar{u})$ for a quantifier-free formula $\varphi(y\bar{x})$ we recursively compute the following information in the decomposition tree of G (bottom-up, if you will). Consider some node t of T and a partial assignment α of \bar{x} to the root path $\beta(P_t)$. The interesting information is: How many vertices underneath t, i.e. in $V(G_t)$, fulfill φ under the "best" choice on completing the assignment α to vertices in $V(G_t)$. Then the answer to the problem can be read off the information for the root node.

Assume we already know this kind of information for every child t' of t. To compute this information for t, we branch how the variables x_i that are not assigned under α are distributed among the children of t. Then the table entries of these children are combined in a suitable way. We do this for every distribution among children and take the maximum of the resulting values. If a vertex corresponding to t fulfills with the assignment the formula φ , it gets counted towards the number of "fulfilling" vertices.

However, we have to take more into consideration. First, branching on the distribution of the unassigned variables x_i s under α among the children of t is not fast enough, as there are around n^k possibilities for that. Instead, we branch on how the unassigned variables are partitioned. For every such partition, we formalize the optimal choice of children t_i such that they contain exactly the unassigned variables from the i-th part, as an optimization problem.

Secondly, the graphs $G_{t'}$ spanned by each child t' of t are in general not disjoint. Combining the counts of two overlapping graphs yields to double counting. We circumvent this in the above optimization problem.

Thirdly, we need to keep track of how the vertices in the root path P_t are adjacent to the variables x_i that are assigned underneath t. We cannot branch on the complete assignment as the number of those is too high.

Before we turn to the dynamic program on the decomposition tree, we consider something simpler:

Let G be a graph and $\varphi(y\bar{x})$ be quantifier-free FO formula. Consider the pair (P,W) which is a set of vertices $P = \{v_1, \ldots, v_k\} \subseteq V(G)$ and a set $W \subseteq V(G)$ that is disjoint with P. We are interested in how many vertices v in $G[P \cup W]$ satisfy $\varphi(v\bar{u})$ for an optimal choice of $\bar{u} \in (P \cup W)^{|\bar{u}|}$. For this, we keep track of $M_{\alpha}^{(P,W)}[S]$, which is the number of fulfilling vertices $v \in W$ wrt. φ , $\hat{\alpha}$ and S, maximizing over S-completions $\hat{\alpha}$ on W.

We can "forget" a vertex v, i.e., derive the information of $(P, W \cup \{v\})$ from the information $(P \cup \{v\}, W)$ as follows: Assume the maximum number of fulfilling vertices in W is x for a given partial assignment α on $P \cup \{v\}$ and adjacency profile S on $P \cup \{v\}$. Then the number of fulfilling vertices in $W \cup \{v\}$ is x+1 if v satisfies φ with the assignment α and adjacency profile S, or x otherwise. However, neither α nor S are valid assignments or adjacency profiles for P. Hence, we need to adjust these so that we can formulate this information for $(P, W \cup \{v\})$. For this, we need to remove v from α and add the neighborhood of v in P to S as S_i , for every i with $\alpha(x_i) = v$. Then, $M_{\alpha}^{(P \cup \{v\}, W)}[S] = M_{\alpha|_P}^{(P, W \cup \{v\})}[S'](+1)$ where $\alpha|_P$ is the assignment α without v and S' is the adjacency profile as described above.

One can also combine the information of two structures (P, W_1) and (P, W_2) to get the information of $(P, W_1 \uplus W_2)$ if W_1 and W_2 are disjoint. This is also known as "merge." Consider some assignment α on P and some adjacency profile S on P. Then the number of fulfilling vertices in $U \uplus W$ wrt φ , α and S is the $\max\{M_{\alpha}^{P,W_1}[S_1] + M_{\alpha}^{P,W_2}[S_2] \mid S_1 \uplus S_2 = S\}$.

Indeed however, the algorithm does not take a quantifier-free formula φ but a set of weighted positive conjunctive clauses. Instead of just counting the fulfilled vertices, it computes the added up weight of them wrt. to the weights of the clauses.

3.4.1 Some definitions

Let $I \subseteq \mathbf{N}$. An *I-adjacency profile* S of a set P is a collection of sets $\{S_i \subseteq P\}_{i \in I}$. For $J \subseteq I$, we denote with $S|_J$ the collection $\{S_i\}_{i \in J}$. Equivalently, S can be interpreted as a function $S \colon P \to 2^I$.

Let G be a graph and T its r-decomposition, t a node in T, α a partial assignment of \bar{x} on P_t and an I-adjacency profile S on P_t , where $I \subseteq [|\bar{x}|] \setminus \text{dom}(\alpha)$. Then $\hat{\alpha}$ is an S-refinement of α if there exists a partial assignment $\bar{\alpha}$ of $(x_i)_{i \in I}$ on $V(G_t)$ and $N(\bar{\alpha}(x_i)) \cap P_t = S_i$ for all $i \in I$ and $\hat{\alpha} = \alpha \uplus \bar{\alpha}$.

Let $\psi(\bar{x})$ be a conjunctive clause and $\bar{z} \subseteq \bar{x}$. The \bar{z} -projection of ψ , $\psi \star \bar{z}$, is the conjunctive clause which contains the literals of ψ that involve at least one variable of \bar{z} . Note that for a graph G and $\bar{u} \in V(G)^{\bar{x}}$ $G \models \psi(\bar{u})$ if and only if $G \models (\psi \star x_i)\psi(\bar{u})$ for all $x_i \in \bar{x}$.

We say v fulfills a positive conjunctive clause $\omega(y\bar{x})$ wrt. a partial assignment α , an I-adjacency profile S and a complete conjunctive clause $\xi(\bar{x})$ if for every literal $E(yx_i)$ in ω

either v is adjacent to $\alpha(x_i)$ (if assigned) or $v \in S_i$ (if S_i exists). The weight of v in Ω wrt. α and S is

$$\sum \{\mu \mid (\mu,\omega) \in \Omega \text{ and } v \text{ fulfills } \omega \text{ wrt. } \alpha,\, S\}.$$

Let $P, W \subseteq V(G), v \in P$ such that $N^G[W \cap \hat{\alpha}(\bar{x})] \subseteq W \cup P$. Consider an assignment $\hat{\alpha}(\bar{x})$, complete conjunctive clause $\xi(\bar{x})$ with $G \models \xi(\hat{\alpha}(\bar{x}))$. Then $\Omega[v, \hat{\alpha}(\bar{x})] = \sum \mu \llbracket \omega(v\hat{\alpha}(\bar{x})) \rrbracket = \Omega[v, G, \alpha, S, \xi]$.

With all the tools at hand, we can formulate Algorithm 1 and show its correctness.

▶ Lemma 26. Let $\xi(\bar{x})$ be a complete conjunctive clause, Ω_{ξ} be a set of weighted complete conjunctive clauses $(\mu, \omega(y\bar{x}))$ where y appears in every literal. Let G be a graph and T be a radius-r decomposition of G with $r = 2^k$. Then Algorithm 1 computes

$$\max_{\bar{u}} \sum_{(\mu,\omega') \in \Omega_{\xi}} \left[\!\!\left[\# y \, \omega'(y\bar{u}) \wedge \xi(\bar{u}) \right]\!\!\right]^G.$$

Proof. Notice that M maps adjacency profiles of P_t to integers. Let S be an I-adjacency profile on P_t for some I. At the end of the recursive call of $algo(t,\alpha)$, For every S, there exists an S-refinement $\hat{\alpha}$ to G_t such that $M[S] = \Omega[G_t, \hat{\alpha}(\bar{x})]$ and $G[G_t \cup P] \models (\xi \star I)(\hat{\alpha}(\bar{x}))$. Let $\hat{\alpha}$ be such a refinement that maximizes M[S].

Let $v = \beta_T(t)$. Assume t is a leaf. Then, v can take the multiple roles of any unassigned x_i under α' or no role at all. Assume $\alpha'(x_i) = v$. Then S_i is the neighborhood of v on P_t and $S = \{S_i \mid \alpha'(x_i) = v\}$. In any case, v fulfills ω wrt. G, S and α if and only if v fulfills ω , G, \emptyset and α' for all $\omega \in \Omega$. This is computed in lines 3-11.

Otherwise, assume t is an internal node of T.

Consider an *I*-adjacency profile *S* of P_t and et $\hat{\alpha}$ be an *S*-refinement of α to G_t which maximizes the weight of vertices in G_t wrt. G and $\hat{\alpha}$, i.e $\Omega[G_t, \hat{\alpha}(\bar{x})]$.

We change our viewpoint from P_t to $P_t \cup \{v\}$. For this, let α' be the restriction of $\hat{\alpha}$ to $P_t \cup \{v\}$. In another words, α' is an refinement of α to $P_t \cup \{v\}$. Let S' be the adjacency profile on $P_t \cup \{v\}$ of $\hat{\alpha}$, i.e., $S'_i = N[\hat{\alpha}(x_i)] \cap (P_t \cup \{v\})$ for $\hat{\alpha}(x_i) \in V(G_t - v)$.

To determine $\Omega[G_t - v, \hat{\alpha}(\bar{x})]$ we want to apply Lemma 25: Setting $\bar{x}' = \text{dom}(\alpha') \subseteq \beta(P_t \cup \{v\})$, by Lemma 25 there exists a cover system \mathcal{Z} in $G_t - v$ of radius r such that

$$\Omega[G_t - v, \hat{\alpha}(\bar{x})] = \Omega|_{y\bar{x}'}[G_t - v, \hat{\alpha}(\bar{x})] + \sum_{Z \in \mathcal{Z}} (\Omega|_{y\bar{x}'\bar{z}_Z}[Z, \hat{\alpha}(\bar{x})] - \Omega|_{y\bar{x}}[G_t - v, \hat{\alpha}(\bar{x})]). \quad (6)$$

Note that by definition $M_{\alpha'}[S] = \Omega[G_t - v, \hat{\alpha}(\bar{x})]$ which equals $\max_{\bar{w}} \Omega[G_t - v, \alpha'(\bar{x})\bar{w}]$ where \bar{w} ranges over tuples \bar{w} whose sets neighborhoods equals S. Both $\Omega|_{y\bar{x}'}[G_t - v, \hat{\alpha}(\bar{x})]$ and $\Omega|_{y\bar{x}'}[Z, \hat{\alpha}(\bar{x})]$ can be easily computed in linear time (without recursion) as their evaluation depends only on α' .

To compute the above sum (Equation (6)), we need to determine $\Omega|_{y\bar{x}'\bar{z}_Z}[Z,\hat{\alpha}(\bar{x})]$ recursively. Consider $Z \in \mathcal{Z}$ and let $S' = \{S_i \in S \mid \bar{w}_i \in Z\}$. As the covering system \mathcal{Z} from Lemma 25 is a subset of \mathcal{X}_r , by construction (Definition 16) there exists a child t' of t with $V(G_{t'}) = Z$ and for that by induction, $M_{t'}[S'] = \Omega|y\bar{x}\bar{z}_Z[Z,\hat{\alpha}(\bar{x})]$.

Finding such a cover system \mathcal{Z} for an optimal choice of $\hat{\alpha}(\bar{x})$ is modeled with an instance of the Disjoint Cluster Maximization problem where the weights are set as described in Equation (6) (lines 16-25). As the form of the cover system is not known beforehand, i.e., it is not know which x_i belong into the same cover system, the algorithm branches over all partitions of unassigned variables.

To recap, before line 26 $M_{\alpha'}[S'] = \Omega[G_t - v, \hat{\alpha}(\bar{x})]$ where S' is an adjacency profile on $P_t \cup \{v\}$. Now note that at this point it is not guaranteed that $\hat{\alpha}(\bar{x})$ does not contradict

 ξ , i.e., $G \models (\xi \star I)(\hat{\alpha}(\bar{x}))$. By induction, we know that $G \models (\xi \star \bar{z}_Z)(\hat{\alpha}(\bar{x}))$ for all $Z \in \mathcal{Z}$. Hence, for the algorithm it remains to make sure whether $G \models (\xi \star J)(\hat{\alpha}(\bar{x}))$ for the variables $J = \hat{\alpha}^{-1}(v) = \alpha'^{-1}(v)$. This can be derived from S' and α' and happens in line 27.

After line 34, $M_{\alpha'}[S] = \Omega[G_t, \hat{\alpha}(\bar{x})]$ where S is an adjacency profile on $\beta(P_t)$ (instead of $\beta(P \cup \{x\})$ as before).

As now all information about v is taken care of, the parts of the assignment which are assigned to v are forgotten and collect the resulting values into M[S] (lines 35-36).

If t is the root of T, we return $M[\varnothing]$ (which is the only entry of M) which is $\max_{\bar{u}} \Omega[\emptyset, \bar{u}] =$ $\max_{\bar{u}} \sum_{(u,\omega) \in \Omega} \mu \llbracket \#y \, \omega(z\bar{u}) \rrbracket^G \wedge \llbracket \psi(\bar{u}) \rrbracket^G.$

▶ Theorem 1. Let C be a nowhere dense graph class. For every $\varepsilon > 0$, every graph $G \in \mathcal{C}$ and every quantifier-free first-order formula $\varphi(y\bar{x})$ we can compute a vertex tuple \bar{u}^* that maximizes $\llbracket \# y \varphi(y\bar{u}^*) \rrbracket^G$ in time $O(n^{1+\varepsilon})$.

Proof. Using Lemma 15 we can compute a set of weighted positive conjunctive clauses Ω with

$$\llbracket \# y \, \varphi(y \bar{u}^*) \rrbracket^G = \sum_{(\mu, \omega) \in \Omega} \llbracket \# y \, \omega(y \bar{u}^*) \rrbracket^G$$

for every $\bar{u} \in V(G)^{\bar{x}}$ in time f(k).

For every complete conjunctive clause $\xi(\bar{x})$, we compute the set Ω_{ξ} . Let $\omega(\bar{x})$ be a conjunctive clause. We decompose ω into $\omega(\bar{x}) \equiv \omega'(y\bar{x}) \wedge \psi(\bar{x})$ where $\psi(\bar{x})$ is the conjunction of literals of ω which contain only \bar{x} as variables and $\omega'(y\bar{x})$ are remaining literals of ω . For every $(\mu, \omega) \in \Omega$, $(\mu, \omega'(y\bar{x}))$ is added to Ω_{ξ} where $\omega(y\bar{x}) \equiv \Delta(y\bar{x}) \wedge \psi(\bar{x})$ as above and $\xi(\bar{x}) \models \psi(\bar{x})$. Note that for every vertex tuple \bar{u} there exists exactly one such ξ with $G \models \xi(\bar{u})$. Also, for that ξ

$$\sum_{(\mu,\omega)\in\Omega} \llbracket \#y\,\omega(y\bar{u}) \rrbracket^G = \sum_{(\mu,\omega')\in\Omega_\xi} \llbracket \#y\,\omega'(y\bar{u}) \rrbracket \wedge \llbracket \xi(\bar{u}) \rrbracket^G.$$

Computing a good ordering π of G with $\operatorname{wcol}_r(G) \leq n^{\varepsilon}$ and a decomposition tree $T_r(G, \pi, \ell)$ tales almost linear time by Lemma 8 and Lemma 18.

Using Algorithm 1 on G, T, Ω_{ξ} and ξ for every complete conjunctive clause ξ and taking the best result of those calls, gives us by Lemma 26 the correct result for the stated problem.

The (non-recursive) computation of a child takes t almost linear time in $V(G_t)$. Also, for every child t' of t, there is a recursive call. We get the following recurrence relation R(j,n)for the time needed to evaluate a node t at level j and $n = |G_t|$:

$$R(0,n) \le c$$

$$R(j,n) \le \sum_{X \in \mathcal{X}_r} cR(j-1,|X|) + cn^{1+\delta} \quad \text{for all } j \ge 1$$

In [22], the authors showed that R(j,n) can be bounded by $c^{\ell}n^{1+\varepsilon}$. As c and ℓ only depend on φ , \mathcal{C} and ε , we get the desired result.

Characterizing Almost Nowhere Dense Graph Classes

In this section, we provide various characterizations of almost nowhere dense classes, i.a. via bounded depth minors and generalized coloring numbers.

Algorithm 1 $algo(t, \alpha)$

```
Input: A graph G with a decomposition T of G, a node t of T, a partial assignment \alpha of \bar{x} on P_t,
               complete conjunctive clause \xi(\bar{x}) and a set \Omega of weighted positive clauses \omega(\bar{x})
    Output: M with M[S] as described above.
 1 M, M_{t'} := are empty associative arrays over the family of subsets of \beta(P_t) for every child t' of t.
      If an entry is not in the array its value is -\infty;
 v := \beta_T(t) (vertex of t);
    /* Base case
                                                                                                                               */
 \mathbf{3} if t is a leaf in T then
          for
each Possible refinement \alpha' of \alpha to
 v do
              if \alpha' and S contradict \xi then skip;
 5
               S := \{\};
 6
              foreach i \in \alpha'^{-1}(v) do
 7
                    S_i := N[v] \cap \beta(P_t);
  8
                    S := S \cup \{S_i\};
              M_{\alpha}[S] := \Omega_{y \operatorname{dom}(\alpha')}[v, \alpha'(\bar{x})]
         return M;
11
12 foreach Possible refinement \alpha' of \alpha to v do
          clear M_{t'}s;
13
          foreach Child\ t' of t do
14
               M_{t'} := algo(t', \alpha'); 
15
          /* combine results from children
          foreach I \subseteq [k] \setminus dom(\alpha') do
16
              /* Not assigned x_is
              foreach I-adjacency profile S on P_t \cup \{v\} do
17
                    foreach Partition \mathcal{I} of I do
18
                         Init w\colon \mathcal{X}_r^{G_t} \times \mathcal{I} \to \mathbf{N} /* w is a weighting function
19
                         foreach Child t' of t and J \in \mathcal{I} do
20
                              \delta := \Omega|_{y\operatorname{dom}(\alpha')}[G_{t'}, \alpha'(\bar{x})];
21
                           w(V(G_t), J) := M_{t'}[S|_J];
22
                         \Delta := \Omega|_{y \operatorname{dom}(\alpha')J} [G_t - v, \alpha'(\bar{x})];
23
                         d^* := \Delta + weight of Disjoint Cluster Maximizer of \mathcal{X}_r^{G_t} and w;
24
                         M_{\alpha'}[S] := \max\{M_{\alpha'}[S], d^*\};
25
          /* forget v
                                                                                                                               */
26
          foreach S \in M_{\alpha'} do
              if \alpha' and S contradict \xi then remove S from M_{\alpha'} and skip;
27
               M_{\alpha}[S] := \text{weight of } v \text{ in } \Omega \text{ wrt. } \alpha' \text{ and } S;
28
               S' := S;
              /* add the n'hood of v to adjacency profile with index of x in part.
                   assignment \alpha'
                                                                                                                               */
               foreach i \in \alpha'^{-1}(v) do
30
                    S_i := N[v] \cap \beta(P_t);
                    S' := S' \cup \{S_i\};
32
               M_{\alpha'}[S'] := M_{\alpha'}[S];
33
              if S \neq S' then remove S from M_{\alpha'};
34
    /* collect
35 foreach adjacency profile S (without v) do
     M[S] = \max\{M_{\alpha'}[S] \mid \alpha' \text{ is an refinement of } \alpha \text{ on } v\};
    /* return
37 if v is the root of T then return M[\varnothing];
38 else return M;
```

- ▶ **Definition 27** (Almost nowhere dense). A graph class C is almost nowhere dense if for every $r \in \mathbb{N}$, $\varepsilon > 0$ there exists n_0 such that no graph $G \in C$ with $|G| \ge n_0$ contains $K_{\lceil |G|^{\varepsilon} \rceil}$ as a depth-r minor.
- **Theorem 28.** Let C be a graph class. The following statements are equivalent.
- 1. C is almost nowhere dense.
- **2.** For every $r \in \mathbb{N}$, $\varepsilon > 0$ there exists n_0 such that no graph $G \in \mathcal{C}$ with $|G| \ge n_0$ contains $K_{\lceil |G|^{\varepsilon} \rceil}$ as a depth-r minor.
- **3.** For every $r \in \mathbb{N}$, $\varepsilon > 0$ there exists n_0 such that no graph $G \in \mathcal{C}$ with $|G| \ge n_0$ contains $K_{\lceil |G|^{\varepsilon} \rceil}$ as a depth-r topological minor.
- **4.** For every $r \in \mathbb{N}$, $\varepsilon > 0$ there exists n_0 such that no graph $G \in \mathcal{C}$ with $|G| \ge n_0$ contains an r'-subdivision of $K_{\lceil |G| \in \rceil}$ as a subgraph for any $r' \le r$.
- **5.** For every $r \in \mathbb{N}$, $\varepsilon > 0$ there exists n_0 such that $\operatorname{wcol}_r(G) \leq |G|^{\varepsilon}$ for every graph $G \in \mathcal{C}$ with $|G| \geq n_0$.
- **6.** For every $r \in \mathbb{N}$, $\varepsilon > 0$ there exists n_0 such that $\operatorname{col}_r(G) \leq |G|^{\varepsilon}$ for every graph $G \in \mathcal{C}$ with $|G| \geq n_0$.

The characterizations from Theorem 28 are very similar to those for nowhere dense classes. The only difference in the characterizations 1. to 4. would be the size of the forbidden cliques: for nowhere dense classes, the size would be f(r) instead of $\lceil |G|^{\varepsilon} \rceil$. Similarly, if we would substitute "for every $G \in \mathcal{C}$ " with "for every subgraph $G \subseteq H \in \mathcal{C}$ " in characterizations 5 and 6 would characterize nowhere dense classes. Note that every almost nowhere dense class which is monotone, i.e. closed under taking subgraphs, is also nowhere dense.

Conversely, if a class \mathcal{C} is almost nowhere dense, then its subgraph-closure \mathcal{C}_{\subseteq} is not almost nowhere dense in general. Consider for this the class of graphs which for every $n \in \mathbb{N}$ contains independent set of size n with a clique of size $\log n$, i.e. the graph $I_n \cup K_{\log n}$. This class is almost nowhere dense but its subgraph-closure contains cliques K_n of every size n as member, and so, all graphs.

We need the following theorem by Grohe, Kreutzer and Siebertz [21], which in turn builds upon the original results of Kierstead and Yang [25] and Zhu [36].

▶ Proposition 29 ([21, Theorem 3.3]). There exists a function $f: \mathbb{N} \to \mathbb{N}$ such that for all $d, r \in \mathbb{N}$ and all classes \mathcal{C} of graphs, if the class of all topological depth-r minors of \mathcal{C} is d-degenerate then $\operatorname{col}_r(G') \leq f(r) \cdot d$ for every subgraph $G' \subseteq G$ of a graph $G \in \mathcal{C}$.

By setting \mathcal{C} to be the class containing only a single graph and reversing the statement, we get the following statement that better suits our needs.

▶ Corollary 30. There exists a function $f \colon \mathbf{N} \to \mathbf{N}$ such that for all $d, r \in \mathbf{N}$ and all graphs G, if $\operatorname{col}_r(G) \geq f(r) \cdot d$ then there exists a topological depth-r minor of G that is not d-degenerate.

We also need the following observations about degeneracy.

▶ Fact 1. A d-core of a graph G is a maximally connected subgraph of G in which all vertices have degree at least d. The degeneracy of a graph G is the largest number d for which G has a d-core. If a graph is not d-degenerate then it has a (d+1)-core and therefore a subgraph in which all vertices have degree at least d+1. A graph with degeneracy d has at most nd edges.

Combining Corollary 30 and Fact 1 yields:

▶ Lemma 31. Let $\rho > 0$ and $r \in \mathbb{N}$. There exists $n_0 = n_0(\rho, r)$ and $\mu = \mu(\rho) > 0$ such that all graphs G on $n \ge n_0$ vertices with $\operatorname{col}_r(G) \ge n^{1/\rho}$ contain $a \le (r+1)(9^{2\rho}+1)$ -subdivision of $K_{\lceil n^{\mu} \rceil}$ as a subgraph.

Proof. Let $G \in \mathcal{C}$ be a graph of order n with $\operatorname{col}_r(G) \geq n^{1/\rho}$ and f(r) be the function of Corollary 30. By Corollary 30, G has a topological depth-r minor that is not $n^{1/\rho}/f(r)$ -degenerate. According to Fact 1, G also has a topological depth-r minor H in which all vertices have degree at least $n^{1/\rho}/f(r)$. Without loss of generality, we can assume $n \geq n_0(\rho, r)$ to be large enough that $n^{1/\rho}/f(r) \geq n^{1/2\rho}$. Then Proposition 32 guarantees that there exists $\mu(\rho)$ such that H has a $\leq 9^{2\rho}$ -subdivision of $K_{\lceil n^{\mu(\rho)} \rceil}$ as a subgraph. By transitivity, this means G has a $\leq (r+1)(9^{2\rho}+1)$ -subdivision of $K_{\lceil n^{\mu(\rho)} \rceil}$ as a subgraph.

We use the following statements about subdivided cliques in graphs with polynomial minimum degree.

- ▶ Proposition 32 ([21, Lemma 2.12]). Let $\rho > 1$. There exists $n_0 = n_0(\rho)$ and $\mu = \mu(\rho) > 0$ such that all graphs G on $n \ge n_0$ vertices with minimum degree at least $n^{1/\rho}$ contain a 9^{ρ} -subdivision of $K_{\lceil n^{\mu} \rceil}$ as a subgraph.
- ▶ Proposition 33 ([14, Lemma 3.14]). There exists n_0 such that every graph G with $n \ge n_0$ vertices and minimum degree at least $4n^{0.6}$ contains a 1-subdivision of $K_{\lceil n^{0.1} \rceil}$ as a subgraph.

Combining these two statements yields the following useful observation.

▶ Lemma 34. There exists n_0 such that every graph G containing an $\leq r$ -subdivision of K_n as a subgraph with $n \geq r \cdot n_0$ also contains an r'-subdivision of $K_{\lceil n/r^{0.05} \rceil}$ as a subgraph for some $r' \leq 2r + 1$.

Proof. Let n'_0 be the constant from Proposition 33 and $n_0 \geq 4n'_0 + 4$. Let G be a $\leq r$ -subdivision of K_n with $n \geq r \cdot n_0$. Then there exists $r' \leq r$, such that at least n(n-1)/2(r+1) edges of K_n are subdivided exactly r'-times in G. This yields a graph H with n vertices and n(n-1)/2(r+1) edges such that G contains an r'-subdivision of H as a subgraph. A graph with degeneracy d has at most nd edges. Thus, by Fact 1, H has a (n-1)/2(r+1)-core, i.e., a subgraph with minimum degree at least (n-1)/2(r+1). Since $n \geq r \cdot n_0 \geq 4rn'_0 + 4r$, we have that $(n-1)/2(r+1) \geq n'_0$. Then by Proposition 33, H contains a 1-subdivision of a complete graph of order $\lceil ((n-1)/2(r+1))^{0.1} \rceil$ as a subgraph. Since G contains an r'-subdivision of H as a subgraph, this means that G contains a 2r' + 1-subdivision of order $\lceil ((n-1)/2(r+1))^{0.1} \rceil$ as a subgraph. Without loss of generality, we can assume $n \geq n_0$ to be large enough that $\lceil ((n-1)/2(r+1))^{0.1} \rceil \geq \lceil n/r^{0.05} \rceil$.

At last, we use all these observations to obtain a characterization of almost nowhere dense classes.

Proof of Theorem 28. The equivalence $1. \Leftrightarrow 2$. is by definition. For convenience, we show equivalence of the inverse of the remaining statements. Let us prove $\neg 2. \Rightarrow \neg 3$. Let $\omega_r(G)$ (or $\tilde{\omega}_r(G)$) be the largest value of t such that G has K_t as depth-r minor (or depth-r topological minor). According to [32, Corollary 2.20],

$$\tilde{\omega}_r(G) \le \omega_r(G) \le 1 + (\tilde{\omega}_{10r}(G) + 1)^{10r}. \tag{7}$$

If $\neg 2$, holds then there exists $r, \varepsilon > 0$ and an infinite sequence of graphs G_1, G_2, \ldots such that $\omega_r(G_i) \ge |G_i|^{\varepsilon}$. Then there also exists $\varepsilon' > 0$ and c such that $\omega_r(G_i) \ge 1 + (|G_i|^{\varepsilon'} + 1)^{10r}$ for all $i \ge c$. By (7), $\tilde{\omega}_{10r}(G_i) \ge |G_i|^{\varepsilon'}$ for $i \ge c$, which implies $\neg 3$.

The implication $\neg 3. \Rightarrow \neg 4$. follows from Lemma 34.

The implication $\neg 4. \Rightarrow \neg 2$. holds, since every graph that contains an r-subdivision of a graph H as a subgraph also contains H as depth-r minor.

Furthermore, $\neg 5. \Leftrightarrow \neg 6.$, since $\operatorname{col}_r(G) \leq \operatorname{wcol}_r(G) \leq \operatorname{col}_r(G)^r$ for every graph G. Next, $\neg 6. \Rightarrow \neg 3.$ follows from Lemma 31.

At last, we prove $\neg 3. \Rightarrow \neg 4.$: Assume a graph G contains K_t as a depth-r topological minor, where the principal vertices are $P \subseteq V(G)$, |P| = k. Let π be an ordering of G and $v \in P$ be maximal with respect to π . For every $w \in P$ let m_w be the smallest vertex with respect to π on the path from v to w in the depth-r topological minor model. Then $\{m_w \colon w \in P\} \subseteq \operatorname{WReach}_{r+1}[G, \pi, v]$. Since all paths from v to P share no vertex except v, the vertices m_w are all distinct. This means $\operatorname{wcol}_{r+1}(G) \geq t$.

5 Approximation on Almost Nowhere Dense

In this section we consider the same problem as before, i.e., finding vertices for \bar{x} that satisfy $\#y\,\varphi(\bar{x}y)>N$ but on almost nowhere dense classes of graphs. Here, we give an approximation algorithm with an additive error. For this, we use completely different techniques compared to Section 3. We first show how to reduce the corresponding model-checking problem to approximate sums over unary functions. The procedure from Lemma 41 is the only source of error. Then we present the approximate optimization algorithm in Theorem 35.

The main result of this section is the following approximate optimization algorithm with additive error.

▶ **Theorem 35.** There exists a computable function f such that for every graph G and every quantifier-free first-order formula $\varphi(y\bar{x})$ we can compute a vertex tuple \bar{u}^* with

$$|\max_{\bar{u}} \left[\!\left[\#y\,\varphi(y\bar{u})\right]\!\right]^G - \left[\!\left[\#y\,\varphi(y\bar{u}^*)\right]\!\right]^G| \leq 4^{|\varphi|}\mathrm{wcol}_2(G)^{O(|\varphi|)}$$

in time $\operatorname{wcol}_{f(|\varphi|)}(G)^{f(|\varphi|)}n$.

For the approximate model-checking problem with an additive error δ , similar to [11], we want an algorithm such that

- 1. the algorithm returns "yes" only if G satisfies the formula,
- 2. returns "no" only if G does not satisfy the formula,
- **3.** returns \perp only if the optimum is within δ to N.

The option \perp can be seen as "I do not know" as the computed result and the desired result are so close that the difference falls into the additive error δ .

Given the approximate optimization algorithm from Theorem 35, we can easily build an approximate model-checking algorithm as described above for the formula $\exists \bar{x} \# y \varphi(y\bar{x}) > N$ by computing a vertex tuple \bar{u}^* from the theorem. If $N - \llbracket \# y \varphi(y\bar{u}^*) \rrbracket^G \leq \delta$, answer \bot . Otherwise, answer "yes" or "no" according whether $\llbracket \# y \varphi(y\bar{u}^*) \rrbracket^G > N$ or not. Note that δ cannot be chosen freely as it depends on the graph (respectively, its weak coloring numbers).

The runtime of the algorithm from Theorem 35 is fpt if the weak r-coloring numbers are bounded by n^{ε} for $r \leq f(|\varphi|)$. This is the case for almost nowhere dense classes. This is in contrast to the results of [11] where the running time of their algorithms is bounded by $f(\text{wcol}_{f(|\varphi|)})||G||$ which is fpt on classes of bounded expansion but is not fpt on nowhere dense and almost nowhere dense classes.

This gives us the following corollary.

▶ Corollary 2. Let C be an almost nowhere dense class of graphs. For every $\varepsilon > 0$, every graph $G \in C$ and every quantifier-free first-order formula $\varphi(y\bar{x})$, we can compute in time $O(n^{1+\varepsilon})$ a vertex tuple $\bar{u} \in V(G)^{|\bar{x}|}$ with

$$|\max_{\bar{u}} \left[\!\left[\#y\,\varphi(y\bar{u})\right]\!\right]^G - \left[\!\left[\#y\,\varphi(y\bar{u}^*)\right]\!\right]^G | \leq n^{\varepsilon}.$$

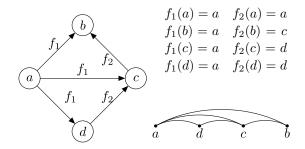


Figure 2 A graph and one of its functional representations.

5.1 Functional structures

We will heavily rely on functional representations of directed graphs, where edges are replaced with functions mapping the endpoint of an edge to its startpoint. For this, we need a functional signature consisting of $\Delta^-(\vec{G})$ many unary functions. They were used by Durand and Grandjean [12], as well as Kazana and Segoufin [24], and also by Dvořák, Kráľ, and Thomas [15]. A big advantage of functional structures is that short paths can be expressed in a quantifier-free way.

We focus on representations of graphs where the arcs are all directed along a fixed vertex ordering. One can imagine that this vertex ordering witnesses that the weak coloring number of the graph is small, which means that the number of symbols in the signature is also small.

▶ **Definition 36.** For a graph G with vertex ordering π we define the functional representation \vec{G}_{π} of G w.r.t. π as a functional structure with universe V(G) and functional signature (f_1, \ldots, f_t) for $t = \text{wcol}_1(G, \pi)$ where $[\![f_i(u)]\!]^{\vec{G}_{\pi}} = v$ if v is the ith weakly 1-reachable vertex from u with regard to π and $[\![f_i(u)]\!]^{\vec{G}_{\pi}} = u$ if $i > |\text{WReach}_1[G, \pi, u]|$.

We denote G as the underlying graph of \vec{G}_{π} and \vec{G}_{π} has multiplicity 1.

In the following we will define an augmentation that is a special case of a transitive fraternal augmentation as defined by Nešetřil and Ossona de Mendez [30]. A transitive fraternal augmentation adds transitive and fraternal arcs to a directed graph. If uv and vw are arcs, then uw is the corresponding transitive arc and if uv and uw are arcs then vw and wv are fraternal arcs. While a transitive arc is unique there are two possible fraternal arcs and in a transitive fraternal augmentation only one of them has to be added. In our case the direction of the fraternal arcs will be determined by an order π , which allows us to bound the indegree of the resulting directed graph by a weak coloring number.

▶ **Definition 37.** For a graph G and a vertex ordering π we define the augmentation \vec{G}_{π}^2 of G as the expansion of \vec{G}_{π} with the functional symbols h_i for $i \in [\operatorname{wcol}_2(G,\pi)]$ where $\llbracket h_i(u) \rrbracket^{\vec{G}_{\pi}^2} = v$ if v is the ith weakly 2-reachable vertex from u w.r.t. π and $\llbracket h_i(u) \rrbracket^{\vec{G}_{\pi}^2} = u$ if $i > |\operatorname{WReach}_2[G,\pi,u]|$.

Note that the underlying graph of \vec{G}_{π}^2 is not G. However, $\operatorname{wcol}_r(\vec{G}_{\pi}^2, \pi) \leq \operatorname{wcol}_{2r}(\vec{G}_{\pi}, \pi)$ for every $r \in [|G|]$. Moreover, the multiplicity of \vec{G}_{π} is 1 and that of \vec{G}_{π}^2 is 2. Let us look again at the graph \vec{G}_{π} in Figure 2. If we construct \vec{G}_{π}^2 then $h_1(b) = a$, $h_2(b) = b$, and $h_3(b) = c$ because a, b, and c are weakly 2-reachable from d. The augmentation has three new functional symbols because $\operatorname{wcol}_2(G,\pi) = 2$. We have $h_3(b) = f_2(b) = c$, which shows that the multiplicity is 2.

▶ Lemma 38. Given a graph G with an ordering π , one can compute \vec{G}_{π} and \vec{G}_{π}^2 in time O(||G||) and $O(\text{wcol}_2(G,\pi)^2||G||)$, respectively.

Proof. Computing \vec{G}_{π} is easy. For this, the edges of G have to be directed from right to left w.r.t. to π . The left neighbors of each vertex v have to be assigned to $f_i(v)$ in order. We only store $f_i(v)$ for the $|\text{WReach}_1(G, \pi, v)|$ many left neighbors of v.

We derive \vec{G}_{π}^2 from \vec{G}_{π} and store it as follows: As \vec{G}_{π} and \vec{G}_{π}^2 agree except for the new functions h_i , we need to store only the latter. For every $u \in V(G)$ we store all $h_i(u)$ with $h_i(u) \neq u$, i.e., only the first $|\mathrm{WReach}_2(u)| \leq d(u) \mathrm{wcol}_1(G)$ ones, by going through all neighbors x of u and then to all neighbors of x on the left of u. All found vertices are then sorted in $O(\mathrm{wcol}_2(G,\pi)\log\mathrm{wcol}_2(G,\pi))$ time. Altogether \vec{G}_{π}^2 can be computed in time

$$\sum_{u \in V(G)} O(d(u) \operatorname{wcol}_1(G, \pi) + \operatorname{wcol}_2(G, \pi) \log(\operatorname{wcol}_2(G, \pi))) = O(||G||\operatorname{wcol}_2(G, \pi)^2).$$

Note that we can compute $h_i(u)$ in constant time if we have this representation of \vec{G}_{π}^2 .

5.2 Multiplicity

▶ **Definition 39.** A structure G with signature σ has multiplicity m if for every distinct pair $u, v \in V(G)$, the number of function symbols $f \in \sigma$ with $u = f^G(v)$ or $v = f^G(u)$ and relation symbols $R \in \sigma$ such that $R^G(u, v)$ is at most m.

A quantifier-free conjunctive clause $\omega(\bar{x}) \in \text{FO}[1, \sigma]$ has multiplicity m if for distinct $i, j \in [|\bar{x}|]$ there are at most m positive literals of the form $f(x_i) = x_j$ or vice versa.

▶ **Definition 40.** We define FO[1, σ , 2] to be all quantifier-free conjunctive clauses with at most one function application, signature σ and multiplicity at most 2 where for i, j the literal $f(x_i) = x_j$ may appear only if $i \neq j$ and there is no literal $x_i = x_j$.

A clause $\omega(\bar{x}) \in \text{FO}[1, \sigma, 2]$ is complete if for every $i \neq j$ and every $f \in \sigma$ either $x_i = x_j$ or $x_i \neq x_j$ is contained in ω . Furthermore, if $x_i \neq x_j \in \omega$ then for every function symbol $f \sigma$ either $f(x_i) = x_j$ or $f(x_i) \neq x_j$ must be contained in ω . If, on the other hand, $x_i = x_j \in \omega$, then no other literal containing x_i and x_j is allowed in ω .

The complicated interaction between literals of the forms $x_i = x_j$ and $f(x_i) = x_j$ stems from the absence of self-loops.

5.3 Decomposing Formulas into Simpler Ones

Let $\omega(y\bar{x})$ be a conjunctive clause in a functional signature σ . Over the course of this section, we will repeatedly decompose such clauses into three conjunctive clauses $\tau(y), \psi(\bar{x}), \Delta^{\pm}(y\bar{x})$ such that $\omega(y\bar{x}) \equiv \tau(y) \wedge \psi(\bar{x}) \wedge \Delta^{\pm}(y\bar{x})$. We will always require that

- $\tau(y)$ has functional depth at most two and contains only the variable y,
- $\psi(\bar{x})$ contains literals of the form $x_p = f_j(x_q)$, and
- $\Delta^{=}(y\bar{x})$ contains only literals of the form $f(y)=x_i, f(x_i)=y$ and $y=x_i$ for $f\in\sigma$.

Formulas with the names $\tau(y), \psi(\bar{x}), \Delta^{=}(y\bar{x})$ will always be conjunctive clauses with the properties above and will refer to a decomposition of a clause ω , even if this is not explicitly mentioned. We call $\Delta^{=}(y\bar{x})$ also the *positive mixed literals* of ω .

The first step of the algorithm is to decompose a relational quantifier-free formula φ into a set of weighted conjunctive clauses with a restricted form. Also, we switch from a relational representation of the graph and the formula to a functional representation. The form of the clauses will be simple in the sense that there is only one literal that contains both y and a

variable from \bar{x} , which will allow us to use Lemma 43. We will use the notion of multiplicity throughout this series of lemmas to be able to apply Lemma 44.

The approximative error occurs in the procedure of Lemma 41. Here, clauses with literal y = f(x) are ignored as these cannot be handled with our techniques. However, their impact on the evaluation is relatively small as the vertices described by this literal have to be in $\operatorname{WReach}_r[G,\pi,v]$ for some small number of vertices $v\in V(G)$. Also, the number of clauses with this literal is quite small.

- **Lemma 41.** Consider a graph G with order π and a quantifier-free first order formula $\varphi(\bar{x}y)$, both with signature σ . In time $(\operatorname{wcol}_2(G,\pi)+1)^{O(|\varphi|)}$, one can construct a set Ω with the following properties:
- 1. The set Ω contains pairs $(\mu, \omega(\bar{x}y))$ where $\mu \in \mathbf{Z}$ and ω is of the form $\tau(y) \wedge \psi(\bar{x}) \wedge f(y) = x_i$
- 2. ψ has only positive literals with at most one function application,
- 3. $|\Omega| < (\text{wcol}_2(G) + 1)^{O(|\varphi|)}$,
- **4.** $|\omega| < 2|\varphi| + 1$ for each $(\mu, \omega) \in \Omega$,
- 5. for all $\bar{u} \in V(G)^{|\bar{x}|}$, and with $\delta := 4^{|\varphi|} \operatorname{wcol}_2(G)^{O(|\varphi|)}$ $\sum_{(\mu,\omega)\in\Omega} \mu[\![\#y\,\omega(\bar{u}y)]\!]^{\vec{G}_{\pi}^2} \delta \leq [\![\#y\,\varphi(\bar{u}y)]\!]^G \leq \sum_{(\mu,\omega)\in\Omega} \mu[\![\#y\,\omega(\bar{u}y)]\!]^{\vec{G}_{\pi}^2} + \delta.$

Before we can prove this lemma we consider first Lemmas 15 and 42. Lemma 15 uses inclusion-exclusion to get rid of negative literals. Then Lemma 42 switches to the functional setting. A clause that results from Lemma 15 is then transformed into a set of clauses with only one mixed positive literal each. The considered graph changes from the relational graph G, to its functional representation \vec{G}_{π} and then its augmentation \vec{G}_{π}^2 . The multiplicity is bounded during this procedure.

Lemma 42. Let \vec{G}_{π} be the functional representation of a graph G and ω be a conjunctive clause of the form $\omega(y\bar{x}) = \tau(y) \wedge \psi(\bar{x}) \wedge \Delta(y\bar{x})$ where $\Delta(y\bar{x})$ is a conjunction of positive literals $f_i(y) = x_p$.

Then a set of clauses Ω with the following properties can be computed in time (wcol₂(G, π)+ $1)^{O(|\omega|)}$:

- 1. Each clause in Ω is of the form $\tau'(y) \wedge \psi'(\bar{x}) \wedge f(y) = x_i$,
- 2. $\psi' \in FO[1, \sigma, 2, \bar{x}]$ contains only positive literals with at most one function application,
- **3.** $|\omega'| \leq |\omega|$ for each $\omega' \in \Omega$,
- **4.** $|\Omega| \leq (\text{wcol}_2(G, \pi) + 1)^{|\omega|}$,

5. for every
$$\bar{u} \in V(G)^{|\bar{x}|}$$
,
$$\llbracket \#y \, \omega(y\bar{u}) \rrbracket^{\vec{G}_{\pi}} = \sum_{\omega' \in \Omega} \llbracket \#y \, \omega'(y\bar{u}) \rrbracket^{\vec{G}_{\pi}^2}.$$

Note that in 5. the interpretation inside the sum is over \vec{G}_{π}^2 and not over \vec{G}_{π} .

Proof. Remember that the mixed positive literals of a clause are those literals contained in the part $\Delta^{=}(y\bar{x})$ of its decomposition. We start with $\Omega = \{\omega\}$ and describe a procedure that picks a clause $\omega' \in \Omega$ with l > 1 mixed positive literals, removes ω' and replaces it with $\operatorname{wcol}_2(G,\pi)+1$ clauses with at most l-1 mixed positive literals. Once this procedure cannot be applied any longer, each clause has exactly one mixed positive literal and the set Ω satisfies 1, i.e., that there is only one mixed positive literal. Since initially, ω has at most $|\omega|$ many mixed positive literals, Ω will have size at most $(\operatorname{wcol}_2(G,\pi)+1)^{|\omega|}$ upon termination of the procedure.

Let us pick a clause $\omega' \in \Omega$ and describe the procedure mentioned above in detail. There are two literals $x_p = f_i(y)$ and $x_q = f_j(y)$ in $\Delta^{=}(y\bar{x})$ with $i \leq j$. This implies that x_p is weakly 2-reachable from x_q .

Let $\Delta'^{=}(y\bar{x})$ be the clause obtained from $\Delta^{=}(y\bar{x})$ by removing the literals $x_p = f_i(y)$ and $x_q = f_j(y)$. We remove ω' from Ω and add the clause

$$x_p = x_q \wedge x_p = y \wedge \tau(y) \wedge \psi(\bar{x}) \wedge \Delta'^{=}(y\bar{x}),$$

as well as for each $k \in [\mathrm{wcol}_2(G, \pi)]$ the clause

$$x_p \neq x_q \wedge x_q = f_j(y) \wedge x_p = h_k(x_q) \wedge f_i(y) = h_k(f_j(y)) \wedge \tau(y) \wedge \psi(\bar{x}) \wedge \Delta'^{=}(y\bar{x}).$$

This means, one clause is removed and $1 + \text{wcol}_2(G, \pi)$ new clauses are added to Ω .

Remember that $h_i(x_q)$ is the *i*th weakly 2-reachable vertex from x_q or x_q itself. Thus, if there are two distinct k and k' such that $x_p = h_k(x_q)$ and $x_p = h_{k'}(x_q)$ then $x_p = x_q$. This implies that the newly added $1 + \text{wcol}_2(G, \pi)$ many clauses are mutually exclusive. The equivalence follows by observation: As x_p is 2-reachable from x_q there exists a $k \in [\text{wcol}_2(G, \pi)]$ such that $x_p = h_k(x_q)$.

With $x_p = h_k(x_q)$ and some syntactic replacements it follow that the literal $x_p = f_i(y)$ is equivalent to $f_i(y) = h_k(f_j(y))$. Hence,

$$x_p = f_i(y) \wedge x_q = f_i(y) \wedge x_p = h_k(x_q)$$

is equivalent to

$$f_i(y) = h_k(f_j(y)) \land x_q = f_j(y) \land x_p = h_k(x_q).$$

The idea of this proof is that the number of mixed literals can be decreased. Two vertices which are weakly 1-reachable from y, are connected by a functional edge in the augmented graph. With a syntactic trick, this can be expressed with fewer mixed literals. Note that we use functional representations to express these distance-2 relationships without needing to resort to quantifiers.

Finally, we are able to prove Lemma 41 by combining Lemmas 15 and 42. The other part of this proof is the transition from relational to functional representations.

Proof of Lemma 41. First, we apply Lemma 15 to G and φ , resulting in the set Ω_1 . Next, we turn to the functional representation \vec{G}_{π} of G. The signature of \vec{G}_{π} is then $\{f_i \mid i \in \text{wcol}_1(G,\pi)\}$. Let $(\mu,\omega_1) \in \Omega_1$. Note that ω contains only positive literals. We construct ω_2 by replacing every adjacency atom E(a,b) of ω_1 for $a,b \in y\bar{x}$ with

$$a = b \vee \bigvee_{i \in [\text{wcol}_1(G)]} (f_i(a) = b \wedge f_i(a) \neq a) \vee (f_i(b) = a \wedge f_i(b) \neq b).$$
(8)

Note that the disjunction in (8) is mutually exclusive in \vec{G}_{π} . Thus, each adjacency atom gets replaced with a mutually exclusive disjunction over at most $2\text{wcol}_1(G)+1$ conjunctive clauses. Therefore, transforming ω_2 into disjunctive normal form yields at most $(2\text{wcol}_1(G)+1)^{|\varphi|}$ many mutually exclusive clauses. We place each of those clauses, with a weight μ into a new set Ω_2 . This procedure is repeated for all $(\mu,\omega_1)\in\Omega_1$.

$$[\![\# y \, \varphi(y\bar{u})]\!]^G = \sum_{(\mu,\omega) \in \Omega_1} \mu [\![\# y \, \omega(y\bar{u})]\!]^G = \sum_{(\mu,\omega) \in \Omega_2} \mu [\![\# y \, \omega(y\bar{u})]\!]^{\vec{G}_{\pi}}. \tag{9}$$

Each clause in Ω_2 to is of the form $\tau(y) \wedge \psi(\bar{x}) \wedge \Delta^{=}(y\bar{x})$.

Consider the set $\Upsilon \subseteq \Omega_2$ of weighted clauses which contain a (positive or negative) literal of the form y = f(x). We claim that $|\sum_{(\mu,\omega)\in\Upsilon}\mu[\![\#y\,\omega(y\bar{u})]\!]^{\vec{G}_{\pi}}| \leq 4^{|\varphi|}\mathrm{wcol}_2(G)^{O(|\varphi|)}$.

The size of Υ is bounded by the size of Ω_2 which is $\operatorname{wcol}_2(G)^{O(|\varphi|)}$. Also, for each $\omega \in \Upsilon$ it holds that $[\![\#y\,\omega(y\bar{u})]\!] \leq \operatorname{wcol}_1(\vec{G}_\pi^2)$ as there is only one choice of y for every fixed tuple \bar{u} (due to the positive literal $y=f(u_i)$). As the weights of a clause in Ω_1 is bounded by $4^{|\varphi|}$ and the disjunction in 8 is mutually exclusive, the weights of Ω_2 are also bounded by $4^{|\varphi|}$. The claim follows directly.

Hence, removing Υ from Ω_2 changes its evaluation by at most $4^{|\varphi|} \operatorname{wcol}_2(G)^{O(|\varphi|)} s$ additively, i.e.,

$$\llbracket \#y\,\varphi(y\bar{u}) \rrbracket^G = \sum_{(\mu,\omega)\in\Omega_2} \mu \llbracket \#y\,\omega(y\bar{u}) \rrbracket^{\vec{G}_\pi} = \sum_{(\mu,\omega)\in\Omega_2\backslash\Upsilon} \mu \llbracket \#y\,\omega(y\bar{u}) \rrbracket^{\vec{G}_\pi} \pm 4^{|\varphi|} \mathrm{wcol}_2(G)^{O(|\varphi|)}.$$

We can apply Lemma 42 to $\Omega_2 \setminus \Upsilon$. The resulting set gives us the desired set Ω : Condition 5 follows from (9) and Lemma 42. The size of Ω is bounded by $|\Omega_2| \cdot (\operatorname{wcol}_2(G, \pi) + 1)^{O(|\varphi|)}$ which again is bounded by $(\operatorname{wcol}_2(G, \pi) + 1)^{O(|\varphi|)}$. Computing Ω is dominated by its size.

5.4 From Formulas to Weights

The next lemma breaks down the evaluation of $\#y \varphi(y\bar{u})$ into evaluating quantifier-free first-order clauses on *single* variables with "weights." Note that there is no counting quantifier or dependence on y in these clauses. The lemma is essentially an adaption from [11] (Theorem 3 and Lemma 6).

▶ Lemma 43. Consider as input \vec{G}_{π}^2 and a set Ω' of conjunctive clauses of the form $\tau(y) \wedge \psi(\bar{x}) \wedge f(y) = x_i$, where $\psi(\bar{x})$ is in $FO[1, \sigma, 2, \bar{x}]$. In time $f(|\varphi|) \text{wcol}_2(G)^{O(|\varphi|)} ||G||$ we can compute a set of conjunctive clauses Ω with free variables \bar{x} , as well as functions $c_{\omega,i}(v) \colon V(G) \to \mathbf{Z}$ for $\omega \in \Omega$ and $i \in \{1, \dots, |\bar{x}|\}$ such that for every $\bar{u} \in V(G)^{|\bar{x}|}$ there exists exactly one formula $\omega \in \Omega$ with $\vec{G}_{\pi}^2 \models \omega(\bar{u})$ and for such a formula ω

$$\sum_{(\mu,\omega')\in\Omega'} [\![\#y\,\omega'(y\bar{u})]\!]^{\vec{G}_{\pi}^2} = \sum_{i=1}^{|\bar{x}|} c_{\omega,i}(u_i).$$

The size of Ω is $\operatorname{wcol}_2(G,\pi)^{O(|\varphi|)}$. The length of each $\omega \in \Omega$ is $O(\operatorname{wcol}_2(G,\pi))$, ω has multiplicity 2, and each literal in ω has at most one function application (e.g., $f(x_i) = x_i$).

Proof. Let ρ be the signature of \vec{G}_{π}^2 (namely, $(f_i)_{i \in [\text{wcol}_1(G,\pi)]} \cup (h_i)_{i \in [\text{wcol}_2(G,\pi)]}$) and $\Omega \subseteq \text{FO}[1,\rho,2]$ be the set of all complete conjunctive clauses with at most one function application per literal, signature ρ , multiplicity 2, free variables $\bar{x}z$. This set has three important properties: First, for every $\bar{u} \in V(G)^{|\bar{x}|}$ there exists exactly one $\omega \in \Omega$ with $\vec{G}_{\pi}^2 \models \omega(\bar{u})$.

Second, for every $\omega \in \Omega$ and conjunctive clause $\psi(\bar{x}) \in \mathrm{FO}[1, \rho, 2]$ either $\omega \models \psi$ or $\omega \models \neg \psi$. The size of Ω is bounded by $|\rho|^{2|\bar{x}|^2}$ as each complete conjunctive clause in $\mathrm{FO}[1, \rho, 2]$ can be identified with its positive literals.

Let now $\bar{u} \in V(G)^{|\bar{x}|}$, $(\mu, \tau(y) \wedge \psi(\bar{x}) \wedge g(y) = x_i) \in \Omega'$ and $\omega \in \Omega$ such that $\vec{G}_{\pi}^2 \models \omega(\bar{u})$. If $\omega \models \neg \psi$ then

$$[\![\# y \, \tau(y) \wedge \psi(\bar{u}) \wedge g(y) = x_i]\!]^{\vec{G}_{\pi}^2} = 0.$$

Otherwise, if $\omega \models \psi$ then

$$\llbracket \#y\,\tau(y) \wedge \psi(\bar{u}) \wedge g(y) = x_i \rrbracket^{\vec{G}_\pi^2} = \llbracket \#y\,\tau(y) \wedge g(y) = x_i \,\rrbracket^{\vec{G}_\pi^2}.$$

Using this observation, we define for every $\omega \in \Omega$ and $i \in \{1, ..., |\bar{x}|\}$ a set $\Gamma_{\omega,i}$ by iterating over all formulas $\omega \in \Omega$ and $(\mu, \tau(y) \wedge \psi(\bar{x}) \wedge g(y) = x_i \in \Omega'$ and adding $(\mu, \tau(y) \wedge g(y) = x_i$ to $\Gamma_{\omega,i}$ if $\omega \models \psi$. Now for every $\bar{u} \in V(G)^{|\bar{x}|}$ there exists exactly one formula $\omega \in \Omega$ with $\vec{G}_{\pi}^2 \models \omega(\bar{u})$, and for such a formula ω

$$\sum_{(\mu,\omega')\in\Omega'} \mu[\![\#y\,\omega'(y\bar{u})]\!]^{\vec{G}_{\pi}^2} = \sum_{i=1}^{|\bar{x}|} \sum_{(\mu,\tau(y)\wedge g(y)=x_i\in\Gamma_{\omega,i}} \mu[\![\#y\,\tau(y)\wedge g(y)=u_i]\!]^{\vec{G}_{\pi}^2}.$$
(10)

Fix one set $\Gamma_{\omega,i}$. For every formula $(\mu, \tau(y) \land g(y) = u_i) \in \Gamma_{\omega,i}$ we construct a function c with $c(v) = \llbracket \# y \, \tau(y) \land g(y) = u_i \rrbracket^{\vec{G}_{\pi}^2}$ in time $O(|\varphi| \cdot ||G||)$ by the following algorithm. Note that τ is quantifier-free and its size is bounded by $O(|\varphi|)$.

1 for $u \in V(\vec{G}_{\pi}^2)$ with $\vec{G}_{\pi}^2 \models \tau(u)$ do 2 $\mid c(g(u)) \leftarrow c(g(u)) + 1$

Let $c_{\omega,i}$ be the sum over all such functions c for formulas in $\Gamma_{\omega,i}$. Then

$$c_{\omega,i}(u_i) = \sum_{(\mu,\tau(y)g(y)=u_i \in \Gamma_{\omega,i}} \mu \llbracket \# y \, \tau(y) \wedge g(y) = u_i \rrbracket^{\vec{G}_{\pi}^2}. \tag{11}$$

Combining Equations (10) and (11) yields our statement.

Computing Ω takes linear time. Computing all $\Gamma_{\omega,i}$ takes $O(|\Omega| \cdot |\bar{x}| \cdot |\Omega'|)$ time. Computing each $c_{\omega,i}$ takes $O(|\Omega'| \cdot |\varphi| \cdot ||\vec{G}_{\pi}^2||)$ time. This gives us in total the desired running time.

For now, the size of the signature of the relational structure and the size of the clauses depend on $\operatorname{wcol}_2(G,\pi)$ which is too large for our application. The following lemma decreases both sizes to a number that only depends on $|\bar{x}|$, the number of free variables of the clause. The underlying structure remains untouched.

This lemma will be essential to make the running time fpt on nowhere and almost nowhere dense graph classes, but is not needed for graph classes of bounded expansion. Because of the bounded multiplicity the number of positive literals in ω is bounded by a function of k and most literals are negative. As ω is complete, we will be able to treat these negative literals equivalently, when necessary. We consider the following lemma together with the notion of multiplicity the main difference between the approach in this section and the one of [11].

▶ Lemma 44. Let G be a relational structure with signature σ with binary, symmetric and irreflexive relations and multiplicity 2 and ω a complete conjunctive clause in FO[1, σ , 2] over free variables $\bar{x} = x_1 \dots x_k$. Then we can compute a relational structure G' and a relational clause ω' both with signature ρ such that $td(G') \leq td(G)$, $|\rho| \leq 2k^2$ and $|\omega'| \leq 2^{2k^4} + k^2$ in time $||G|| + |\omega'|$ and for all $\bar{u} \in V(G)^k$

$$\llbracket \omega(\bar{u}) \rrbracket^G = \llbracket \omega'(\bar{u}) \rrbracket^{G'}.$$

Proof. Let L^+ be the positive literals of ω with two distinct variables (e.g., $E(x_i, x_j)$ and not $E(x_i, x_i)$), σ^+ be the set of relational symbols occurring in L^+ and $\sigma^- := \sigma \setminus \sigma^+$. As the multiplicity of G is at most 2, $|\sigma^+| \le 2k^2$.

We define a relational structure G' on vertices V(G) where for each $E' \in \sigma^+$ the edge relation E' is preserved. Additionally, we introduce the edge relation $E^- := \bigcup_{E' \in \sigma^-} E'$. Note that the underlying graph of G and G' is the same. Hence, their tree-depths are identical.

Moreover, the literals of the clause ω can be partitioned into sets ω_{σ^+} and ω_{σ^-} such that $\omega(\bar{x}) \equiv \omega_{\sigma^+}(\bar{x}) \wedge \omega_{\sigma^-}(\bar{x})$ where ω_{σ^+} is the conjunction of literals of ω with (positive

and negative) edge relations from σ^+ and equality and ω_{σ^-} the conjunction of literals using negative edge relations from σ^- .

Also note that $|\omega_{\sigma^+}| \leq 2^{2k^4}$ as there are at most k^2 pairs x_i and x_j and at most $2k^2$ choices for $f \in \sigma_+$. Also $\omega_{\sigma^-}(\bar{x}) \equiv \bigwedge_{f \in \sigma^-} \bigwedge_{i,j \in [k]} \neg E^-(x_i,x_j)$.

We define a new complete, conjunctive clause ω' with signature $\rho := \sigma^+ \cup \{E^-\}$

$$\omega'(\bar{x}) := \omega_{\sigma^+}(\bar{u}) \wedge \bigwedge_{i,j \in [k]} \neg E^-(x_i, x_j).$$

It is easy to see that for each $\bar{u} \in V(G)^k$ that

$$G \models \omega(\bar{u}) \iff G' \models \omega'(\bar{u}).$$
 (12)

The formula ω' has length at most $2^{2k^4} + k^2$ and that $\omega' \in FO[1, \rho, 2]$

We are now able to prove our main result of this section. The proof idea works as follows: Using Lemmas 41 and 43 we can break down the counting formula into a sum of vertex weights that depend only on single vertices. Using low treedepth colorings and an optimization variant of Courcelle's theorem, we can optimize it in fpt time.

However, this approach is not yet possible as both the signature and the length of the clauses are not bounded by a function of $|\varphi|$, but they depend on the weak coloring number of G. Hence, it is not suited as an input for Courcelle's theorem. We solve this problem by applying Lemma 44 which yields a shorter, equivalent formula of size f(k).

Proof of Theorem 35. We use Lemma 8 to compute a vertex ordering π with $\operatorname{wcol}_r(G,\pi) \leq \operatorname{wcol}_{g(r)}(G)^{g(r)}$ for every $r \in \mathbf{N}$ in linear time for a computable function g. Note that if have a running time or some structure of size bounded by $\operatorname{wcol}_{h(r)}(G,\pi)^{h(r)}$ for some computable function h, then it is also bounded by $\operatorname{wcol}_{f(r)}(G)^{f(r)}$ for some computable function f. This bound is good enough for most of our cases.

We use Lemma 38 to construct \vec{G}_{π} and \vec{G}_{π}^2 Lemma 41 to construct a set Ω' such that for every $\bar{u} \in V(G)^{|\bar{x}|}$

$$\sum_{(\mu,\omega)\in\Omega'} \mu \llbracket \#y\,\omega(\bar{u}y) \rrbracket^{\vec{G}_{\pi}^2} - \delta \le \llbracket \#y\,\varphi(\bar{u}y) \rrbracket^G \le \sum_{(\mu,\omega)\in\Omega'} \mu \llbracket \#y\,\omega(\bar{u}y) \rrbracket^{\vec{G}_{\pi}^2} + \delta \tag{13}$$

where $\delta := 4^{|\varphi|} \operatorname{wcol}_2(G)^{O(|\varphi|)}$.

Applying Lemma 43 to Ω' gives us a set Ω , and functions $c_{\omega,i}(v)$ with $c_{\omega,i}(v) = O(|\vec{G}_{\pi}^2|)$ such that for every $\bar{u} \in V(G)^{|\bar{x}|}$

$$\sum_{(\mu,\omega)\in\Omega'} \mu [\![\#y\,\omega(\bar{u}y)]\!]^{\vec{G}_{\pi}^2} = \sum_{i=1}^{|\bar{x}|} c_{\omega,i}(u_i),$$

where $\omega \in \Omega$ is the formula with $\vec{G}_{\pi}^2 \models \omega(\bar{u})$. Assume for now that we can compute for a given $\omega \in \Omega$ a tuple $\bar{u}^* \in V(G)^{|\bar{x}|}$ such that

$$\sum_{i=1}^{|\bar{x}|} c_{\omega,i}(u_i^*) = \max_{\bar{u}} \Big\{ \sum_{i=1}^{|\bar{x}|} c_{\omega,i}(u_i) \mid \vec{G}_{\pi}^2 \models \omega(\bar{u}) \Big\}.$$
 (14)

Then we could cycle through all $\omega \in \Omega$, compute a solution \bar{u}^* satisfying (14), and return the optimal \bar{u}^* among all of them. This gives us a solution to our original optimization

problem up to an additive error of δ resulting from Equation (13). Thus, from now on, we will concentrate on one formula $\omega \in \Omega$ and solve the optimization problem (14).

It will now be easier for us to work with relational instead of functional structures. We transform \vec{G}_{π}^2 into a relational undirected structure G' with the same universe via standard methods: The unary relations are preserved. Additionally, for every function symbol f we add the relation symbol E_f with $E_f^{G'} = \{(v, f_{\vec{G}_{\pi}^2}(v)), (f_{\vec{G}_{\pi}^2}(v), v) \mid v \in V(\vec{G}'), f_{\vec{G}_{\pi}^2}(v) \neq v \}$, a symmetric and irreflexive binary relation.

The resulting structure is isomorphic to an undirected graph with both vertex- and edge-labels, without self-loops¹, and has the same weak coloring numbers as \vec{G}_{π}^2 . We further construct a relational conjunctive clause $\omega'(\bar{x})$ such that $\vec{G}_{\pi}^2 \models \omega(\bar{u})$ iff $G' \models \omega'(\bar{u})$ for every $\bar{u} \in V(G)^{|\bar{x}|}$. This can be done by replacing each literal $f(x_i) = x_j$ by $E_f(x_i, x_j)$. Note that this preserves the multiplicity of the clauses, i.e., ω' has multiplicity 2.

As $\operatorname{wcol}_r(G') \leq \operatorname{wcol}_{2r}(G)$ for every r (see Definition 37) we can compute a low tree-depth coloring with few colors by Proposition 14, i.e., an r-treedepth coloring with at $\operatorname{most} \chi \leq \operatorname{wcol}_{2^{r-2}}(G') \leq \operatorname{wcol}_{2^{r-1}}(G)$ colors. Remember that we fixed an ω' such that $\vec{G}_{\pi}^2 \models \omega'(\bar{u})$. As the tuple \bar{u} is contained in some subgraph induced by at most $|\bar{x}|$ colors, we can from now on assume that we are working with graphs of bounded tree-depth: Define \mathcal{H} as the set of graphs induced by at most $|\bar{x}|$ colors in G. The size of \mathcal{H} is bounded by $\binom{\chi}{|\bar{x}|} \leq \operatorname{wcol}_{2^{|\bar{x}|}-1}(G,\pi)^{|\bar{x}|}$.

For every $\bar{u} \in V(G)^{|\bar{x}|}$ with $\vec{G}_{\pi}^2 \models \omega'(\bar{u})$ there exists $H \in \mathcal{H}$ such that $\bar{u} \in V(H)^{|\bar{x}|}$ and $H \models \omega'(\bar{u})$. In order to optimize (14), it is therefore sufficient to consider every graph $H \in \mathcal{H}$ and compute $\bar{u}^* \in V(H)^{|\bar{x}|}$ such that

$$\sum_{i=1}^{|\bar{x}|} c_{\omega,i}(u_i^*) = \max_{\bar{u} \in V(H)^{|\bar{x}|}} \Big\{ \sum_{i=1}^{|\bar{x}|} c_{\omega,i}(u_i) \mid H \models \omega'(\bar{u}) \Big\},$$
(15)

and then return the best value found for \bar{u}^* . The input H to the optimization problem (15) comes from a graph class with bounded tree-depth. Using Courcelle's theorem [4] one can solve a wide range of problems on these graphs in fpt time. Since we want to solve an optimization problem we require an extension of the original theorem. Courcelle, Makowsky, and Rotics define LinEMSOL [5] as an extension of monadic second order logic allowing one to search for sets of vertices with weights that are optimal with respect to a linear evaluation function.

Before we can apply the result of LinEMSOL to H and ω' , note that the size of the signature of H and ω' and the size of ω do not depend only on $|\bar{x}|$, but on the weak coloring number of G. Hence, it is unsuitable as an input in this form. Instead, we apply Lemma 44 to H and ω' resulting in a graph H^* and a complete conjunctive clause ω^* , both with a signature ρ where the size of ρ and ω^* is bounded by a function depending only on $|\bar{x}|$ and $H \models \omega'(\bar{u})$ iff $H^* \models \omega^*(\bar{u})$ for every $\bar{u} \in V(G)^{|\bar{x}|}$. Our linear evaluation function is $\sum_{i=1}^{|\bar{x}|} c_{\omega^*,i}(u_i)$ and our formula $\omega^*(\bar{x})$ clearly lies in monadic second order logic.

We find a solution \bar{u}^* to our optimization problem (15) in linear fpt time with LinEMSOL. Cycling through each $\omega \in \Omega$ and $H \in \mathcal{H}$ increases the running time by a factor of $|\Omega| \cdot |\mathcal{H}| \leq \operatorname{wcol}_2(G)^{O(|\varphi|)} \cdot \operatorname{wcol}_{2|\bar{x}|-1}(G)^{|\bar{x}|}$. Transforming G into G' and H into H^* takes linear time.

¹ We disregard self-loops, as they do not contain any additional information.

6 Hardness Results

In this section, we try to see how far the above result can or cannot be extended to either a bigger class of problems or to more general graph classes. Exemplary, we examine the distance-r versions of the dominating set, independent set and clique problem. Note that in contrast to the section before, we do not consider the partial problem versions. We see that each of these problems behave differently in this context. The distance-r dominating set problem is already hard for distance 1 on some almost nowhere dense graph classes, whereas distance-r independent set and distance-r clique are both fpt on almost nowhere dense graph classes.

As for graph classes, we consider classes that are closed under removing edges because monotone graph classes are very well understood and the notions of nowhere dense and almost nowhere dense coincide on those classes. Interestingly, for graph classes closed under removing edges the distance-r clique problem is fpt for all distances r if and only if the class is almost nowhere dense (under some complexity theoretic assumptions). However, there exist graph classes which are closed under removing edges but not almost nowhere dense that allow for fpt algorithms for the distance-r independent set problem. The difference of behavior between distance-r clique and distance-r independent set is also intriguing as the FO-formulation of these problems has exactly one quantifier alternation for both. Before we continue, we give a formal definition of distance-r clique.

▶ **Definition 45.** A set K of vertices is a distance-r clique in a graph G if there exist pairwise vertex disjoint paths of length at most r between each pair of vertices in K.

Note that K is a distance-r clique if and only if K are the principle vertices of an (r-1)-subdivision of a clique appearing as subgraph in G. A distance-1 clique is exactly a "usual" clique. Note that stars are *not* distance-2 cliques.

We also consider the generalization of an independent set.

▶ **Definition 46.** A set I of vertices is a distance-r independent set in a graph G if every distinct pair of vertices from I has distance strictly larger than r in G.

Note that a usual independent set is exactly a distance-1 independent set.

6.1 Exact Evaluation Beyond Nowhere Dense Classes

The following lemma proves that dominating set is W[1]-hard on almost nowhere dense classes.

The class of bipartite graphs with sides L and R where L has polylogarithmic size is almost nowhere dense: A witness for this is a vertex ordering that starts with L and starts . Only the vertices from L are weakly r-reachable from any vertex. Hence, $\operatorname{wcol}_r(G) \leq |L| + 1$ for each r.

▶ **Theorem 47.** In bipartite graphs whose left side has $2k(k-1)\lceil \log(n) \rceil$ vertices and whose right side has n vertices it is W[1]-hard to decide whether there are $\binom{k}{2}$ right-side vertices dominating all left-side vertices.

Proof. We reduce from colorful clique. Assume we have a k-partite graph G of size n consisting of parts V_0, \ldots, V_{k-1} (each of a different color) and want to find a colorful clique of size k. Without loss of generality, we can assume n to be large enough that $\binom{2\lceil \log(n)\rceil}{\lceil \log(n)\rceil-1} \ge n$. This means, we can find for each $v \in V(G)$ a unique binary encoding $\operatorname{enc}(v)$ of length $2\lceil \log(n)\rceil$ such that the first bit is set to one and in total exactly half the bits are set to one.

Let $\overline{\text{enc}}(v)$ be the binary complement of enc(v). We construct a bipartite graph H, whose left side is partitioned into cells C_{ij} for $0 \le i \ne j < k$, each of size $2\lceil \log(n) \rceil$, and whose right side will be specified soon. The vertices of each cell are ordered. When we say for a given vertex v from the right side and cell C that v is connected to C according to a specified encoding, we mean that for $1 \le l \le 2\lceil \log(n) \rceil$, v is connected to the lth vertex of C if and only if the lth bit in the encoding is set to one. For $0 \le i < k$ we define

$$\operatorname{succ}_{i}(j) = \begin{cases} j+1 \mod k & i \neq j+1 \mod k \\ j+2 \mod k & \text{otherwise.} \end{cases}$$

For all $0 \le i < j < k$ and all $u \in V_i$ and $v \in V_j$ such that $uv \in E(G)$, add a vertex $x_{u,v}$ to the right side and

- \blacksquare connect $x_{u,v}$ to $C_{i,j}$ according to $\operatorname{enc}(u)$,
- \blacksquare connect $x_{u,v}$ to $C_{i,\operatorname{succ}_i(j)}$ according to $\overline{\operatorname{enc}}(u)$,
- \blacksquare connect $x_{u,v}$ to $C_{j,i}$ according to $\operatorname{enc}(v)$,
- \blacksquare connect $x_{u,v}$ to $C_{j,\operatorname{succ}_i(i)}$ according to $\overline{\operatorname{enc}}(v)$.

Correctness: We claim the correctness of our construction: G contains a colorful clique of size k if and only if H contains a set of at most $\binom{k}{2}$ right-side vertices dominating all left-side vertices.

The forward direction is easy. If G contains a colorful clique v_0, \ldots, v_{k-1} then it is easy to see that the set $\{x_{v_i,v_j} \mid 0 \le i < j < k\}$ dominates all left-side vertices.

For the backward direction, assume there exists a set S of at most $\binom{k}{2}$ right-side vertices that dominates all left-side vertices. We say a vertex touches a cell if it is adjacent to at least one vertex from the cell. There are k(k-1) cells, each right-side vertex touches most four cells, and each cell needs to be touched by at least two vertices from S. Thus, with $|S| \leq k(k-1)/2$ and by a simple counting argument, S can only dominate all left-side vertices is if all cells are touched by exactly two vertices from S.

Let us fix a cell C. There exist exactly two vertices $x,y \in S$ touching C. Both x and y are adjacent to exactly half the vertices of C, meaning that every vertex in C has exactly one neighbor from x and y. For every vertex $v \in V(G)$, the encoding $\operatorname{enc}(v)$ has the first bit set to one. Thus, there exists a vertex $v \in V(G)$ such that one vertex from x and y is connected to C according to $\operatorname{enc}(v)$ and the other vertex from x and y is connected to C according to $\operatorname{enc}(v)$.

Assume a cell $C_{i,j}$ is connected to a vertex $x \in S$ according to $\operatorname{enc}(v)$ for some vertex $v \in V(G)$. By the way the adjacency of x is defined, $C_{i,\operatorname{succ}_i(j)}$ is connected to x according to $\overline{\operatorname{enc}}(v)$. By the previous paragraph, there exists a vertex from S such that $C_{i,\operatorname{succ}_i(j)}$ is connected to this vertex according to $\operatorname{enc}(v)$. By induction, for all $0 \le i < k$, there exists a vertex v_i such that for all $0 \le j \ne i < k$, each cell $C_{i,j}$ is connected to some vertex from S according to $\operatorname{enc}(v_i)$.

For all $0 \le i < j < k$, the vertex that touches $C_{i,j}$ according to $\operatorname{enc}(v_i)$ also touches $C_{j,i}$ according to $\operatorname{enc}(v_j)$. This guarantees that there is an edge between v_i and v_j in G. Therefore, the vertices v_0, \ldots, v_{k-1} form a clique of size k in G.

We can reduce the aforementioned dominating set variation to the classical dominating set problem by connecting the right side to a fresh vertex.

▶ Corollary 48. There exists an almost nowhere dense graph class \mathcal{C} where the dominating set problem is W[1]-hard and cannot be solved in time $n^{o(k)}$ assuming ETH. This implies also the hardness of the fragments PDS-like, $FOC_1(\mathbf{P})$, and $FOC(\{>\})$ of $FOC(\mathbf{P})$ on \mathcal{C} .

Note that this result does not follow from the intractability result of FO-logic on subgraphclosed somewhere dense classes, i.e. not nowhere dense classes.

6.2 Beyond Distance One

We showed that the dominating set problem is W[1]-hard on some almost nowhere dense graph class. However, this is not true for the distance-r clique and independent set problem.

Distance-r clique and independent set on the other hand are fpt on almost nowhere dense graph classes. Here, we use low treedepth colorings to solve existential FO formulas. With the right formulation and inclusion-exclusion this works even for distance-r independent set which cannot be expressed as a purely existential FO formula.

▶ **Theorem 49.** There exists a computable function f such that for every graph G the distance-r clique problem can be solved in time $\operatorname{wcol}_{f(k,r)}(G)^{f(k,r)}n$.

Proof. We can solve this problem with the help of subgraph queries where each subgraph is an $\leq r$ -subdivision of a k-clique. These subgraphs have less than $k^2(r+1)$ vertices and there are at most $(r+1)^{k^2}$ of them. Subgraph queries can be done by checking an existential FO-formula using Theorem 35.

▶ **Theorem 50.** There exists a computable function f such that for each graph G the distance-r independent set problem can be solved in time $\operatorname{wcol}_{f(k,r)}(G)^{f(k,r)}n$.

Proof. A distance-r k-subrelation is a function $D: \binom{[k]}{2} \to [r] \cup \{\infty, *\}$. We write $(G, h) \models D$ for a graph G and an injective function $h: [k] \to V(G)$ if for every $vw \in \binom{[k]}{2}$,

- 1. if $D(vw) = l \in [r]$, then $\operatorname{dist}_G(h(v), h(w)) \leq l$,
- **2.** if $D(vw) = \infty$, then $\operatorname{dist}_G(h(v), h(w)) \ge r + 1$.

Let $\llbracket D \rrbracket^G$ be the number of functions h with $(G,h) \models D$. In essence, D encodes whether some graph H appears in G as a subgraph with conditions on the distances of non-adjacent vertices.

Let D^{∞} be the distance-r k-subrelation with $D^{\infty}(vw) = \infty$ for every vw. There exists a distance-r independent set in a graph G if and only if there is some h such that $(G,h) \models D^{\infty}$, in particular, if and only if $\llbracket D^{\infty} \rrbracket^G > 0$.

To compute this value, we use the inclusion-exclusion principle. Let D be some distance-r k-subrelation with an entry vw such that $D(vw) = \infty$. Then the value of $\llbracket D \rrbracket^G$ can be computed as $\llbracket D \rrbracket^G = \llbracket D^* \rrbracket^G - \llbracket D^r \rrbracket^G$ where D^* and D^r are distance-r k-subrelations equal to D except for the value of vw which is v and v respectively. We apply this rule exhaustively until v does not appear in the images of the subrelations.

Distance-r k-subrelations without ∞ can be expressed by a disjunction of subgraph queries where each graph is an r-subdivision of graph on k vertices. These graphs have less than $k^2(r+1)$ vertices and there are at most $(r+2)^{k^2}$ such graphs. Using Lemma 8 and Proposition 14 we can compute a $k^2(r+1)$ -treedepth coloring with $\operatorname{wcol}_{f(k^2(r+1))}(G)^{f(k^2(r+1))}$ many colors. Using these low treedepth colorings and [8, Theorem 6 and 8] one can count how often such graphs appear as subgraphs in time $\operatorname{wcol}_{f'(k,r)}(G)^{f'(k,r)}n$ for some computable function f'.

6.3 Beyond Almost Nowhere Dense

For graph classes that are closed under removing vertices and edges, i.e., monotone graph classes, we know a lot already. Most importantly, FO-model checking is fpt on such classes if and only if the class is nowhere dense (unless FPT = W[1]) [22]. We now want to consider

graph classes that are only closed under removing edges. Here the concept of almost nowhere dense graph classes becomes interesting.

The following observation follows directly from characterization 6 in Theorem 28. If \mathcal{P} is a parameterized problem that can be solved in time $\operatorname{col}_{f(k)}(G)^{f(k)}n$ and \mathcal{C} is an almost nowhere dense graph class, then \mathcal{P} can be solved on \mathcal{C} in almost linear fpt time $f(k,\varepsilon)n^{1+\varepsilon}$ for every $\varepsilon > 0$. We complement this by showing that the distance-r clique problem is most likely not fpt on all graph classes that are not almost nowhere dense, but closed under removing edges. Hence, under certain common complexity theoretic assumptions, if a graph class $\mathcal C$ is closed under removal of edges then distance-r clique is fpt on $\mathcal C$ iff $\mathcal C$ is almost nowhere dense.

 \triangleright Theorem 51. Let \mathcal{C} be a graph class that is not almost nowhere dense, but closed under removing edges. Then there exists a number r, such that one cannot solve the distance-r' clique problem parameterized by solution size in fpt time on C for all $r' \leq r$ unless i.o.W[1] \subseteq FPT.

Similar hardness results in parameterized complexity are usually built on the hardness assumption FPT $\neq W[1]$. The complexity class i.o.W[1] should be read as "infinitely often in W[1]" and needs to be explained.

▶ **Definition 52.** For a language L and an integer n let $L_n = L \cap \{0,1\}^n$. A language L is in i.o.C for a complexity class C if there is some $L' \in C$ such that $L'_n = L_n$ for infinitely many input lengths n.

Considering the infinite often variant i.o.C of a complexity class C is an established technique in complexity theory (i.e., [3, 2]). To prove our result, we show that a graph class \mathcal{C} that is not almost nowhere dense, contains an infinite sequence of graphs having cliques of polynomial size as bounded depth topological minors. If $\mathcal C$ is also closed under removal of edges then having bounded depth topological clique minors of size n implies the existence of subdivisions of arbitrary graphs H of size n as induced subgraphs. Extra care needs to be taken to make sure that all paths connecting the principal vertices should be of equal length, since otherwise a reduction would need to try out an exponential number of possible length combinations to finally find the correct subdivision of H that is contained in C. The following corollary is a direct consequence of Theorem 28.4.

▶ Corollary 53. Let C be some graph class that is not almost nowhere dense. Then there are r, ε and an infinite sequence of strictly ascending numbers n_0, n_1, \ldots such that for all $i \in \mathbf{N}$ there is a graph $G \in \mathcal{C}$ of order at most n_i that contains an r'-subdivision of $K_{\lceil n \rceil \rceil}$ as a subgraph for some $r' \leq r$.

The consequence i.o. W[1] \subseteq FTP is weaker than W[1] \subseteq FPT. We could use the latter in Theorem 51 if we required a stronger precondition, i.e., that \mathcal{C} has "witnesses" for input lengths n_0, n_1, n_2, \ldots such that the gap between n_i and n_{i+1} is only polynomial. This approach has been used, e.g., in proving lower bounds on the running time of MSO-model checking in graph classes where the treewidth grows polylogarithmically [27, 17].

Proof of Theorem 51. Let r and ε be the constants (depending on \mathcal{C}) from Corollary 53. Assume that the distance-(r+1) clique problem on \mathcal{C} is fpt when parameterized by solution size. We will present a Turing reduction showing that the (usual) clique problem on the class of all graphs is infinitely often in FPT.

By Corollary 53 for infinitely many $n_0, n_1, \dots \in \mathbb{N}$ there exists a graph from \mathcal{C} of size at most $n_i^{1/\varepsilon}$ that contains an r'-subdivision of a clique of size n_i as a subgraph for some

 $r' \leq r$. Let us pick one $n = n_i$. Suppose we want to decide whether a graph G with n vertices contains a clique of size k. Since \mathcal{C} is closed under removal of edges, there exist $r' \leq r$, and $n \leq N \leq n^{1/\varepsilon}$ such that \mathcal{C} contains a graph $H_{r',N}$ consisting of an r'-subdivision of G together with N isolated vertices. Now for all k, G contains a clique of size k iff $H_{r',N}$ contains a distance-(r'+1) clique of size k. Assume for contradiction we had an algorithm that decides in time at most $f(r',k)n^c$ whether a graph in \mathcal{C} of size n contains an distance-(r'+1) clique for $r' \leq r$. (For graphs not in \mathcal{C} , the algorithm may give a wrong answer, but we can modify it to construct and test a witness of a distance-(r'+1) clique on yes-instances. Hence, we can assume that the algorithm never returns "no" on yes-instances.)

The existence of such an algorithm yields us an FPT algorithm for the k-clique problem on general graphs: For all $r' \leq r$, and $n \leq N \leq n^{1/\varepsilon}$, we run this (hypothetical) fpt algorithm in parallel on $H_{r',N}$ for $f(r',k)N^c$ time steps. Then G contains a clique of size k iff for at least one value of r' and N we have $H_{r',N} \in \mathcal{C}$ and $H_{r,N}$ contains a distance-(r'+1) k-clique.

As the k-clique problem is W[1]-hard, we get the desired result.

Note that this result does not extend to the distance-r independent set problem. Consider the class of graphs where at least half of its vertices are isolated. Then the distance-r independent set problem is trivially FPT for this graph class. However, this graph class is closed under removing edges, but it is not almost nowhere dense.

References

- Omid Amini, Fedor V. Fomin, and Saket Saurabh. Implicit branching and parameterized partial cover problems. *J. Comput. Syst. Sci.*, 77(6):1159–1171, 2011. doi:10.1016/j.jcss.2010.12.002.
- 2 Richard Beigel, Lance Fortnow, and Frank Stephan. Infinitely-often autoreducible sets. SIAM Journal on Computing, 36(3):595–608, 2006.
- 3 Leonard Berman. On the structure of complete sets: Almost everywhere complexity and infinitely often speedup. In 17th Annual Symposium on Foundations of Computer Science (sfcs 1976), pages 76–80, 1976. doi:10.1109/SFCS.1976.22.
- Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. Inf. Comput., 85(1):12-75, 1990. doi:10.1016/0890-5401(90)90043-H.
- 5 Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000. doi:10.1007/s002249910009.
- 6 Anuj Dawar, Martin Grohe, and Stephan Kreutzer. Locally excluding a minor. In 22nd IEEE Symposium on Logic in Computer Science (LICS 2007), 10-12 July 2007, Wroclaw, Poland, Proceedings, pages 270-279. IEEE Computer Society, 2007. doi:10.1109/LICS.2007.31.
- 7 Anuj Dawar and Stephan Kreutzer. Domination problems in nowhere-dense classes. In Ravi Kannan and K. Narayan Kumar, editors, IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2009, December 15-17, 2009, IIT Kanpur, India, volume 4 of LIPIcs, pages 157–168. Schloss Dagstuhl Leibniz-Zentrum für Informatik, 2009. doi:10.4230/LIPIcs.FSTTCS.2009.2315.
- 8 Erik D. Demaine, Felix Reidl, Peter Rossmanith, Fernando Sánchez Villaamil, Somnath Sikdar, and Blair D. Sullivan. Structural sparsity of complex networks: Bounded expansion in random models and real-world graphs. *J. Comput. Syst. Sci.*, 105:199–241, 2019. doi: 10.1016/j.jcss.2019.05.004.
- 9 Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999. doi:10.1007/978-1-4612-0515-9.
- Rodney G. Downey and Michael R. Fellows. Fundamentals of Parameterized Complexity. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

- Jan Dreier and Peter Rossmanith. Approximate evaluation of first-order counting queries. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 13, 2021*, pages 1720–1739. SIAM, 2021. doi:10.1137/1.9781611976465.104.
- Arnaud Durand and Etienne Grandjean. First-order queries on structures of bounded degree are computable with constant delay. *ACM Trans. Comput. Log.*, 8(4):21, 2007. doi:10.1145/1276920.1276923.
- Arnaud Durand, Nicole Schweikardt, and Luc Segoufin. Enumerating answers to first-order queries over databases of low degree. *Log. Methods Comput. Sci.*, 18(2), 2022. doi:10.46298/lmcs-18(2:7)2022.
- 14 Z. Dvořák. Asymptotical Structure of Combinatorial Objects. PhD thesis, Charles University, Faculty of Mathematics and Physics, 2007.
- Zdenek Dvorák, Daniel Kráľ, and Robin Thomas. Testing first-order properties for subclasses of sparse graphs. J. ACM, 60(5):36:1–36:24, 2013. doi:10.1145/2499483.
- Markus Frick and Martin Grohe. Deciding first-order properties of locally tree-decomposable structures. J. ACM, 48(6):1184–1206, 2001. doi:10.1145/504794.504798.
- 17 Robert Ganian, Petr Hliněný, Alexander Langer, Jan Obdržálek, Peter Rossmanith, and Somnath Sikdar. Lower bounds on the complexity of MSO1 model-checking. *Journal of Computer and System Sciences*, 80(1):180–194, 2014.
- Petr A. Golovach and Yngve Villanger. Parameterized complexity for domination problems on degenerate graphs. In Hajo Broersma, Thomas Erlebach, Tom Friedetzky, and Daniël Paulusma, editors, Graph-Theoretic Concepts in Computer Science, 34th International Workshop, WG 2008, Durham, UK, June 30 July 2, 2008. Revised Papers, volume 5344 of Lecture Notes in Computer Science, pages 195–205, 2008. doi:10.1007/978-3-540-92248-3_18.
- Martin Grohe. Generalized model-checking problems for first-order logic. In Afonso Ferreira and Horst Reichel, editors, STACS 2001, 18th Annual Symposium on Theoretical Aspects of Computer Science, Dresden, Germany, February 15-17, 2001, Proceedings, volume 2010 of Lecture Notes in Computer Science, pages 12–26. Springer, 2001. doi:10.1007/3-540-44693-1_2.
- Martin Grohe, Stephan Kreutzer, Roman Rabinovich, Sebastian Siebertz, and Konstantinos S. Stavropoulos. Coloring and covering nowhere dense graphs. SIAM J. Discret. Math., 32(4):2467–2481, 2018. doi:10.1137/18M1168753.
- 21 Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Characterisations of nowhere dense graphs (invited talk). In Anil Seth and Nisheeth K. Vishnoi, editors, IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2013, December 12-14, 2013, Guwahati, India, volume 24 of LIPIcs, pages 21-40. Schloss Dagstuhl Leibniz-Zentrum für Informatik, 2013. doi:10.4230/LIPIcs.FSTTCS.2013.21.
- Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *J. ACM*, 64(3):17:1–17:32, 2017. doi:10.1145/3051095.
- Martin Grohe and Nicole Schweikardt. First-order query evaluation with cardinality conditions. In Jan Van den Bussche and Marcelo Arenas, editors, *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Houston, TX, USA, June 10-15, 2018*, pages 253–266. ACM, 2018. doi:10.1145/3196959.3196970.
- Wojciech Kazana and Luc Segoufin. First-order queries on classes of structures with bounded expansion. *Log. Methods Comput. Sci.*, 16(1), 2020. doi:10.23638/LMCS-16(1:25)2020.
- 25 Henry A. Kierstead and Daqing Yang. Orderings on graphs and game coloring number. Order, 20(3):255-264, 2003. doi:10.1023/B:0RDE.0000026489.93166.cb.
- 26 Joachim Kneis, Daniel Mölle, and Peter Rossmanith. Partial vs. complete domination: t-dominating set. In Jan van Leeuwen, Giuseppe F. Italiano, Wiebe van der Hoek, Christoph Meinel, Harald Sack, and Frantisek Plasil, editors, SOFSEM 2007: Theory and Practice of Computer Science, 33rd Conference on Current Trends in Theory and Practice of Computer Science, Harrachov, Czech Republic, January 20-26, 2007, Proceedings, volume 4362 of Lecture

- Notes in Computer Science, pages 367–376. Springer, 2007. doi:10.1007/978-3-540-69507-3_31.
- 27 Stephan Kreutzer and Siamak Tazari. Lower bounds for the complexity of monadic second-order logic. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*, pages 189–198. IEEE Computer Society, 2010. doi:10.1109/LICS.2010.39.
- 28 Dietrich Kuske and Nicole Schweikardt. First-order logic with counting. In 32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017, pages 1–12. IEEE Computer Society, 2017. doi:10.1109/LICS.2017.8005133.
- Jaroslav Nešetřil and Patrice Ossona de Mendez. Grad and classes with bounded expansion I. decompositions. Eur. J. Comb., 29(3):760-776, 2008. doi:10.1016/j.ejc.2006.07.013.
- 30 Jaroslav Nešetřil and Patrice Ossona de Mendez. Sparsity Graphs, Structures, and Algorithms, volume 28 of Algorithms and combinatorics. Springer, 2012. doi:10.1007/978-3-642-27875-4.
- Jaroslav Nešetřil and Patrice Ossona de Mendez. On nowhere dense graphs. European Journal of Combinatorics, 32(4):600 617, 2011. URL: http://www.sciencedirect.com/science/article/pii/S0195669811000151, doi:https://doi.org/10.1016/j.ejc.2011.01.006.
- 32 Marcin Pilipczuk, Michał Pilipczuk, and Sebastian Siebertz. Lecture notes for the course "Sparsity" given at Faculty of Mathematics, Informatics, and Mechanics of the University of Warsaw, Winter semesters 2017/18 and 2019/20. Available https://www.mimuw.edu.pl/~mp248287/sparsity2.
- Detlef Seese. Linear time computable problems and first-order descriptions. *Math. Struct. Comput. Sci.*, 6(6):505-526, 1996. doi:10.1017/s0960129500070079.
- 34 Sebastian Siebertz. Nowhere Dense Classes of Graphs: Characterisations and Algorithmic Meta-Theorems. PhD thesis, TU Berlin, 2016.
- Alexandre Vigny. Dynamic query evaluation over structures with low degree. CoRR, abs/2010.02982, 2020. URL: https://arxiv.org/abs/2010.02982, arXiv:2010.02982.
- Xuding Zhu. Colouring graphs with bounded generalized colouring number. *Discret. Math.*, 309(18):5562–5568, 2009. doi:10.1016/j.disc.2008.03.024.